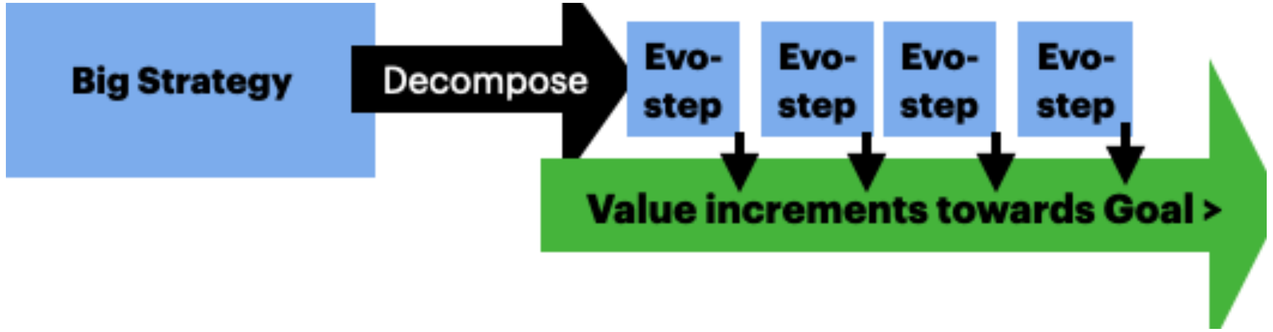
EVO

Evolutionary Value Optimization





© 2024 Tom@Gilb.com

Start Monday 2nd September 2024 16:27, 1ST DRAFT (292 PAGES) done 3 Sept 04:13 (by copying from my other 2024 books about Evo), 4.Sept major edit and add one (303 pages)
4th Sept 23:40 330 pages (added Cleanroom, Flyvbjerg) Ready to put up on Researchgate and Leanpub now.

EVO BOOK CONTENTS

Contents

1. Introduction

2. Basic Evo Cycle

3. Efficient Planning

4. Value Viability First

5. Knowing When to Pivot or Quit

6. Stakeholders

7. Stakeholder Values

8 Product and Service Qualities

9. Constraints

10. Design of Product and Services

11. Strategy (business model, organization)

12. Decomposition (of design or strategy)

13. Prioritization of delivery sequence

14. Value Management

15. Value Stream

16. Income Stream (Managing Costs)

17. Feedback

18. Specialist Roles

19. Presentations

Skipped

24. Case Studies

25. Templates

26. Skipped

27. Processes

28. Rules

29. Nicks Concerns

References (to papers, books slides, videos)

Note considerable relevant Evo sections were copied from [Startup] and [Plan-gineering] books 2024. ‘Re-use is ‘in’.

1.0. ‘EVO’ - Introduction

My first ‘Evo’ Project 1960

EVO is Both a concatenation of ‘Evolutionary’, and an acronym for Evolutionary Value Optimization.

I started doing it (‘it’ had no name), about 1960 intuitively.

I was responsible for new IBM ‘electronic computers’ (IBM 604 +), and a new application, and new business (my first consulting gig since leaving IBM), and I was 20 years old. I had not been taught by IBM or Oslo University how to manage projects, or how to design systems.

But I guess my common sense told me to *do one small useful step at a time. Lock it in if it worked. ‘Fix’ if it didn’t.* Keep going towards the longer-term objective *(a fully automated invoicing system, replacing the older one, for a clothing wholesaler, Dobloug Brothers)*. I did it in about 20 steps, and it was a success, nobody suggested otherwise. [Success]

So, I made a mental note, and continued to use it, and to teach it; until this very day.

We named it and formalized it along the way.

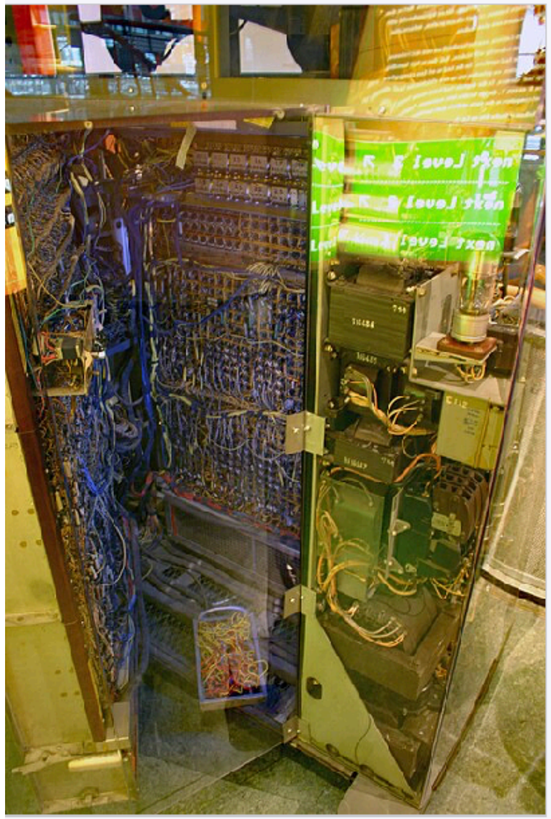
IBM 604 Electronic Calculating Punch

IBM



IBM 604 as part of IBM CPC assembly

Manufacturer IBM

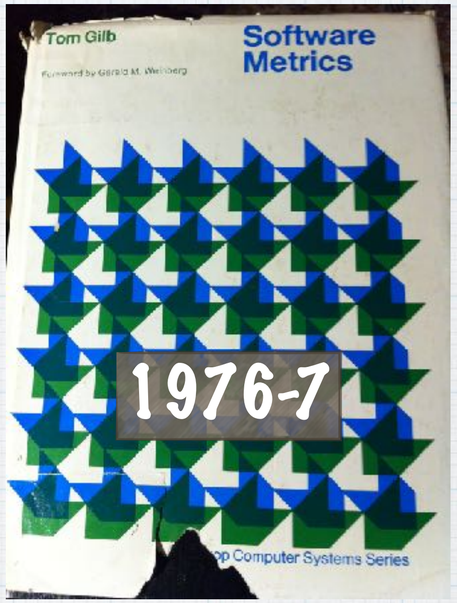


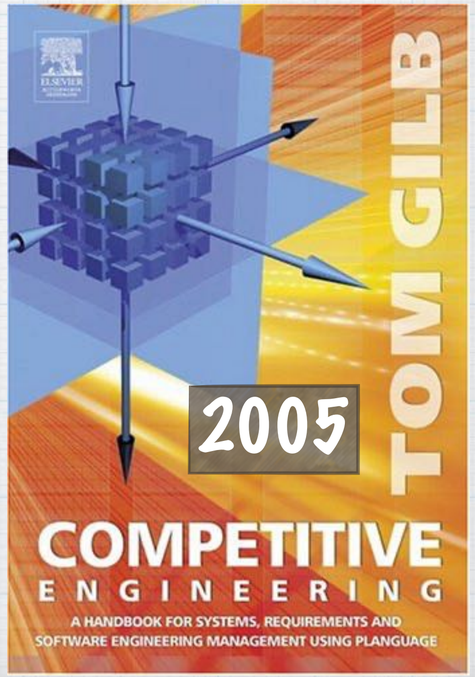
IBM 604 Electronic Calculator at NEMO national science museum in Amsterdam. Note plugboard control panel used to program the 604, at bottom.

1.1 16 Years Later, 1976, I generalized my experience

‘On step-results measurement, and retreat possibility’

'A complex system will be most successful
if it is implemented in small steps
and if each step has a clear measure of
successful achievement
as well as a "retreat" possibility to a
previous successful step upon failure.'





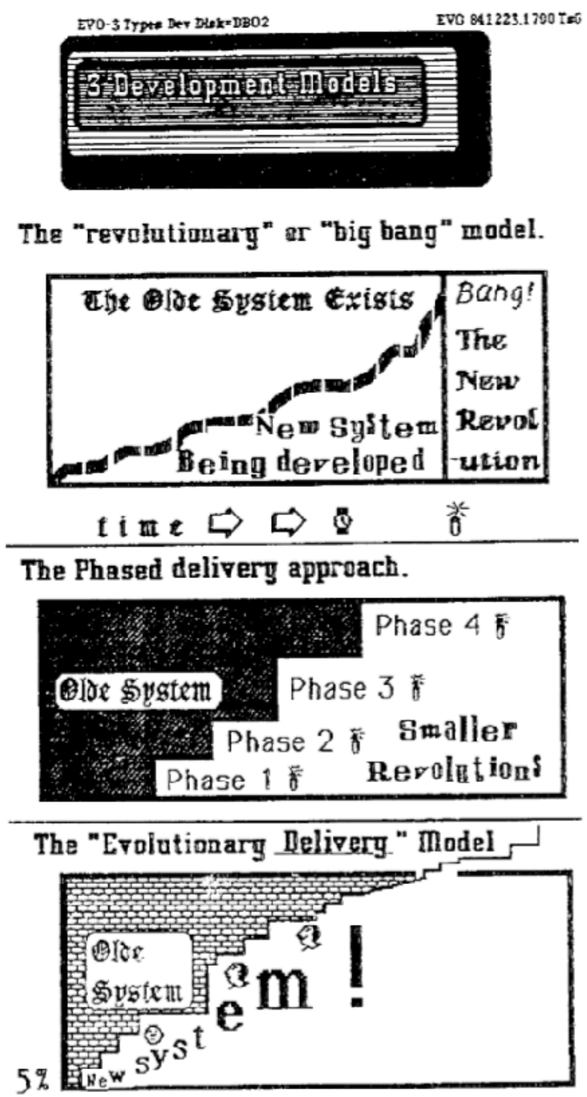
Source: (p. 214), Software Metrics [SM] 1976-7. And in [PoSEM] 1988 Evo History Ch. 15, and as Evo Chapter 10 in [CE] 2005

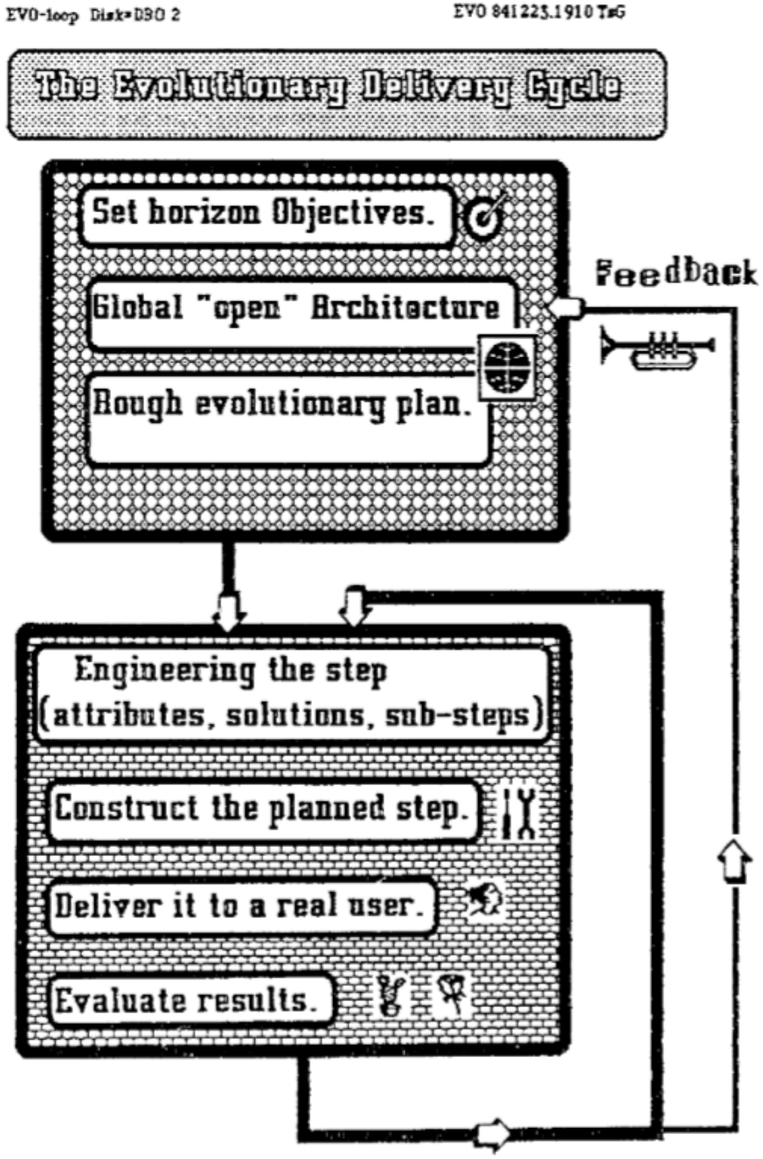
See this Presenter Note for more detail, or ask me tom@Gilb.com

I was 35 years old when this was published. I had 16 years ‘agile’ (or ‘Evo’, ‘deliver value to stakeholder in small increments, measure value and retreat if necessary’) experience by then (from 1960 Dobloug Case, and UiOslo Publisher/Admin (1968) etc).

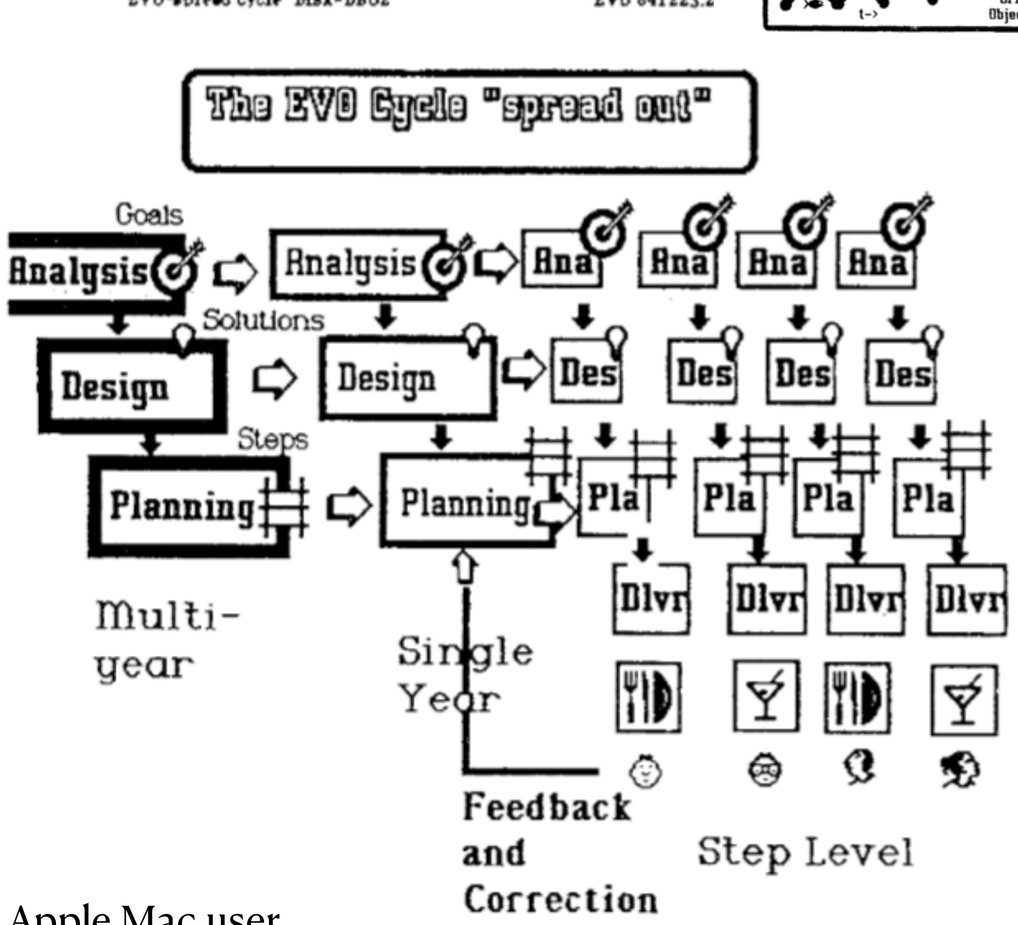
1.2 Early Publication in Journal (ACM SEN [Evo1985])

But notice this is 9 years after my 1976 Software Metrics book, so I have a quarter of century experience to derive my models from









Notice a subtle detail here. We start with the Old system, and through an incremental process of improvement, we deliver the 'new system'

And yes I was an early Apple Mac user

Principles of Evo

Source [CE] 2005 Evo Chapter 10

1. **The Principle of 'Capablanca's next move'**

There is only one move that really counts, the next one.

2. **The Principle of "Do the juicy bits first'**

Do whatever gives the biggest gains. Don't let the other stuff distract you!

3. **The Principle of 'Better the devil you know'**

Successful visionaries start from where they are, what they have and what their customers have.

4. **The Principle of 'You eat an elephant one bite at a time'**

System stakeholders need to digest new systems in small increments.

5. **The Principle of 'Cause and Effect'**

If you change in small stages, the causes of effects are clearer and easier to correct.

6. **The Principle of The early bird catches the worm'**

Your customers will be happier with an early long-term stream of their priority improvements, than years of promises, culminating in late disaster.

7. **The Principle of 'Strike early, while the iron is still hot'**

Install small steps quickly with people who are most interested and motivated.

8. **The Principle of 'A bird in the hand is worth worth two in the bush'**

Your next step should give the best result you can get

9. **The Principle of 'No plan survives first contact with the enemy'**

A little practical experience beats a lot of committee meetings.

10. **'The Principle of 'Adaptive Architecture'**

Since you cannot be sure where or when you are going, your first priority is to equip yourself to go almost anywhere, anytime.





Chapter 2 The Basic Evo Cycle

The EVO Process, the One-Page Summary

A Systems Engineering Process
For incremental value delivery management

A *PLANNING* CYCLE
Clarify your environment: critical-stakeholders' territory

1. **STAKEHOLDERS:** Identify *critical stakeholder's* and their critical **values**
2. **VALUES: Quantify** and **clarify** y*our project's* critical values: what *degree* of values do you expect the design to deliver, to which stakeholders
3. Identify design **constraints**: legality, political, cultural, policy, other plans
4. Identify design **resource-limitation**s: time, money, operational costs for example.

Design (Solutions): Architecture Level

6. **SOLUTIONS:** use the many Value and Resource requirements, AS EXPLICIT TOOLS, to find solutions, to deliver Values, within constraints. Aka Designs, Strategies, Architecture
7. **DECOMPOSE:** Decomposition of top-level architecture into design components
8. **PRIORITY:** Identification, *detailed* specification and *prioritization* of top-level architecture, and of any *decomposed* design solutions..

A *VALUE DELIVERY* CYCLE: An 'Evo-step'.

9. The **Evo Value delivery steps** (for example, a week, or 2% of total project budget)
   1. Select your highest **priority value**, and the most-critical scale-parameter attributes (*the who, what, where, when - slices of reality*).
   2. **DEVELOP**: Find a design component which will deliver good **value-for-resources** to your critical priority requirements.
   3. Ready the design-component for delivery: for ***integration*** **to the existing system.**
   4. **DELIVER: Deliver** the design component to the **real system**
   5. **MEASURE: Measure** the results (values and costs) of the design increment
   6. **Dynamic-Design-To-Requirements:** If results are **not OK**, attempt design **improvement, and redeliver.**
   7. **KEEP CYCLING:** If results *are* **OK,** then **repeat** this 'value delivery cycle', scale up, until 'done'.

Project Completion: All Value is delivered

10. **DONE?**: When **all value requirements are reached**, or when all **critical resources are used up**. Stop.



Gilb Value Cycle ('EVO')

Copyright 2020 kai@Gilb.com



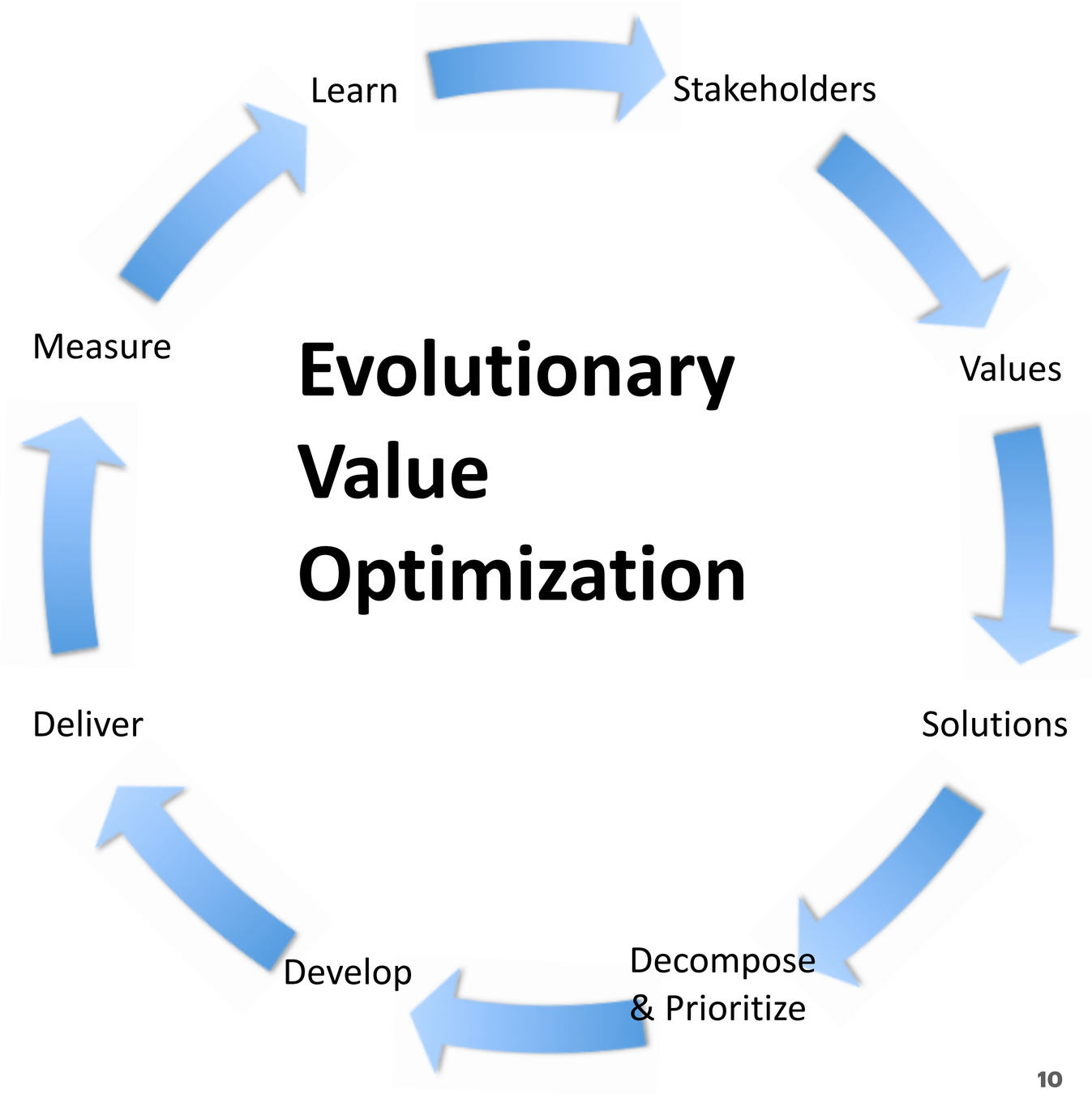
2. The Planning Cycle

Sub-processes

2.1.1. A Planning Cycle

The 'EVO' Cycle

- There is a clear logic to planning. Certain planning elements cannot be done safely, without unnecessary risks, until other elements have been done fairly well. [DL].
- The most common planning fault, is to work on the 'means' - before we have clarified the 'ends' - which those means will support.
- For example the term 'strategic planning' is missing emphasis on the main points.
- The main point is delivering values to the stakeholders. Is it possible to agree on that simple planning-foundational idea?
- I prefer to call it **'Value Planning'** [Value Planning]. Were I to be even logically stricter with myself, I should probably call it **'Stakeholder Satisfaction' planning** [SE].



2.2.1. Stakeholders

The stakes are 'potential' requirements

- Once we have agreed on some basic ideas of what we are planning for, like area, market, people, tasks: which we can call the 'Plan Scope', the 'environment', we can take a next logical planning step: identify potential stakeholders. 'Potential' means we reserve the right to exclude or to down-prioritize them later. But, we do not know enough about them, to do so - yet.
- One 'tool' I use, to quickly make some planning decisions, is to use the 'priority' term '**critical**'. I want to deal with the **critical stakeholders** first, because, by definition, if I do *not*, then *they* have the **power** to destroy my planned system.
- This power can be legal, market, preference, economic, military: but they *have* that power, and have been seen to use it, or to *threaten* to use it, or suspected of being *willing* to use it - if we are not 'nice enough' to them. [SE]
- Many methods of planning are doomed to fail, because they have too narrow an idea of 'stakeholders'. They talk about 'customers' and 'users' (User Stories, Use Cases, UX Design) but planners totally forget to consider dozens of real existing potential *critical* stakeholders, like European Law, which is not a user, nor a customer. But stakeholders like that can impose huge fines, and shut you out of a large market, if you do not consider them.
- There are even 'official standards 'with defined stakeholder concepts, that do not *explicitly* recognize that entities 'holding a big critical stake' are not all 'humans!. Big mistake. If it has power and needs respect it!

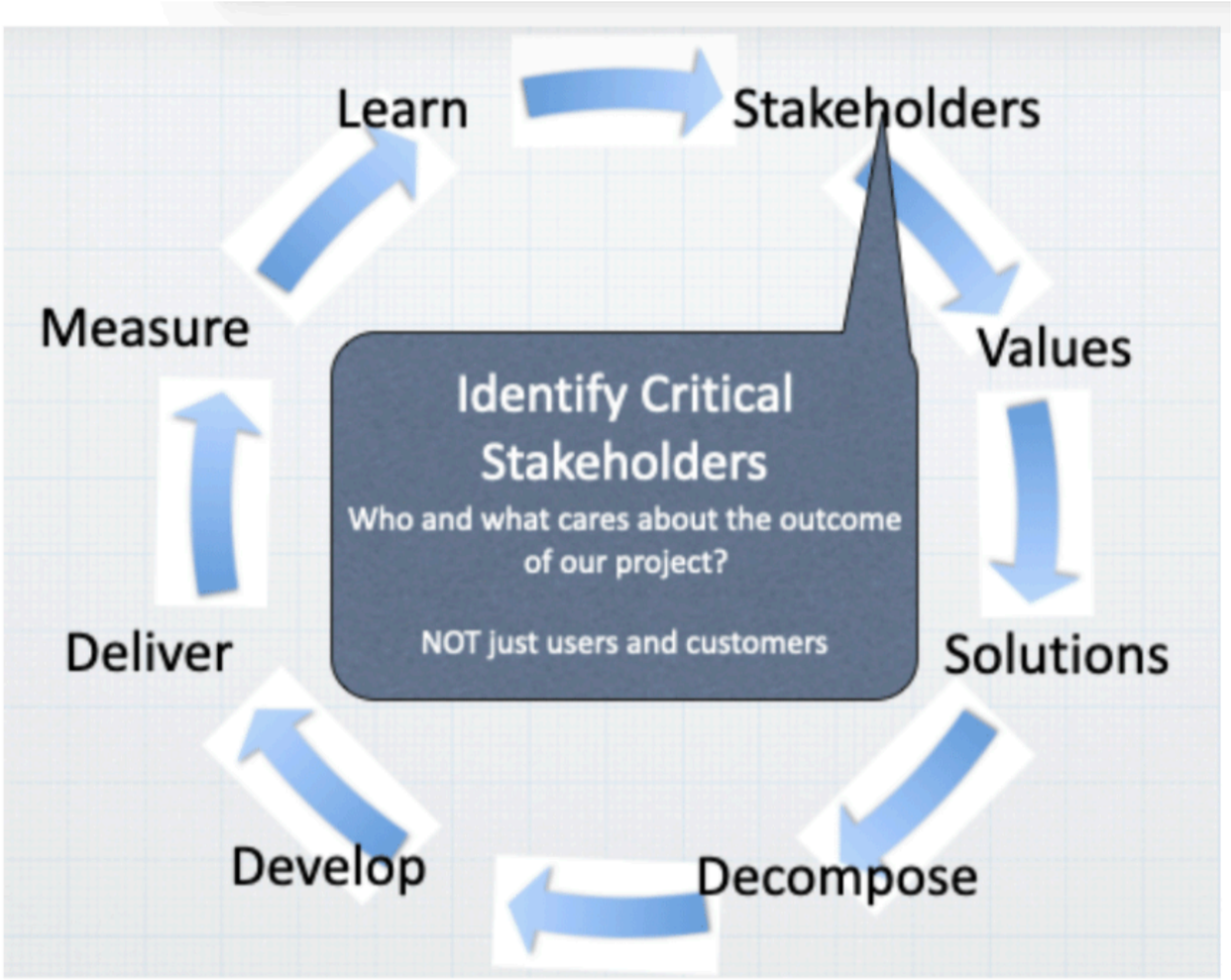


Figure 10.1.1 M. The Gilb Evo Cycle, where stakeholders are the logical prelude to value, and Scale analysis. But there is no simple sequence. It repeats and we learn better.

2.2.2. Stakeholders

The stakes are ‘potential’ requirements

- Stakeholders, by definition, have *their* ‘values’. Things they *care* about to some degree. These values can be articulated as ‘potential’ requirements.
- They are of two general types: ‘**Performance Values**’ (*how good* the system is for them, [Value Planning]). And **Constraints**, *limits to what they will tolerate.*
- If you miss a *critical* stakeholder, or if you miss *one* of their *critical* values; then you are far-less likely to satisfy the critical value. Guess what? Failure to deliver the ‘unknown value’ is probable.
- So, it is worth *some* discipline in identifying stakeholders. It might only take an hour!.



Figure 10.1.1 M. The Gilb Evo Cycle, where stakeholders are the logical prelude to value, and Scale analysis. But there is no simple sequence. It repeats and we learn better.

2.2.3 Simple Stakeholder Example

Here is an example an initial draft stakeholder list, below

- ‘Imagine an international software AI startup, based in UK’
- Stakeholders.
  - Investors
  - Owners
  - Founders
  - Customers
  - Users
  - AI Laws in UK

Here's a list of 20 critical stakeholders for an international software AI startup based in the UK, along with one critical requirement for each:

1. Founders: Clear vision and strategy for the company's AI technology and market positioning[1][4].

2. Investors: Strong potential for return on investment and scalability of the AI solution[1][2].

3. Employees: Competitive compensation and opportunities for professional growth in AI[1].

4. Customers: Reliable, effective AI solutions that address their specific needs[4].

5. Users: User-friendly interface and demonstrable value from the AI product[4].

6. UK Government: Compliance with AI regulations and contribution to the UK's AI leadership[1][3].

7. Data Protection Authorities: Adherence to data protection laws, especially GDPR[3].

8. AI Ethics Boards: Ethical development and deployment of AI technologies[1][3].

9. Competitors: Differentiation in AI capabilities and market approach[4].

10. Technology Partners: Seamless integration and compatibility with existing systems[4].

11. Academic Institutions: Collaboration opportunities for AI research and talent pipeline[1].

12. Media and Press: Positive coverage of the company's AI innovations and impact[2].

13. Industry Regulators: Compliance with sector-specific regulations for AI applications[3].

14. Legal Advisors: Guidance on intellectual property protection for AI algorithms[3].

15. Cloud Service Providers: Reliable and scalable infrastructure for AI operations[4].

16. AI Talent Pool: Access to skilled AI developers and researchers[1][2].

17. International Markets: Adaptability of AI solutions to different cultural and regulatory environments[1].

18. Cybersecurity Experts: Robust security measures to protect AI systems and data[3].

19. Environmental Stakeholders: Energy-efficient AI solutions to minimize carbon footprint[1].

20. Local Communities: Job creation and positive socio-economic impact through AI innovation[1].

This list incorporates the stakeholders you mentioned (Investors, Owners, Founders, Customers, Users) and considers AI Laws in the UK as part of the regulatory environment. Each stakeholder has a unique and critical requirement that directly impacts the success and sustainability of the AI startup.

Citations:

[1] https://api.startupcoalition.io/u/2023/10/The-UKs-AI-Startup-Roadmap-FOR-RELEASE.pdf

[2] https://www.techuk.org/resource/tech-nation-s-2024-report-gives-the-spotlight-exploring-uk-tech-in-the-age-of-ai.html

[3] https://assets.publishing.service.gov.uk/media/641d71e732a8e0000cfa9389/artifical_intelligence_sector_study.pdf

[4] https://www.f6s.com/companies/artificial-intelligence/united-kingdom/co

[5] https://sifted.eu/articles/scaleup-policy-wins-uk-government

I asked Perplexity-AI to help me out.

2.2.4 So, what was I missing out on?

I asked, and got 14 extra 'critical' stakeholders, and 14 corresponding critical value objective Goals. (Wishes)

Certainly! Here are the tags of the critical requirements for the 14 stakeholders you didn't initially list:

1. EMPLOYEE_SATISFACTION
2. UK_GOV_COMPLIANCE
3. DATA_PROTECTION
4. AI_ETHICS
5. COMPETITOR_DIFFERENTIATION
6. TECH_PARTNER_INTEGRATION
7. ACADEMIC_COLLABORATION
8. MEDIA_COVERAGE
9. INDUSTRY_COMPLIANCE
10. IP_PROTECTION
11. CLOUD_RELIABILITY
12. AI_TALENT_ACQUISITION
13. INTERNATIONAL_ADAPTABILITY
14. CYBERSECURITY_ROBUSTNESS
15. ENERGY_EFFICIENCY
16. COMMUNITY_IMPACT

Aspect	6 Goals	20 Goals
Survival Probability	30-35%	80-85%
Performance in Core Areas	Strong	Exceptional
Risk Level	High	Low
Long-term Viability	Uncertain	Strong
Scalability	Limited	High
Investor Attractiveness	Moderate	High
Market Positioning	Focused	Comprehensive
Adaptability	Low	High

Figure 2.2.4 I asked perplexity to estimate the startup survival, 3 years, Difference between 6 goals (for 6 stakeholders) and all 20 goals (for 20 stakeholders)

See AI detail in Presenter's Notes

2.3.1 Shifting to a different organization, for variation, A Transport Company, with Critical Stakeholders, and some of their Critical Values: in 'Planguage'

- Based on the scenario you've described, here are the 10 most critical stakeholders for your medium-sized Polish road and lorry transport brokerage, along with up to 3 critical value objectives for each, specified in Planguage format:

- **1. Trucking Companies/ Carriers**

- Tag: **CarrierSatisfaction**
- **Ambition**: Improve carrier satisfaction with our brokerage services
- **Scale**: % of carriers rating their satisfaction as "high" or "very high"
- **Meter**: Annual carrier satisfaction survey
- **Status**: 65%
- **Tolerable**: 70%
- **Wish**: 90%
- **Stakeholders**: Carriers, Operations Team

- Tag: CarrierPaymentSpeed
- Ambition: Reduce time to pay carriers
- Scale: Average days to pay after delivery
- Meter: Financial system reports
- Status: 45 days
- Tolerable: 30 days
- Wish: 15 days
- Stakeholders: Carriers, Finance Team

- **2. Shippers/Clients**

- Tag: ShipperRetention
- Ambition: Increase shipper retention rate
- Scale: % of shippers who use our services again within 12 months
- Meter: CRM system reports
- Status: 60%
- Tolerable: 75%
- Wish: 90%
- Stakeholders: Shippers, Sales Team

- Tag: ShipperSatisfaction
- Ambition: Improve shipper satisfaction with our brokerage services
- Scale: Net Promoter Score (NPS)
- Meter: Quarterly NPS survey
- Status: 30
- Tolerable: 50
- Wish: 70
- Stakeholders: Shippers, Customer Service Team

- **3. Employees**

- Tag: EmployeeRetention
- Ambition: Reduce employee turnover
- Scale: Annual employee turnover rate
- Meter: HR system reports
- Status: 25%
- Tolerable: 15%
- Wish: 5%
- Stakeholders: Employees, HR Team

- Tag: EmployeeSatisfaction
- Ambition: Improve employee job satisfaction
- Scale: % of employees reporting high job satisfaction
- Meter: Annual employee satisfaction survey
- Status: 60%
- Tolerable: 75%
- Wish: 90%
- Stakeholders: Employees, Management Team

- **4. Financial Institutions/Lenders**

- Tag: DebtReduction
- Ambition: Reduce company debt
- Scale: Total debt in PLN
- Meter: Financial statements
- Status: 5,000,000 PLN
- Tolerable: 3,000,000 PLN
- Wish: 1,000,000 PLN
- Stakeholders: Lenders, Finance Team

- Tag: CreditRating
- Ambition: Improve company credit rating
- Scale: Credit rating score
- Meter: Credit rating agency reports
- Status: BB
- Tolerable: BBB
- Wish: A
- Stakeholders: Lenders, Finance Team

- **5. Regulatory Bodies**

- Tag: ComplianceRate
- Ambition: Maintain high compliance with transport regulations
- Scale: % of compliance in annual audits
- Meter: Regulatory audit reports
- Status: 95%
- Tolerable: 98%
- Wish: 100%
- Stakeholders: Regulatory Bodies, Legal Team

- **6. Technology Providers**

- Tag: SystemModernization
- Ambition: Upgrade internal office systems
- Scale: % of systems modernized
- Meter: IT department reports
- Status: 30%
- Tolerable: 70%
- Wish: 100%
- Stakeholders: IT Team,

2.3.2. A ‘Value’ Specification

A basic ‘Planguage’ specification

- Tag: **CarrierSatisfaction**
- **Ambition**: Improve carrier satisfaction with our brokerage services
- **Scale**: % of carriers rating their satisfaction as "high" or "very high"
- **Meter**: Annual carrier satisfaction survey
- **Status**: 65%
- **Tolerable**: 70%
- **Wish**: 90%
- **Stakeholders**: Carriers, Operations Team

Explanations

- Tag: **CarrierSatisfaction**
  - **THIS IS A PERMANENT ID FOR THIS VALUE REQUIREMENT**
- **Ambition**: Improve carrier satisfaction with our brokerage services
  - AMBITION LEVEL IS A SIMPLE VISION, OFTEN FROM MANANGERS
- **Scale**: % of carriers rating their satisfaction as "high" or "very high”
  - THE ‘SCALE OF MEASURE’ IS THE FOUNDATION FOR QUANTIFICATION AND CLARIFICATION OF THE VALUE (LIKE VOLTS, MPH)
- **Meter**: Annual carrier satisfaction survey.
  - METER IS THE PLANNED MEASUREMENT METHOD (LIKE VOLTMETER, SPEEDOMETER)
- **Status**: 65%
  - STATUS IS THE CURRENT LEVEL OF THE VALUE
- **Tolerable**: 70%
  - TOLERABLE IS THE MINIMUM FUTURE LEVEL
- **Wish**: 90%
  - WISH IS A DESIRED POTENTIAL FUTURE SUCCESSFUL LEVEL
- **Stakeholders**: Carriers, Operations Team
  - MORE THAN ONE STAKEHOLDER CAN SUPPORT A VALUE IDEA, EVEN AT DIFFERENT LEVELS OF PERFORMANCE.

2.3.3 Value Objectives Principles

Source: [Decision-eering] book 2022

3.4 Principles of clear objectives for decisions.

© 2022 Tom Gilb for Decision-eering book

1.QUANTIFIED OBJECTIVES IS BASIC: If you do not have clear quantified stakeholder value objectives specified, then all related decisions about suitable strategies to meet the objectives are totally meaningless, and cannot be understood, or reviewed.

2. **DEFINED AGREED CLEAR VALUE IDEAS**: Absolutely all stakeholder value objectives are variables, by our definition of value, therefore a well-defined **Scale** of measure is a necessity and always possible if you know how [36, QU].

Example of a Wish statements, Tagged 'Short Term' and 'Long Term'



Source [11, SE] *Figure 5.3.3*
The nearest term Wish statement has the highest initial priority, by that date. The long term Wish level can wait a few y

3. **INCREMENTAL MEASUREMENT FEEDBACK**: If you expect to use the Objective in practice to get feedback as you incrementally build your value stream, you will need to define a **Meter,** which cost effectively allows you to measure incremental value in practice; and to make decisions about how well things are going and if change in strategy is needed immediately.

4. **BENCHMARK:** It is not enough to specify a Target level or your objective. You reply need to do enough analysis of the current situation to specify a **Status** level. Otherwise you will not be able to make good decisions select strategies for moving above that level.

5. **SUCCESS DECISION:** you need to define, decide on, one or more levels of the value objective which constitute successful level, where effort can stop, and be redeployed to other unfulfilled objectives. This supports dynamic decision-making.

6. **FAILURE DECISION:** you need to define, decide on, one or more numeric constraint levels ('Tolerable' levels) to officially define the border between failure and not-failure levels. This will motivate planners to design and deliver until at least these levels are reached. It will help you decide on your priorities for resources.

2.4.1 The Means/Strategies/ Architecture/Solutions

=

'The stuff we can *do* and *make* which will deliver Values'

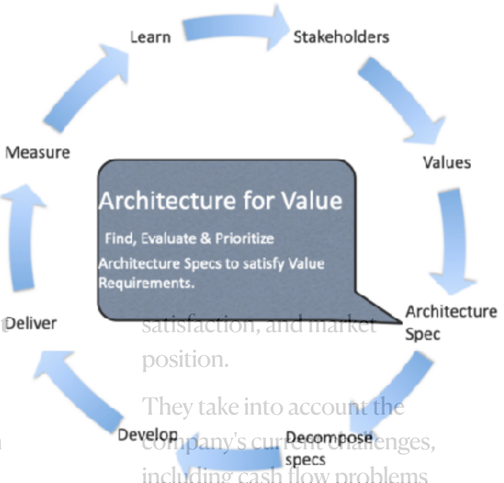
(At acceptable budgets and deadlines)



EVO Cycle

2.4.1 ‘Means to our Ends’ aka Strategies

For Polish Road Transport Values



Based on the value objectives you've outlined, here are **10 effective strategie**s to help your Polish road and lorry transport brokerage achieve its goals:

1. **Digital Transformation Initiative**

Implement a comprehensive digital transformation of your internal systems to address the SystemModernization objective. This will improve operational efficiency, enhance data analytics capabilities, and provide better service to both carriers and shippers.

2. **Carrier-Centric Payment System**

Develop a faster, more efficient payment system for carriers to improve CarrierPaymentSpeed. This could include options like early payment discounts or supply chain financing, which can also help with cash flow issues.

3. **Customer Experience Enhancement Program**

Launch a program focused on improving both CarrierSatisfaction and ShipperSatisfaction. This could include personalized service, real-time tracking, and a user-friendly mobile app for both carriers and shippers.

4. **Strategic Debt Restructuring**

Work with financial institutions to restructure debt, aiming to improve DebtReduction and CreditRating. This may involve negotiating better terms or exploring alternative financing options.

5. **Compliance and Risk Management System**

Implement a robust compliance and risk management system to maintain and improve ComplianceRate. This will help avoid regulatory issues and potentially improve relationships with financial institutions.

6. **Employee Development and Retention Program**

Create a comprehensive program to improve EmployeeSatisfaction and EmployeeRetention. This could include training opportunities, career advancement paths, and improved work-life balance initiatives.

7. **Market Expansion Strategy**

Develop a targeted strategy to increase MarketShare. This could involve expanding into new geographic areas or offering specialized services to niche markets within the transport industry.

8. **Industry Leadership Initiative**

Increase AssociationInvolvement by actively participating in industry associations and events. This can help build the company's reputation and create networking opportunities for business growth.

9. **Community Engagement Plan**

Implement a structured community engagement plan to improve CommunityImpact. This could include local hiring initiatives, partnerships with educational institutions, or supporting local environmental projects.

10. **Operational Efficiency Drive**

Launch a company-wide initiative to improve operational efficiency, targeting the Profitability objective. This could involve process optimization, cost reduction measures, and better resource allocation.

These strategies are designed to address **multiple objectives simultaneously**, focusing on improving financial health, stakeholder satisfaction, and market position.

They take into account the company's current challenges, including cash flow problems and competitive pressure, while also looking towards future growth and sustainability[1][2][3][4][5].

Citations:

[1] https://lgigroup.com/glossary/transport-logistics/

[2] https://transportlogistic.de/en/trade-fair/

[3] https://transportlogistic.de/en/

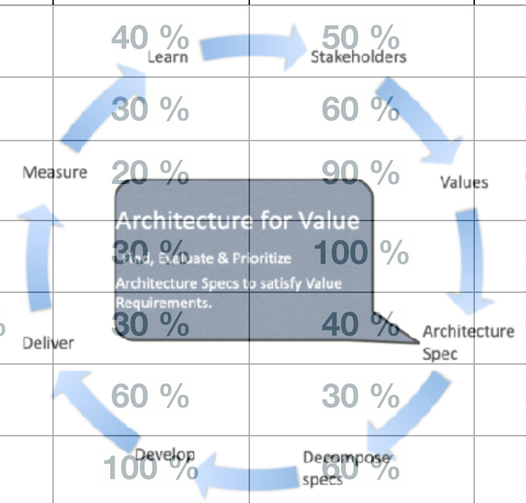
[4] https://transport-logistikk.no/welcome-to-transport-and-logistics-2023/

[5] https://www.kristiania.no/en/syllabus/school-of-economics-innovation-and-technology/first-cycle-degree/tlo3100/transport-logistics/

2.4.2. How good are the strategies at meeting our Wish level Value Objective?

100% means that the strategy would reach the Wish level

Values / Objectives	Digital Transformation	Carrier-Centric Payment	Customer Experience	Debt Restructuring	Compliance System	Employee Development	Market Expansion	Industry Leadership	Community Engagement	Operational Efficiency	Total Effectiveness
CarrierSatisfaction	70 %	90 %	80 %	10 %	30 %	40 %	50 %	60 %	20 %	50 %	500 %
CarrierPaymentSpeed	80 %	100 %	40 %	60 %	20 %	30 %	-20 %	10 %	0 %	70 %	390 %
ShipperRetention	60 %	20 %	90 %	20 %	40 %	50 %	70 %	50 %	30 %	60 %	490 %
ShipperSatisfaction	70 %	30 %	90 %	10 %	30 %	60 %	60 %	50 %	40 %	50 %	490 %
EmployeeRetention	50 %	10 %	30 %	30 %	20 %	90 %	40 %	30 %	50 %	40 %	390 %
EmployeeSatisfaction	60 %	20 %	40 %	40 %	30 %	100 %	50 %	50 %	60 %	50 %	500 %
DebtReduction	30 %	-20 %	50 %	100 %	30 %	40 %	60 %	20 %	10 %	80 %	400 %
CreditRating	40 %	30 %	50 %	90 %	60 %	30 %	50 %	40 %	30 %	70 %	490 %
ComplianceRate	80 %	40 %	30 %	20 %	100 %	60 %	30 %	70 %	40 %	60 %	530 %
SystemModernization	100 %	60 %	70 %	30 %	50 %	40 %	60 %	30 %	20 %	80 %	540 %
Total Impact	640 %	380 %	570 %	410 %	410 %	540 %	450 %	410 %	300 %	610 %	4720 %



This IET Evaluation is an integral part of the ‘Strategy’ or ‘architecture’ part of the Evo Cycle

2.4.3 *Decomposition* of an Effective Strategy

So that we identify sub-strategies which can deliver results, soon, and frequently: 'Evo'

Based on the search results and the given query, I'll decompose System Modernization into 10 effective, independently implementable sub-strategies and provide a brief description for each. Then, I'll construct an Impact Evaluation Table (IET) using these sub-strategies and the provided values.

1. API Integration: Implement APIs to connect legacy systems with modern applications, enabling better data flow and interoperability. This approach allows for gradual modernization without completely replacing existing systems[1].

2. Cloud Migration (Rehosting): Move existing applications to cloud infrastructure without significant changes to the code, often called "lift and shift." This strategy provides quick wins in terms of scalability and cost savings[2].

3. Refactoring: Optimize existing code for better performance in the cloud environment without changing its external behavior. This improves application efficiency and maintainability while preserving core functionality[2][3].

4. Microservices Adoption: Break down monolithic applications into smaller, independently deployable services. This architecture enhances scalability, flexibility, and ease of maintenance[3].

5. Container Adoption: Implement containerization technologies like Docker to package applications and their dependencies, improving portability and consistency across different environments[2].

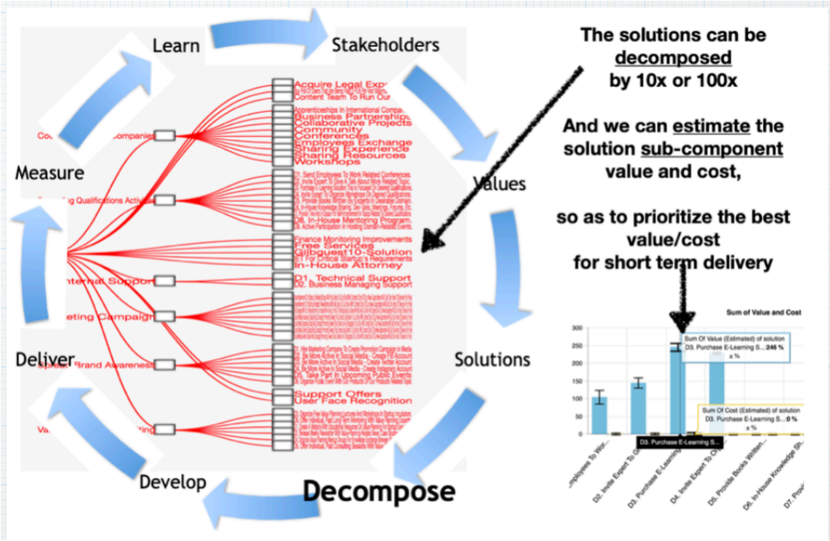
6. Serverless Computing: Transition suitable components to serverless architectures, reducing infrastructure management overhead and improving scalability[3].

7. Data Modernization: Upgrade databases and data storage systems to modern, cloud-native solutions, enhancing data accessibility, security, and analytics capabilities[4].

8. DevOps Implementation: Adopt DevOps practices and tools to streamline development, testing, and deployment processes, improving collaboration and reducing time-to-market[3].

9. Legacy Application Replacement: Identify and replace outdated applications with modern, off-the-shelf solutions or custom-built alternatives that better meet current business needs[4][5].

10. Continuous Integration/Continuous Deployment (CI/CD): Implement automated CI/CD pipelines to facilitate frequent, reliable software updates and reduce manual intervention in the deployment process[3].

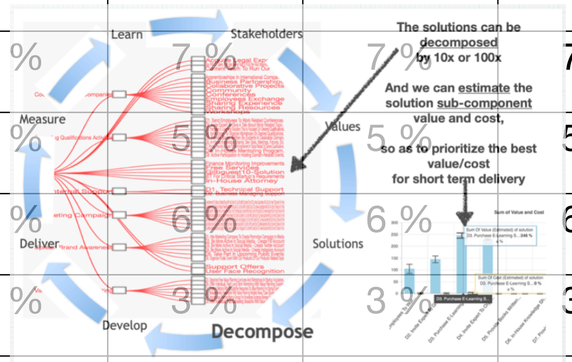


Citations:
[1] https://www.linkedin.com/pulse/6-key-strategies-legacy-systems-modernization-koukiali-mba
[2] https://stratusgrid.com/blog/rs-of-application-modernization
[3] https://www.redhat.com/en/blog/modenization-strategies-modernizing-existing-code
[4] https://www.trianz.com/insights/application-modernization-strategies-7-rs-of-transformation
[5] https://guidehouse.com/insights/advanced-solutions/2024/setting-a-strategy-for-successful-technology-modernization

2.4.4. How good are the sub-strategies at meeting our Wish-level of the Value Objective?

100% means that the strategy would reach the Wish level

Values (below) / Sub-Strategies ->	API Integration	Cloud Migration	Refactoring	Microservices Adoption	Container Adoption	Serverless Computing	Data Modernization	DevOps Implementation	Legacy App Replacement	CI/CD Implementation	Total Effectiveness
CarrierSatisfaction	7 %	7 %	7 %	7 %	7 %	7 %	7 %	7 %	7 %	7 %	70 %
CarrierPaymentSpeed	8 %	8 %	8 %	8 %	8 %	8 %	8 %	8 %	8 %	8 %	80 %
ShipperRetention	6 %	6 %	6 %	6 %	6 %	6 %	6 %	6 %	6 %	6 %	60 %
ShipperSatisfaction	7 %	7 %	7 %	7 %	7 %	7 %	7 %	7 %	7 %	7 %	70 %
EmployeeRetention	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	5 %	50 %
EmployeeSatisfaction	6 %	6 %	6 %	6 %	6 %	6 %	6 %	6 %	6 %	6 %	60 %
DebtReduction	3 %	3 %	3 %	3 %	3 %	3 %	3 %	3 %	3 %	3 %	30 %
CreditRating	4 %	4 %	4 %	4 %	4 %	4 %	4 %	4 %	4 %	4 %	40 %
ComplianceRate	8 %	8 %	8 %	8 %	8 %	8 %	8 %	8 %	8 %	8 %	80 %
SystemModernization	10 %	10 %	10 %	10 %	10 %	10 %	10 %	10 %	10 %	10 %	100 %
Total Impact	64 %	64 %	64 %	64 %	64 %	64 %	64 %	64 %	64 %	64 %	640 %



This is part of the 'Decomposition' Planning



Decompose
Estimate
Prioritize

2.5.1 ‘Develop’ : Plan example

Doing everything necessary for successful sub-strategy delivery

Based on the search results and the query, here are 10 activities or tasks needed to develop Digital Transformation r later delivery to the live system to achieve measurable results:

1. Perform a Digital Readiness Assessment: Evaluate the current state of your organization's digital capabilities, infra and processes. This will help identify gaps and prioritize areas for improvement[3].

2. Define Clear Transformation Goals: Establish specific, measurable objectives for your digital transformation initiat aligning them with overall business strategy[4].

3. Create a Detailed Roadmap: Develop a step-by-step plan that outlines the journey from the current state to the de future, including milestones and timelines[3].

4. Identify and Prioritize Key Projects: Select strategic initiatives and quick-win projects that align with organizationa can demonstrate value early in the transformation process[4].

5. Develop a Skills Acquisition Plan: Outline strategies for upskilling existing employees and recruiting new talent to digital transformation efforts[4].

6. Establish Cross-functional Teams: Form dedicated teams that bring together diverse skill sets and perspectives to transformation projects[2].

7. Implement Data Analytics Infrastructure: Set up systems and processes to collect, analyze, and leverage data for i decision-making and improved customer insights[3].

8. Develop API Integration Strategy: Plan for the integration of legacy systems with modern applications through API enhance data flow and interoperability[1].

9. Design and Implement Pilot Projects: Start with small-scale implementations to test new technologies, processes, and approaches before full-scale rollout[1].

10. Establish Key Performance Indicators (KPIs): Define clear metrics to measure the success and impact of digital transformation initiatives, including both tangible and intangible goals[4].

These activities will help prepare your organization for digital transformation, ensuring a structured approach that can deliver measurable results when implemented in the live system. Each task contributes to building a solid foundation for transformation, from assessing current capabilities to measuring outcomes.

Citations:
[1] https://www.thebusinessdesk.com/eastmidlands/news/2077085-digital-transformation-10-ideas-to-begin-your-journey
[2] https://chazeypartners.com/article/six-things-businesses-can-do-to-enable-digital-transformation/
[3] https://www.techtarget.com/searchcio/tip/How-to-build-a-digital-transformation-roadmap-in-6-steps
[4] https://www.forbes.com/sites/bernardmarr/2023/08/22/9-vital-steps-to-create-a-digital-transformation-strategy/
[5] https://www.apty.io/blog/7-best-practices-for-a-successful-digital-transformation/

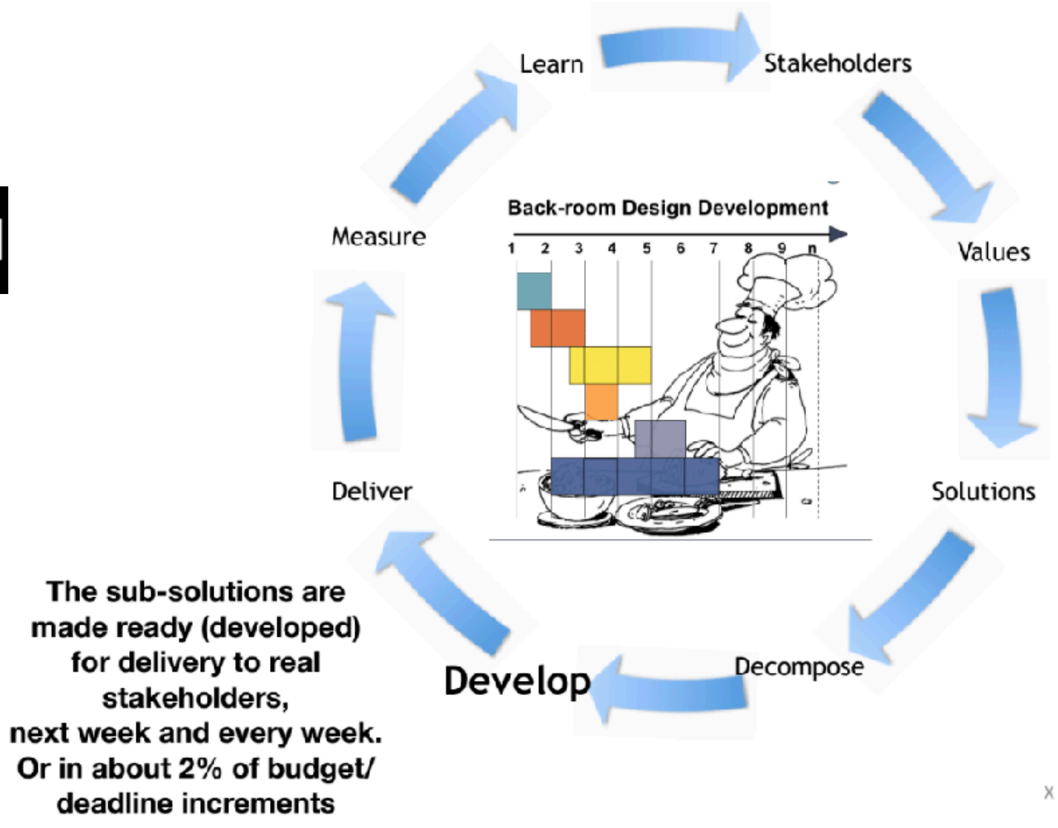


Figure: 4.5 The Gilb Evo 'Value Delivery' Cycle

2.6.1 ‘Deliver’ : Plan

Tasks to plan for delivering the developed sub-solution to the system

Critical tasks to Deliver Digital Transformation to the live system:

1. Implement API Integration: Connect legacy systems with modern applications to enable better data flow and interoperability.

2. Execute Cloud Migration: Move existing applications to cloud infrastructure using a "lift and shift" approach.

3. Refactor Key Applications: Optimize existing code for better performance in the cloud environment.

4. Adopt Microservices Architecture: Break down monolithic applications into smaller, independently deployable services.

5. Deploy Containerization: Implement containerization technologies like Docker for improved portability and consistency.

6. Transition to Serverless Computing: Move suitable components to serverless architectures to reduce infrastructure management overhead.

7. Modernize Data Systems: Upgrade databases and data storage systems to cloud-native solutions.

8. Implement DevOps Practices: Adopt DevOps tools and methodologies to streamline development and deployment processes.

9. Replace Legacy Applications: Identify and replace outdated applications with modern alternatives.

10. Set Up CI/CD Pipelines: Implement automated CI/CD pipelines for frequent, reliable software updates.

11. Conduct User Training: Provide comprehensive training to employees on new systems and processes.

12. Perform Security Audits: Ensure all new systems and integrations meet security standards.

The sub-solutions are *delivered* to real stakeholders, in order to experiment, to test, to pilot, to get reactions, NUMERICALLY and to allow for potential corrections in design, in implementation process, and in lower-priority requirements

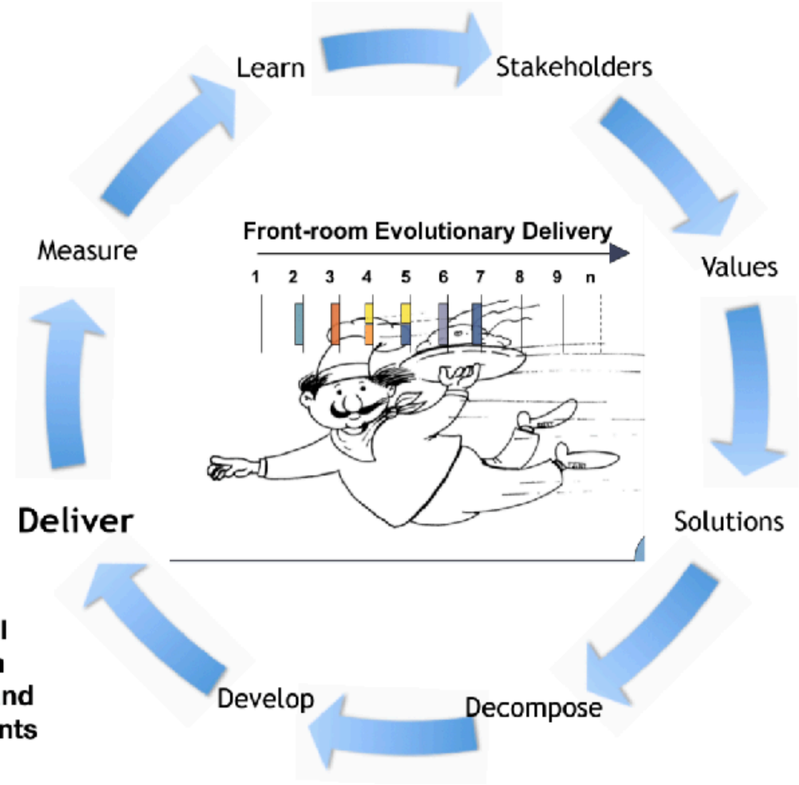
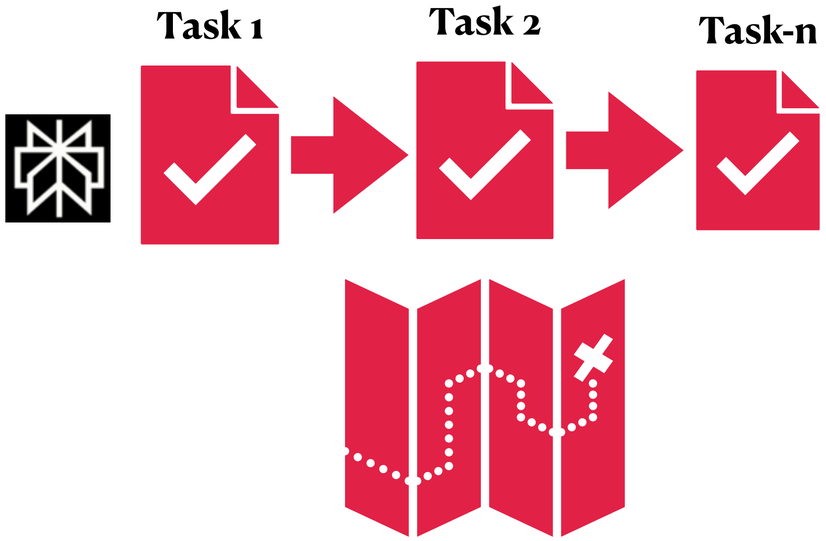


Figure: 4.6. The Gilb Evo 'Value Delivery' Cycle



2.7.1. ‘Measure’: Planning

Planned steps to get feedback and measurability from the sub-strategy delivery

Tasks for Measuring Results (on 10 Objectives):

1. Carrier Satisfaction: Conduct regular surveys and analyze feedback from carriers.

2. Carrier Payment Speed: Track and compare payment processing times before and after implementation.

3. Shipper Retention: Monitor shipper churn rates and compare to pre-transformation baseline.

4. Shipper Satisfaction: Implement Net Promoter Score (NPS) surveys and analyze trends.

5. Employee Retention: Track employee turnover rates and compare to industry benchmarks.

6. Employee Satisfaction: Conduct regular employee engagement surveys and analyze results.

7. Debt Reduction: Monitor debt levels and track progress towards reduction goals.

8. Credit Rating: Track changes in credit rating from financial institutions.

9. Compliance Rate: Measure adherence to regulatory requirements and internal policies.

10. System Modernization: Assess the percentage of systems modernized and their performance improvements.

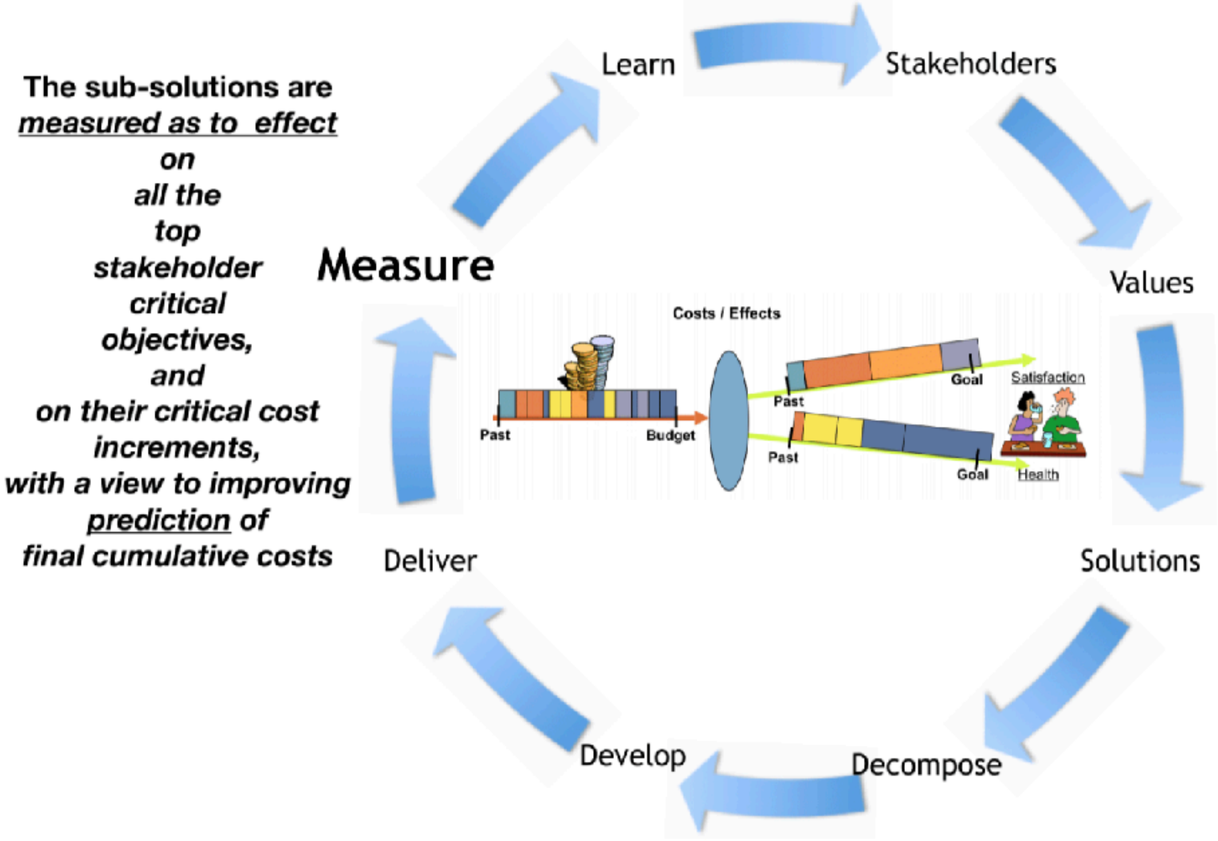
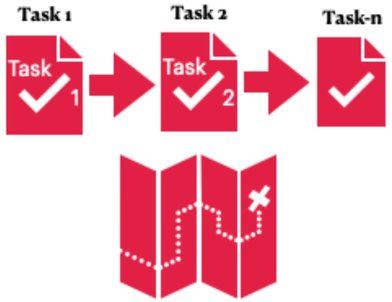


Figure: 4.7 The Gilb Evo ‘Value Delivery’ Cycle. Courtesy Kai Gilb.



This is a task list, (of 10 different measure processes, Meters) not an Evo value delivery step

2.7.2 Principles of Evo Measurement (2007 TG)

Principles of Evo Measurement

1. Perfect accuracy is Not necessary. Keep it simple – but get feedback enough to get confidence that you are moving forward

2. order of magnitude accuracy is usually enough

3. let the measurement be done by domain experts

4. *design* the initial measures to fit within the week cycle (design to cost)

5. you can refine measurement accuracy in the backroom (after evo step delivery)

6. sampling is a good way to reduce the cost of measurement

7. when final indicators would take too long, use early indicators

8. if necessary use multiple parallel measuring processes, to increase confidence

9. final acceptance test measures can differ from (more costly, more confidence) initial weekly cycle measures

10. measures should as far as possible reflect the stakeholder results

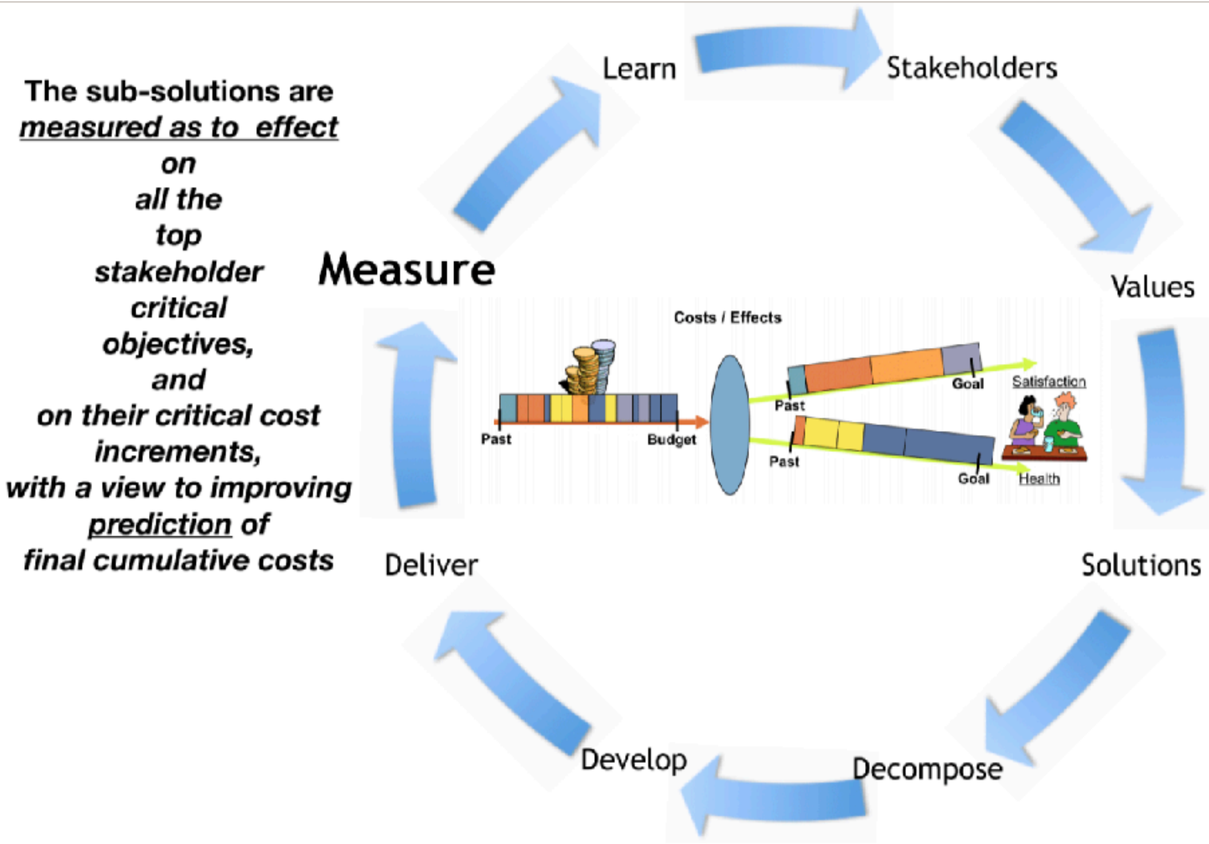


Figure: 4.7 The Gilb Evo 'Value Delivery' Cycle. Courtesy Kai Gilb.

2.8.1 Learn (and improve)

Improve quickly, based on facts

Steps for Learning from Feedback:

1. Establish Feedback Channels: Create multiple avenues for stakeholders to provide feedback (e.g., surveys, focus groups, suggestion boxes).

2. Collect Data Systematically: Implement tools and processes to gather feedback data consistently.

3. Analyze Feedback: Use data analytics tools to identify patterns and trends in the feedback.

4. Prioritize Issues: Rank identified problems based on impact and frequency.

5. Develop Action Plans: Create specific, measurable plans to address prioritized issues.

6. Implement Changes: Execute the action plans and track progress.

7. Communicate Results: Share findings and actions taken with stakeholders to close the feedback loop.

8. Measure Impact: Assess the effectiveness of implemented changes through follow-up surveys or metrics.

9. Iterate and Refine: Continuously adjust strategies based on new feedback and measured results.

10. Foster a Learning Culture: Encourage open discussion of successes and failures to promote organizational learning.These tasks and steps provide a comprehensive approach to delivering Digital Transformation, measuring its impact, and continuously improving based on feedback.

Citations:
[1] https://criticalsoftware.com/en/expertise/digital-transformation
[2] https://chazeypartners.com/article/six-things-businesses-can-do-to-enable-digital-transformation/
[3] https://www.unitygroup.com/blog/10-critical-digital-transformation-challenges-to-overcome/
[4] https://whatfix.com/blog/digital-transformation-challenges/
[5] https://businessmap.io/digital-transformation/examples

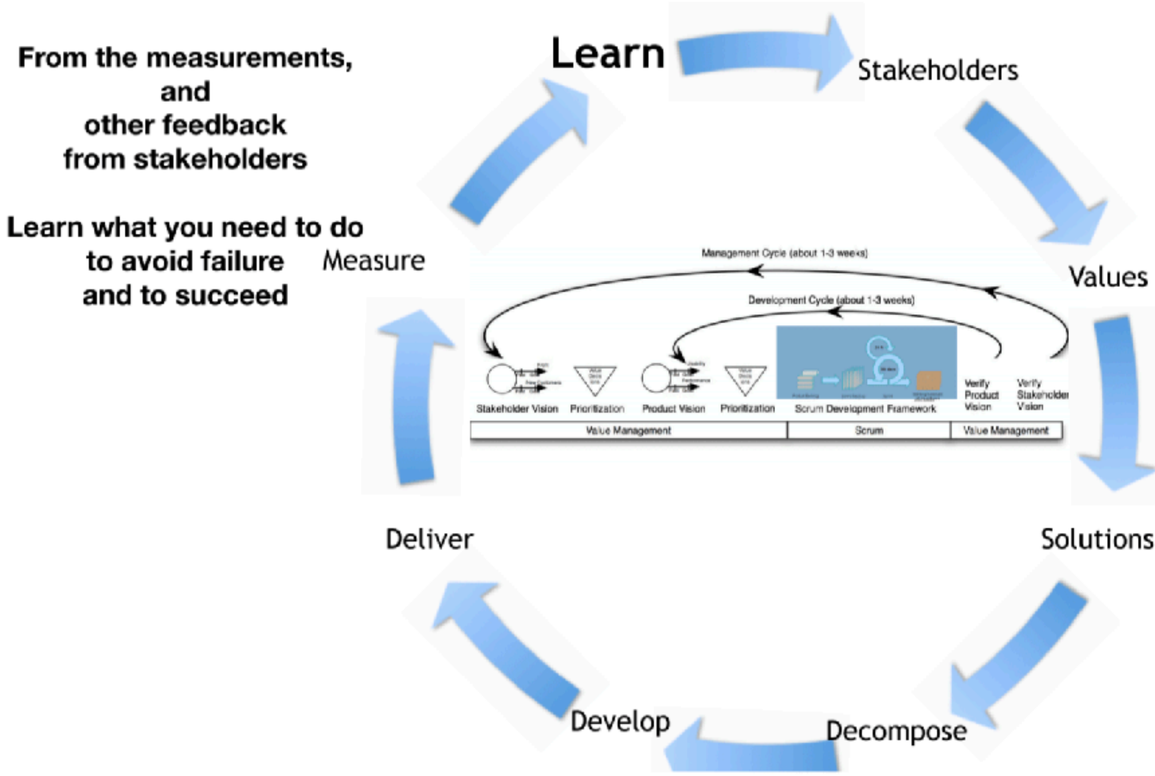
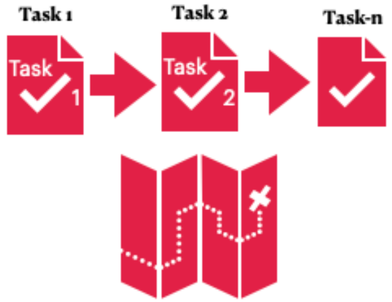


Figure: 4.8. The Gilb Evo 'Value Delivery' Cycle . Original diagrams Kai Gilb.



This is a task list, (of 10 Learning sub-processes) not an Evo value delivery step

2.9 Evolving from an Existing System

Should you start from an existing system, or from scratch with a ‘clean slate’ ?

If at all possible, your early Evo steps should be taken on an existing system. Notice I did not say YOUR existing system, because maybe you re a new startup and there is no history. But you can probably find something out there, an existing product or service that you can use as a base.

I think it is tempting to ignore the awful old system. But I do not think that is a good idea. There are many good reasons for this. But they vary a great deal, according to your exact circumstances.

1. The Base System might already have a ‘market’, they just need to convert to your higher value product or service.

2. You can get going much earlier with results, because the Base System already exists, You just have to make each increment improve values and costs. You do not have to build the base system first. That is non-productive- You need to focus on value improvement, values first, not construction of a framework, that already exists! You cannot get high value prioritization when there is no incremental value.

3. This approach forces the developers to recognize they are dealing with a multi-fated live system, with al stakeholders, and experience of the base system ‘culture’.

4. You can focus on improvements, not exiting base systems.

5. Be aware that there are several forces of evil that will try to get you assume you have to build a new system, at great cost. These are the usuall suspected consultants, and hardware suppliers. Sometimes even internal people who want a big budget, and the power that come with it. They will make plausible arguments. Some of them have decades of lying experience, and decades of false credibility. Get yourself consultants who will seriously discuss short term continuous improvements. [**Twelve Tough**] .

6. Of course, in the long term, you will maybe replace all hardware and software, and production supply chain, and organization: nBut, why not get early proven results for 5 years, and avoid a big failure project blackening your personal reputation and career?

7. I have analyzed a large number of big failure projects [Akson] and every one of them has exactly this big bang problem. My Evoethod has no such problem! The reason is that if you encounter one or more steps that cannot do what you plan of value and costs, you stop and fix it, or you stop. But there is still a working system in place to revert to, maybe with 50 weeks steps of improvement behind it, that first year. I remind your of [Flyvbjerg] 16,000 projects, 99.5% failure . Obviously the traditional project management methods are not working!

3. Efficient Planning

A conceptual framework for Evo

Designs —-> Values/Costs = Efficiency

Where Scope is a ‘given’ constraint

Imagine a system possessing all required ‘scope’ (functionality, within binary constraints)

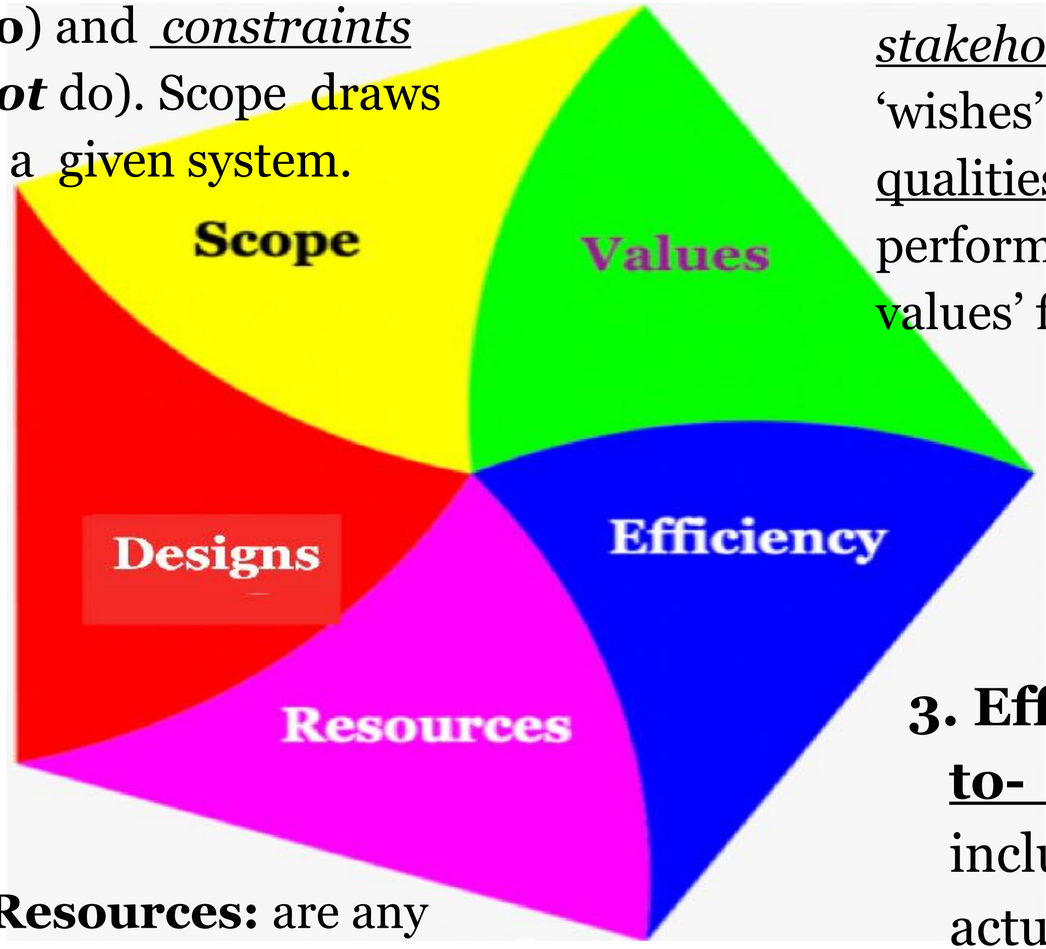
And 0% Availability, and $ 1,000,000,000,000 costs

Now you can start the Evo process of reducing costs, and improving Values, like availability

1. **Scope**: is the specified set of stakeholder- and system- *functions* (what it must **do**) and *constraints* (what it must ***not*** do). Scope draws a border around a given system.

2. **Values**: is the specified set of *stakeholder* values ('wants', 'needs', 'wishes', 'visions') and *system* qualities, including system performance attributes ('*potential* values' for stakeholders).

5. Designs: are any types of 'implementable ideas' (designs, strategies, architecture, solutions) which we use, in order to deliver Values, within Constraints.



1.3.1 The 'Penta' defines 5 critical factors we use in Evo. Many project management systems fail to recognize their importance to project success. They focus instead on function and costs, but forget Values, Efficiency, and Design

4. Resources: are any *critical* and *prioritized*, set of *limited* resources for the system lifecycle, such as time, money, people, space.

3. Efficiency: is '**effectiveness-to- costs ratio**'. *Effectiveness* includes all stakeholder-values actually *delivered*. The costs are life-cycle costs, not just 'capital' costs. This is a view outside the black box of Designs.

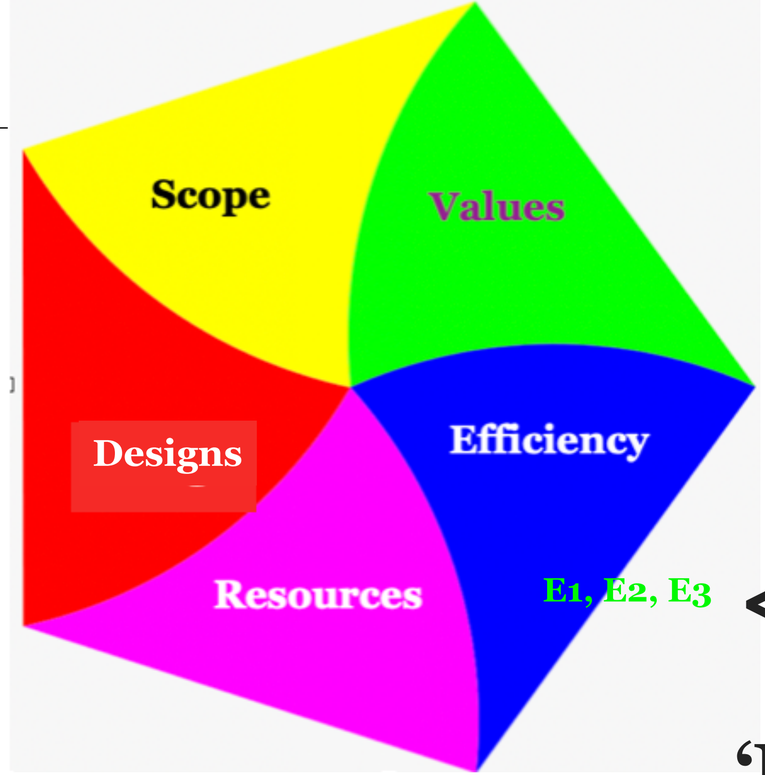
© Tom Gilb 2022

1.3.2. Level 2 Penta: Efficiency = Values/Costs

E1: **Generic Defined Efficiency**. Any selected critical set of Stakeholder Values divided by any selected, useful, or available, critical set of budgeted constrained resources. {V1, V2, Vn}/{R1, R2, Rn}.

E2: **Profitability**: any defined notions of profitability. Example before taxes, long-term average.

E3: **Value for Money**: useful for government-like institutions, charities, ideal organizations. 'Citizen/Stakeholder Values' for 'financial expenditure', like taxes, donations. For system lifecycle or selected stages or aspects.



'Efficiency'

Efficiency is any selected useful ratio of 'good stuff' in relation to 'their costs'

Components

1.3.2 'Values' alone are worthless, if their costs are unaffordable by the stakeholders, or if costs exceed the values. So we need to get in the habit of thinking, estimating, measuring, the 'values/costs ratio'. The efficiency, of the emerging new system (values and costs). Evo includes very specific tools *(quantified values, Impact Estimation Tables, Efficiency Prioritization rules)* *for helping us consider efficiency early, and incrementally*

1.3.3 Efficiency: Good project Values without wasting limited resources

Balancing the Good, with their Costs.

- Our focus is not on 'maximizing' values, at 'any costs'.
- Our focus is on delivering, 'satisficing', a desired level of values, *without using unnecessary resources* to do so.
- This is '**balance**'. [Optima]
- The primary priority is to reach planned levels, targets, Goals of **values**. [Value Management]
  - Then *secondarily*, to reduce the **resources** needed, to continuously deliver planned value levels. *'Resource reduction'* [Cost Engineering]
- The **Values/Resources ratio,** which we can quantify in practice, using an Impact Estimation Table [CE, VIET] is how we will measure the 'efficiency' of our projects and processes.
- Evo is constantly, and continuously, monitoring this 'efficiency'. [Success] in the [Flyvbjerg] sense (on time, on budget, benefits delivered) is not an accident in Evo, it is designed into the Evo process, using Planguage tools.

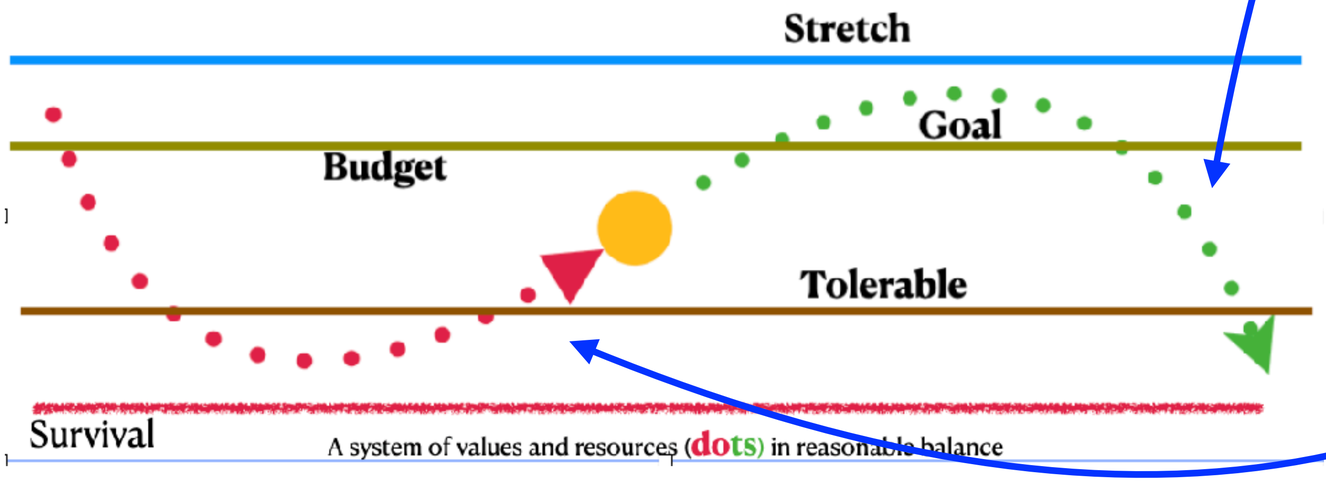
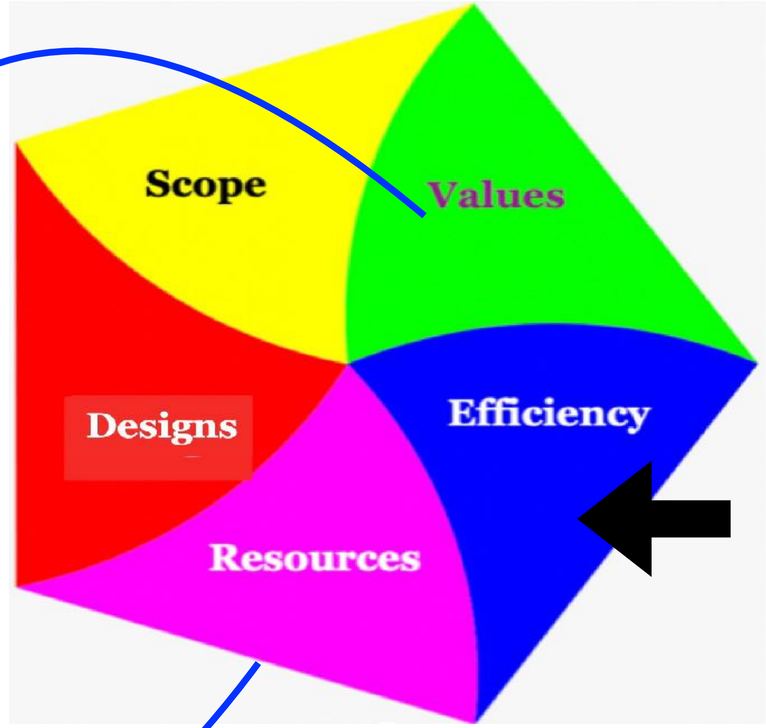


Figure 1.3.3 Balancing multiple (the dots) values (green dots) and costs (red dots) see [Optima] book for detail on balancing multiple factors .



[Penta]



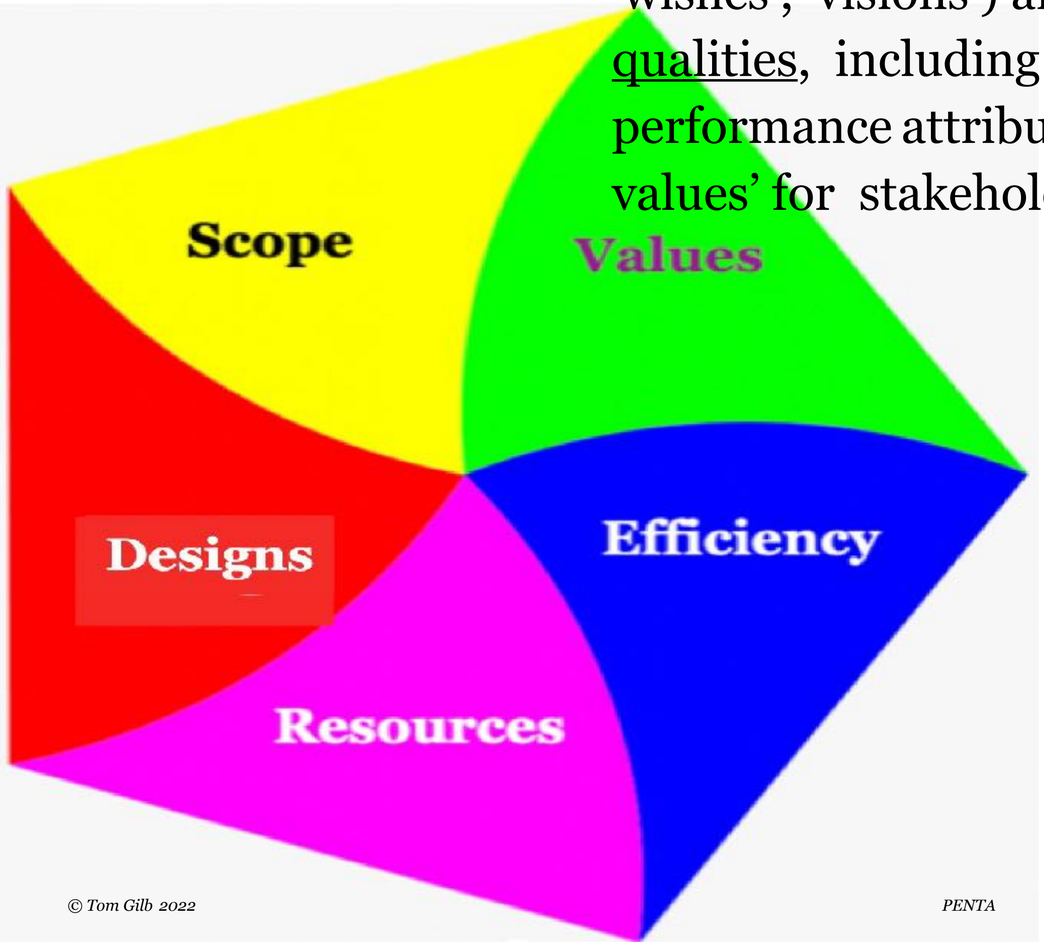
4. Values Viability First

Stakeholder Values First [Value First]

Improving Stakeholder Needs [Stakeholder Engineering]

- ***Values***: the word 'Value' can easily trigger a notion of *financial* value.
- But *we* use it to describe any **stakeholder** values.
- And in particular, in this book, we are concerned with stakeholder 'values' of *any* kind
  - We are, for example, interested in values that help the less-advantaged, less wealthy humans. [Startup]
    - For example in health, education, food, climate, children, families, war-affected, and economics.

Values: is the specified set of *stakeholder* values ('wants', 'needs', 'wishes', 'visions') and *system* qualities, including system performance attributes ('*potential* values' for stakeholders).

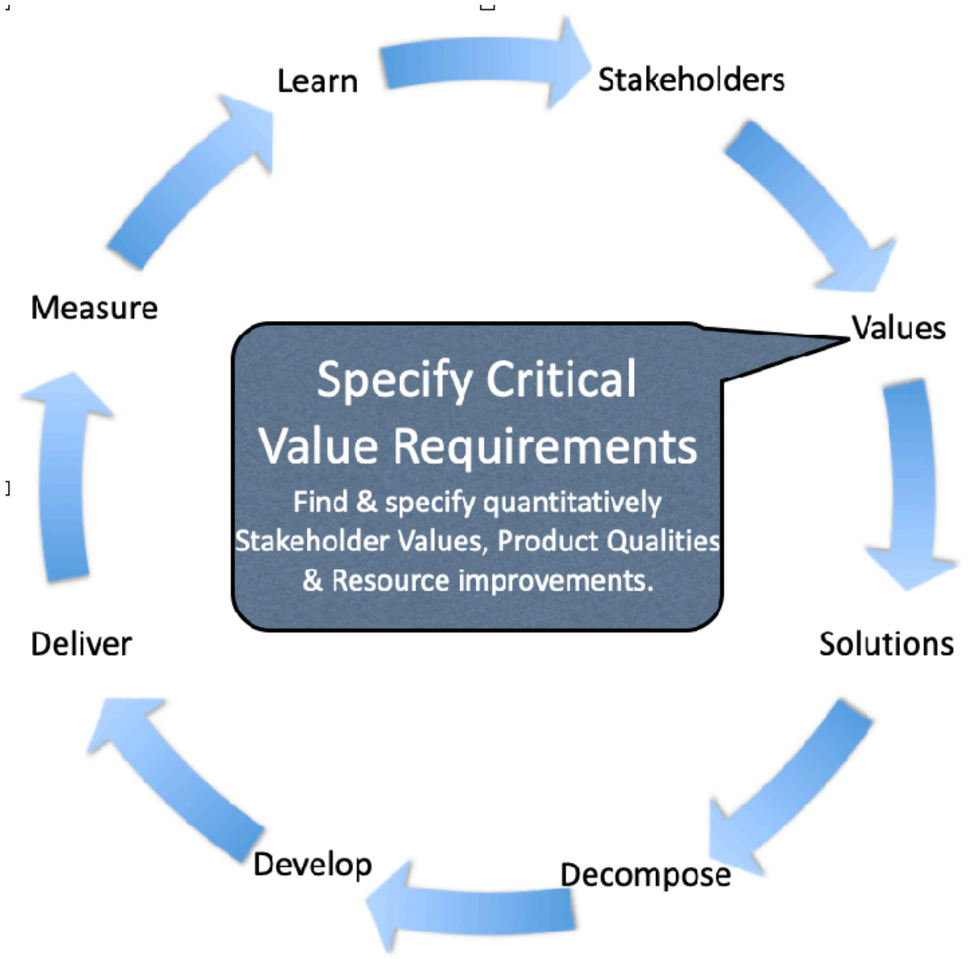


© Tom Gilb 2022

Values Via the Evo Process

One or more *value increments* are planned, and (hopefully) delivered, for *every* Evo Cycle.

- This book will focus on specific methods for planning value, and for delivering value. Evolutionarily.
- The methods are 'Planguage' [CE] and one instrument that ties Planguage ('**Plan**ning Lan**guage**') together is the 'Evo-cycle'.
- 'Evo' is a concatenation of the longer word '**Evo**lutionary'.
  - Sometimes it is an acronym: *Evolutionary Value Optimization*.
  - *'Evo' was widely used by my client [HP], who adopted corporate-wide.*
  - *Evolutionary is widely used by the US [DoD] in standards.*
    - https://ac.cto.mil/wp-content/uploads/2022/02/Systems-Eng-Guidebook_Feb2022-Cleared-slp.pdf *(2022)*
- 'Evolutionary' is formally defined, by me, as a step-by-step *value delivery* to a system, with *feedback* and *adjustment*, to *maintain progress* towards long-term planned values, within planned resources. This is unchanged since 1960-76 [SM]
  - This is quite different from:
    - Iterative (tic, toc, 0:1): no progress.
    - Incremental (1, 2 , 3 , 4, n): No feedback and change.



The EVO Process, the One-Page Summary

A Systems Engineering Process For incremental value delivery management

A *PLANNING* CYCLE

Clarify your environment: critical-stakeholders' territory

1. **STAKEHOLDERS:** Identify *critical stakeholder's* and their critical **values**
2. **VALUES: Quantify** and **clarify** y*our project's* critical values: what *degree* of values do you expect the design to deliver, to which stakeholders
3. Identify design **constraints**: legality, political, cultural, policy, other plans
4. Identify design **resource-limitation**s: time, money, operational costs for example.

Design (Solutions): Architecture Level

6. **SOLUTIONS:** use the many Value and Resource requirements, AS EXPLICIT TOOLS, to find solutions, to deliver Values, within constraints. Aka Designs, Strategies, Architecture
7. **DECOMPOSE:** Decomposition of top-level architecture into design components
8. **PRIORITY:** Identification, *detailed* specification and *prioritization* of top-level architecture, and of any *decomposed* design solutions..

A *VALUE DELIVERY* CYCLE: An 'Evo-step'.

9. The **Evo Value delivery steps** (for example, a week, or 2% of total project budget)
   1. Select your highest **priority value**, and the most-critical scale-parameter attributes (*the who, what, where, when - slices of reality*).
   2. **DEVELOP**: Find a design component which will deliver good **value-for-resources** to your critical priority requirements.
   3. Ready the design-component for delivery: for ***integration*** **to the existing system.**
   4. **DELIVER: Deliver** the design component to the **real system**
   5. **MEASURE: Measure** the results (values and costs) of the design increment
   6. **Dynamic-Design-To-Requirements:** If results are **not OK**, attempt design **improvement, and redeliver.**
   7. **KEEP CYCLING:** If results *are* **OK,** then **repeat** this 'value delivery cycle', scale up, until 'done'.

Project Completion: All Value is delivered

10. **DONE?**: When **all value requirements are reached**, or when all **critical resources are used up**. Stop.



Gilb Value Cycle ('EVO')

Copyright 2020 kai@Gilb.com



Examples of Values

Learn

Stakeholders

Values

Solutions

Decompose & Prioritize

Develop

Deliver

Measure

Specify Critical Value Requirements

Find & specify quantitatively Stakeholder Values, Product Qualities & Resource improvements.

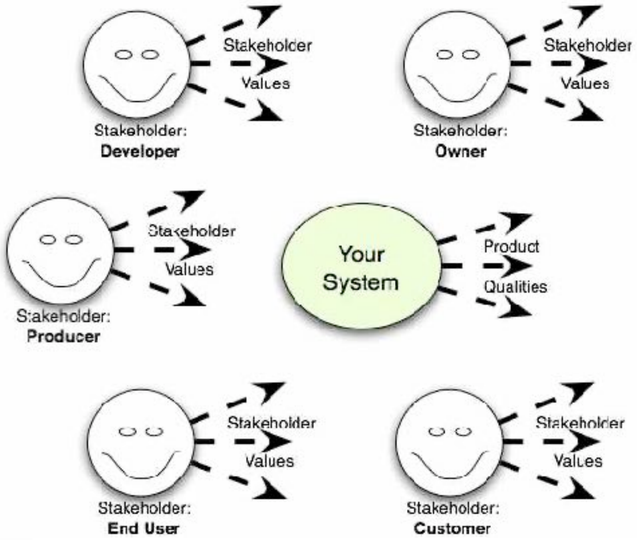
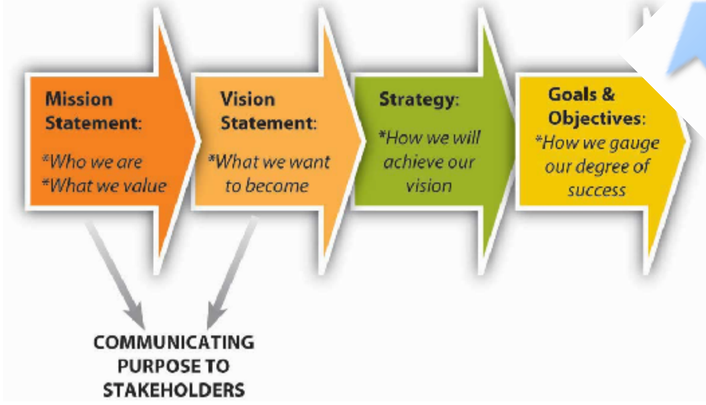
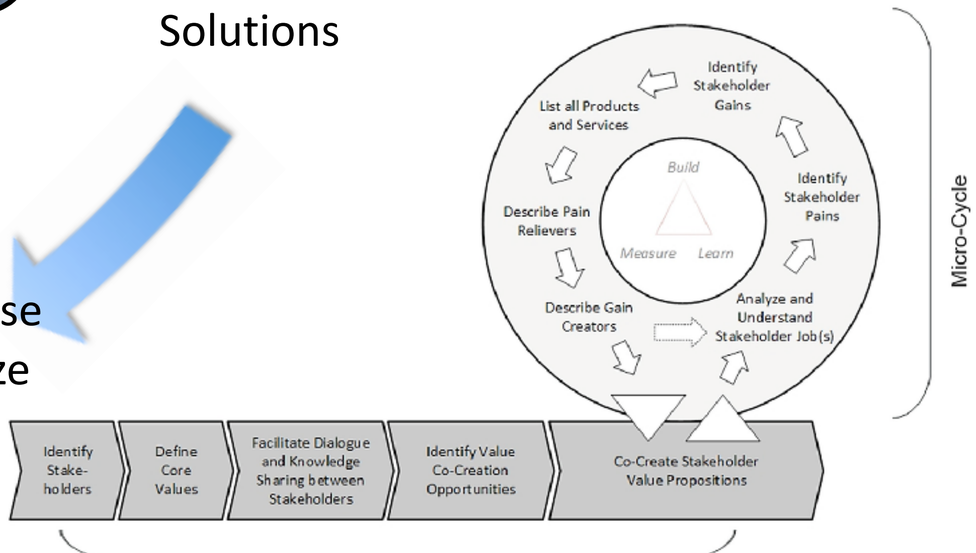


Figure 4.5 *Key Roles of Mission and Vision*





20 Examples of Social Values <- Perplexity

Here are some examples of specifying, and defining Social Values, in Planguage

Here is a list of 20 critical social value objectives for the poor and underprivileged people in our country, presented in Planguage format:

1. Education Access

Ambition: Increase access to quality education for underprivileged children

Scale: Percentage of underprivileged children enrolled in school

Meter: Annual school enrollment rates

Goal: 95% enrollment rate within 5 years

Stakeholders: Children, parents, educators, government

2. Healthcare Coverage

Ambition: Expand affordable healthcare coverage for low-income families

Scale: Percentage of low-income population with health insurance

Meter: Health insurance enrollment data

Goal: 90% coverage within 3 years

Stakeholders: Low-income families, healthcare providers, insurers

3. Job Creation

Ambition: Generate employment opportunities in economically disadvantaged areas

Scale: Number of new jobs created

Meter: Employment statistics in target areas

Goal: Create 100,000 new jobs annually

Stakeholders: Unemployed individuals, local businesses, government

4. Affordable Hou

Ambition: In
of afforda

Scale: N
affor

Me
re

Scal
popul
insecur

Meter: Food

Goal: Reduce fo
by 50% within 5 yea

Example

11. Child Nutrition

Ambition: Improve nutritional status of children in poverty-stricken areas

Scale: Percentage reduction in child malnutrition rates

Meter: Child health surveys

Goal: Reduce child malnutrition by 60% within 4 years

Stakeholders: Children, parents, nutritionists, healthcare providers

ecure

Meter: Banking sector data

Goal: 95% financial inclusion within 4 years

Stakeholders: Unbanked individuals, banks, fintech companies

ean Water Access

tion: Provide clean
ng water to underserved
unities

ercentage of
tion with access to
ater

Water infrastructure

100% clean water access
in 5 years

keholders: Rural
ommunities, water
authorities, NGOs

9. Skill Development

Ambition: Enhance employability through

5 of 20 Examples of Social Values <- Perplexity

Here is a list of 5 more critical social value objectives for the poor and underprivileged people in our country, presented in Planguage format:

1. Education Access

Ambition: Increase access to quality education for underprivileged children

Scale: Percentage of underprivileged children enrolled in school

Meter: Annual school enrollment rates

Goal: 95% enrollment rate within 5 years

Stakeholders: Children, parents, educators, government

2. Healthcare Coverage

Ambition: Expand affordable healthcare coverage for low-income families

Scale: Percentage of low-income population with health insurance

Meter: Health insurance enrollment data

Goal: 90% coverage within 3 years

Stakeholders: Low-income families, healthcare providers, insurers

3. Job Creation

Ambition: Generate employment opportunities in economically disadvantaged areas

Scale: Number of new jobs created

Meter: Employment statistics in target areas

Goal: Create 100,000 new jobs annually

Stakeholders: Unemployed individuals, local businesses, government

4. Affordable Housing

Ambition: Increase availability of affordable housing units

Scale: Number of new affordable housing units built

Meter: Housing development records

Goal: Construct 50,000 new units per year

Stakeholders: Low-income families, housing developers, local authorities

5. Food Security

Ambition: Reduce hunger and malnutrition among vulnerable populations

Scale: Percentage of population experiencing food insecurity

Meter: Food insecurity surveys

Goal: Reduce food insecurity by 50% within 5 years

Another 5 of 20 Examples of Social Values <- Perplexity

6. Digital Literacy

Ambition: Improve digital skills among underprivileged communities

Scale: Percentage of target population with basic digital skills

Meter: Digital literacy assessments

Goal: 80% digital literacy rate within 3 years

Stakeholders: Underprivileged individuals, tech companies, educators

7. Financial Inclusion

Ambition: Expand access to banking and financial services for the unbanked

Scale: Percentage of adults with a bank account

Meter: Banking sector data

Goal: 95% financial inclusion within 4 years

Stakeholders: Unbanked individuals, banks, fintech companies

8. Clean Water Access

Ambition: Provide clean drinking water to underserved communities

Scale: Percentage of population with access to clean water

Meter: Water infrastructure data

Goal: 100% clean water access within 5 years

Stakeholders: Rural communities, water authorities, NGOs

9. Skill Development

Ambition: Enhance employability through vocational training programs

Scale: Number of individuals completing vocational training

Meter: Training program completion rates

Goal: Train 500,000 individuals annually

Stakeholders: Unemployed youth, vocational institutes, employers

10. Women's Empowerment

Ambition: Increase economic participation of women from low-income backgrounds

Scale: Percentage of women in formal employment

Meter: Labor force participation data

Goal: 50% increase in women's employment within 5 years

Stakeholders: Women, employers, women's rights organizations

Another 5 of 20 Examples of Social Values <- Perplexity

11. Child Nutrition

Ambition: Improve nutritional status of children in poverty-stricken areas

Scale: Percentage reduction in child malnutrition rates

Meter: Child health surveys

Goal: Reduce child malnutrition by 60% within 4 years

Stakeholders: Children, parents, nutritionists, healthcare providers

12. Mental Health Support

Ambition: Enhance access to mental health services for underprivileged communities

Scale: Number of individuals receiving mental health support

Meter: Mental health service utilization data

Goal: Provide support to 1 million individuals annually

Stakeholders: Individuals with mental health issues, mental health professionals, community organizations

13. Sustainable Livelihoods

Ambition: Promote sustainable income-generating activities in rural areas

Scale: Number of households with improved livelihoods

Meter: Household income surveys

Goal: Improve livelihoods of 1 million households within 5 years

Stakeholders: Rural communities, agricultural experts, microfinance institutions

14. Energy Access

Ambition: Expand access to clean and affordable energy in off-grid areas

Scale: Percentage of rural population with electricity access

Meter: Energy infrastructure data

Goal: 95% rural electrification within 4 years

Stakeholders: Rural communities, energy providers, government

15. Legal Aid

Ambition: Improve access to legal services for low-income individuals

Scale: Number of low-income individuals receiving legal assistance

Meter: Legal aid service records

Goal: Provide legal aid to 500,000 individuals annually

Stakeholders: Low-income individuals, legal professionals, justice system

Another 5 of 20 Examples of Social Values <- Perplexity

Here is a list of 5 more critical social value objectives

16. Public Transportation

Ambition: Enhance affordable public transportation in underserved areas

Scale: Percentage increase in public transport coverage

Meter: Public transportation usage data

Goal: 50% increase in coverage within 3 years

Stakeholders: Commuters, transport authorities, urban planners

17. Disability Inclusion

Ambition: Improve accessibility and opportunities for people with disabilities

Scale: Percentage of public spaces with disability-friendly infrastructure

Meter: Accessibility audits

Goal: 100% accessibility in public spaces within 5 years

Stakeholders: People with disabilities, urban planners, disability rights organizations

18. Elder Care

Ambition: Enhance support systems for low-income elderly individuals

Scale: Number of elderly individuals receiving adequate care

Meter: Elder care service utilization data

Goal: Provide care to 1 million elderly individuals annually

Stakeholders: Elderly population, caregivers, healthcare providers

19. Environmental Justice

Ambition: Reduce environmental hazards in low-income neighborhoods

Scale: Percentage reduction in pollution levels in target areas

Meter: Environmental quality assessments

Goal: 70% reduction in pollution within 5 years

Stakeholders: Residents of affected areas, environmental agencies, local authorities

20. Digital Connectivity

Ambition: Expand internet access in rural and low-income urban areas

Scale: Percentage of population with reliable internet access

Meter: Internet connectivity data

Goal: 95% internet penetration within 4 years

Stakeholders: Underserved communities, telecom companies, government

These social value objectives address various aspects of poverty and under privilege, focusing on improving the quality of life, opportunities, and overall well-being of disadvantaged populations in our country[1][2][5].

Citations:

[1] https://www.fastercapital.com/content/Social-Value--Creating-Social-Value--The-Mission-that-Drives-Social-Enterprises.html

[2] https://socialvaluecompany.com/insights/key-components-of-a-successful-social-value-strategy/

[3] https://ijcrt.org/papers/IJCRT2308108.pdf

[4] https://www.linkedin.com/posts/edward-dunn-_sustainable-procurement-consultant-ref-30180-activity-7165294416465752064-BJ9W

[5] https://www.samtaler.co.uk/social-value-hub/what-is-social-value-model-asking-for

Clarity of Value Objectives

Quantified, and 'structured' into 'Parameters' (like 'Scale: ....')

- From these examples you can see that we do not condone just nice-sounding political phrases (like the Ambition Level parameter is).
- We insist on clarity of our critical purposes.
- That means we *always* quantify it (Scale and Goal)
- And we use other well-defined [CG] parameters to describe it.
- This is a major part Evo, the idea that we base our project work on well-defined objectives
- Poorly-defined objectives are a threat to our success in serving stakeholder needs.

11. Child Nutrition

Ambition: Improve nutritional status of children in poverty-stricken areas

Scale: Percentage reduction in child malnutrition rates

Meter: Child health surveys

Goal: Reduce child malnutrition by 60% within 4 years

Stakeholders: Children, parents, nutritionists, healthcare providers



5.1 Knowing When To Pivot

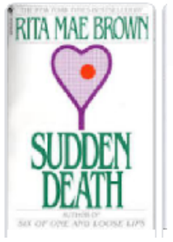
Control over the Value Improvement Effort

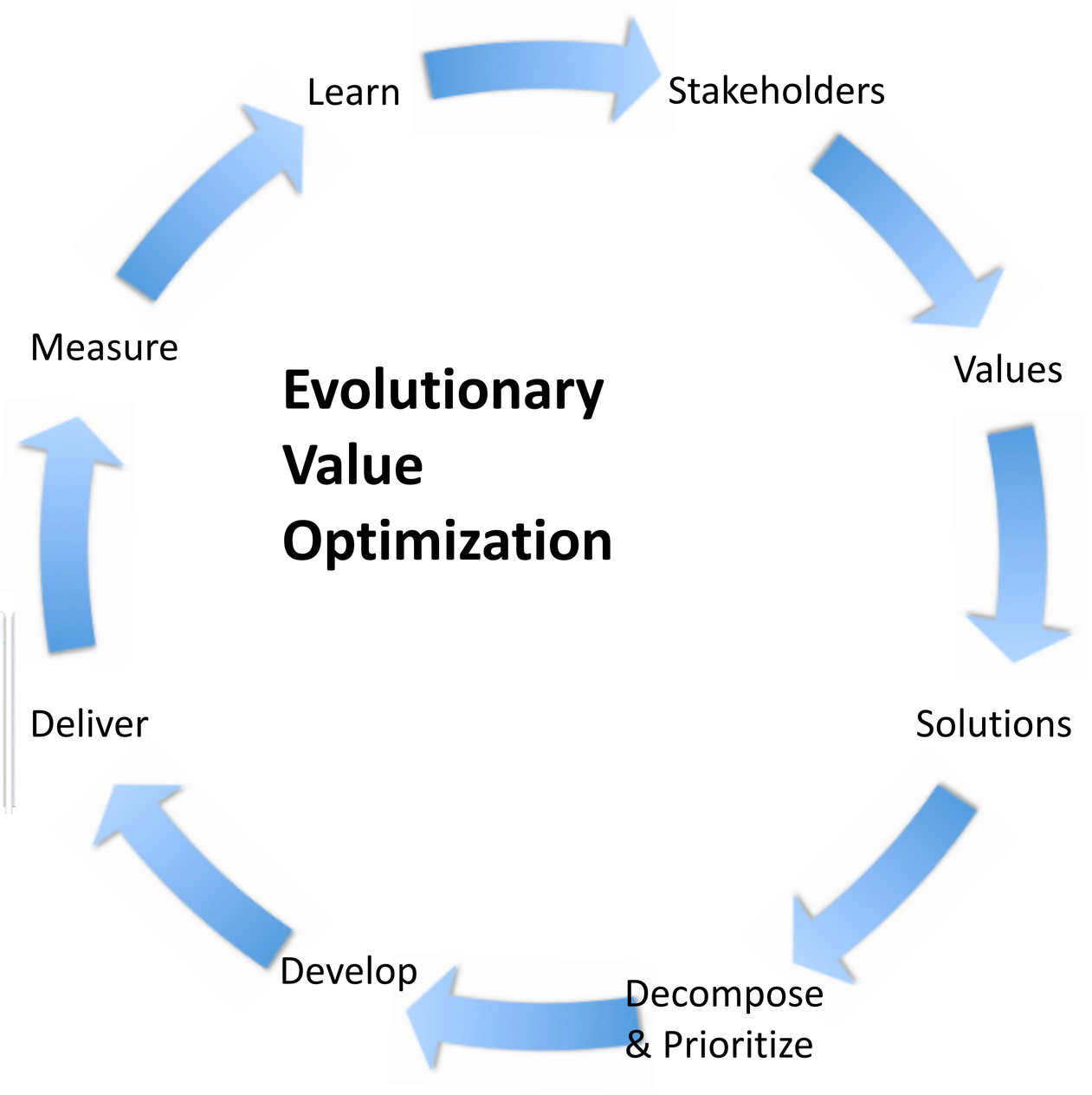
Sensing Problems Early

- The Evo-cycle, plus the 'quantification' of Values, is the key to sound management, of your project.
- The Evo-cycle frequency is the key to *this*. A reasonable cycle for most purposes is a *week*. ± 6 days (or more, it varies)
- That means that we are going to get *some* feedback, some *measures* and complaints, every week, all year long, 50 times a year
- If that feedback is 'negative', we need to do something about it, and try again.
- Like in scientific experiments, and in engineering.
- If the negativity persists (no value, negative values, costs mounting), then, at what point should you give up, and admit you do not know what you are doing?

'Insanity is doing the same thing over and over again and expecting different results.' (R M Brown), 1983

- An Evo-step is not merely a *work* cycle. It is an *implementation* of a strategy or solution in a *real* system. Perhaps under safe conditions. But with feedback from *reality*. And the intent, always, to deliver some value.
  - This is similar, in spirit to, but *very* different from, Lean Startup, asking people in street if they will buy a product that does not exist yet.
- So we cannot waste too many weeks of resources, figuring out that we are wasting our time, and something big has to change (Pivot) or we have to stop wasting resources. [Evo Week]



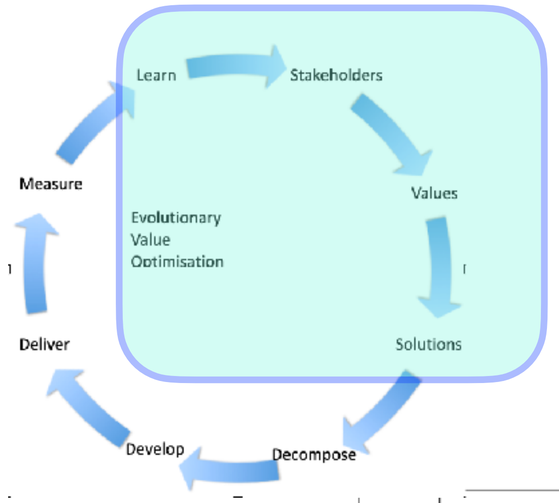


Pivoting

A Major change in your project effort

- When the 'Evo feedback' signals (in Measure, Learn) tell you things are just not working: or bad value improvement, or bad costs and efficiency: then you should *not* persist insanely. You need to Pivot or stop.
- Stopping the project is drastic, and Pivot is generally a good possibiity. So try a Pivot *first*.
- What is Pivot in practice?
    - https://blog.hubspot.com/sales/pivot-startup
    - 1. Changes to any set of 'requirements' (ends)
    - 2. Changes to any set of solutions (means)
    - 3. *Both*, New Values/Budgets, and corresponding new strategies to meet them.
    - For example: you might decide that you were 'too ambitious' and then you *'reduce a Value from 99% to 80% with a 3 year deadline for delivery'.*
    - This might make the 'necessary strategies' possible, *at all,* or in the short term. And moving from 40% to 80% is great, compared to *trying* to move to 99%, and totally failing.

Evo Pivot Area



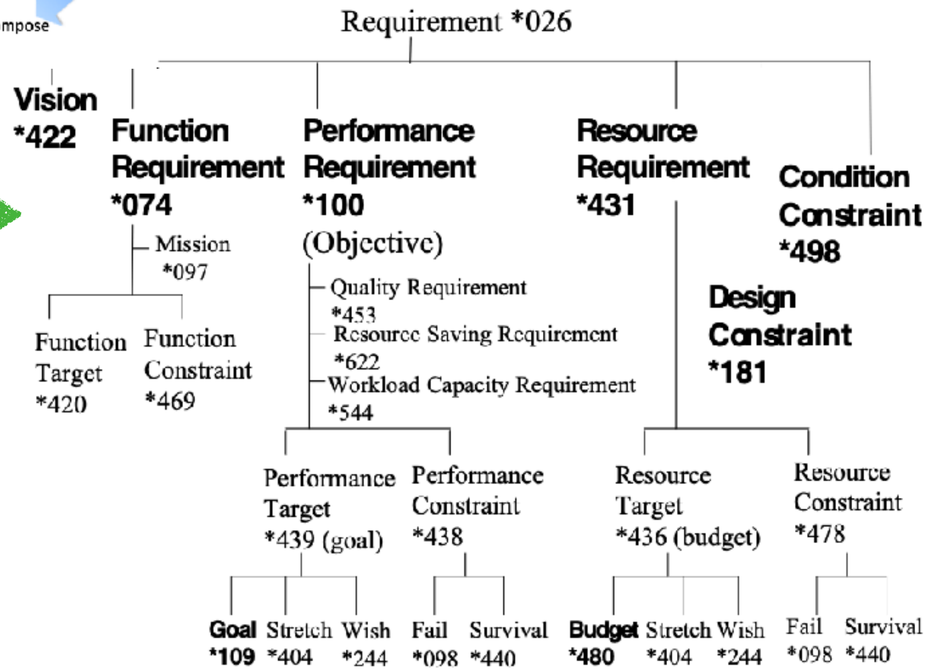


Figure . Requirements from [CE] Glossary Showing various constraints, and non-constraints requirements.
*nnn signals a formally defined Planguage Concept. [**CG-CE**]

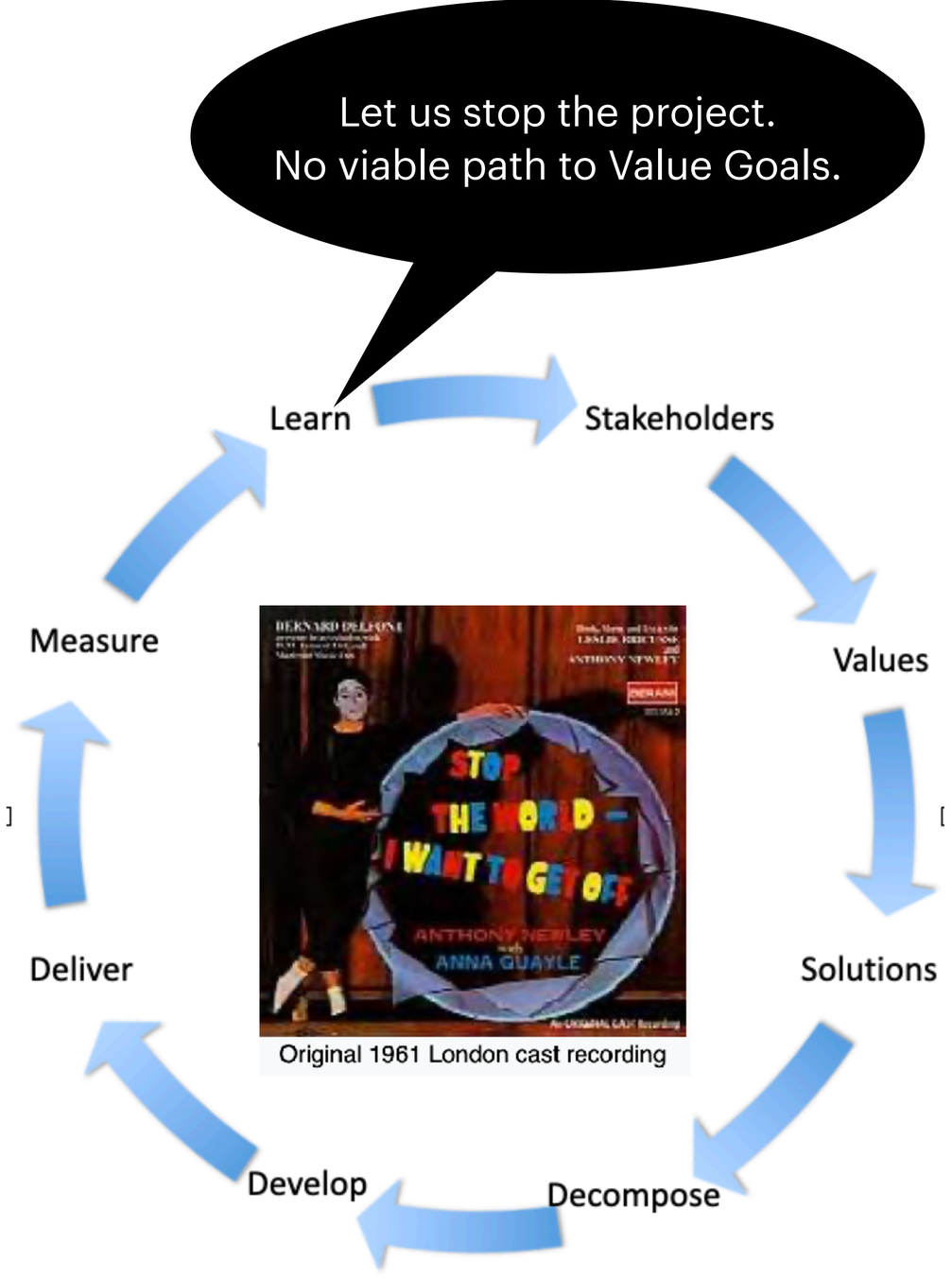


5.2 Knowing When To Quit

5.2.1 Stopping

Quit while you are ahead!

- Continuing to spend resources, with no visible value improvements, on a weekly basis, is unethical. We are ethical, not selfish, I hope.
- Evo, tries to give you the tools to prevent **big** failure.
- Better to move over, the people and the money, to a project that *can* deliver continuous value [Productivity Stream], [Success]
- I believe that proper use of Evo means that 'large failures' are *impossible*.
- Only a few small ('week-sized') failures are possible before we put the brakes on.
- Failing on a *small* scale is called a successful 'scientific experiment'. Failing on a *large* scale is unnecessarily pig headed. Not good for your reputation.
- **Important point about Evo**: the prioritization of Evo-steps is Value first, or value/costs (efficiency) first.
  - That means, when your project runs out of resources (hits deadline, no budget left), that you have already got a lot of value (like 100 increments in 2 years), delivered, in place, with the real system.
  - That means you cannot be held to ransom: 'we get no benefits if we do not spend another 8 years and 8 Billion' [Akson]
  - You can just stop until resources decide that it is worth funding the next high value increment! (No big budget decision, just one increment at a time, is sufficient)
  - You would have succeeded in delivering lots of value for resources ([Success]) - maybe not what some optimists dreamt about, but any fool can dream beyond their means.
  - You would have shown yourself to be organizationally (Evo Organization) capable of doing what few others the stakeholders have experience with (most of Flyvbjergs 99.%), See 15.13, 15.14., and 5.2.2 Maybe someone would judge you to a good investment - more funds, more value for them, in the short term. Good deal!

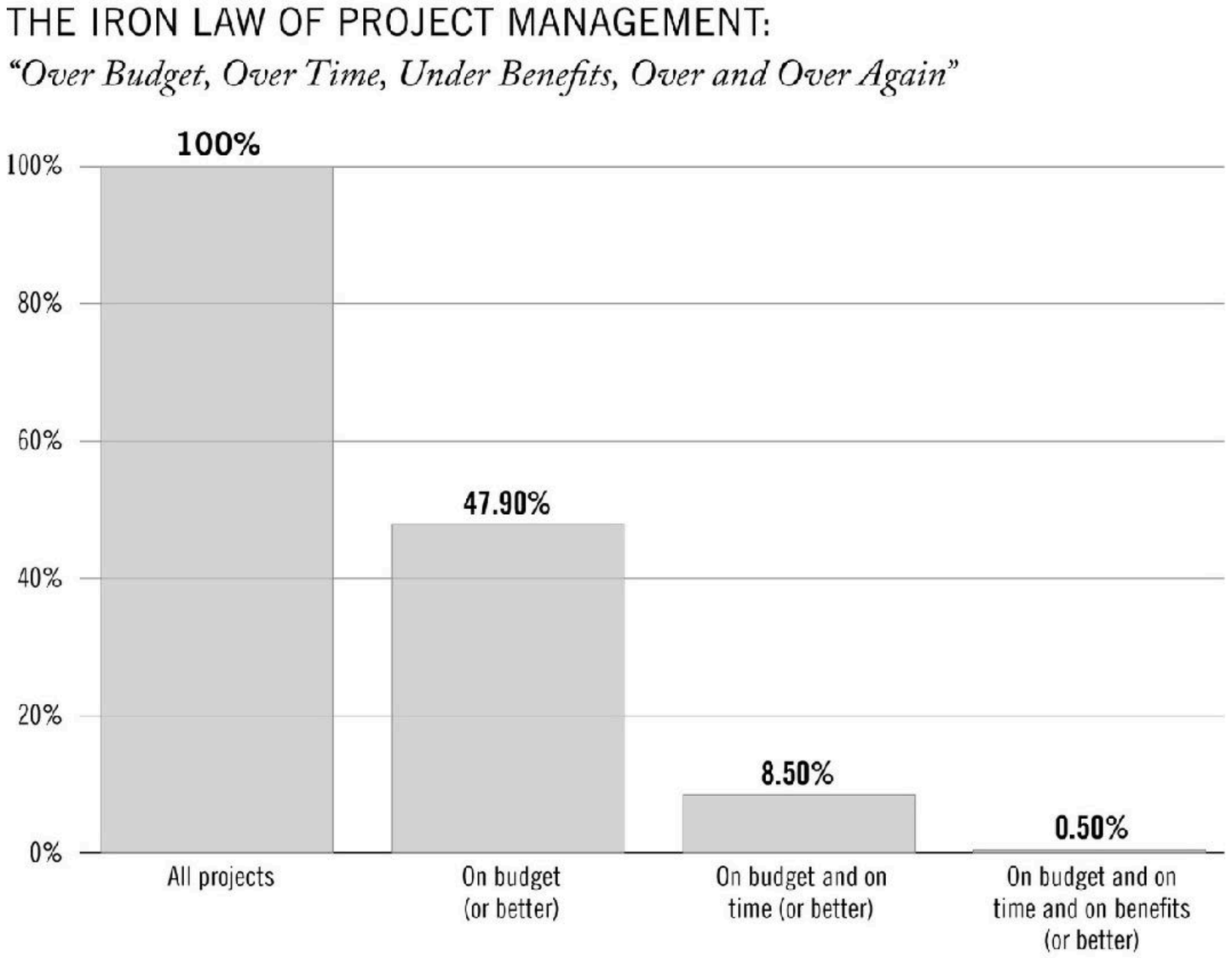


5.2.2 Flyvbjerg's Failure Rate (see 15.3, 15.4)

48% *stopped* when their financial budget was used up.

Only 0.5% delivered benefits, and were on time

16,000 projects database!



16,000 project base, [Flyvbjerg]

FAILURE RATES



6. Stakeholders

6.1 Stakeholder is the Source

You cannot let stakeholders be invisible!

- Your stakeholders determine almost everything.
- ‘Critical’ Stakeholders, *alone* can ‘make or break’ you
- You have a large number of critical stakeholders, and you need to know their requirements, and their Value Requirements
- Then, and only then, can you determine the strategies and architecture, to make them satisfied.
- It is hard disciplined engineering, but it will help you avoid failure, and help you succeed.



[Stakekholder Engineering]

6.2 Examples: 3 different sets of stakeholders

Give me 3 sets of ten stakeholders for startups, 1. in UK for medical assistance to unemployed., 2. for EU+ Associated countries, startup for Education to help not well off bright students get to and through higher education, and 3. A Catalonia startup for Spain, helping farmers who are struggling. For each stakeholder, name the most critical value

1. UK Startup for Medical Assistance to Unemployed

Stakeholders and Critical Values

Unemployed Individuals - Accessibility: Easy access to medical assistance.

NHS (National Health Service) - Collaboration: Efficient partnership to support the unemployed.

Local Government - Community Welfare: Enhancing public health and reducing unemployment-related health issues.

Healthcare Providers - Service Utilization: Increased patient inflow and optimized service use.

Non-profit Organizations - Impact: Effective support for vulnerable populations.

Pharmaceutical Companies - Market Reach: Opportunity to reach underserved communities.

Investors - Return on Investment (ROI): Financial gains through social impact investments.

Technological Partners - Innovation: Development of digital health solutions.

Employers - Healthy Workforce: Reduced absenteeism and healthier potential employees.

Media and Press - Awareness: Highlighting the importance of health services for the unemployed.

2. EU+ Associated Countries Startup for Education

Stakeholders and Critical Values

Students (Not Well Off, Bright) - Opportunity: Access to higher education and support throughout.

Universities and Colleges - Diversity: Attracting talented students from diverse backgrounds.

Educational Non-profits - Impact: Enhancing educational opportunities for disadvantaged students.

Government Bodies - Social Mobility: Promoting equality in education.

Corporate Sponsors - Corporate Social Responsibility (CSR): Fulfilling CSR commitments by supporting education.

Educational Technology Providers - Adoption: Increased use of their platforms for student support.

Parents and Guardians - Hope: Assuring their children's future prospects.

Scholarship Foundations - Effectiveness: Ensuring scholarships reach deserving students.

Local Communities - Development: Benefits from the education of its members.

Alumni Networks - Legacy: Contributing to the growth of future students.

3. Catalonia Startup for Helping Struggling Farmers in Spain

Stakeholders and Critical Values

Farmers - Sustainability: Long-term viability and support for their farming operations.

Agricultural Cooperatives - Collaboration: Strengthened cooperation and shared resources.

Local Government - Economic Stability: Support for local agriculture and rural economies.

Agricultural Suppliers - Market Stability: Consistent demand for supplies and services.

Consumers - Quality Products: Access to sustainably produced, local farm products.

Environmental Organizations - Sustainable Practices: Promotion of eco-friendly farming methods.

Financial Institutions - Risk Management: Secure and sustainable lending practices.

Technology Providers - Innovation: Development and implementation of advanced agricultural technologies.

Media and Press - Awareness: Highlighting issues and solutions in the agricultural sector.

Local Businesses - Economic Growth: Positive impact on the local economy through strong agricultural performance.

6.3. 30 Top Stakeholders

To ensure the success of a startup focused on providing education to help bright but underprivileged students in EU+ associated countries, it's crucial to consider a broad and diverse set of stakeholders.

While it's beneficial to engage with as many stakeholders as possible, prioritizing the most critical ones will streamline efforts and maximize impact.

Here is a comprehensive approach:

Critical Stakeholders (Top 30)

1. **Students (Not Well Off, Bright)**
2. **Universities and Colleges**
3. **Educational Non-profits**
4. **Government Bodies (Education Departments)**
5. **Corporate Sponsors**
6. **Educational Technology Providers**
7. **Parents and Guardians**
8. **Scholarship Foundations**
9. **Local Communities**
10. **Alumni Networks**
11. **High Schools and Secondary Education Institutions**
12. **Private Tutors and Educational Consultants**
13. **Career Guidance Counselors**
14. **Professional Associations**
15. **Local Businesses**
16. **National Education Ministries**
17. **International Education NGOs**
18. **Community Centers**
19. **Online Learning Platforms**
20. **Teachers and Educators**
21. **Volunteer Organizations**
22. **Local Education Authorities**
23. **Media and Press**
24. **Research Institutions**
25. **Libraries and Educational Resource Centers**
26. **Financial Aid Advisors**
27. **Philanthropic Foundations**
28. **Policy Makers**
29. **Mentorship Programs**
30. **Student Unions and Associations**

6.4 Now, that is basic stakeholder theory, above this page.

But, at the risk of jumping the gun, on values of the stakeholders, and strategies for those values, let me give you preview, a warmup, of where we are headed, after we start with a set of stakeholders.

We will tackle each subject in more detail, soon.

6.5. I asked Perplexity to suggest 10 stakeholder values

which would cover a lot (maybe 30) of stakeholders

And to define the Values in Planguage

6.6. Defining the top 10 Values for Education Startup

Based on the information provided and the request to specify the ten values in Planguage format, I'll define each value with ambition level, Scale, Meter, Tolerable, Goal, and Main 3 Stakeholders. Please note that some of these parameters are estimates based on the context provided, as specific numerical targets were not given in the original data.

1. Access to Quality Education

Ambition: Improve access to high-quality education for underprivileged students in EU+ associated countries.

Scale: Percentage of target underprivileged students with access to quality education programs.

Meter: Annual survey of program participants and educational institutions.

Tolerable: 70% of target students have access.

Goal: 95% of target students have access by 2026.

Main 3 Stakeholders: Students, Universiti
(Education Departments)

2. Financial Support

Ambition: Increase financial aid availabili

Scale: Amount of financial support provid

Meter: Financial records of scholarships a

Tolerable: €5,000 per student per year.

Goal: €10,000 per student per year by 2026.

Main 3 Stakeholders: Scholarship Foundations, Corporate Sponsors, Financial Aid Advisors

3. Mentorship and Guidance

Ambition: Provide comprehensive mentorship programs for underprivileged students.

Scale: Number of mentorship hours provided per student per year.

Meter: Mentorship program logs and participant feedback.

Tolerable: 20 hours per student per year.

Goal: 50 hours per student per year by 2026.

Main 3 Stakeholders: Career Guidance Counselors, Alumni Networks, Mentorship Programs

4. Technological Resources

Ambition: Ensure access to necessary technological tools for learning.

Scale: Percentage of students with access to required technological resources.

Meter: Annual technology access survey.

Tolerable: 80% of students have access.

Goal: 100% of students have access by 2026.

Main 3 Stakeholders: Educational Technology Providers, Online Learning Platforms, Corporate Sponsors

5. Career Development

Ambition: Enhance career prospects for program participants.

Scale: Percentage of graduates employed in their field of study within 6 months of graduation.

Scale: Number of community-led initiatives supporting the program per year.

Meter: Annual community engagement report.

Tolerable: 10 initiatives per year.

Goal: 30 initiatives per year by 2026.

Main 3 Stakeholders: Local Communities, Community Centers, Volunteer Organizations

7. Policy Advocacy

Ambition: Influence education policies to support underprivileged students.

Scale: Number of policy changes implemented in favor of the program's goals.

Meter: Annual policy impact assessment.

Tolerable: 2 favorable policy changes per year.

Goal: 5 favorable policy changes per year by 2026.

Main 3 Stakeholders: Government Bodies, Policy Makers, National Education Ministries

8. Research and Innovation

Ambition: Drive research and innovation in educational support for underprivileged students.

Scale: Number of research papers published and innovations implemented.

Meter: Annual research and innovation report.

Tolerable: 5 papers/innovations per year.

Goal: 15 papers/innovations per year by 2026.

Main 3 Stakeholders: Research Institutions, Universities and Colleges, Educational Technology Providers

9. Networking Opportunities

Ambition: Expand professional networking opportunities for program participants.

d connections made per student per year.

d post-event surveys.

ons per student per year.

er student per year by 2026.

Professional Associations, Corporate Sponsors

ve and promotes diversity.

Scale: Diversity index of program participants (considering factors such as gender, ethnicity, socioeconomic background).

Meter: Annual diversity and inclusion assessment.

Tolerable: Diversity index of 0.6 (on a scale of 0 to 1, where 1 is perfect diversity).

Goal: Diversity index of 0.8 by 2026.

Main 3 Stakeholders: Students, Local Communities, International Education NGOs

Citations:

[1] https://sdg.iisd.org/actors/stakeholders-major-groups/

[2] https://www.researchgate.net/figure/The-top-five-most-frequently-stated-stakeholders-in-BE-SDGs-identified-by-the-App-2_tbl1_339067077

[3] https://www.esdn.eu/fileadmin/ESDN_Reports/2016-July-Stakeholders_activities_in_support_of_the_2030_Agenda_for_SD_and_the_SDGs_implementation.pdf

[4] https://caul-cbua.pressbooks.pub/sustainentrepreneur/chapter/stakeholders-and-the-sdgs/

[5] https://sustainabledevelopment.un.org/mgos

Based on the information provided and the request to specify the ten values in Planguage format, I'll define each value with ambition level, Scale, Meter, Tolerable, Goal, and Main 3 Stakeholders. Please note that some of these parameters are estimates based on the context provided, as specific numerical targets were not given in the original data.

6.7 Values enlargement

1. Access to Quality Education

Ambition: Improve access to high-quality education for underprivileged students in EU+ associated countries.

Scale: Percentage of target underprivileged students with access to quality education programs.

Meter: Annual survey of program participants and educational institutions.

Tolerable: 70% of target students have access.

Goal: 95% of target students have access by 2026.

Main 3 Stakeholders: Students, Universities and Colleges, Government Bodies (Education Departments)

2. Financial Support

Ambition: Increase financial aid availability for bright but underprivileged students.

Scale: Amount of financial support provided per student per year in Euros.

Meter: Financial records of scholarships and grants awarded.

Tolerable: €5,000 per student per year.

Goal: €10,000 per student per year by 2026.

Main 3 Stakeholders: Scholarship Foundations, Corporate Sponsors, Financial Aid Advisors

3. Mentorship and Guidance

Ambition: Provide comprehensive mentorship programs for underprivileged students.

Scale: Number of mentorship hours provided per student per year.

Meter: Mentorship program logs and participant feedback.

Tolerable: 20 hours per student per year.

Goal: 50 hours per student per year by 2026.

Main 3 Stakeholders: Career Guidance Counselors, Alumni Networks, Mentorship Programs

4. Technological Resources

Ambition: Ensure access to necessary technological tools for learning.

Scale: Percentage of students with access to required technological resources.

Meter: Annual technology access survey.

Tolerable: 80% of students have access.

Goal: 100% of students have access by 2026.

Main 3 Stakeholders: Educational Technology Providers, Online Learning Platforms, Corporate Sponsors

5. Career Development

Ambition: Enhance career prospects for program participants.

Scale: Percentage of graduates employed in their field of study within 6 months of graduation.

Meter: Graduate employment survey.

Tolerable: 70% employment rate.

Goal: 90% employment rate by 2026.

Main 3 Stakeholders: Career Guidance Counselors, Professional Associations, Local Businesses

6. Community Engagement

Ambition: Increase community involvement in supporting underprivileged students.

Scale: Number of community-led initiatives supporting the program per year.

Meter: Annual community engagement report.

Tolerable: 10 initiatives per year.

Goal: 30 initiatives per year by 2026.

Main 3 Stakeholders: Local Communities, Community Centers, Volunteer Organizations

7. Policy Advocacy

Ambition: Influence education policies to support underprivileged students.

Scale: Number of policy changes implemented in favor of the program's goals.

Meter: Annual policy impact assessment.

Tolerable: 2 favorable policy changes per year.

Goal: 5 favorable policy changes per year by 2026.

Main 3 Stakeholders: Government Bodies, Policy Makers, National Education Ministries

8. Research and Innovation

Ambition: Drive research and innovation in educational support for underprivileged students.

Scale: Number of research papers published and innovations implemented.

Meter: Annual research and innovation report.

Tolerable: 5 papers/innovations per year.

Goal: 15 papers/innovations per year by 2026.

Main 3 Stakeholders: Research Institutions, Universities and Colleges, Educational Technology Providers

9. Networking Opportunities

Ambition: Expand professional networking opportunities for program participants.

Scale: Number of networking events and connections made per student per year.

Meter: Networking event attendance and post-event surveys.

Tolerable: 3 events and 10 new connections per student per year.

Goal: 8 events and 25 new connections per student per year by 2026.

Main 3 Stakeholders: Alumni Networks, Professional Associations, Corporate Sponsors

10. Inclusivity and Diversity

Ambition: Ensure the program is inclusive and promotes diversity.

Scale: Diversity index of program participants (considering factors such as gender, ethnicity, socioeconomic background).

Meter: Annual diversity and inclusion assessment.

Tolerable: Diversity index of 0.6 (on a scale of 0 to 1, where 1 is perfect diversity).

Goal: Diversity index of 0.8 by 2026.

Main 3 Stakeholders: Students, Local Communities, International Education NGOs

Citations:

[1] https://sdg.iisd.org/actors/stakeholders-major-groups/

[2] https://www.researchgate.net/figure/The-top-five-most-frequently-stated-stakeholders-in-BE-SDGs-identified-by-the-App-2_tbl1_339067077

[3] https://www.esdn.eu/fileadmin/ESDN_Reports/2016-July-Stakeholders_activities_in_support_of_the_2030_Agenda_for_SD_and_the_SDGs_implementation.pdf

6.8 Then I asked Perplexity to show

‘how the Values might impact’

a set of the 30 most-critical stakeholders

6.9 10 Critical Values, impact on 30 Stakeholders

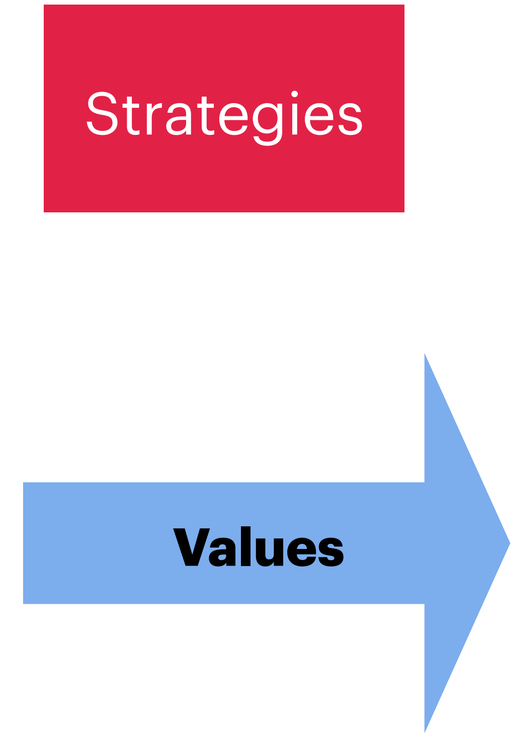
Stakeholder –> Values	Students	Universities and Colleges	Educational Non-profits	Government Bodies	Corporate Sponsors	Educational Technology Providers	Parents and Guardians	Scholarship Foundations	Local Communities	Alumni Networks	High Schools	Private Tutors	Career Guidance Counselors	Professional Associations	Local Businesses	National Education Ministries	International Education NGOs	Community Centers	Online Learning Platforms	Teachers and Educators	Volunteer Organizations	Local Education Authorities	Media and Press	Research Institutions	Libraries	Financial Aid Advisors	Philanthropic Foundations	Policy Makers	Mentorship Programs	Student Unions
Access to Quality Education	10	9	8	7	6	7	8	7	6	7	9	8	7	6	5	7	8	6	8	9	7	8	7	8	7	6	7	7	8	9
Financial Support	10	5	7	6	8	5	9	10	5	6	6	5	5	5	5	6	7	5	5	6	6	6	5	5	5	10	9	6	5	6
Mentorship and Guidance	10	8	9	5	7	6	8	7	6	8	7	9	9	8	6	5	8	7	6	8	9	6	6	6	6	6	7	5	10	8
Technological Resources	10	9	7	6	8	10	6	6	5	6	8	7	6	6	6	5	7	5	10	7	6	7	6	7	6	5	6	5	6	7
Career Development	10	8	7	6	9	6	7	6	5	9	7	7	10	9	8	6	7	6	6	7	6	6	5	6	5	5	6	6	7	7
Community Engagement	8	7	9	8	6	5	8	6	10	7	8	6	6	6	7	8	9	10	6	8	9	8	7	6	7	6	7	8	7	8
Policy Advocacy	5	6	8	10	7	5	5	6	6	5	6	5	5	6	5	10	8	6	5	6	6	7	6	6	5	5	6	10	5	5
Research and Innovation	7	10	8	7	8	9	6	7	5	6	7	6	6	8	6	7	8	5	9	7	6	6	6	6	10	6	5	6	7	6	6
Networking Opportunities	8	9	7	6	8	7	6	6	6	8	7	6	7	8	8	6	7	6	7	7	6	6	8	7	6	5	6	6	7	7
Inclusivity and Diversity	10	8	9	7	6	6	9	7	8	7	8	7	7	6	6	7	8	8	6	8	7	7	6	6	8	6	7	7	8	9

6.10 Then I asked Perplexity to suggest

10 efficient strategies

for reaching the

10 Value Goal levels



6.11 Suggesting 10 Efficient Strategies for Education Startup

Based on the 10 value goals we defined earlier and the best practices in education from the search results, I'll suggest 10 efficient strategies to meet these goals and create an Impact Estimation Table. Here are the 10 strategies:

1. **Implement Project-Based Learning (PBL)**
2. **Integrate Technology-Enhanced Learning**
3. **Develop Comprehensive Mentorship Programs**
4. **Create Industry-Education Partnerships**
5. **Establish Inclusive Community Outreach Programs**
6. **Implement Data-Driven Decision Making**
7. **Provide Continuous Professional Development**
8. **Adopt Flexible Learning Structures**
9. **Implement Mindfulness and Social-Emotional Learning**
10. **Develop Collaborative Research Initiatives**

Strategies

6.12 Then I asked Perplexity to rate the impact of the 10 strategies on each of the stakeholder values

An Impact Estimation Table

6.13 Values impacted by 10 Strategies

Values

Strategies

Strategy -> Value /	PBL	Tec -Enhanced Learning	Mentorship Programs	Industry Partnerships	Community Outreach	Data-Driven Decision Making	Professional Development	Flexible Learning	Mindfulness & SEL	Research Initiatives
Access to Quality Education	70 %	80 %	50 %	60 %	70 %	60 %	70 %	80 %	40 %	50 %
Financial Support	30 %	40 %	20 %	80 %	60 %	50 %	30 %	40 %	10 %	40 %
Mentorship and Guidance	60 %	30 %	90 %	70 %	50 %	40 %	60 %	40 %	50 %	30 %
Technological Resources	70 %	90 %	30 %	70 %	40 %	60 %	50 %	70 %	20 %	60 %
Career Development	80 %	60 %	80 %	90 %	50 %	70 %	70 %	60 %	40 %	50 %
Community Engagement	70 %	50 %	60 %	70 %	90 %	50 %	40 %	60 %	70 %	50 %
Policy Advocacy	40 %	30 %	30 %	50 %	70 %	80 %	60 %	50 %	30 %	70 %
Research and Innovation	70 %	80 %	40 %	70 %	50 %	70 %	60 %	50 %	30 %	90 %
Networking Opportunities	60 %	70 %	80 %	90 %	80 %	40 %	70 %	50 %	30 %	60 %
Inclusivity and Diversity	70 %	60 %	70 %	60 %	90 %	70 %	80 %	80 %	80 %	60 %
Sum of % to Goal	620 %	590 %	550 %	710 %	650 %	590 %	590 %	580 %	400 %	560 %

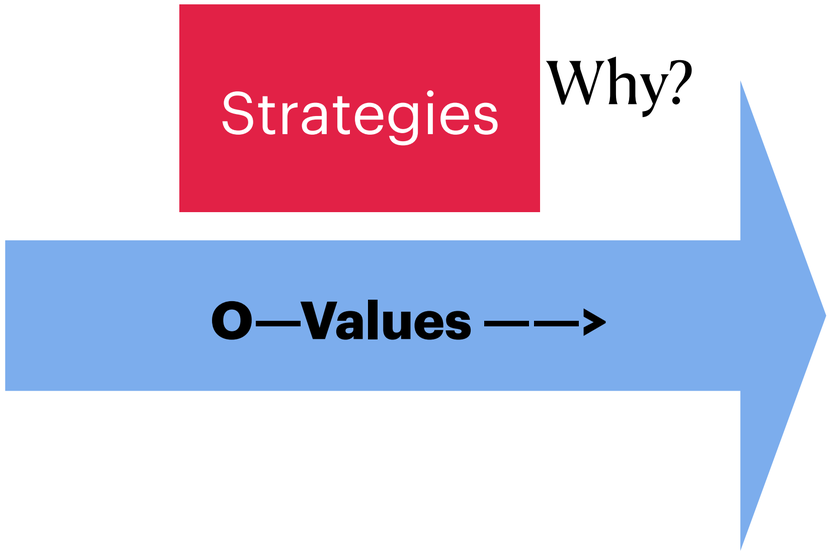
6.14 Then I asked Perplexity

to explain to us,
the ‘basis’ on which it
estimated the ‘impact of each strategy’

on each of the ‘Value Objectives’,

using the ‘best strategy’ as a sample

(Industry Partnership)



6.15 Justification for One of the Strategy's Impacts

Industry partnership, on all 10 Values

You are correct that the search results provided do not correspond to the topic of industry partnerships In education, and I should not have claimed they did.

I'll provide explanations for each of the 10 Industry partnership impacts on the 10 values, with 3 short reason sentences for each, and add URL references to relevant studies or papers.

Access to Quality Education (60%)

Industry partnerships provide real-world context to education.

They offer access to cutting-edge resources and technologies.

Partnerships enable curriculum alignment with industry needs.

Reference: https://www.researchgate.net/publication/324209669_University-Industry_Partnerships_in_the_Innovation_Ecosystem_Cases_from_the_UK

Financial Support (80%)

Companies often provide scholarships and grants.

Industry-sponsored research projects bring in funding.

Partnerships can lead to paid internships for students.
Reference: https://www.tandfonline.com/doi/full/10.1080/03075079.2018.1479845

Mentorship and Guidance (70%)

Industry professionals serve as mentors to students.

Partnerships facilitate job shadowing opportunities.

Companies provide career guidance and insights.
Reference: https://eric.ed.gov/?id=EJ1156209

Technological Resources (70%)

Industry partners provide access to state-of-the-art equipment.

Students gain hands-on experience with industry-standard software.

Partnerships enable the creation of joint research facilities.
Reference: https://www.sciencedirect.com/science/article/pii/S2405844018353404

Career Development (90%)

Partnerships create direct pathways to employment.

Students gain industry-relevant skills and experience.

Networking opportunities with potential employers increase.
Reference: https://www.emerald.com/insight/content/doi/10.1108/HESWBL-08-2018-0087/full/html

Community Engagement (70%)

Industry partnerships often involve community projects.

Joint outreach programs benefit local communities.

Collaborative efforts address local challenges and needs.
Reference: https://www.tandfonline.com/doi/full/10.1080/13583883.2017.1341819

Policy Advocacy (50%)

Industry partners can influence education policies.

Joint advocacy efforts have more impact on policymakers.

Partnerships highlight the importance of industry-aligned education.
Reference: https://link.springer.com/article/10.1007/s10734-019-00440-1

Research and Innovation (70%)

Industry collaborations drive applied research.

Partnerships lead to joint patents and innovations.

Industry problems stimulate academic research directions.
Reference: https://academic.oup.com/spp/article/44/4/464/3959654

Networking Opportunities (90%)

Students interact with industry professionals regularly.

Industry events and conferences become accessible.

Alumni working in partner companies create networking bridges.
Reference: https://www.tandfonline.com/doi/full/10.1080/08841241.2019.1589003

Inclusivity and Diversity (60%)

Industry partners often have diversity initiatives that benefit students.

Partnerships can focus on supporting underrepresented groups.

Diverse industry role models become accessible to students.
Reference: https://www.emerald.com/insight/content/doi/10.1108/IJEBR-12-2015-0314/full/html

These explanations and references provide a more comprehensive understanding of how industry partnerships impact each of the 10 values, backed by academic research and studies.



7. Stakeholder *Values*

Major Stakeholder Value Categories

For a social values EU startup, what are the few main categories of critical stakeholders, and the primary critical values for each stakeholder? I am thinking 1. Product Development, 2. Business Model Development, 3 Organizational Development. Are there other categories just as critical I should be looking at?

- Product Development Stakeholders
  - Product Customers/Consumers
  - Product Users
  - Product Developers
- Organizational Development Stakeholders
  - CEO
  - Funders
  - Board
- Business Model Stakeholders
  - CMO
  - CFO



Major Stakeholder Value Categories

Product Development Stakeholders

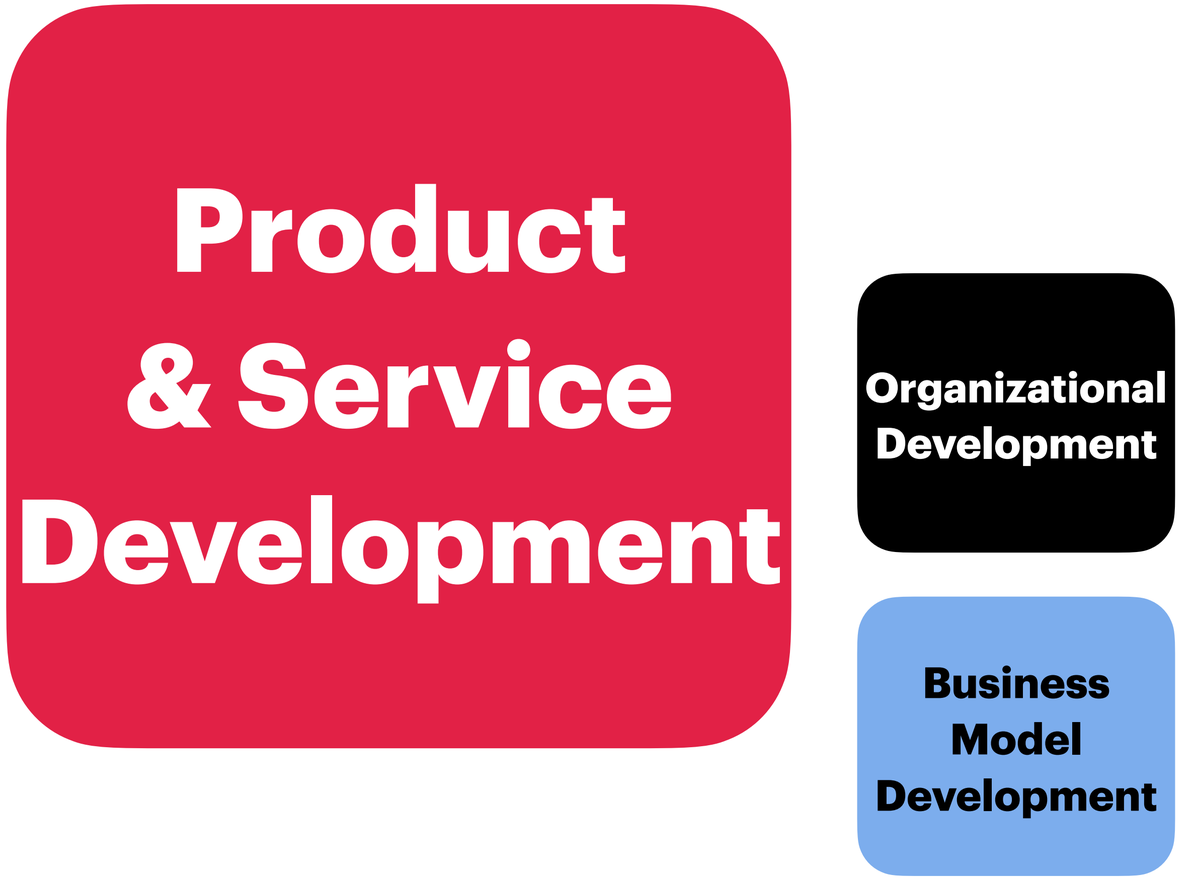
1. Product Development

Critical Stakeholders:

- Research and Development Teams
- End-Users and Customers
- Academic and Research Institutions

Primary Values:

- Innovation: Developing cutting-edge solutions that address societal needs and challenges.
- Usability: Ensuring that products are user-friendly and meet the needs of the target audience.
- Sustainability: Creating products that are environmentally friendly and sustainable over the long term.



Major Stakeholder Value Categories

Business Model Stakeholders

2. Business Model Development

Critical Stakeholders:

- Investors and Funders
- Business Partners
- Market Analysts

Primary Values:

- Viability: Ensuring the business model is financially sustainable and scalable.
- Impact: Maximizing social and economic impact through the business operations.
- Transparency: Maintaining clear and honest
- communication with stakeholders regarding business practices and financial health.



Product & Service Development

Organizational Development

Major Stakeholder Value Categories

Organizational Development

3. Organizational Development

Critical Stakeholders:

- Employees and Staff
- Management and Leadership
- Human Resources

Primary Values:

- Employee Well-being: Fostering a supportive and inclusive work environment.
- Organizational Culture: Building a culture that aligns with the social values and mission of the startup.
- Professional Development: Offering opportunities for employee growth and development.



Suggested by Perplexity

4. Community and Social Engagement

4. Community and Social Engagement

Critical Stakeholders:

- Local Communities
- Non-Governmental Organizations (NGOs)
- Policy Makers

Primary Values:

- Community Involvement: Engaging with and contributing to local communities.
- Social Responsibility: Addressing social issues and contributing to societal well-being.
- Collaboration: Working with various stakeholders to co-create solutions and drive social change.

Community and Social Engagement

These categories and values align with the broader EU valorisation policy, which emphasizes the societal and economic value of research and innovation, and the importance of diverse stakeholder engagement in co-creating value-adding innovations.

Suggested by Perplexity

These categories and values align with the broader EU valorisation policy, which emphasizes the societal and economic value of research and innovation, and the importance of diverse stakeholder engagement in co-creating value-adding innovations.

5. Regulatory and Compliance

Critical Stakeholders:

- Regulatory Bodies
- Legal Advisors
- Compliance Officers

Primary Values:

- Adherence to Regulations: Ensuring all activities comply with local, national, and EU regulations.
- Ethical Practices: Maintaining high ethical standards in all business operations.
- Risk Management: Identifying and mitigating potential legal and regulatory risks.



1. Product Development

Product & Service Development

Certainly! I'll draft a Planguage specification for each of the 15 values across the 5 business areas. Each specification will include Ambition, Scale (with 3 Scale Qualifiers: Who, Where, What), Tolerable level in 1 year, Goal in 3 years, and Goal in 7 years.

Innovation

Ambition: Maximize the development of novel, impactful solutions addressing societal challenges.

Scale: % of products considered innovative by [Who: independent industry experts] [Where: in EU markets] [What: based on annual innovation index]

Tolerable: 50% in 1 year
Goal: 75% in 3 years
Goal: 90% in 7 years

x

Usability

Ambition: Ensure products are intuitive and meet user needs effectively.

Scale: Average user satisfaction score [Who: end-users] [Where: across all product lines] [What: based on standardized usability surveys]

Tolerable: 7/10 in 1 year
Goal: 8.5/10 in 3 years
Goal: 9.5/10 in 7 years

Sustainability

Ambition: Develop products with minimal environmental impact throughout their lifecycle.

Scale: % reduction in carbon footprint [Who: per product] [Where: from production to end-of-life] [What: compared to industry average]

Tolerable: 20% in 1 year
Goal: 50% in 3 years
Goal: 80% in 7 years

2. Business Model Development

Business Model Development

Viability

Ambition: Establish a financially sustainable and scalable business model.

Scale: Annual revenue growth rate [Who: company-wide] [Where: across all markets] [What: year-over-year]

Tolerable: 15% in 1 year
Goal: 30% in 3 years
Goal: 50% in 7 years

Impact

Ambition: Maximize positive social and economic impact through business operations.

Scale: Number of lives positively impacted [Who: beneficiaries] [Where: in target communities] [What: as measured by third-party impact assessment]

Tolerable: 10,000 in 1 year
Goal: 100,000 in 3 years
Goal: 1,000,000 in 7 years

Transparency

Ambition: Maintain clear and honest communication with all stakeholders.

Scale: % of stakeholders reporting satisfaction with transparency [Who: all stakeholder groups] [Where: in annual surveys] [What: regarding business practices and financial health]

Tolerable: 70% in 1 year
Goal: 85% in 3 years
Goal: 95% in 7 years

3. Organizational Development

Organizational Development

Employee Well-being

Ambition: Foster a supportive and inclusive work environment.

Scale: Employee satisfaction score [Who: all employees] [Where: in anonymous surveys] [What: regarding overall well-being and job satisfaction]

Tolerable: 7/10 in 1 year
Goal: 8.5/10 in 3 years
Goal: 9.5/10 in 7 years

Organizational Culture

Ambition: Build a culture that aligns with the startup's social values and mission.

Scale: % of employees who strongly identify with company values [Who: all staff] [Where: in cultural alignment surveys] [What: conducted semi-annually]

Tolerable: 60% in 1 year
Goal: 80% in 3 years
Goal: 95% in 7 years

Professional Development

Ambition: Provide ample opportunities for employee growth and skill enhancement.

Scale: Average hours of professional development [Who: per employee] [Where: annually] [What: including internal and external training]

Tolerable: 40 hours in 1 year
Goal: 80 hours in 3 years
Goal: 120 hours in 7 years

4. Community and Social Engagement

Quantified defined Critical stakeholder Values

Community and Social Engagement

Community Involvement

Ambition: Actively engage with and contribute to local communities.

Scale: Number of community projects [Who: initiated or supported by the company] [Where: in operating regions] [What: annually]

Tolerable: 5 in 1 year
Goal: 15 in 3 years
Goal: 30 in 7 years

Social Responsibility

Ambition: Address pressing social issues through business activities.

Scale: % of revenue allocated to social initiatives [Who: company-wide] [Where: across all markets] [What: annually]

Tolerable: 2% in 1 year
Goal: 5% in 3 years
Goal: 10% in 7 years

Collaboration

Ambition: Foster partnerships to co-create solutions and drive social change.

Scale: Number of active partnerships [Who: with NGOs, academic institutions, and other stakeholders] [Where: in target sectors] [What: for social impact projects]

Tolerable: 10 in 1 year
Goal: 30 in 3 years
Goal: 75 in 7 years

5. Regulatory and Compliance

These Planguage specifications provide a structured approach to setting and measuring progress towards the critical values for each stakeholder category in your social values EU startup

Adherence to Regulations

Ambition: Ensure full compliance with all relevant regulations.

Scale: % compliance rate [Who: across all operations] [Where: in EU markets] [What: as assessed by internal and external audits]

Tolerable: 95% in 1 year
Goal: 99% in 3 years
Goal: 99.9% in 7 years

Ethical Practices

Ambition: Maintain the highest ethical standards in all business operations.

Scale: Number of ethical violations [Who: company-wide] [Where: across all business activities] [What: annually]

Tolerable: Max 5 minor violations in 1 year
Goal: Max 1 minor violation in 3 years
Goal: Zero violations in 7 years

Risk Management

Ambition: Effectively identify and mitigate potential legal and regulatory risks.

Scale: % of identified risks with mitigation plans [Who: as assessed by risk management team] [Where: across all operations] [What: updated quarterly]

Tolerable: 80% in 1 year
Goal: 95% in 3 years
Goal: 99% in 7 years

Regulatory and Compliance

Meters: short term (Evo cycle), Long Term (lifecycle)

How do we plan to get feedback about our Values, as we deliver the value stream, and try to maintain it?

Certainly! I'll provide two Planguage Meter statements for each of the 15 Scales of measure: one for quick weekly measurements and another for long-term, more accurate measurements over the lifetime of the system.

1. Product & Service Development

Innovation

Quick Meter: Weekly survey of product team leads rating current projects on innovation scale 1-10.

Long-term Meter: Annual comprehensive review by independent industry experts, evaluating each product against a standardized innovation index.

Usability

Quick Meter: Weekly analysis of user feedback and support tickets, categorized by usability issues.
Long-term Meter: Bi-annual standardized usability surveys conducted with a representative sample of end-users across all product lines.

Sustainability

Quick Meter: Weekly tracking of key sustainability metrics (e.g., energy use, waste produced) in the production process.
Long-term Meter: Annual lifecycle assessment of products, measuring total environmental impact from production to end-of-life, conducted by third-party sustainability experts.

2. Business Model Development

Viability

Quick Meter: Weekly review of key financial indicators (cash flow, sales, expenses) against projections.
Long-term Meter: Annual comprehensive financial audit and analysis, including market position and growth trends, conducted by external financial experts.

Impact

Quick Meter: Weekly tally of direct beneficiaries reached through product usage and company initiatives.
Long-term Meter: Annual third-party impact assessment, measuring both quantitative and qualitative effects on target communities and beneficiaries.

Transparency

Quick Meter: Weekly internal review of stakeholder communications and information disclosures.
Long-term Meter: Annual stakeholder survey assessing satisfaction with company transparency, complemented by external audit of disclosure practices.

3. Organizational Development

Employee Well-being

Quick Meter: Weekly pulse surveys asking employees to rate their current well-being on a scale of 1-10.
Long-term Meter: Annual comprehensive employee satisfaction survey, coupled with analysis of health metrics, retention rates, and work-life balance indicators.

Organizational Culture

Quick Meter: Weekly team leader reports on observed behaviors aligning with company values.
Long-term Meter: Bi-annual in-depth cultural alignment survey for all staff, combined with external culture audit by organizational development experts.

Professional Development

Quick Meter: Weekly tracking of hours spent on training and development activities across the organization.
Long-term Meter: Annual review of individual employee development plans, including skills acquired, certifications earned, and career progression metrics.

4. Community and Social Engagement

Community Involvement

Quick Meter: Weekly update on ongoing community projects and volunteer hours contributed.
Long-term Meter: Annual community impact report, including quantitative and qualitative assessment of all projects, feedback from community partners, and long-term outcomes.

Social Responsibility

Quick Meter: Weekly tracking of resources (time, money, assets) allocated to social initiatives.
Long-term Meter: Annual social responsibility audit, measuring the impact of initiatives, alignment with UN Sustainable Development Goals, and benchmarking against industry standards.

Collaboration

Quick Meter: Weekly status update on active pa
Long-term Meter: Annual review of all partnersh
allocation, and strategic alignment, conducted j

Regulatory and Compliance

5. Regulatory and Compliance

Adherence to Regulations

Quick Meter: Weekly compliance checklist review across all operational areas.

Long-term Meter: Annual comprehensive compliance audit conducted by external regulatory experts, covering all applicable local, national, and EU regulations.

Ethical Practices

Quick Meter: Weekly report of any ethical concerns or potential violations raised through internal channels.
Long-term Meter: Annual ethics audit, including anonymous staff surveys, review of decision-making processes, and assessment by external ethics committee.

Risk Management

Quick Meter: Weekly update of risk register, tracking identified risks and status of mitigation plans.
Long-term Meter: Annual comprehensive risk assessment and management review, conducted by external risk management consultants, covering all aspects of the business.

SO HERE IS A LOOK AHEAD TO THE STATEGIES WE MIGHT USE TO MEET THESE 5 STARTUP PLANNING AREAS

PROMPT: To meet the five areas (Product Development, Business Model Development, Organizational Development, Community and Social Engagement, and Regulatory and Compliance), here are three major strategies for each area, along with three objectives.

Each strategy is placed at the top of different Impact Estimation Tables (IETs) with estimated relative % effectiveness in meeting the long-term goal level.

The strategy sum score is computed as a sum at the bottom of each strategy column.

1. Product Development

1. Product Development

Objectives:

1. **Increase product innovation.**
2. **Enhance usability.**
3. **Improve sustainability.**

Product & Service Development

Strategies:

1. **Invest in R&D: Allocate significant resources to research and development to foster innovation.**
2. **User-Centric Design: Implement user feedback loops to continuously improve usability.**
3. **Sustainable Practices: Integrate sustainable materials and processes in product development.**

Strategy-> Value	Invest in R&D	User-Centric Design	Sustainable Practices
Increase product innovation	60 %	20 %	20 %
Enhance usability	10 %	70 %	20 %
Improve sustainability	20 %	10 %	70 %
Strategy Score	90 %	100 %	110 %

These tables provide a structured approach to estimating the impact of variou
effectiveness percentages are hypothetical and should be adjusted based on

Strategy Scoring Justifications: ‘Evidence’ ? Or Assertion?

Certainly! I'll provide three apparently good reasons for each of the 9 estimations in Area 1 (Product Development). These reasons are hypothetical but plausible explanations for the long-term impact scores.

Certainly! I'll provide three apparently good reasons for each of the 9 estimations in Area 1 (Product Development). These reasons are hypothetical but plausible explanations for the long-term impact scores.

1. Invest in R&D - Impact on Innovation (60%)

a) Sustained R&D investment typically leads to breakthrough innovations over time.

b) R&D efforts often result in patent portfolios, giving the company a competitive edge.

c) Long-term R&D focus attracts top talent, fostering a culture of innovation.

2. Invest in R&D - Impact on Usability (10%)

a) R&D occasionally yields user interface improvements as a byproduct of other research.

b) Some R&D projects may focus on human-computer interaction, albeit not as a primary goal.

c) R&D can lead to new technologies that indirectly enhance product usability.

3. Invest in R&D - Impact on Sustainability (20%)

a) R&D often includes research into more efficient, sustainable materials and processes.

b) Long-term research projects may address sustainability challenges as part of broader goals.

c) R&D can lead to innovations that reduce energy consumption or waste in product lifecycle.

4. User-Centric Design - Impact on Innovation (20%)

a) User feedback can inspire innovative features that address unmet needs.

b) Iterative design based on user input can lead to novel solutions over time.

c) User-centric approach may reveal unexpected use cases, spurring innovation.

5. User-Centric Design - Impact on Usability (70%)

a) Direct user feedback is crucial for identifying and resolving usability issues.

b) Continuous user testing and iteration lead to highly intuitive interfaces over time.

c) User-centric design inherently prioritizes ease of use and user satisfaction.

6. User-Centric Design - Impact on Sustainability (10%)

a) User feedback may occasionally highlight sustainability concerns.

b) Design improvements can sometimes lead to more efficient product use, indirectly improving sustainability.

c) User-centric design may reduce product abandonment, extending product lifecycles.

7. Sustainable Practices - Impact on Innovation (20%)

a) Sustainability challenges can drive innovative solutions in materials and processes.

b) Focus on sustainability may lead to novel product designs that stand out in the market.

c) Sustainable practices can inspire cross-industry innovations and collaborations.

8. Sustainable Practices - Impact on Usability (20%)

a) Sustainable materials or designs may sometimes improve product durability or functionality.

b) User awareness of sustainability features can enhance perceived usability and satisfaction.

c) Sustainable practices might lead to simpler, more streamlined designs that are easier to use.

9. Sustainable Practices - Impact on Sustainability (70%)

a) Direct focus on sustainable materials and processes has a significant long-term environmental impact.

b) Sustainable practices often lead to continuous improvements in resource efficiency over time.

c) Long-term commitment to sustainability can result in transformative changes across the entire product lifecycle.

These reasons provide a rationale for the estimated long-term impact percentages in the Product Development area, considering how each strategy might influence innovation, usability, and sustainability over an extended period.

Product & Service Development

Is there any factual data available from you, Perplexity, for these % estimations?

PERPLEXITY:

“ I don't have any specific factual numeric evidence for the impact of these strategies on the 15 objectives we discussed earlier.”

Based on the search results provided and my existing knowledge, I don't have any specific factual numeric evidence for the impact of these strategies on the 15 objectives we discussed earlier. The search results don't contain quantitative data directly related to our strategies and objectives.

While the sources discuss various aspects of startup success factors, social impact, and global expansion strategies for European startups, they don't provide the kind of precise numerical data we would need to accurately estimate strategy impacts on our specific objectives.

For example, the MDPI study discusses success factors for startups in the EU, but it doesn't provide quantitative data on the effectiveness of specific strategies. Similarly, the LinkedIn article about social impact in startups offers qualitative advice but no numerical evidence.

To obtain factual numeric evidence for strategy impacts, we would need to conduct specific research or gather data from startups that have implemented similar strategies. This might involve surveys, case studies, or analysis of performance data from multiple companies over time.

Without such data, it would be inappropriate to provide numerical estimates, as doing so would involve making up figures rather than relying on factual evidence. If you need specific quantitative data on strategy impacts, I would recommend conducting targeted research or consulting industry reports that focus on measuring the outcomes of similar strategies in startup environments

Product & Service Development

We can test the hypothesis that these suggested strategies are effective by using Evo Feedback

so, can I conclude that if I use your suggested strategies, in the hope of reaching the suggested target levels, I would be best off if I decomposed the larger strategies into, say, weekly value increments, and measured incrementally to see if they worked as we hoped?

Yes, that's an excellent conclusion and approach. Your suggestion aligns well with best practices in startup management and agile methodologies. Here's why this approach is beneficial:

1. Incremental Progress: Breaking down larger strategies into weekly value increments allows you to make steady, measurable progress. This approach is often referred to as "chunking" in project management.

2. Rapid Feedback: By measuring incrementally, you can quickly see if your strategies are working as intended. This allows for rapid course correction if needed.

3. Data-Driven Decision Making: Regular measurements provide you with actual data, rather than relying solely on assumptions or long-term projections.

4. Flexibility and Adaptability: Weekly increments allow you to be more agile and responsive to changes in your market or operating environment.

5. Risk Mitigation: By testing strategies in smaller increments, you reduce the risk associated with committing to a long-term strategy that may not yield the expected results.

6. Learning and Iteration: This approach facilitates a culture of continuous learning and improvement within your startup.

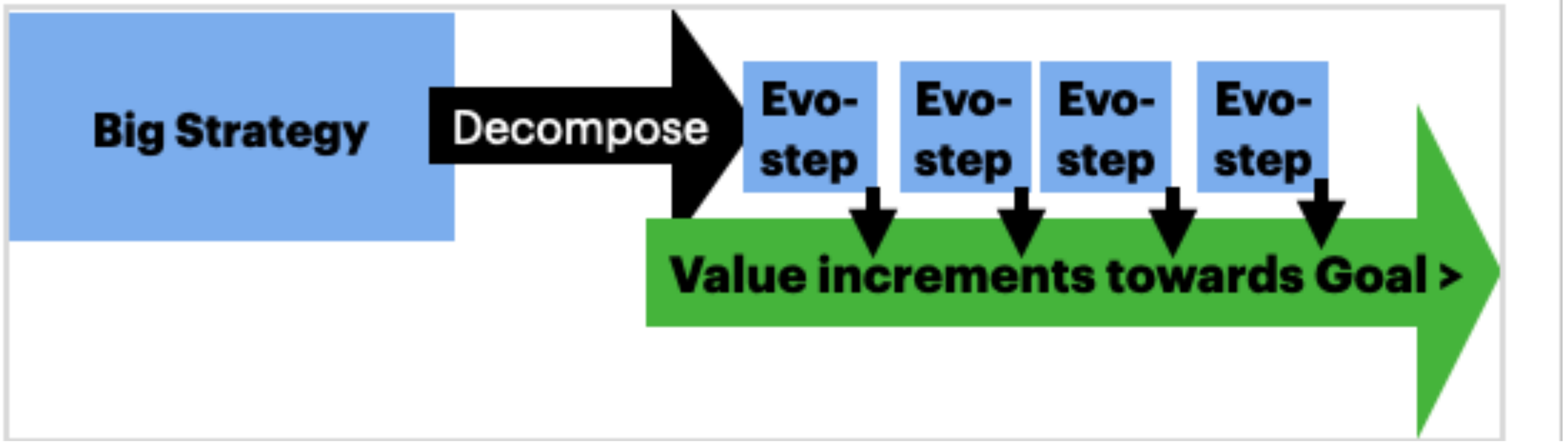
7. Motivation and Morale: Seeing regular, incremental progress can boost team morale and maintain motivation over the long term.

8. Resource Optimization: You can allocate resources more efficiently by focusing on what's working and adjusting what isn't.

This incremental, measurement-focused approach is particularly valuable in the context of social impact startups, where the effects of strategies may take time to manifest fully. By breaking down your strategies and measuring regularly, you're creating a framework for evidence-based growth and impact.

Remember to ensure that your weekly increments align with your overall long-term goals, and be prepared to adjust your strategies based on the data you gather. This approach will help you navigate the uncertainty inherent in startup environments while working towards your ambitious social impact goals.

Lets plan weekly value increases!



Source [Optima] Figure 4.2.3 B. **Decomposing into Evo-steps, not 'tasks', enables us to get measurable results, feedback, and planning improvement faster**

Wrong answer: Just tasks, not value. (*This* is 'Waterfall')

Maybe I was not clear enough, when I said Value Steps?

Certainly. I'll decompose the strategy "Sustainable Practices: Integrate sustainable materials and processes in product development" for an EU medical apparatus product and service into weekly value delivery increments. I'll provide 20 weeks of increments, with 3 lines of description for the first 10, and 10 lines of detail for the first increment. Then, I'll estimate the cumulative impact towards the short-term goal.

Weekly Value Delivery Increments:

Week 1: Material Assessment
- Conduct an inventory of current materials used in the product
- Research sustainable alternatives for high-impact materials
- Prioritize materials for replacement based on environmental impact and feasibility

Week 2: Supplier Engagement
- Identify potential suppliers of sustainable materials
- Initiate conversations with suppliers about sustainable options
- Evaluate supplier sustainability practices and certifications

Week 3: Design for Disassembly
- Review current product design for ease of disassembly
- Identify components that hinder recycling or reuse
- Propose design modifications to improve end-of-life recyclability

Week 4: Energy Efficiency Analysis
- Assess current energy consumption in manufacturing processes
- Identify high-energy consumption areas in the production line
- Research energy-efficient alternatives for manufacturing equipment

Week 5: Packaging Redesign
- Evaluate current packaging materials and design
- Research biodegradable or recyclable packaging alternatives
- Develop prototypes for sustainable packaging solutions

Week 6: Life Cycle Assessment (LCA) Initiation
- Define scope and boundaries of LCA study
- Gather data on current product life cycle impacts
- Begin modeling of product life cycle in LCA software

Week 7: Waste Reduction Strategy
- Analyze current waste streams in production
- Identify opportunities for waste reduction or recycling
- Develop a plan for implementing waste reduction measures

Week 8: Sustainable Manufacturing Processes
- Review current manufacturing processes for sustainability
- Research cleaner production techniques applicable to the product
- Propose pilot tests for sustainable manufacturing methods

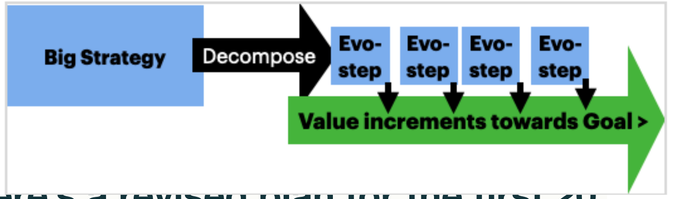
Week 9: Product Longevity Enhancement
- Analyze product failure modes and lifetime
- Identify components that limit product lifespan
- Propose design changes to extend product life and reduce replacement frequency

Week 10: Circular Economy Integration
- Assess potential for product take-back or refurbishment programs
- Research successful circular economy models in medical devices
- Develop a preliminary circular economy strategy for the product

Detailed description of Week 1: Material Assessment

1. Conduct a comprehensive inventory of all materials currently used in the product, including primary components, secondary materials, and packaging.
2. Create a database listing each material, its quantity, supplier, and known environmental impacts.
3. Research sustainable alternatives for materials identified as having high environmental impact, focusing on bio-based, recycled, or more easily recyclable options.
4. Consult with material scientists and sustainability experts to understand the latest advancements in sustainable materials suitable for medical devices.
5. Analyze the performance characteristics of potential sustainable materials to ensure they meet medical-grade requirements and regulations.
6. Prioritize materials for replacement based on a matrix considering environmental impact, cost, availability, and technical feasibility.
7. Develop a report detailing the findings of the material assessment and recommendations for sustainable material substitutions.
8. Initiate preliminary testing of top priority sustainable material alternatives to assess compatibility with the product's requirements.
9. Engage with the product development team to discuss potential design modifications that could facilitate the use of more sustainable materials.
10. Prepare a presentation for stakeholders outlining the potential environmental and economic benefits of transitioning to sustainable materials.

Planning Evo Value delivery steps



Understood. Let's focus on delivering incremental value *each week with tangible outcomes*. Here's a revised plan for the first 20 weeks, ensuring each step delivers a real value increment towards integrating sustainable materials and processes in product development for an EU medical apparatus product.

Week 1: Material Assessment and Initial Replacement

1. **Conduct a quick inventory of the top 3 high-impact materials currently used.**
2. **Identify and procure small quantities of sustainable alternatives for these materials.**
3. **Replace one non-critical component with a sustainable material in a small batch of products.**
4. **Test this small batch for compliance with medical standards.**
5. **Document the results and prepare to scale up successful replacements.**

Week 2: Supplier Engagement and Pilot Batch

1. Finalize agreements with suppliers for sustainable materials identified in Week 1.
2. Order sufficient quantities for a pilot production run.
3. Produce a pilot batch of products using the sustainable materials.
4. Collect feedback from initial users and internal testers.
5. Make necessary adjustments based on feedback.

Week 3: Design for Disassembly - Initial Implementation

1. Modify the design of one component to facilitate easier disassembly.
2. Produce a small batch of products with this new design.
3. Test the disassembly process and measure improvements.
4. Document the results and prepare to implement further design changes.

Week 4: Energy Efficiency - Quick Wins

1. Identify and implement one quick energy-saving measure in the production line (e.g., switching to LED lighting).
2. Measure the immediate reduction in energy consumption.
3. Document the energy savings and prepare a report for stakeholders.

Week 5: Packaging Redesign - Initial Rollout

1. Design and produce a small batch of biodegradable or recyclable packaging.
2. Use this new packaging for a subset of products.
3. Collect feedback from customers and logistics partners.
4. Measure the reduction in packaging waste.

Week 6: Life Cycle Assessment (LCA) - Initial Findings

1. Complete a quick LCA for the current product.
2. Identify the top 3 areas for improvement.
3. Implement one small change based on LCA findings (e.g., reducing material usage in one component).
4. Measure the immediate impact of this change.

Week 7: Waste Reduction - Initial Steps

1. Implement a recycling program for one type of waste in the production process.
2. Measure the amount of waste diverted from landfill in the first week.
3. Document the results and prepare to scale up the program.

Week 8: Sustainable Manufacturing - Pilot Test

1. Implement a pilot test of one cleaner production technique (e.g., water-based cleaning instead of solvent-based).
2. Measure the reduction in hazardous waste production.
3. Document the results and prepare to scale up the new technique.

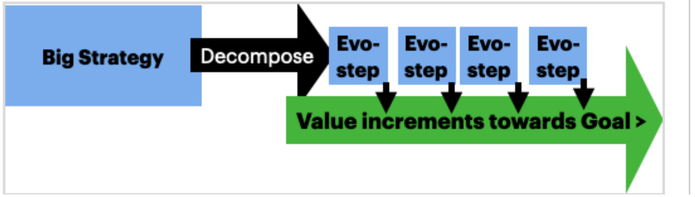
Week 9: Product Longevity - Quick Fix

1. Identify one component with a high failure rate.
2. Replace it with a more durable alternative in a small batch of products.
3. Measure the initial improvement in product longevity.
4. Collect feedback from users and document the results.

Week 10: Circular Economy - Initial Pilot

1. Launch a small-scale product take-back program.
2. Collect and refurbish a small number of returned products.
3. Measure the success of the refurbishment process.
4. Document the results and prepare to expand the program.

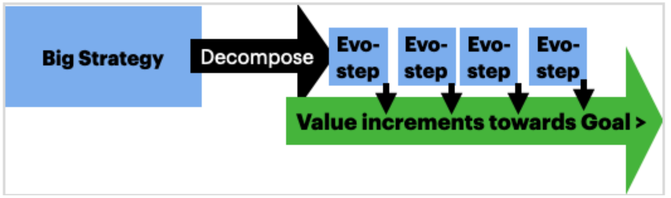
Detailed Description of Week 1: 'Evo-step "Material Assessment and Initial Replacement"



Detailed Description of Week 1: Material Assessment and Initial Replacement

1. **Conduct a quick inventory of the top 3 high-impact materials currently used in the product, focusing on those with the highest environmental impact.**
2. **Identify sustainable alternatives for these materials by consulting with suppliers and researching industry best practices.**
3. **Procure small quantities of these sustainable alternatives for testing purposes.**
4. **Choose one non-critical component (e.g., a casing or housing part) for initial replacement with a sustainable material.**
5. **Manufacture a small batch of products incorporating the sustainable material in the chosen component.**
6. **Conduct compliance testing to ensure the new material meets medical standards and regulatory requirements.**
7. **Collect initial feedback from internal testers and early adopters regarding the performance and quality of the new material.**
8. **Document the results of the compliance tests and feedback, highlighting any issues or areas for improvement.**
9. **Prepare a report summarizing the findings and outlining the next steps for scaling up the use of sustainable materials.**
10. **Present the report to stakeholders, including recommendations for further material replacements and potential impacts on sustainability goals.**

Weeks 11-20: Continued *Incremental* Value Delivery



Week 11: Scale Up Successful Material Replacement

1. Increase the production volume using the successfully tested sustainable material.
2. Monitor production for any issues.
3. Measure the environmental impact reduction.

Week 12: Further Design for Disassembly

1. Implement design changes for another component to facilitate disassembly.
2. Produce and test a small batch.
3. Measure improvements in recyclability.

Week 13: Energy Efficiency - Additional Measures

1. Implement another quick energy-saving measure (e.g., optimizing machine settings).
2. Measure the energy savings.
3. Document the results.

Week 14: Packaging Redesign - Full Rollout

1. Roll out the new sustainable packaging to all products.
2. Measure the reduction in packaging waste.
3. Collect customer feedback.

Week 15: LCA - Further Improvements

1. Implement another change based on LCA findings.
2. Measure the impact.
3. Document the results.

Week 16: Waste Reduction - Expand Program

1. Expand the recycling program to include another type of waste.
2. Measure the amount of waste diverted.
3. Document the results.

Week 17: Sustainable Manufacturing - Full Implementation

1. Fully implement the cleaner production technique tested in Week 8.
2. Measure the reduction in hazardous waste.
3. Document the results.

Week 18: Product Longevity - Further Improvements

1. Identify another component with a high failure rate.
2. Replace it with a more durable alternative.
3. Measure the improvement in product longevity.

Week 19: Circular Economy - Expand Program

1. Expand the product take-back program.
2. Collect and refurbish more returned products.
3. Measure the success of the expanded program.

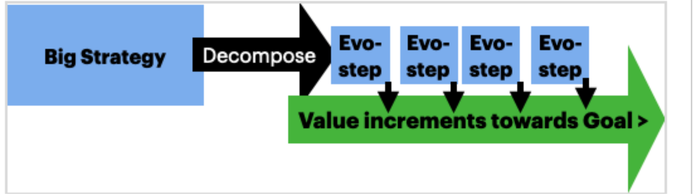
Week 20: Review and Plan Next Steps

1. Review the progress and impact of all implemented changes.
2. Measure the cumulative environmental impact reduction.
3. Plan the next set of incremental improvements.

Estimated Cumulative Impact Towards Short-Term Goal

impact of 40% ± 10% towards the short-term goal

After 20 weeks of implementing these incremental value delivery steps, we estimate a cumulative impact of 40% ± 10% towards the short-term goal of integrating sustainable materials and processes in product development.



This estimate considers:

- Successful initial replacements of high-impact materials.
- Implementation of energy-saving measures and waste reduction programs.
- Positive feedback from initial users and internal testers.
- Measurable reductions in environmental impact from packaging redesign and cleaner production techniques.
- Early successes in product take-back and refurbishment programs.

The uncertainty range (±10%) reflects potential variability in the success of each increment, regulatory compliance challenges, and the time required for some changes to fully manifest their impact.

2. Business Model Development

Business Model Development

2. Business Model Development

Objectives:

1. Ensure financial viability.
2. Maximize social and economic impact.
3. Maintain transparency.

Strategies:

1. Diversify Revenue Streams: Develop multiple revenue sources to ensure financial stability.
2. Impact Investment: Focus on investments that yield high social and economic returns.
3. Transparent Reporting: Implement rigorous reporting standards for transparency.

Strategy ->	Diversify Revenue Streams	Impact Investment	Transparent Reporting
Ensure financial viability	70 %	20 %	10 %
Maximize social impact	20 %	60 %	20 %
Maintain transparency	10 %	20 %	70 %
Strategy Score	100 %	100 %	100 %

3. Organizational Development

Organizational Development

3. Organizational Development

Objectives:

1. Enhance employee well-being.
2. Foster a strong organizational culture.
3. Promote professional development.

Strategies:

1. Well-being Programs: Implement comprehensive employee well-being programs.
2. Cultural Initiatives: Develop initiatives that reinforce organizational values.
3. Training Programs: Offer extensive training and development opportunities.

Strategy -> Objectives	Well-being Programs	Cultural Initiatives	Training Programs
Enhance employee well-being	60 %	20 %	20 %
Foster organiza-tional culture	20 %	60 %	20 %
Promote professi-nal develop-ent	20 %	20 %	60 %
Strategy Score	100 %	100 %	100 %

4. Community and Social Engagement

4. Community and Social Engagement

Objectives:

1. Increase community involvement.
2. Enhance social responsibility.
3. Strengthen collaboration.

Strategies:

1. Community Projects: Initiate and support community projects.
2. Social Initiatives: Allocate resources to social responsibility initiatives.
3. Partnerships: Form strategic partnerships with NGOs and other stakeholders.

Community and Social Engagement

Strategy	Community Projects	Social Initiatives	Partnerships
Increase community involvement	60 %	20 %	20 %
Enhance social responsibility	20 %	60 %	20 %
Strengthen collaboration	20 %	20 %	60 %
Strategy Score	100 %	100 %	100 %

5. Regulatory and Compliance

Objectives:

1. Ensure adherence to regulations.
2. Maintain high ethical standards.
3. Improve risk management.

Strategies:

1. Compliance Training: Regular training on regulatory requirements.
2. Ethical Audits: Conduct regular ethical audits.
3. Risk Assessment: Implement comprehensive risk assessment procedures.

Regulatory and Compliance

Strategy ->	Compliance Training	Ethical Audits	Risk Assessment
Ensure adherence to regulations	60 %	20 %	20 %
Maintain high ethical standards	20 %	60 %	20 %
Improve risk management	20 %	20 %	60 %
Strategy Score	100 %	100 %	100 %



8. Product and Service Qualities

8.1 Suggesting EU Medical Product and Services Qualities

With a suggested defined Scale of Measure

Based on the search results and the context of EU medical products and services, here are 10 critical qualities (5 for EU Medical Product and 5 for Corresponding medical product services) along with suggested scales of measure for each:

For EU Medical Product:

1. Safety

Scale: Number of adverse events reported per 10,000 units sold annually.

2. Effectiveness

Scale: Percentage improvement in relevant health outcomes compared to baseline or standard treatment.

3. Sustainability

Scale: Percentage of product components that are recyclable or biodegradable.

4. User-Centricity

Scale: User satisfaction score (1-10) based on ergonomics and ease of use surveys.

5. Compliance with EU Regulations

Scale: Percentage compliance with applicable EU medical device regulations and standards.

For Corresponding Medical Product Services:

1. Timeliness

Scale: Average response time for service requests in hours.

2. Competence

Scale: Percentage of service issues resolved on the first attempt.

3. Communication Effectiveness

Scale: Customer satisfaction score (1-10) regarding clarity and helpfulness of communication.

4. Empathy

Scale: Patient-reported empathy score (1-10) based on standardized surveys.

5. Continuous Improvement

Scale: Number of service process improvements implemented per quarter based on feedback and data analysis.

These qualities and scales are derived from the WHO's definition of quality healthcare [1], critical attributes of medical professionals [2][4], and essential skills for customer service in the medical industry [5]. They align with the need for effective, safe, and people-centered care, while also considering the specific requirements of EU medical products and services.

8.2 15 Medical Product Qualities

Certainly! I'll expand the critical qualities to 15 for both the EU Medical Product and the Corresponding Medical Product Services, including scales, 3-year goals, and core stakeholders for each. Then, I'll estimate the startup success with these expanded goals compared to using just the initial 5 required qualities.

EU Medical Product:

1. **Safety**
   Scale: Number of adverse events reported per 10,000 units sold annually
   Goal: <1 adverse event per 10,000 units
   Stakeholders: Patients, Regulatory Bodies, Healthcare Providers
2. **Effectiveness**
   Scale: Percentage improvement in relevant health outcomes compared to baseline
   Goal: 30% improvement
   Stakeholders: Patients, Healthcare Providers, Insurance Companies
3. **Sustainability**
   Scale: Percentage of product components that are recyclable or biodegradable
   Goal: 80% recyclable/biodegradable
   Stakeholders: Environmental Agencies, Consumers, Waste Management Companies
4. **User-Centricity**
   Scale: User satisfaction score (1-10) based on ergonomics and ease of use surveys
   Goal: Average score of 8.5/10
   Stakeholders: Patients, Healthcare Providers, Product Designers
5. **Compliance with EU Regulations**
   Scale: Percentage compliance with applicable EU medical device regulations
   Goal: 100% compliance
   Stakeholders: Regulatory Bodies, Legal Department, Quality Assurance Team
6. **Durability**
   Scale: Average product lifespan in years
   Goal: 5 years
   Stakeholders: Patients, Healthcare Providers, Manufacturing Team
7. **Precision**
   Scale: Percentage of measurements within accepted margin of error
   Goal: 99.9% precision
   Stakeholders: Healthcare Providers, Patients, Quality Control Team
8. **Interoperability**
   Scale: Number of compatible systems/devices
   Goal: Compatible with 90% of relevant systems
   Stakeholders: IT Departments, Healthcare Providers, Software Developers
9. **Energy Efficiency**
   Scale: Energy consumption compared to industry standard (percentage)
   Goal: 25% less energy consumption than standard
   Stakeholders: Environmental Agencies, Hospitals, Energy Consultants
10. **Cost-Effectiveness**
    Scale: Cost per unit of health outcome improvement
    Goal: 20% lower cost per unit of improvement than competitors
    Stakeholders: Insurance Companies, Healthcare Providers, Patients
11. **Portability**
    Scale: Weight in kilograms
    Goal: 30% lighter than current market standard
    Stakeholders: Healthcare Providers, Patients, Logistics Team
12. **Maintenance Ease**
    Scale: Average time required for routine maintenance (hours)
    Goal: 50% reduction in maintenance time
    ***Stakeholders*: Maintenance Staff, Healthcare Providers, Product Engineers**
13. **Data Security**
    Scale: Number of security breaches per year
    Goal: Zero breaches
    Stakeholders: IT Security Team, Patients, Regulatory Bodies
14. **Customizability**
    Scale: Number of customizable features
    Goal: 10 customizable features
    Stakeholders: Healthcare Providers, Patients, Product Development Team
15. **Noise Level**
    Scale: Decibel level during operation
    Goal: 20% reduction in noise level
    Stakeholders: Patients, Healthcare Providers, Acoustic Engineers

8.3 15 Medical Service Qualities

Certainly! I'll expand the critical qualities to 15 for both the EU Medical Product and the Corresponding Medical Product Services, including scales, 3-year goals, and core stakeholders for each. Then, I'll estimate the startup success with these expanded goals compared to using just the initial 5 required qualities.

Corresponding Medical Product Services:

1. **Timeliness**
   Scale: Average response time for service requests in hours
   Goal: <2 hours
   Stakeholders: Patients, Healthcare Providers, Service Team
2. **Competence**
   Scale: Percentage of service issues resolved on the first attempt
   Goal: 95% first-time resolution
   Stakeholders: Service Technicians, Healthcare Providers, Quality Assurance Team
3. **Communication Effectiveness**
   Scale: Customer satisfaction score (1-10) regarding clarity and helpfulness
   Goal: Average score of 9/10
   Stakeholders: Patients, Healthcare Providers, Communication Team
4. **Empathy**
   Scale: Patient-reported empathy score (1-10) based on standardized surveys
   Goal: Average score of 9/10
   Stakeholders: Patients, Healthcare Providers, Customer Service Team
5. **Continuous Improvement**
   Scale: Number of service process improvements implemented per quarter
   Goal: 5 improvements per quarter
   Stakeholders: Service Team, Quality Assurance, Management
6. **Availability**
   Scale: Percentage of time services are available (uptime)
   Goal: 99.9% uptime
   Stakeholders: Patients, Healthcare Providers, IT Team
7. **Training Effectiveness**
   Scale: Post-training competency assessment scores (percentage)
   Goal: 95% average score
   Stakeholders: Healthcare Providers, Training Team, Human Resources
8. **Remote Support Capability**
   Scale: Percentage of issues resolved remotely
   Goal: 80% remote resolution
   Stakeholders: IT Support Team, Healthcare Providers, Patients
9. **Multilingual Support**
   Scale: Number of languages supported
   Goal: Support for 10 major EU languages
   Stakeholders: Patients, Healthcare Providers, Language Specialists
10. **Documentation Quality**
    Scale: Comprehension score of provided documentation (1-10)
    Goal: Average score of 9/10
    Stakeholders: Healthcare Providers, Technical Writers, Regulatory Bodies
11. **Proactive Maintenance**
    Scale: Percentage of potential issues identified and resolved before failure
    Goal: 90% proactive resolution
    Stakeholders: Service Team, Healthcare Providers, Data Analysis Team
12. **Customer Feedback Integration**
    Scale: Percentage of customer suggestions implemented
    Goal: 30% implementation rate
    Stakeholders: Patients, Healthcare Providers, Product Development Team
13. **Service Level Agreement (SLA) Adherence**
    Scale: Percentage of SLAs met
    Goal: 99% SLA adherence
    Stakeholders: Healthcare Providers, Legal Team, Service Management
14. **Knowledge Base Effectiveness**
    Scale: Percentage of queries resolved using knowledge base
    Goal: 70% self-service resolution
    Stakeholders: Healthcare Providers, IT Team, Technical Writers
15. **Environmental Impact of Service**
    Scale: Carbon footprint of service operations (CO_2 equivalent)
    Goal: 30% reduction in carbon footprint
    Stakeholders: Environmental Agencies, Service Team, Sustainability Officers

8.4 Estimated Startup Success:

Using 15 quality goals instead of 5 for each product and service would likely increase the startup's chances of success significantly

1. **Comprehensive Coverage: The expanded set of qualities addresses a broader range of stakeholder needs and market requirements, potentially leading to a more robust and competitive offering.**
2. **Detailed Focus: With more specific goals, the startup can allocate resources more effectively and track progress more precisely.**
3. **Stakeholder Satisfaction: By considering a wider array of stakeholders, the startup is more likely to create products and services that meet diverse needs, potentially leading to higher adoption rates and customer satisfaction.**
4. **Risk Mitigation: The expanded set of qualities includes factors like data security and environmental impact, which can help the startup avoid potential pitfalls and regulatory issues.**
5. **Market Differentiation: With a more comprehensive approach to quality, the startup can potentially differentiate itself from competitors who might focus on fewer aspects.**
6. **Long-term Vision: The 3-year goals provide a clear long-term vision, which can guide strategic planning and investment decisions.**

Estimated success increase:

Using just the initial 5 required qualities might result in a 60% chance of startup success.

Using the expanded 15 qualities for each product and service could increase the chances of success to approximately 80-85%.

This estimate assumes that the startup effectively implements strategies to meet these goals and adapts to market feedback.

The increased chance of success comes from a more thorough understanding of the market needs, better alignment with stakeholder expectations, and a more comprehensive approach to product and service quality.

8.5 How many product qualities do we need to manage?

The relationship between the number of quality requirements and market success/ profitability can be described as an inverted U-shaped curve:

Based on the search results and general principles of quality management, I'll create a table and describe a curve that illustrates the relationship between the number of quality requirements and market success/ profitability.

Curve Description:

The relationship between the number of quality requirements and market success/profitability can be described as an inverted U-shaped curve:

1. Initial Rise (1-10 requirements):
   - As the number of quality requirements increases from 1 to 10, both market success and profitability rise sharply.
   - This represents the basic quality standards necessary for market entry and customer satisfaction.
2. Peak (11-15 requirements):
   - The curve reaches its peak around 11-15 quality requirements.
   - This represents the optimal balance where comprehensive quality management leads to high market success and profitability.
   - This aligns with our earlier discussion of 15 quality requirements for both products and services.
3. Plateau (16-20 requirements):
   - Market success continues to rise slightly but at a slower rate.
   - Profitability begins to decline due to increased costs of implementing and maintaining additional quality measures.
4. Decline (21+ requirements):
   - Beyond 20 requirements, market success plateaus or slightly declines.
   - Profitability decreases more sharply due to diminishing returns and increased complexity.

The curve illustrates that while increasing quality requirements generally improves market success, there's a point of diminishing returns where the cost of implementing additional quality measures outweighs the benefits. The optimal range appears to be around 11-15 quality requirements, balancing comprehensive quality management with operational efficiency and profitability.

This aligns with the principle of focusing on core quality management principles (as outlined in ISO 9001) while avoiding over-complication that can lead to reduced efficiency and increased costs. It also supports our earlier recommendation of using 15 quality goals for products and services to maximize startup success

Number of Quality Requirements	Market Success	Profitability	Overall Impact
1-5	Low	Low	Limited
6-10	Moderate	Moderate	Improving
11-15	High	High	Optimal
16-20	Very High	Moderate	Diminishing
21	Very High	Low	Decreasing

Table: Number of Quality Requirements vs. Market Success and Profitability

8.6 20 Most useful medical product qualities

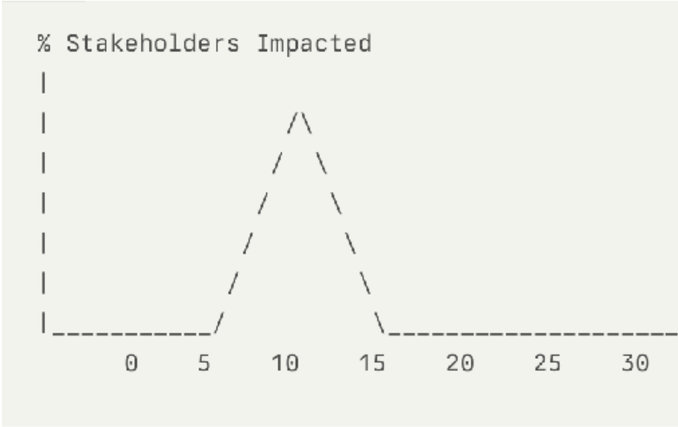
Based on the search results and the context of EU medical products and services, here are the 20 most useful combined EU Product & Service quality requirements, specified with Planguage parameters, and ordered by the percentage of stakeholders they would satisfy. Each requirement includes Ambition, Scale, Meter, Tolerable, Goal in 3 Years, % of 100 Stakeholders Impacted, and Rough ROI for the Quality.

Quality Requirement	Ambition	Scale	Meter	Tolerable	Goal (3 Years)	% of 100 Stakeholders Impacted	Rough ROI for the Quality
Safety	Minimize adverse events	Number of adverse events per 10,000 units sold annually	Weekly reports from healthcare providers	<2 adverse events per 10,000 units	<1 adverse event per 10,000 units	95 %	High
Effectiveness	Maximize health outcome improvement	Percentage improvement in health outcomes compared to baseline	Quarterly clinical trial results	20% improvement	30% improvement	90 %	High
Compliance with EU Regulations	Ensure full compliance with EU medical device regulations	Percentage compliance with applicable regulations	Annual regulatory audits	95% compliance	100% compliance	85 %	High
Timeliness	Respond to service requests promptly	Average response time for service requests in hours	Monthly service logs	<4 hours	<2 hours	80 %	High
Communication Effectiveness	Ensure clear and helpful communication	Customer satisfaction score (1-10) regarding communication	Bi-annual customer surveys	Average score of 8/10	Average score of 9/10	75 %	Medium
Durability	Extend product lifespan	Average product lifespan in years	Annual product durability tests	3 years	5 years	70 %	Medium
User-Centricity	Enhance user satisfaction	User satisfaction score (1-10) based on ergonomics and ease	Bi-annual user surveys	Average score of 7/10	Average score of 8.5/10	65 %	Medium
Data Security	Ensure data protection	Number of security breaches per year	Annual IT security audits	<2 breaches per year	Zero breaches	60 %	High
Sustainability	Increase use of sustainable materials	Percentage of product components that are recyclable	Annual sustainability reports	50% recyclable/biodegradable	80% recyclable/biodegradable	55 %	Medium
Competence	Resolve service issues effectively	Percentage of service issues resolved on the first attempt	Monthly service performance reviews	85% first-time resolution	95% first-time resolution	50 %	Medium
Training Effectiveness	Improve staff competency	Post-training competency assessment scores (percentage)	Quarterly training assessments	85% average score	95% average score	45 %	Medium
Interoperability	Ensure compatibility with other systems	Number of compatible systems/ devices	Annual compatibility tests	Compatible with 70% of relevant systems	Compatible with 90% of relevant	40 %	Medium
Proactive Maintenance	Prevent issues before they occur	Percentage of potential issues identified and resolved before failure	Monthly maintenance logs	70% proactive resolution	90% proactive resolution	35 %	Medium
Customizability	Offer customizable features	Number of customizable features	Annual product reviews	5 customizable features	10 customizable features	30 %	Low
Remote Support Capability	Resolve issues remotely	Percentage of issues resolved remotely	Monthly remote support logs	60% remote resolution	80% remote resolution	25 %	Medium
Availability	Ensure service availability	Percentage of time services are available (uptime)	Monthly service availability reports	98% uptime	99.9% uptime	20 %	Medium
Noise Level	Reduce operational noise	Decibel level during operation	Annual noise level tests	30% reduction in noise level	20% reduction in noise level	15 %	Low
Packaging Redesign	Use sustainable packaging	Percentage of packaging that is biodegradable or recyclable	Annual packaging audits	50% biodegradable/recyclable	80% biodegradable/recyclable	10 %	Low
Life Cycle Assessment	Improve product life cycle impact	Number of life cycle assessments conducted	Annual life cycle assessment reports	1 assessment per product	3 assessments per product	5 %	Low
Customer Feedback Integration	Implement customer suggestions	Percentage of customer suggestions implemented	Quarterly customer feedback reviews	20% implementation rate	30% implementation rate	5 %	Low

8.7 So, how many quality requirements should we handle?

Summary:

- **Initial 5 Quality Requirements: Addressing these would likely impact around 50-60% of stakeholders.**
- **Expanded 15 Quality Requirements: This would maximize stakeholder satisfaction, potentially impacting up to 85-90% of stakeholders.**
- **Beyond 20 Quality Requirements: Adding more requirements may lead to diminishing returns, impacting fewer additional stakeholders while increasing complexity and costs.**



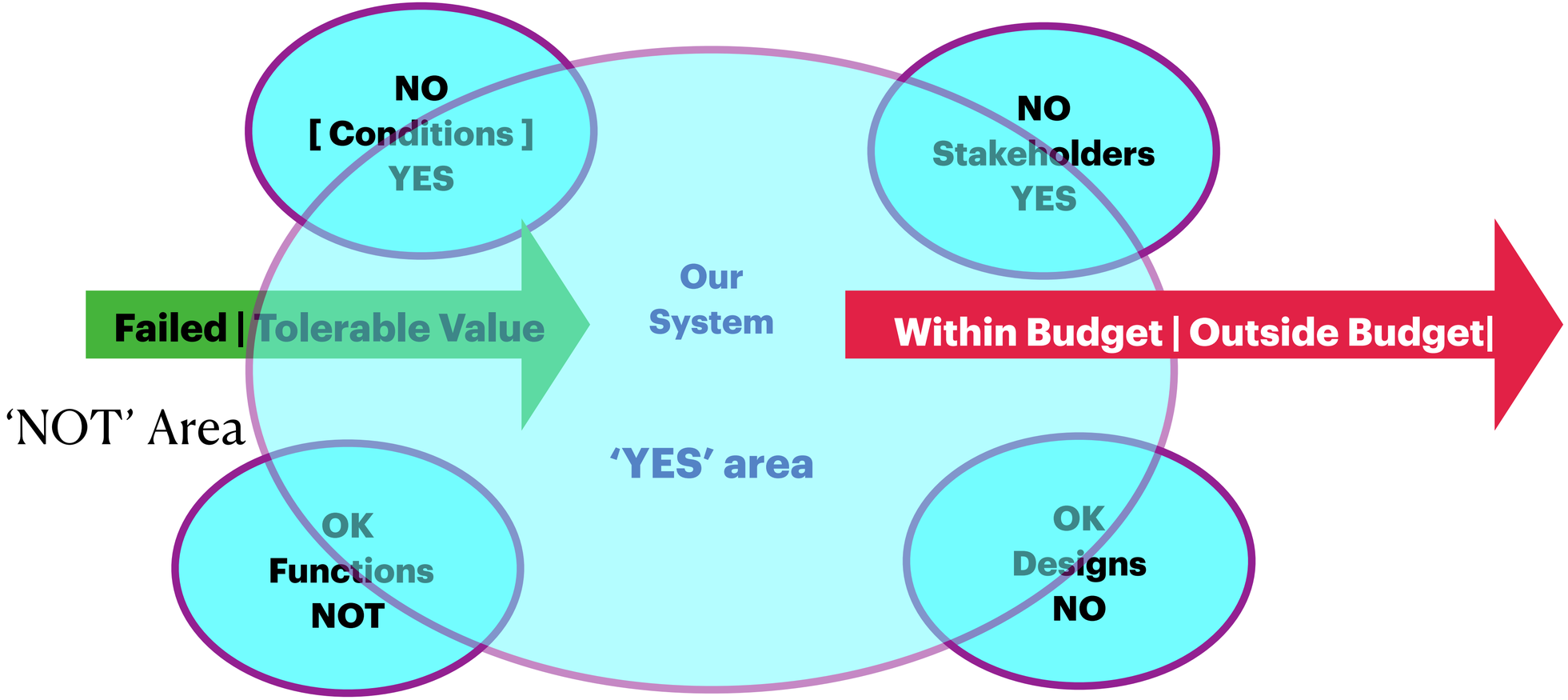
By focusing on the 15 most impactful quality requirements, the startup can achieve optimal market success and profitability, balancing comprehensive quality management with operational efficiency.



9. Constraints

9.0 Constraints

The system border, big oval, is the border between what is 'allowed' (YES) and what is forbidden (NOT) .
***Constraints* are *intentional* (*not 'accidental'*): there is a *stakeholder* and a *reason* behind them.**
All planning, decisions, and reviews, must 'respect' intentional constraints, until they are officially lifted, or adjusted.
For all other planning and decision-making: everything is 'allowed', if not officially constrained.



9.1 [Guides] Defined

Constraints & Tools -> Guides

In Planguage terminology [CE] I have pinned down a ‘Constraint’ as something that **intentionally restricts** us. As opposed to inadvertent, or accidental, non-intended restrictions.

I initially (maybe before 2000) believed that ‘all requirements are constraints’.

But that did not sit well with a concept like ’99.98% Availability’ as a ‘quality goal’. It is *not* a restriction, nor *intended* as one. It is a motivation to *get* to that level. And, if you accidentally get to 99.998% availability, nobody is going to restrict you to 99.98%.

Value Goals, and other targets, **guide** you in a **direction** (Availability, % Uptime, in this case) and a **degree** (happy with 99.98%), by a **time** (deadline).

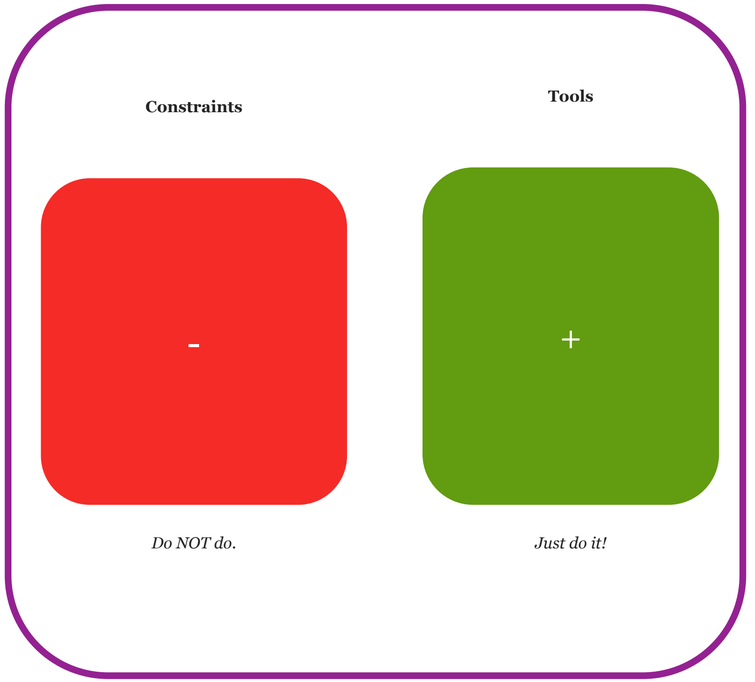
I knew that a Goal/Target/Objective, *does* affect behavior, of designers and testers. But I could not categorize it as a *restriction*, or *constraint*.

Now, I am going to categorize it as a **Tool,** and note that Tools belong, with Constraints, to a category of ideas that *guide* behavior: ‘**Guides**’.

Both *Constraints* (Do NOT, Restrict); and *Tools* (Do, Motivate Towards) are ‘**Guides’.**

And, as the term indicates, they both *guide* behavior;
behavior of what we ‘can do’ *(should do)* and ‘can *not* do’.

Guides



They guide our behaviour, with good intent

9.2 Constraints Logic

- Constraints are included in the very many objects (like stakeholders, values, strategies) that we need to 'balance', in planning and decision-making.
- The fact that constraints are *from* stakeholders, means they can be changed at the stakeholder's *will*, and also changed because of the *power* level changes of stakeholders. [SE]
- The fact that there is a reason, a justification, a 'rationale', behind a constraint, means that when that basis for the reason changes, then the constraint *can* change or be *eliminated*.
- If a particular constraint, is clearly standing in the way of some efficiency (Values/Costs) achievement, then the planner can seek to remove or relax the constraint, by appealing to the stakeholder, or ignoring their will.
    - I have found that 'questioning a constraint' which 'everybody assumes is valid', and removing or relaxing the constraint, by design, is powerful in enabling me to create dramatic improvements quickly. Don't be shy in challenging the 'accepted wisdom' !
- New constraints, can emerge, at any time, from any stakeholder source, and for any reason. They can render a current system invalid. So, keep your eyes open for *'constraint emergence'*.
- And keep your eyes open for *constraint 'decline'*. That means: the power of the constraint stakeholder is <u>declining</u>, and/or the *constraint reason basis is* changing.

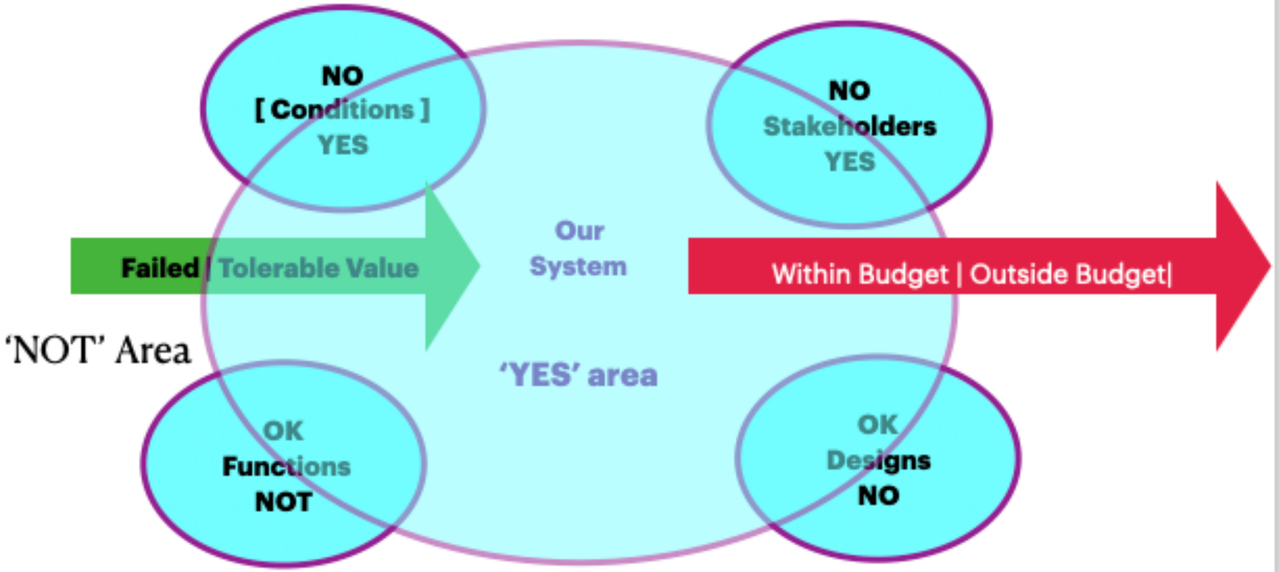


Figure 9.2 Source Tom Gilb 24July 2024 for Optima book.

9.3 What about things that are 'not' Constraints: inside the big oval

- **Values**
  - Any **Tolerable, or better**, value level can be planned and hopefully achieved, including any Goal and Stretch level.
- **Resources**
  - Any resources **within a Budget** can be used, at any time, for any purpose. Until their use depletes the Budget level.
- **Stakeholders**
  - Any non-constrained stakeholders, can be considered as valid, regarding any of their non-constrained values, and non-constrained value levels, or not, *whenever*.
- **Strategies (Designs)**
  - Any non-constrained strategies can be planned and deployed, however unconventional or shocking, or original. To serve the system efficiency purposes. As long as they do not threaten value Tolerable status, or Budgets.
- **Conditions**
  - Non-constrained Conditions ([when], [where], [who], [what], [why], [how]) can be planned, and taken into consideration, in any set or combination, when we analyze that their priority serves our higher purposes (system *efficiency*, generally, and higher level Fundamental Objectives).
- **Functions**
  - Any non-constrained functions (what a system '*does*') can be proposed, and deployed, whenever it suits the priorities of powerful stakeholders.

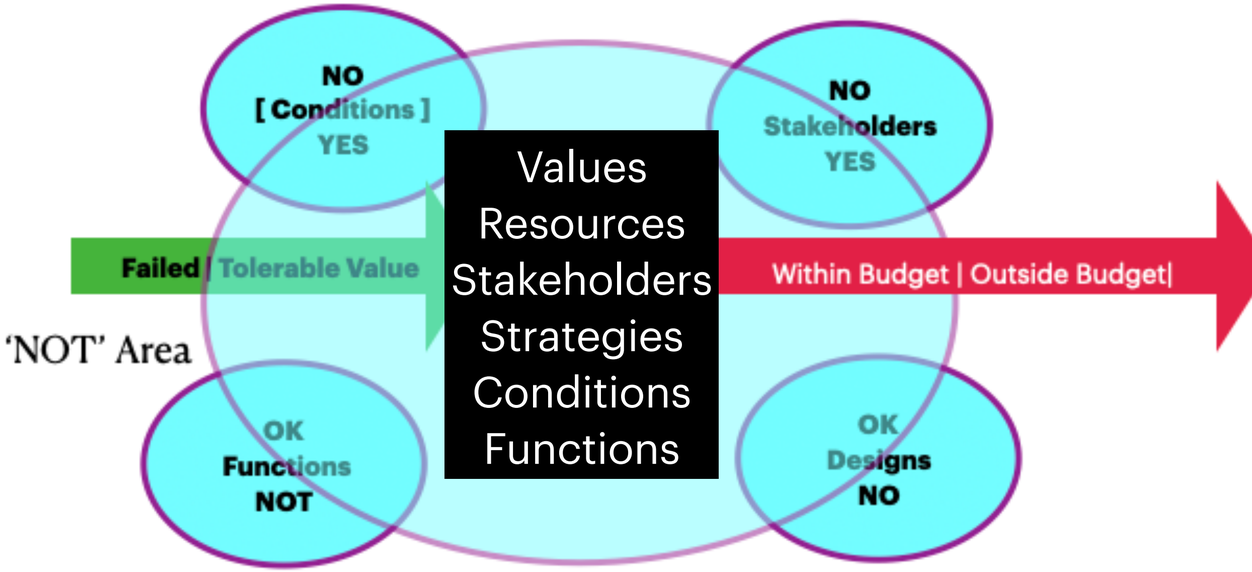


Figure 9.2 Source Tom Gilb 24July 2024 for Optima book.

The planning and decision-making consequence of these observations, is that there is usually an enormously large quantity, and combination, of potential factors, the 'multi-dimensions', which we can and might consider. This makes formal engineering methods, such as Planguage, a necessary tool to keep track of these many factors' potential, and the many factors' real interactions. These need some 'Optimization', some balance, the theme of this book.

9.4 Constraints Consequences

- There are several practical consequences of the constraints.
- Constraints, a type of requirements [CE], Figure 9.6, must be in **writing**, not in our minds, nor can we assume that *'surely everybody knows we cannot do X'.*
- Constraints need to be '**official**' in some sense; **approved**, in a quality controlled version, so that they are not misunderstood.
- Contraints need to be **easily available**, and **communicated** to all planners, and to all stakeholders, who need to know about them, in order to be able to heed them
- Constraints need to have **clearly-identified stakeholders**, and their ***rationale* needs to be documented,** in their constraint specification. (See 9.14, 9.15, 9.16 for examples of documenting the rationale.
- Constraints need to be a **formal consideration,** in any *quality control or reviews* of plans. On a checklist, in the rules, in a defined process.
- There needs to be a **policy** regarding, ignoring or bending, constraints. Who can do it, who can permit it, documenting that it is done. This can also more specifically be put in the individual constraint specification. See 9.14 here, 'Authority', and 'Responsible' parameters.
- When unusual or controversial planning decisions are made, due to constraints which other people are not explicitly aware of, yet, the *causal constraint, which the Planner was aware of,* needs to be identified, justified and explained - to people; so they can *understand*, and *accept*, the *apparently 'strange'* planning decisions. This *causal constraint* needs to be in writing, embedded in, or attached to, the 'strange decisions', for *posterity*. Put it in the specification, perhaps as a '*Rationale*' parameter, or an '*Assumption*'. See 9.14 here for an example.

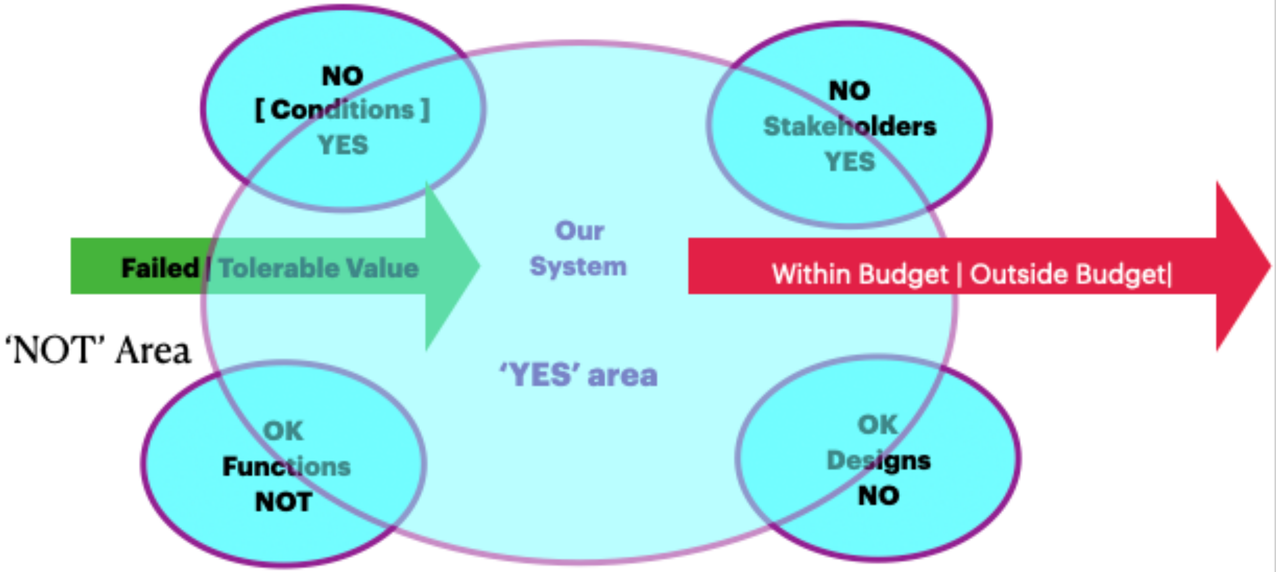


Figure 9.2 Source Tom Gilb 24July 2024 for Optima book.

9.6 Constraints as Requirements

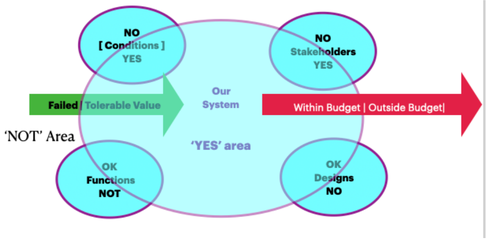


Figure 9.2 Source Tom Gilb 24July 2024 for Optima book.

- Constraints are a sub-set of ***requirements***' categories. There are two basic ways of expressing a constraint.
  - **Binary**: YES or NO conditions. *(Simple!)*
  - **Scalar**: a level or degree, on a defined Scale of measure, for a Value or Resource. *(Much more tricky!)*
  - Requirements have some '*states*':
    - **Proposed** future states (example: 'Wish')
      - Might *not be committed* to, in practice
    - **Agreed,** committed, balanced, future states (example: 'Goal')
      - Might *not be delivered* in practice
    - **Achieved** or, it is the 'current **Status**'
      - Requirement delivered, or not, in real world system.
- Notice in Fig. 9.6 how many of the requirements categories have the word 'Constraint' in them.

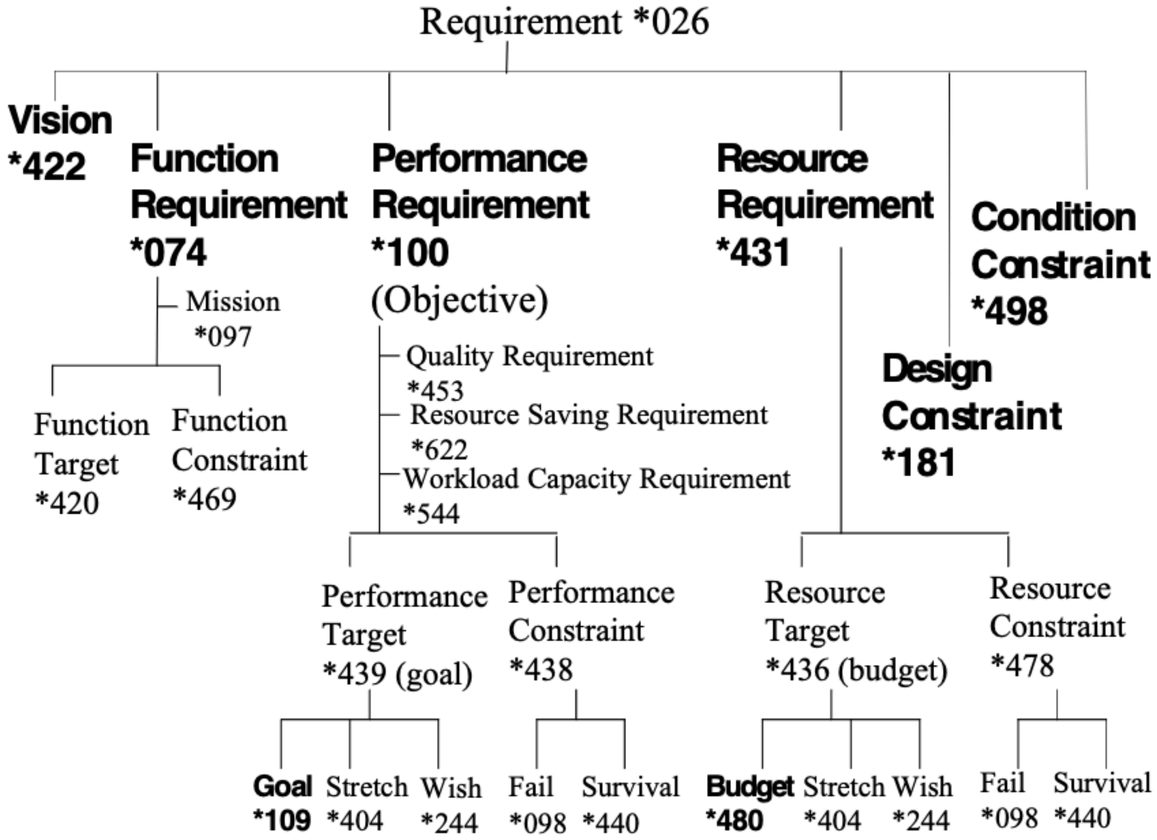


Figure 9.6. Requirements from [CE] Glossary Showing various constraints, and non-constraints requirements.
*nnn signals a formally defined Planguage Concept. [**CG-CE**]

9.7 What about 'accidental' Constraints

- An 'accidental constraint' is a constraint which is *not* the *intention* of any defined *stakeholder*, for any defined *intentional reason*.
  - It could be a *misunderstanding*
  - It could be an *obsolete* constraint, and we are not updated
  - The *conditions* that led the stakeholder to once demand the constraint, might *no longer exist*
  - The stakeholder behind it, has *lost power*, or is 'constrained', out of our scope.
  - Or, for any number of other similar reasons: the constraint is *not valid*, and we should not 'accidentally' adhere to it.
  - An accidental constraint is a ***threat***. It can hypothetically destroy, or hurt, our project, or system.

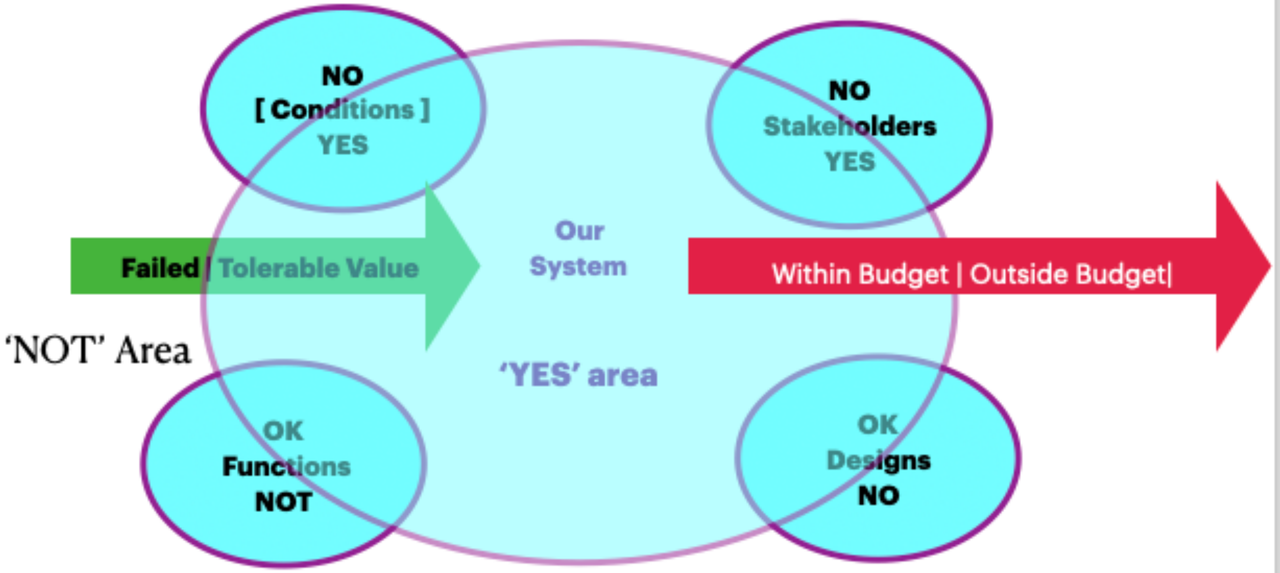


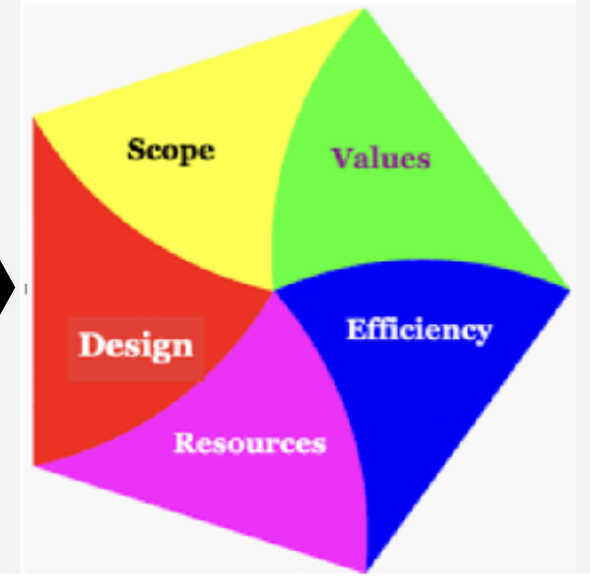
Figure 9.2 Source Tom Gilb 24July 2024 for Optima book.

The consequence of this 'accidental constraint' concept is that we need to have a quality assurance process that reduces the probability, that we waste resources, dealing with an invalid, accidental, constraint.

9.8 'Constraints': *Penta* factors, and the *Evo Cycle*

Constraints permeate the entire planning process

All factors can be constrained



Basic '[Penta]' Model of system attributes

- All Penta factors, and combinations of them, *can* be constraints.
- That implies that the potential, and real, presence of applicable constraints is a part of concern for all stakeholders, who deal with any Penta factors: *for example decision makers, planners, implementers, business analysts, architects, system engineers, quality assurance.*
- Any and all Evo cycle sub-processes, will probably need to consider potential applicable constraints.
  - In particular, during any '*Learn*' phase of the evo Cycle, there needs to be a question of- *if, valid constraints have been respected*, and *if invalid constraints have been 'accidentally' applied*.



Evo cycle

9. 9. Tools as Guides

Constraints + Tools = 'Guides'

- Constraints are like a fence, within which, you *must play*. They restrict your freedom to go outside of the boundaries.
- There is another, *similar to constraints,* 'guide': more of a 'positive' nature [Guides]. I call these guides '**Tools**'.
  - Tools are things like *standards, templates, patterns, principles, policies, visions, target objectives, processes, and rules.*
  - Tools are intended to be 'wisdom' about smart ways to work. They are of the 'DO THIS' category. Whereas *constraints* are the '**DON'T** DO THIS' category
  - Tools are the 'plus', +. The *'here is a good practice !'*
  - Constraints are the 'minus' - the '*watch out !'*
- Both are a **'voice of experience'** - they are '*safe guides'* [Guides]

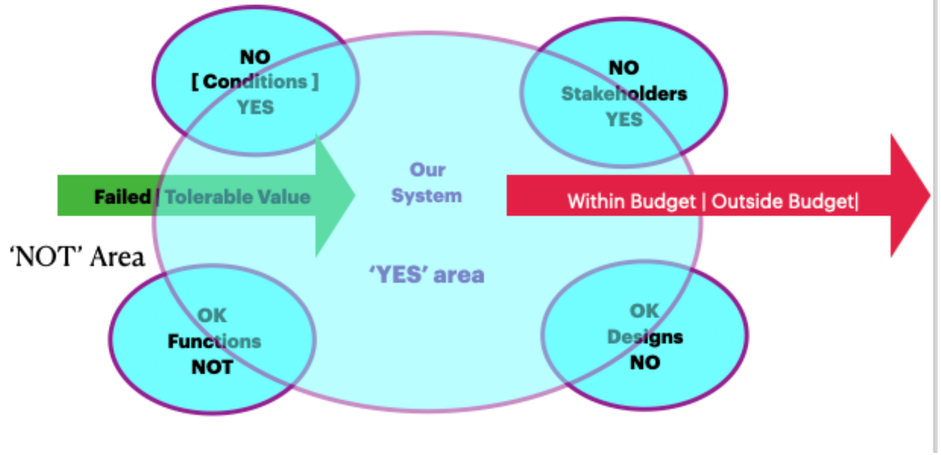


Figure 9.2 'Constraints are a 'fence', saying don't go outside me. Constraints are *hopefully* 'wisdom', and 'tribal knowledge'.

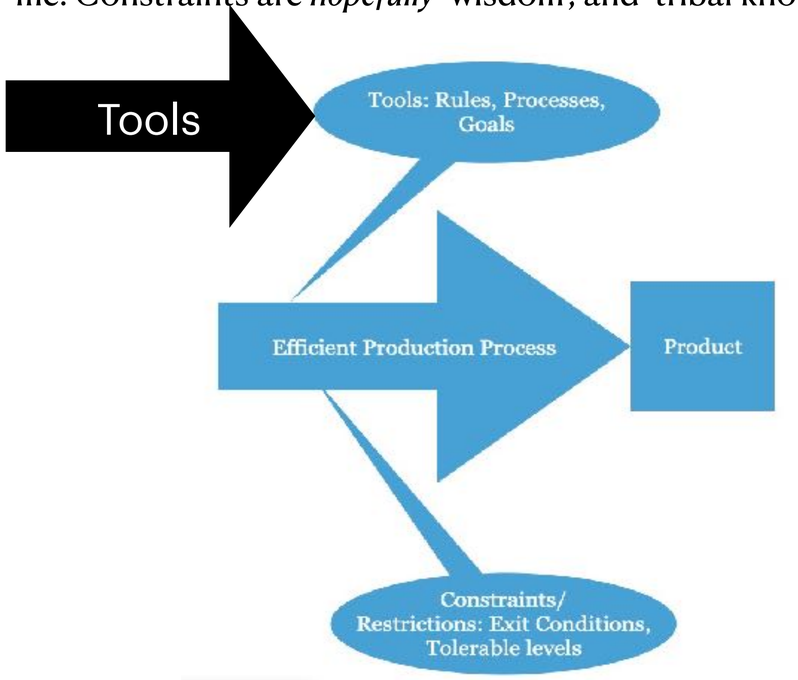
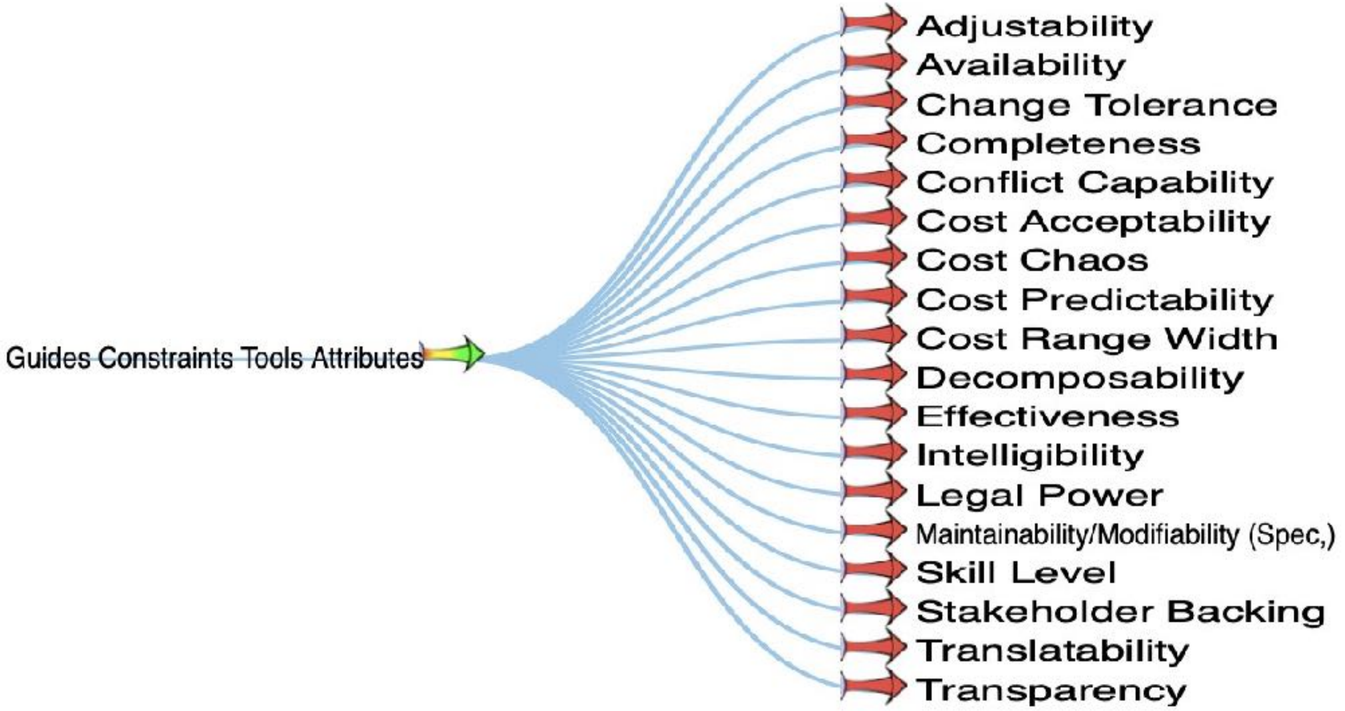


Figure 9.9 Tools are a partner guide, a partner with 'Constraints. Both help to guide us towards successful practices and directions.

9.10 Constraints and Tools are not so simple, they have many *Value* attributes!

An example of a set of attributes for Constraints (and later for Tools): for all Guides

- The main spirit of this book, is to bring out the *multiplicity of factors* we can consider, early.
- So, I will now add to your *treasures of insights*, the simple observation, that all constraints, and all tools, like every thing in this universe, have a *set of performance characteristics*, or 'valued attributes', or 'qualities'. See the sample list, in the diagram at right.
- Take any *real for you* instance of a constraint, or tool (example a 'planning method'), and go through this attribute list, and ask yourself if the constraint or tool (Guide) *has* such a property.
- Hopefully you see, that possibility, or connection to your Constraint/Tool for *all,* or some, of the attributes. And maybe you can see that if the attribute were *extremely* good or bad, it would make a material difference to your planning.
  - The consequence of this *(all constraints and tools have many critical attributes)*,
    - and the fact that *these* values *can* be quantified and measured;
    - and the fact that extreme values can make a 'Guide' useless or useful;
    - means - that in 'large-scale critical situations',
    - we can *make use* of these facts, in order to **make smart choices, *early*. Better planning.**
  - We can specify some or all of these attributes, for - for example *'a training session for all teachers'*
  - And then use an Impact Estimation Table, to decide *which* Training Course and its attributes, best match *our* value objectives, as specified by these attribute objectives. See 9.22, and 9.23 for an example.
  - We can avoid the problem, of *learning about bad attributes of the training course, **after** 10,000 angry teachers are subjected to it*. We *can* do smarter decisions, and planning, and choices.



Source: Gilb Mac/Cases/ValPlan Clips/CONSTRAINTS GUIDES TOOLS BOOK 2023/Guides Constraints Tools Attributes 2023-08-03 at 00.17.04

9.11 Defining Constraint Attributes

Constraint Qualities Defined*:*

Transparency: the degree that we can see the complete reasoning, behind the constraint, including stakeholders, history, purposes, high level objectives, experience in applying the constraint. *See 9.12 for detailed spec. Example*

Intelligibility: the constraint specification is fully and correctly understood, by the intended readership, as far as it is written. I.e. we are not looking at Completeness, *yet*.

Completeness: the Constraint Specification has all necessary information for specific defined purposes, *such as 'deciding to waiver it this time'*.

Maintainability/Modifiability (Spec,): The required total systems effort to update, or change a Constraint Specification, including hearings, and reviews, with all critical stakeholders. Mean Time to Modify.

Decomposability (like with [Scale Qualifiers]): the inherent or explicit ability to decompose a Constraint into sub-constraints, for purposes of independent prioritization. *For example by stating it as a hierarchy or set of sub-constraints, or by using '[Scale Qualifiers]' or other '[Qualifiers]'*

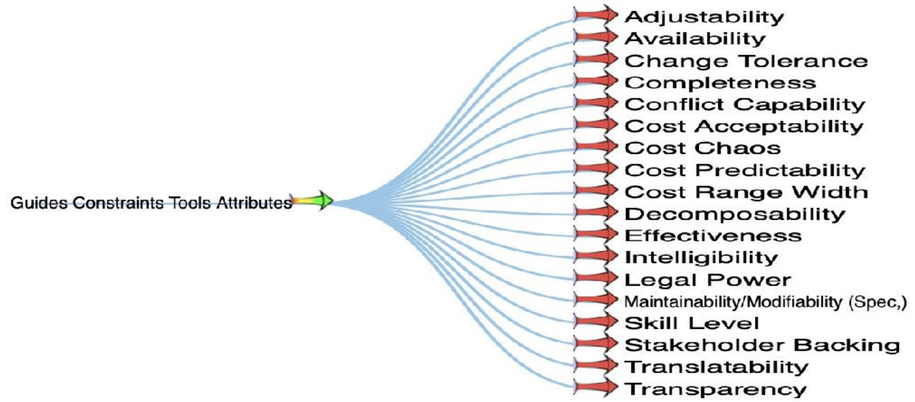
Translatability: Ease of translation into any human language, or any other specification language, by automatic or manual means.

Effectiveness: the inherent power of the Constraint to inhibit, constrain or restrain as intended, in practice. *For example by automated means, by legal enforcement, by organizational pressures*.

Adjustability: the degree to which we can initially, then during the system life cycle, adjust the constraint, by designs or other specifications, to suit the real and intended environment of the Constraint application. *For example by ratcheting up a lower numeric constraint in tact with market competition*.

Availability: the % of calendar time actually available for productive work or action

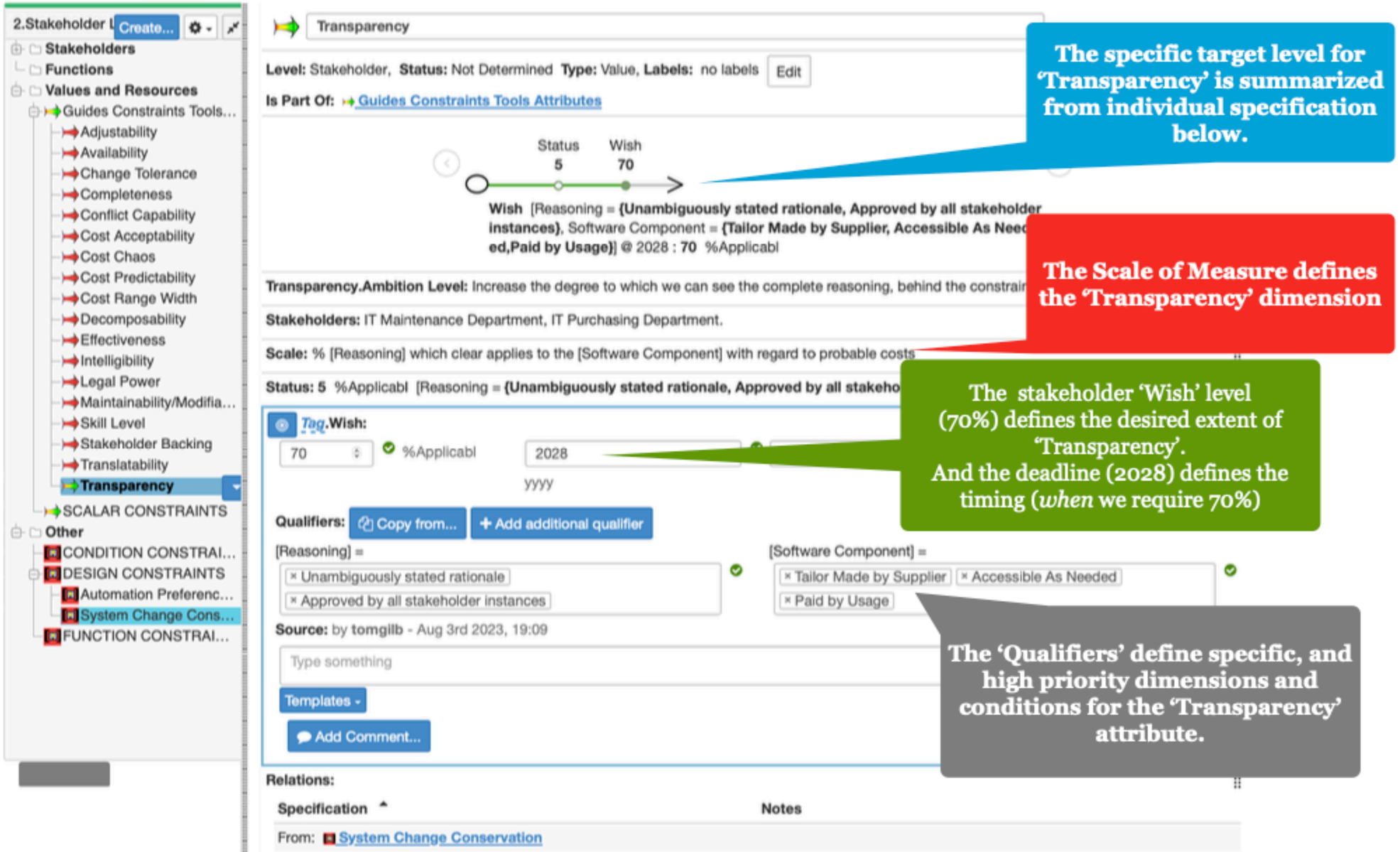
Skill Level: the degree of competence or skills which can improve value production



Constraint (and Tool) Guides Attributes

9.11.2 Defining 'Constraint attribute', '*Transparency*' in Planguage

'Transparency': an attribute of the 'System Change Conservation' Constraint



- Source
- [Guides, p 29]

9.12 Stakeholders are the source of all constraints, tools ; 'Guides'

Stakeholders Create, and maintain Constraints and Tools: 'Guides'

Stakeholders are the '*energy*' behind all systems, and this includes their 'Guides'; 'constraints and tools'.

May the Stakeholder '*force*' be with you, and improve your system efficiency.

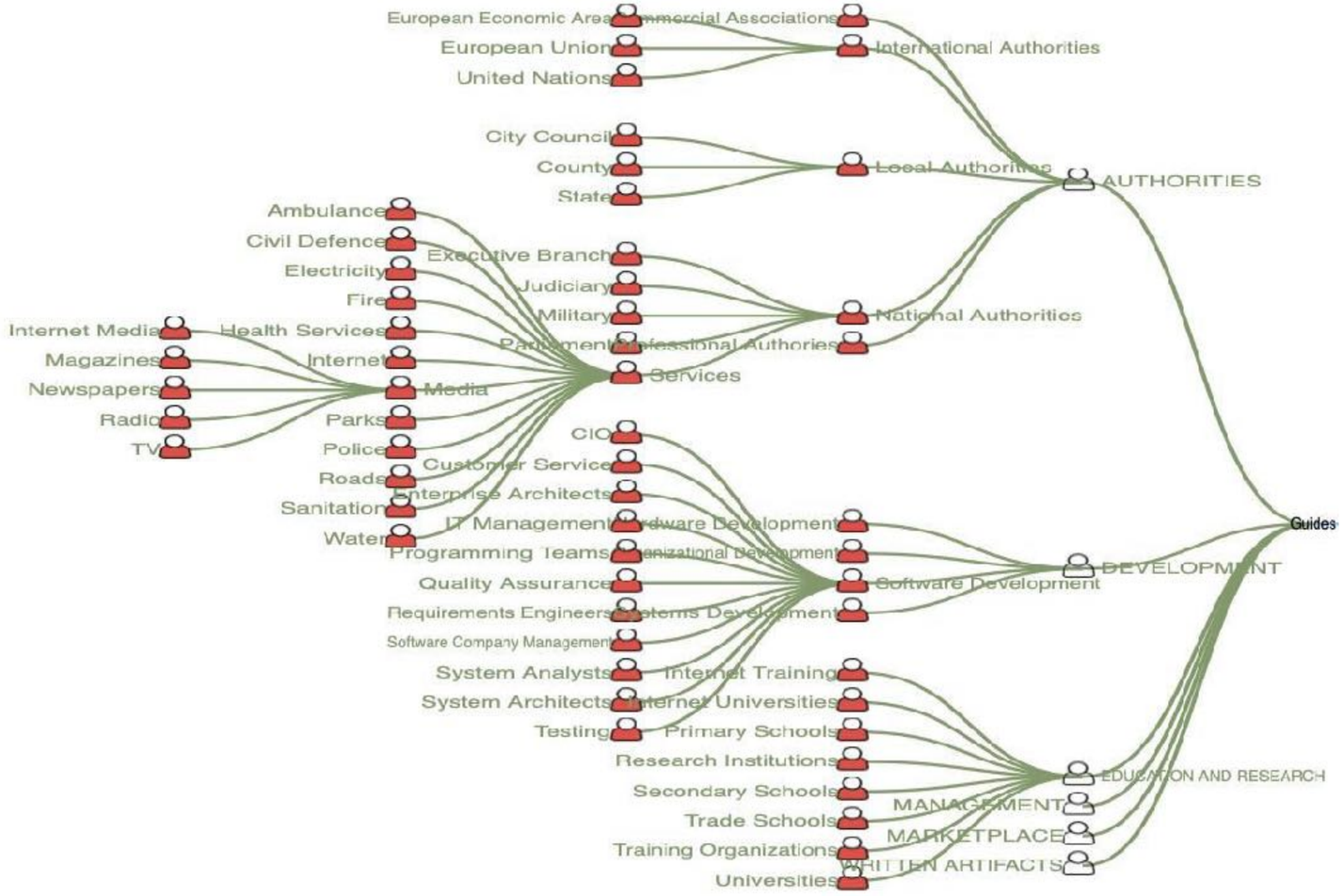
By *mapping* relevant stakeholders, **you will:**

1. Be able to analyze and understand the **attributes** of a Guide (constraint, or tool).

2. Be more-able to understand the **intent** of a constraint. Perhaps be more-able to understand the 'efficiency usefulness' (better values and lower costs) of a Tool.

And 3. Be able to relax, or drop, a Constraint under given circumstances. Maybe, be able to select, or use, a Tool, better.

A Generic Stakeholder Diagram



[Guides]

9.13 Constraint Principles

1. **RESTRICTION INTENDED**: Constraints are stakeholder requirements, which restrict your architecture choices and options, for stakeholder reasons.

2. **CONSTRAINT PRIORITY**: Stakeholders are not 'all powerful', they have different levels of power, and thus consequent, different levels of priority, to impose constraint requirements; so constraints might be dropped, waived or modified, to satisfy higher-priority stakeholders, and to satisfy higher-priority requirements (values and resources).

3. **QUESTION THEM**: Stakeholders may have 'changed their minds 'or 'never really intended to require a constraint', so when constraints are blocking better architecture, the architect needs to question the constraints, to never blindly accept them.

4. **REAL CONSTRAINT**: The architect can probe, to find the underlying justification or reason for constraints, from the stakeholder point-of-view: and they can potentially find an 'architecture-specification' way, to satisfy the stakeholder's *real* needs, while ignoring, or modifying, the constraint.

5. **NARROWER CONSTRAINTS**: Constraints may have been formulated *too widely in scope* and *time*, inadvertently. The architect can suggest narrower constraint formulations, which still fully-satisfy the current stakeholders, but which then allow the architect greater ability to find better architecture, and to satisfy the other requirements.

6. **ADJUST LEVELS**: Scalar constraints might have 'arbitrary', but useful, levels required (Tolerable Levels), which nobody thought would stop good architecture, at the time they were stated: so the architect should feel free to question them, and to re-specify levels, and conditions (who, where, what, etc.); and get approval for the requirement-change, from appropriate stakeholders.

7. **FUZZY CONSTRAINTS**: the terms used in the constraint specification are very likely ambiguous. They should, ideally, be **un-**ambiguous, of course; but people are undisciplined. So, if an unfavorable interpretation, is stopping better architecture, then you can try to make, and get-approved, a clearer interpretation.

9. **ARCH-TEST**: In theory, the architecture process requires all specified, and all changed architecture specifications to be subjected to a *constraints test*: does the 'architecture spec' violate, or threaten to violate, any constraint. You will need a clear requirements checklist of all constraints, to do this; and it is a *clear responsibility* for the *responsible architect* to check constraints. The *architecture constraint check process* should be *noted, with results, directly in the architecture specification object*. The 'Test' parameter is suitable, for this purpose.

9. **REVIEW**: the same process of checking architecture against constraints should be done by independent (of the architect) reviewers. Using the rule-based Spec QC process [CE, Spec QC Chapter].

10. **UPDATES**: all changes, however seemingly slight, need to be checked by the responsible-change-owner architect against all current constraints. And such checks should be noted in the architecture specification object (Test: Parameter). 'Need to be', and 'should be' means, it will probably pay off. The alternative is higher 'downstream' costs, for specification defects, that you do not catch earlier.

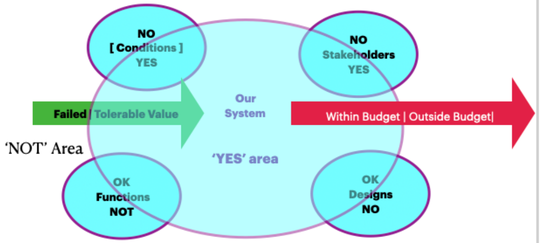
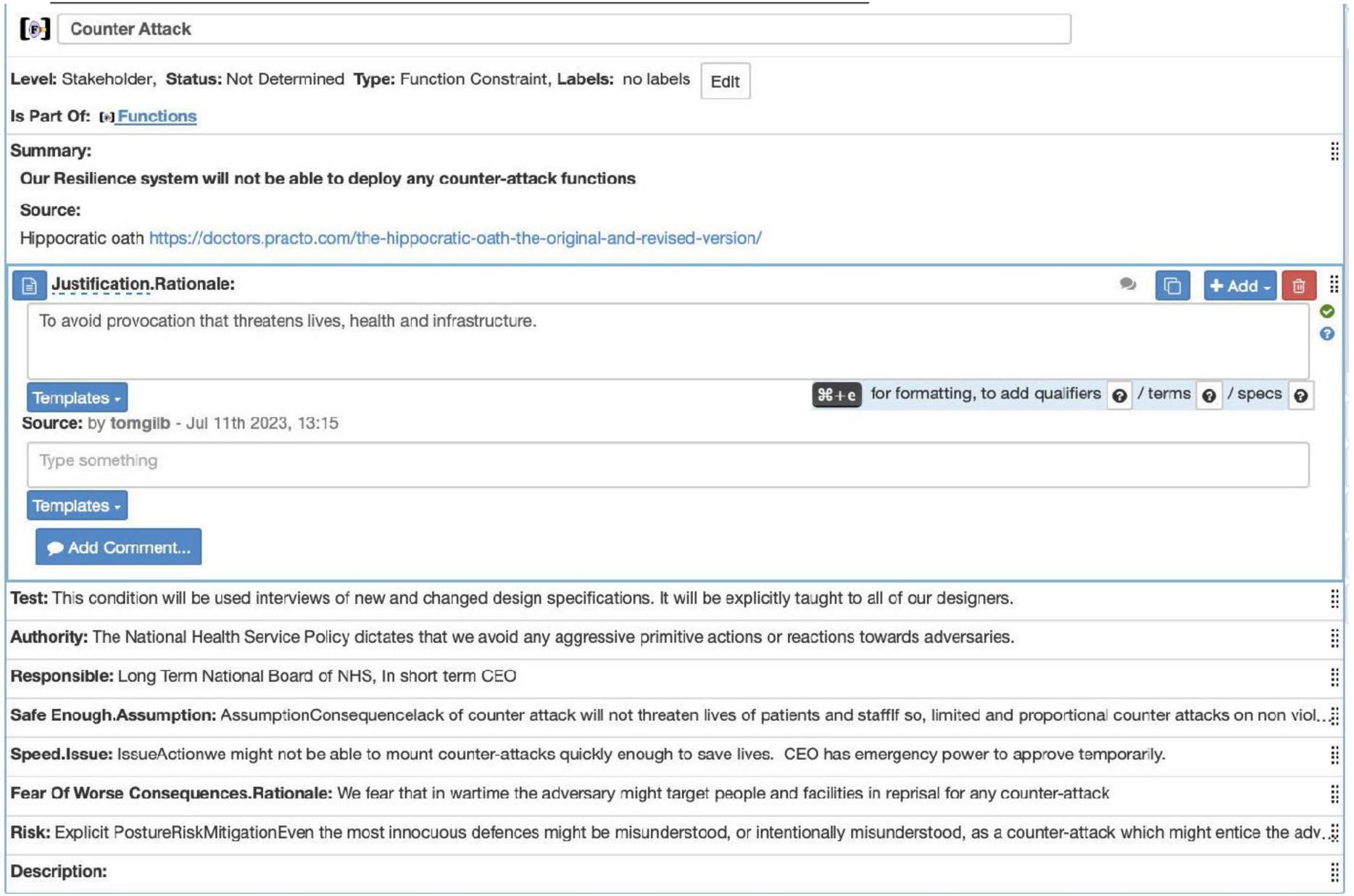


Figure 9.2 'Constraints are a 'fence', saying don't go outside me. Constraints are hopefully 'wisdom', and 'tribal knowledge'.

Architecture Constraint Principles [SEA], © Gilb, 2020-2023

9.14 Function Constraint Spec. Example 'Counter Attack'



Source: ValPlan gilb.com Project: BCS Resilience Book "Cyber Resilience Planning", Stakeholder Level, [Resilience]

9.15 'Condition Constraint' Example 'British Sourced'

[ ] British Sourced

Level: Stakeholder, **Status:** Not Determined **Type:** Constraint, **Labels:** no labels Edit

Is Part Of: [ ] Condition Constraints

Show Sidebar

Summary:

All Resilience system materials, software, data storage, consulting, programming, testing, and operational support will as far as possible be from British Sources, subsidiarily British Commonwealth, even with significant costs and delays, and initial quality degradation.

Source:

British Law on Civil Service Acquisition.

50% Excess Rule.Assumption:

Assumption	Consequence
Costs do not exceed 50% more	Competitive Bidding outside UK is permitted for costly components.

Test:

This Condition will be reviewed with all major changes and initial issues of Resilience Architecture.

In Britain Capability.Rationale: Aside from the Legal Requirement, we want to stimulate British Resilience Capability and keep knowledge in use, maybe for potential export.

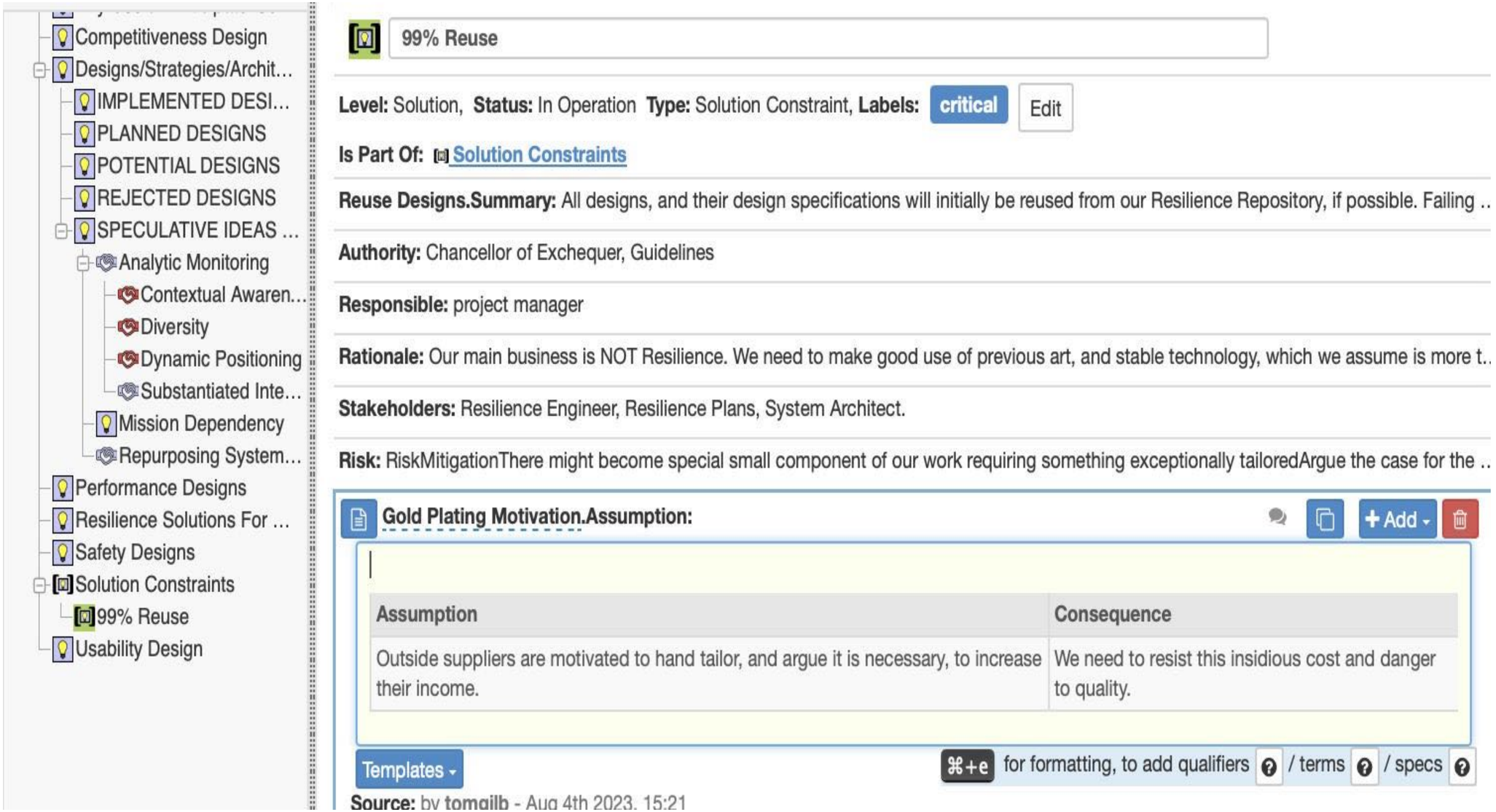
Description:

Responsible: Project Manager, Chief Civil Servant, Resilience Systems Architect

Source: ValPlan gilb.com Project: BCS [Resilience] Book "Cyber Resilience Planning", Stakeholder Level,

9.16 *Design* Constraint. Example '99% Reuse'

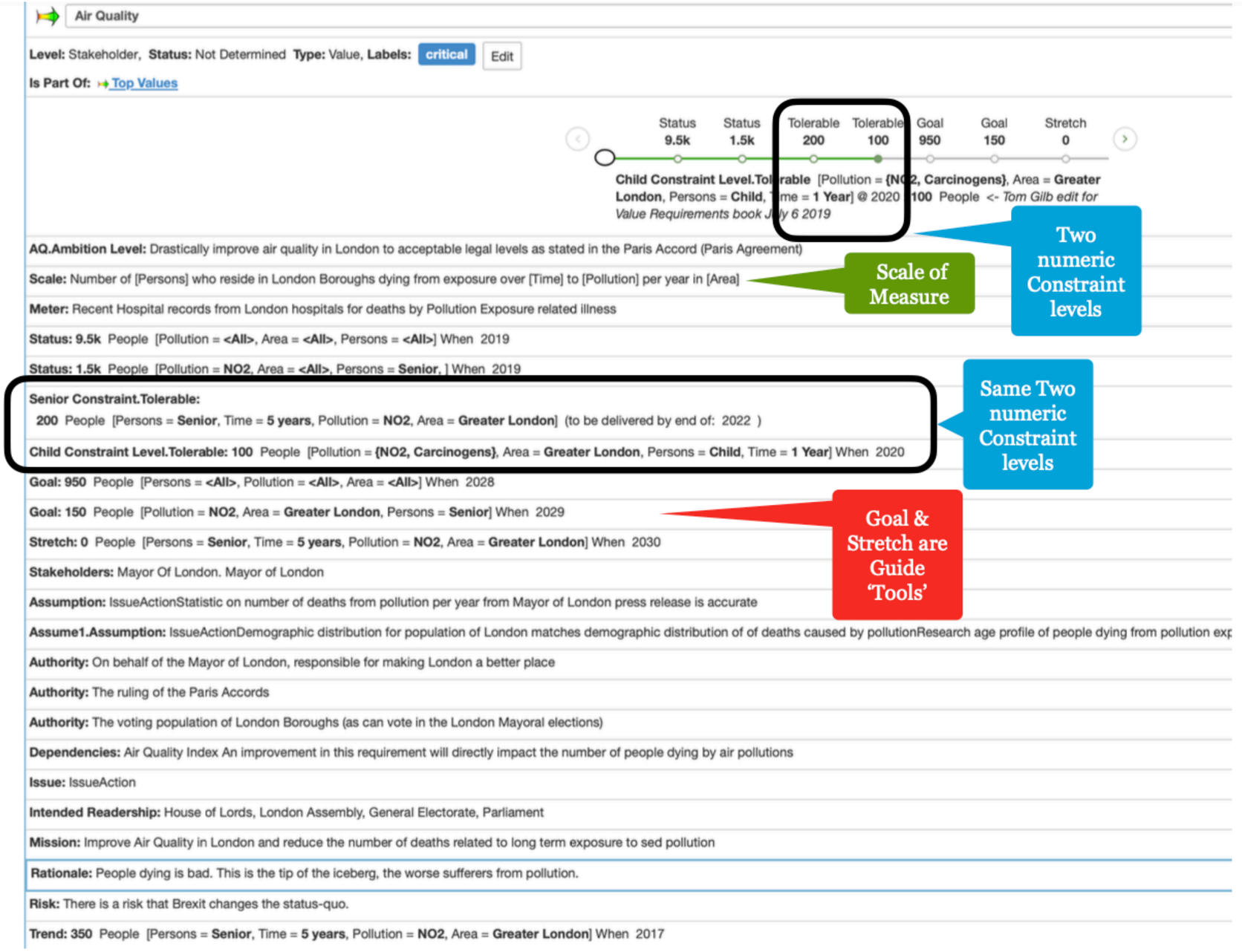
Title



Source: ValPlan gilb.com Project: BCS [Resilience] Book "Cyber Resilience Planning" , Canvas Solution Level, Solution Constraint.

9.17 Both 'Scalar Constraints', and 'Scalar Tools'

- Here is an example of defining a climate quality, in terms of both a numeric constraint (Planguage 'Tolerable' level),
  - and 'Guide Tools' (the *Goal* and *Stretch*, 'Target Levels'.
  - These 'Tools' are positive motivation to be better than the constraint level.
- This is also a rich example of Planguage specification, using [ValPlan].
  - Look at the number of Planguage parameters we are using, as a *set*, to describe 'Air Quality'. ————>
  - Do you think our climate problems, in London alone, deserve this level of planning rigor?



9.18 A 'Design Constraint' specification, in Planguage. *'System Change Specification'*

System Change Conservation

Level: Stakeholder, **Status:** In Planning **Type:** Solution Constraint, **Labels:** no labels Edit

Is Part Of: DESIGN CONSTRAINTS

Show Sidebar

Summary:

The design must use, if possible, current suppliers for the system

Description: The design must use, if possible, current suppliers for the system. This apples only to software, and only if the assumptions about life cycle cost savings are reasonably assured by estim..

Stakeholders:

Stakeholder	Roles	Notes
From: IT Maintenance Department	Expert	They have to train people to use new software components
From: IT Purchasing Department	Authority	They have to create new suppliers and contracts

Authority:

Corporate Purchasing

Responsible: IT Manager

Assumption: AssumptionConsequenceIt is possible to do this for no more than 25% more lifecycle costs.If not then make exception, but in Western Supply Chain (EU, NATO)

Dependencies: NATO Restrictions, and EU Restrictions currently in place.

Implementation Instance: Systems Architect

Issue: IssueActionSystem might not have an architectThen use Project Manager or CTO

Rationale: We need to reduce costs of onboarding new suppliers, and costs of training maintenance people.

Risk:

Risk	Mitigation
Annual costs explode due to frequency of use or new demands, for this component	If severe, switch to alternate supplier.

Relations:

Specification	Notes
To: Transparency	

Notice the large number of useful Constraint 'background' information, we can specify in Planguage. They also make you ask things like 'is anybody responsible for this?'

9.19 What are we doing with these above specifications, (in ‘Planguage’)?

A Design Constraint, indeed any Constraint type, or including ‘Tools’, **cannot be judged, in depth, by the specification of the *constraint itself*.**

If we are to have a hope of understanding the constraints for decision-making purposes *(like; shall we approve in this case?),*

particularly when it *alone* can cause significant costs, or other critical attributes
(like; qualities like the safety, of - for example, a weapons system, or a medical system).

We should consider having rigorous specification and decision-making processes.

If the ‘Constraint Analysis’ cost, per attribute, is for example, one work-day of effort, to get a deeper insight, and the consequence of a wrong decision is ‘one work-year of loss’, then *that* is a value/cost ratio of 364/1

This effort is ‘engineering’. In the sense of rigorous, but ‘risk proportional’ analysis - in advance of significant and critical decisions.

For smaller systems, and for smaller risks, this is not necessary. We will just ‘take a chance’, and then, maybe using ‘Evo’ ([CE], Incremental Value Delivery) we can try to find out ‘how **good**, decisions **are**’, by getting early, realistic systems feedback, in small measurable increments.

The Stakeholder Value attributes of a Specific Design Constraint, expressed in Planguage ‘Drawn Icons’. [CE]

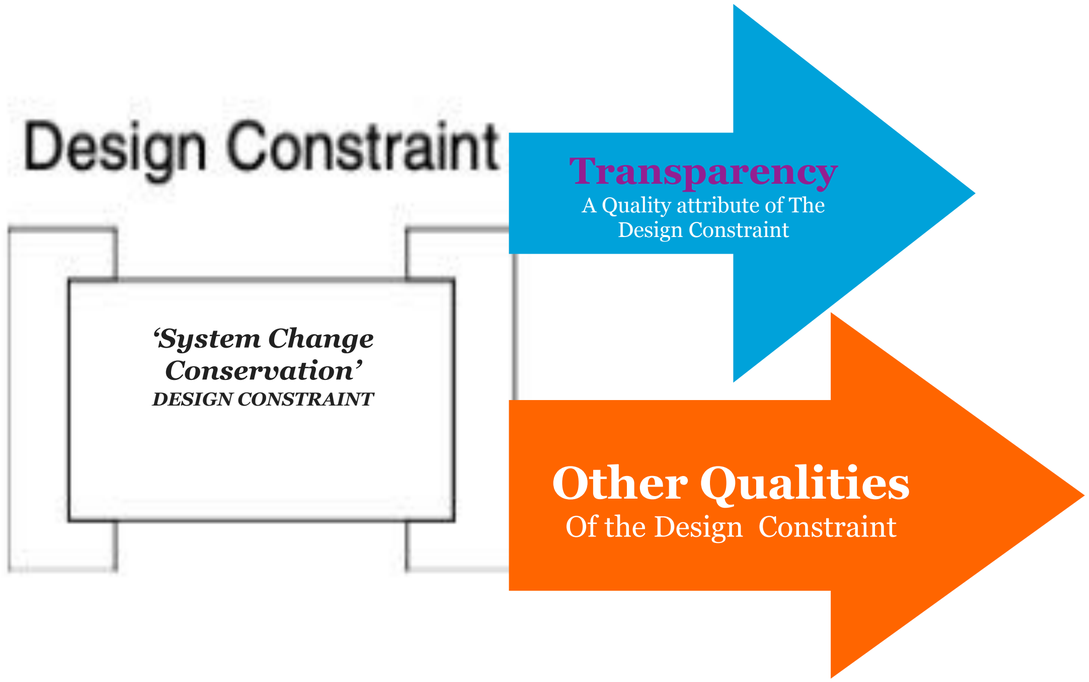
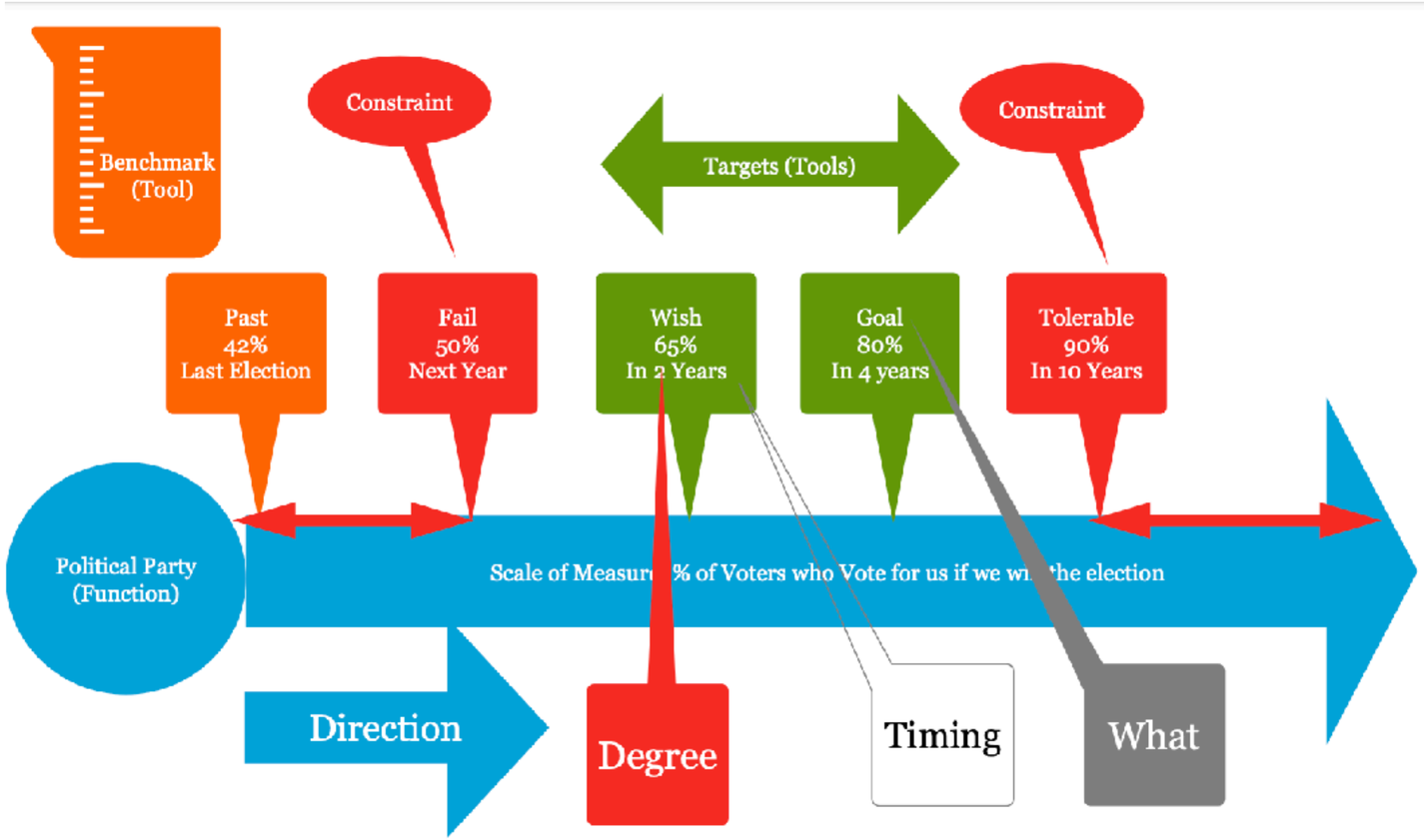


Figure: 8.19
Source: TsG for [Guides] 03Aug23

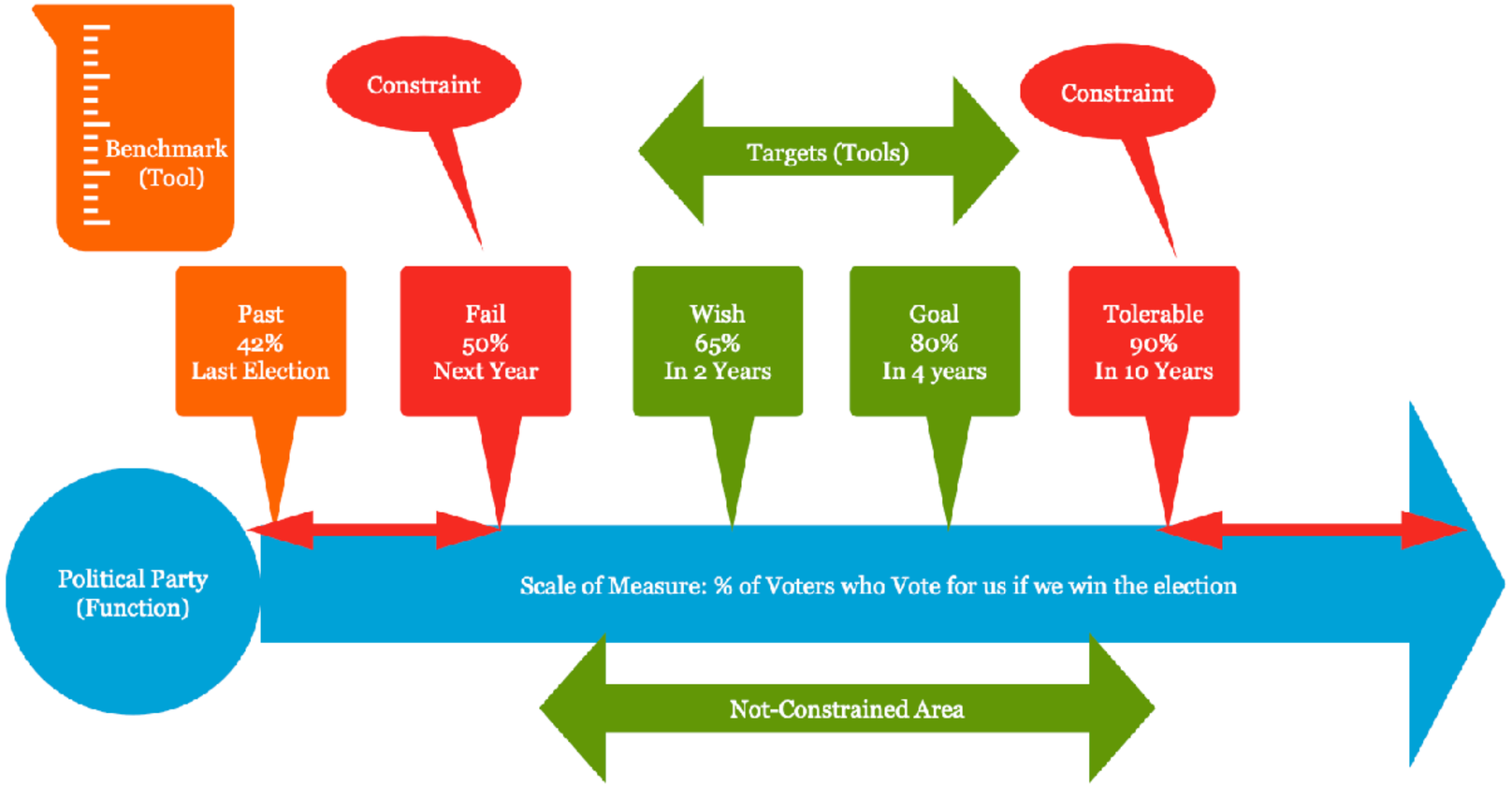
And, although we may know of 20 interesting Constraint Attributes, we do not have to analyze all of them, every time. It is usually sufficient to select some of the few, clearly critical, increments, for our decisions about the constraints. Enough to satisfy ‘reasonable risk analysis’, for the case in hand. Maybe ‘only one attribute’ might suffice. Keeping in mind that there are many complimentary, and alternative methods, for dealing with the ‘risks of accepting a Constraint’: such as *incremental feedback, self-insurance, long experience, and deeper research into world-wide case studies*. **Our main point here is to exemplify exactly, how we can articulate the Constraint Attributes, so that we can *see* them, and *discuss* them.**

9.20 Various Scalar points as Constraint, or Tool, Targets



Example of Guides (Tools and Constraints) in the core of Planguage. Fail and Tolerable levels are the borders of a Constraint Area (red double arrow). Next year, a level of 50% or worse, is deemed a failure. And in 10 years, anything over 90% is considered intolerable (suspicion of election rigging, as in Russia, North Vietnam.). The Past level is a Tool (Benchmark), which helps us understand all the other levels, in relation to it. The Wish (stakeholder dream) and the Goal (project commitment) Target levels are Tools to give us **direction** (the Scale is defined direction), and **degree** (the Wish and Goal level) and **timing** (in X years). All of these concepts are 'Guides' for planners. Source [Guides] book 050823 TG)

9.21 The role of scalar points in guiding our actions



Guides give us **Direction** (*Where* should we head off to?), **Degree** (*How far* should we go?), **Timing** (W*hen* should we go so far, or not), and **What** (*Consequence* of reaching the Degree, like Fail or Goal). The red arrows indicate a constraint area, or 'range'.

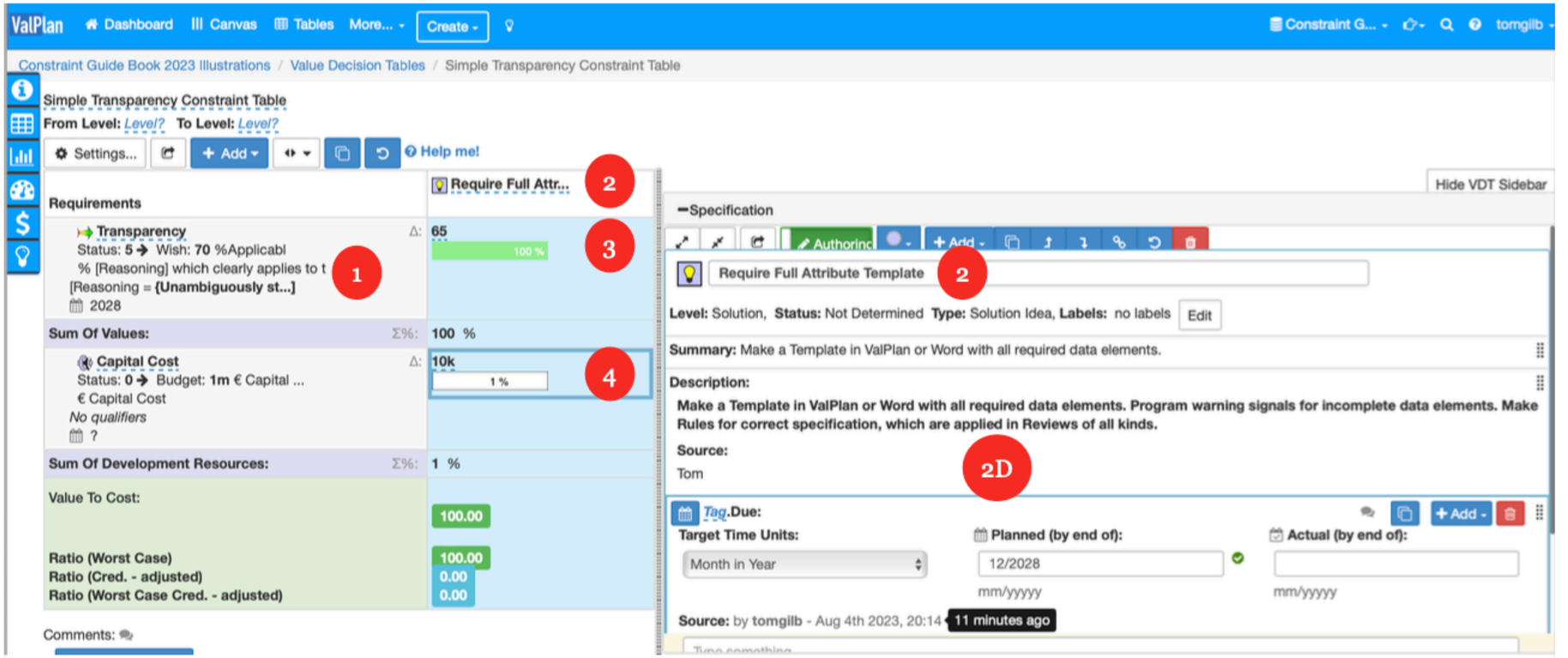
Source Guides book 050823 TG)

9.22 IET

Evaluating strategies for satisfaction of the '*Transparency*' guide

- The Impact estimation process.
- What is the impact of the strategy, on the specific Target level (Wish 70%)
  - which is an improvement from the current Status level 5%
- The estimate is, that with this one strategy *alone*, *'we will get to 70%'*.
  - which is also '100% of the way' from 5% to 70%
- So, if we have estimated correctly, 'Problem Solved'.
- The next question is, can we afford this solution, from our budget? (a Scalar Financial Constraint)
  - If yes: GO
  - If not, try to find a cheaper solution, or set of solutions.

Evaluation of a simple design to satisfy the 'Transparency' (Constraint Quality) of 70% by 2028 within reasonable cost (1%)



Source: ValPlan ... IET Simple Transparency w one Solution 2023-08-04 at 20.15.17

How can we be sure that we have enough 'Transparency' for the 'System Change Conservation' DESIGN CONSTRAINT ?

1. (See red circle 1, in Figure above) We have defined 'enough' as 70% on the Scale of measure (% Reasoning which applies to... etc) by the year 2028.

2. To actually, credibly, achieve that level, we have to do something. To design a credible solution and implement it in practice, and see how it really works (test it, measure it).

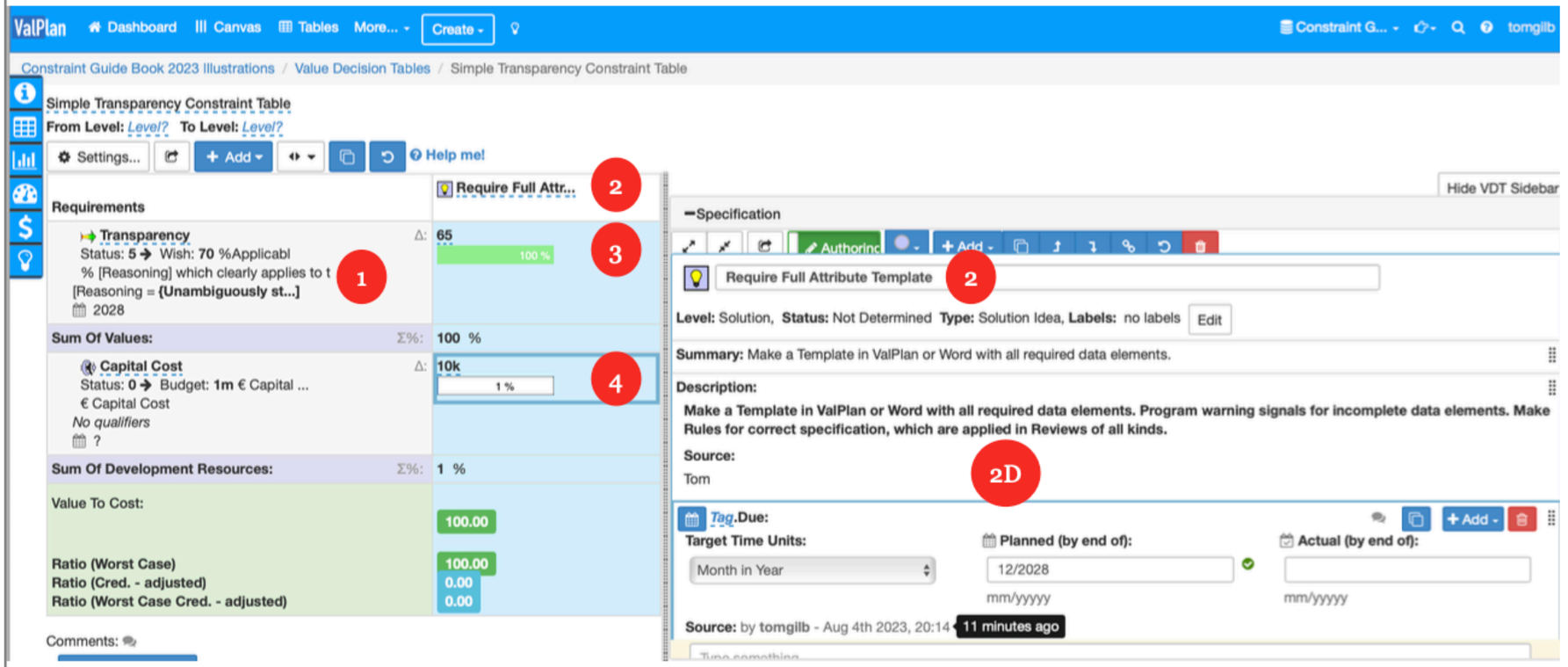
3. So the 'Require Full Attribute Template' design solution is specified (2D), and we believe, and estimate, (3. Red circle) it will get us to 70% (5+65 Increment = 70%). Which (another viewpoint) is '100% of the way to our Wish target level, by the deadline 2028'. A 100% complete 'design effect'. No other design needed, to get to target.

4. We also estimate the rough magnitude of the cost at €10,000 or 1% of our total budget. So that seems acceptable, and we do not feel we need to look at other, more efficient, solutions.

9.23 IET

- Using the Impact Estimation Table to 'discuss' how good, and at what cost, the design strategy *'Require Full Attribute Template'* will help us reach the 'Wish' level (a tool level, to inspire and motivate)

Evaluation of a simple design to satisfy the 'Transparency' (Constraint Quality) of 70% by 2028 within reasonable cost (1%)



Source: ValPlan ... IET Simple Transparency w one Solution 2023-08-04 at 20.15.17

A discussion of this requirements-design-analysis process, using an IET (Impact Estimation Table [2, CE, IET Chapter]).

The IET is a 'Tool' Guide. In this case it guides us to think systematically and locally about a problem, (about, 'better Transparency' (1) for the Design Constraint).

Our *Engineering Logic*, says we cannot get there, by just *hoping* for it, or *just specifying* our Wish. We have to have a *specific idea* (a Design Solution (2, 2D) which some technical expert (architect, engineer, reviewer) can read, and then determine what effect they *believe* the design will have (3)

Once a specific written, and detailed, design Description (2D) is made in *writing*, a technical expert, is, for the first time (engineering logic), able to estimate our design's approximate costs. (4).

This IET Tool, strongly Guides the Engineers to follow this engineering logic, and to document their opinions, with their name on it. This 'engineer evidence' can be used to do many things, like approve, try again, review, present suggestions, use in legal evidence as sound practice.

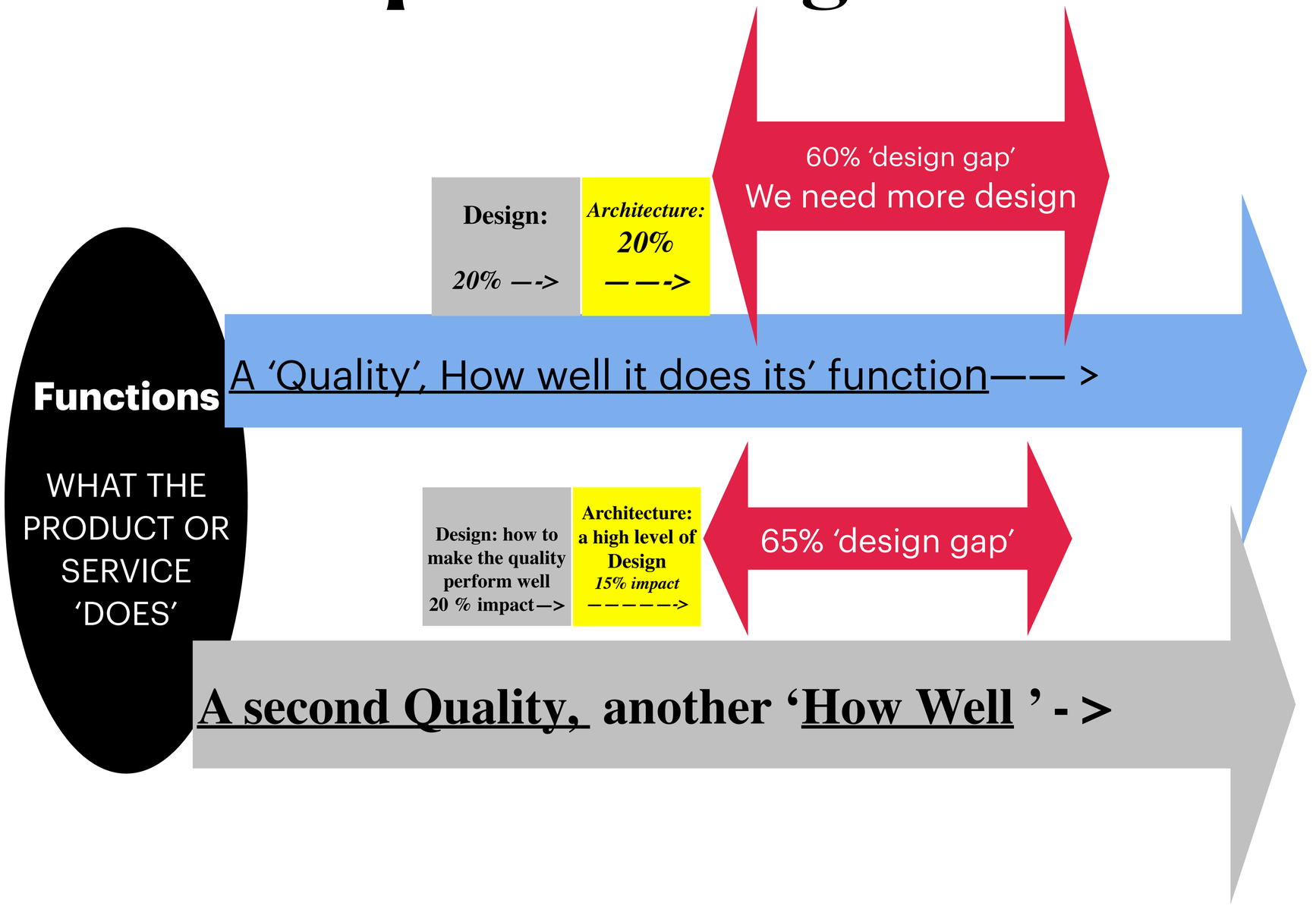
These are examples of decomposition, and of using AI for decomposition

10. Design (Product, Service)



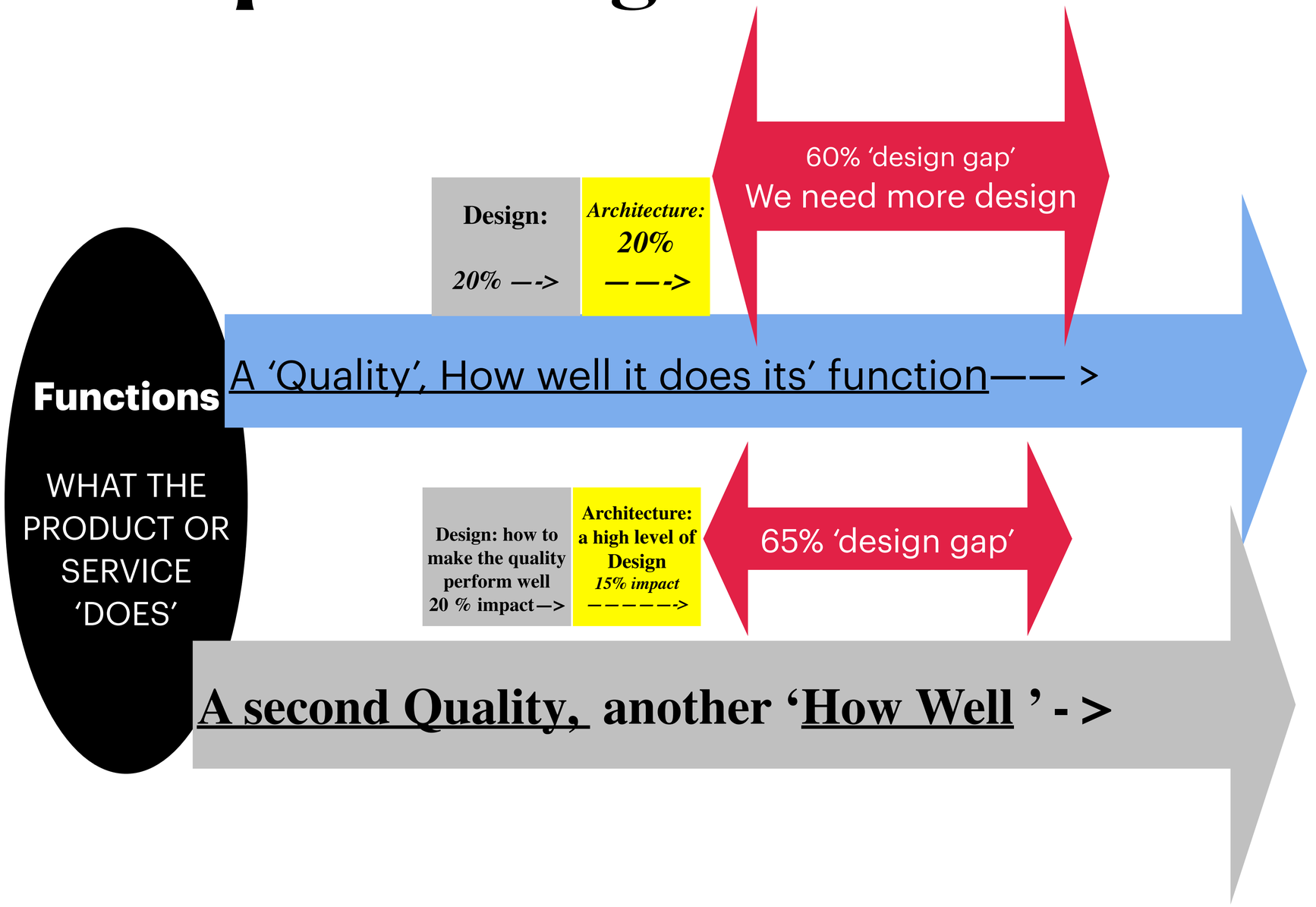
10.1 The Concept of 'Design'

- **Design Process**: *defined as* the *activity* of finding, and verifying 'Design Ideas' which we can '***do***', in order to improve stakeholder value qualities, in the direction of their Target levels, on time.
- **Design Idea**: *defined as:* a specific concept that can be implemented in a system, with the purpose of delivering improved qualities or other planned stakeholder values. Design Ideas contain a *set of many qualities, and many costs.*
- **Design**: *Synonyms*: 'Means' (to ends), Architecture Components, Strategies, Solutions, Do-able Ideas, Implementations.



10.2 The Concept of 'Design' Process

- **Qualities are *conceptual* ideas, of <u>how well</u> we will function. The '-ilities' like reliability, usability, and security.**
- **They are clearest expressed with a number on a Scale of Measure, which defines *what* the quality is, and the *level* we are designing towards.**
  - *99.9% of scheduled uptime, on weekdays.*
- The job of the designer (engineer, architect, strategy planner) is to find a set of designs (1 or more) which will allow us to reach future desired levels of quality (Tolerable, Wish, Goal, Stretch)
- The design process is initially theoretical. We pick a design, and ask what effect we believe it will have on our quality level plans. **'Will we reach our Goal on time?'**
- If 'yes'; we think we will reach our planned levels, we can stop 'designing', and try it out.
- If not, *replace* the design, or *add* to it, until we *think* we have a 'set of designs', which will get us to our 'planned performance' levels. Then try them out, in practice.



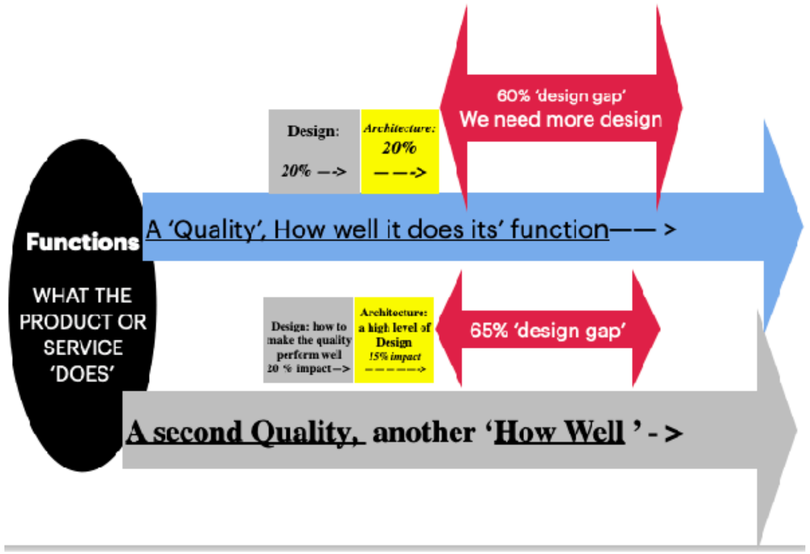
10.3 The *Multi-dimensional* Concept of 'Design' Process

Eternal Competitive Design

- Unfortunately for the designer, it is not that simple. There are 'side-effects' !
- 'Side-effects of a design' are all those quality and cost impacts, that 'just come along with the design', whether you like it or not. Nature is like that, always, everywhere.'
- So, the systems 'engineer' *knows* this, and is mature enough to *not* 'wait until later', when the 'unexpected' side-effects, destroy your product. Engineers take the long term, life cycle view.
- The mature, well trained, 'systems engineer' considers the side-effects immediately, and systematically. Quantitative estimates, of *course*. Using an Impact Estimation Table [CE, VIET], or similar tool, to visualize the many dimensions of competing interests..
- If side-effects are unacceptable (*negative quality levels, too much costs*) then *dump* this 'bad' design and search for better ones
- Then, if the *design set* looks OK, and it is promising *improved value* for your efforts, *Deliver, Measure, Learn* (see Evo Cycle diagram at right)
- If things are *still* acceptable, even if reality is different from estimations, then continue the design process (which now ultimately includes *Decompose and Prioritize, Develop, Deliver, Measure and Learn*).
- Keep on designing and implementing, to move all critical qualities, initially, to **Tolerable** levels. *System Survival.*
- Then, keep designing, until you get all critical qualities to your **Goal** levels (*satisfactory, committed, quite competitive levels*).
- Then, now that you have a 'successful system' (*all Goals reached*) you can safely turn your attention to the multiple ***cost*** levels (people, duration, money, space) and keep going through the design and experiment (Deliver, Measure, Learn). Let me call this *'cost balancing'* design. Also known as 'Design-to-Cost' [Cost Engineering]
- The next stage of the design process is ***optimization***, pushing quality levels even higher, and costs even lower as Tesla and SpaceX do [MM], in daily parallel increments.
- As Deming told me (1983, London), there **is no end** to this 'Plan Do Study Act' cycle, *"as long as you are in competition".*
- If *you* end the design cycle, prematurely, you have 'dropped out of the race'.
- You probably need to continue the design cycle, for the life of the product or service. It is *not* a one-off event.



10.3 A 'Design' cycle



10.4.1.2 Two sets of design strategies, for Water, and for Food

These can easily be detailed, on demand, see. 10.4.3

Detailed Strategies for Food Supply Reliability

1. **Local Farming Training Programs**
   - **Description**: Provide education and training on modern farming techniques to local farmers.
   - **CapEx**: £1.5 million for setting up training centers and resources.
   - **OpEx**: £0.3 million/year for trainers, materials, and ongoing support.
2. **Improved Seed Varieties**
   - **Description**: Introduce and distribute high-yield and drought-resistant seed varieties.
   - **CapEx**: £0.8 million for research, procurement, and distribution.
   - **OpEx**: £0.1 million/year for maintenance and additional seeds.
3. **Efficient Irrigation Systems**
   - **Description**: Implement drip irrigation and other water-efficient systems.
   - **CapEx**: £2.0 million for infrastructure and installation.
   - **OpEx**: £0.4 million/year for maintenance and operations.
4. **Soil Health Improvement**
   - **Description**: Use organic fertilizers and soil conditioners to enhance soil fertility.
   - **CapEx**: £1.2 million for initial soil treatment and organic material procurement.
   - **OpEx**: £0.2 million/year for ongoing soil health programs.
5. **Community Cooperative Formation**
   - **Description**: Establish cooperatives to manage resources and share knowledge.
   - **CapEx**: £1.0 million for setup, legal support, and initial operations.
   - **OpEx**: £0.15 million/year for administration and support.
6. **Agricultural Subsidies**
   - **Description**: Provide financial support to farmers for purchasing inputs and equipment.
   - **CapEx**: £1.8 million for subsidy fund establishment.
   - **OpEx**: £0.5 million/year for disbursing subsidies.
7. **Market Access Facilitation**
   - **Description**: Create channels for farmers to access markets and sell their produce.
   - **CapEx**: £1.0 million for setting up market infrastructure and logistics.
   - **OpEx**: £0.2 million/year for market operations and support.
8. **Storage and Preservation Solutions**
   - **Description**: Build facilities for grain storage and preservation to reduce post-harvest losses.
   - **CapEx**: £2.2 million for construction and equipment.
   - **OpEx**: £0.6 million/year for maintenance and operation.
9. **Renewable Energy for Farming**
   - **Description**: Use solar and wind energy to power farming equipment and irrigation systems.
   - **CapEx**: £1.5 million for renewable energy installations.
   - **OpEx**: £0.4 million/year for energy system maintenance.
10. **Disaster Resilience Planning**
    - **Description**: Develop plans and infrastructure to withstand and recover from natural disasters.
    - **CapEx**: £2.0 million for resilience infrastructure and training.
    - **OpEx**: £0.3 million/year for plan updates and drills.

Strategy 3: Efficient Irrigation 20%

Strategy 8: Storage & Preservation Solutions 20%

20%

Detailed Strategies for Water Supply Availability

1. **Local Farming Training Programs**
   - **Description**: Provide education and training on efficient water use in agriculture.
   - **CapEx**: £1.5 million for setting up training centers and resources.
   - **OpEx**: £0.3 million/year for trainers, materials, and ongoing support.
2. **Improved Seed Varieties**
   - **Description**: Introduce and distribute high-yield and drought-resistant seed varieties to [illegible] water-efficient crops.
   - **CapEx**: £0.8 million for research, procurement, and distribution.
   - **OpEx**: £0.1 million/year for maintenance and additional seeds.
3. **Efficient Irrigation Systems**
   - **Description**: Implement drip irrigation and other water-efficient systems.
   - **CapEx**: £2.0 million for infrastructure and installation.
   - **OpEx**: £0.4 million/year for maintenance and operations.
4. **Soil Health Improvement**
   - **Description**: Use organic fertilizers and soil conditioners to enh[illegible] water retention.
   - **CapEx**: £1.2 million for initial soil treatment and organic ma[illegible] procurement.
   - **OpEx**: £0.2 million/year for ongoing soil health programs.
5. **Community Cooperative Formation**
   - **Description**: Establish cooperatives to manage wate[illegible]rces and share knowledge.
   - **CapEx**: £1.0 million for setup, legal support, and [illegible] operations.
   - **OpEx**: £0.15 million/year for administration and [illegible]ort.
6. **Agricultural Subsidies**
   - **Description**: Provide financial support to [illegible]rs for purchasing water-efficient equipment.
   - **CapEx**: £1.8 million for subsidy fund [illegible]shment.
   - **OpEx**: £0.5 million/year for disburs[illegible]bsidies.
7. **Market Access Facilitation**
   - **Description**: Create channels [illegible]rmers to access markets and sell their water-efficient produce.
   - **CapEx**: £1.0 million for se[illegible]p market infrastructure and logistics.
   - **OpEx**: £0.2 million/year [illegible]arket operations and support.
8. **Storage and Preservation So[illegible]s**
   - **Description**: Buil[illegible]ties for water storage and purification to ensure a steady supply.
   - **CapEx**: £2.2 m[illegible]or construction and equipment.
   - **OpEx**: £0.6 m[illegible]/year for maintenance and operation.
9. **Renewable Energy for [illegible]arming**
   - **Description**: Use solar and wind energy to power water purification systems.
   - **CapEx**: £1.5 million for renewable energy installations.
   - **OpEx**: £0.4 million/year for energy system maintenance.
10. **Disaster Resilience Planning**
    - **Description**: Develop plans and infrastructure to ensure water supply during natural disasters.
    - **CapEx**: £2.0 million for resilience infrastructure and training.
    - **OpEx**: £0.3 million/year for plan updates and drills.
11.

10.4.1 *Food* Supply Objective, Priority 1

Table 1: Food Supply Reliability. Priority 1 Goal: [Namibia, Grain, Local Farming] - 90% reliability within 3 years

Objectives	Strategy 1: Local Farming Training Programs	Strategy 2: Improved Seed Varieties	Strategy 3: *Efficient Irrigation 20%*	Strategy 4: Soil Health Improvement	Strategy 5: Community Cooperative Formation	Strategy 6: Agricultural Subsidies	Strategy 7: Market Access Facilitation	Strategy 8: *Storage & Preservation Solutions 15%*	Strategy 9: Renewable Energy for Farming	Strategy 10: Disaster Resilience Planning
Food Supply Reliability	15 %	10 %	20 %	10 %	5 %	10 %	5 %	15 %	5 %	5 %
CapEx Estimate (£ million)	1.5	0.8	2.0	1.2	1.0	1.8	1.0	2.2	1.5	2.0
OpEx Estimate (£ million/year)	0.3	0.1	0.4	0.2	0.15	0.5				

Detailed Strategies for Food Supply Reliability

1. **Local Farming Training Programs**
   - **Description**: Provide education and training on modern farming techniques to local farmers.
   - **CapEx**: £1.5 million for setting up training centers and resources.
   - **OpEx**: £0.3 million/year for trainers, materials, and ongoing support.
2. **Improved Seed Varieties**
   - **Description**: Introduce and distribute high-yield and drought-resistant seed varieties.
   - **CapEx**: £0.8 million for research, procurement, and distribution.
   - **OpEx**: £0.1 million/year for maintenance and additional seeds.
3. **Efficient Irrigation Systems**
   - **Description**: Implement drip irrigation and other water-efficient systems.
   - **CapEx**: £2.0 million for infrastructure and installation.
   - **OpEx**: £0.4 million/year for maintenance and operations.
4. **Soil Health Improvement**
   - **Description**: Use organic fertilizers and soil conditioners to enhance soil fertility.
   - **CapEx**: £1.2 million for initial soil treatment and organic material procurement.
   - **OpEx**: £0.2 million/year for ongoing soil health programs.
5. **Community Cooperative Formation**
   - **Description**: Establish cooperatives to manage resources and share knowledge.
   - **CapEx**: £1.0 million for setup, legal support, and initial operations.
   - **OpEx**: £0.15 million/year for administration and support.
6. **Agricultural Subsidies**
   - **Description**: Provide financial support to farmers for purchasing inputs and equipment.
   - **CapEx**: £1.8 million for subsidy fund establishment.
   - **OpEx**: £0.5 million/year for disbursing subsidies.
7. **Market Access Facilitation**
   - **Description**: Create channels for farmers to access markets and sell their produce.
   - **CapEx**: £1.0 million for setting up market infrastructure and logistics.
   - **OpEx**: £0.2 million/year for market operations and support.
8. **Storage and Preservation Solutions**
   - **Description**: Build facilities for grain storage and preservation to reduce post-harvest losses.
   - **CapEx**: £2.2 million for construction and equipment.
   - **OpEx**: £0.6 million/year for maintenance and operation.
9. **Renewable Energy for Farming**
   - **Description**: Use solar and wind energy to power farming equipment and irrigation systems.
   - **CapEx**: £1.5 million for renewable energy installations.
   - **OpEx**: £0.4 million/year for energy system maintenance.
10. **Disaster Resilience Planning**
    - **Description**: Develop plans and infrastructure to withstand and recover from natural disasters.
    - **CapEx**: £2.0 million for resilience infrastructure and training.
    - **OpEx**: £0.3 million/year for plan updates and drills.

Reasons for Value Percentages in Food Supply Reliability

1. **Local Farming Training Programs (15%)**
   - **Reason 1**: Educating farmers on modern techniques can significantly increase productivity and efficiency.
   - **Reason 2**: Training programs help ensure the adoption of best practices, leading to more reliable food supply.
2. **Improved Seed Varieties (10%)**
   - **Reason 1**: High-yield and drought-resistant seeds directly contribute to increased food production.
   - **Reason 2**: Improved seeds can help mitigate the effects of poor soil and climate conditions.
3. **Efficient Irrigation Systems (20%)**
   - **Reason 1**: Efficient irrigation ensures optimal water usage, crucial in arid regions like Namibia.
   - **Reason 2**: Reliable irrigation systems can double or triple crop yields by providing consistent water supply.
4. **Soil Health Improvement (10%)**
   - **Reason 1**: Healthy soil is fundamental to sustainable farming and higher crop yields.
   - **Reason 2**: Improving soil health can lead to better water retention and nutrient availability.
5. **Community Cooperative Formation (5%)**
   - **Reason 1**: Cooperatives can pool resources and knowledge, increasing overall farming efficiency.
   - **Reason 2**: Collective efforts can improve market access and bargaining power for farmers.
6. **Agricultural Subsidies (10%)**
   - **Reason 1**: Financial support helps farmers invest in necessary equipment and inputs.
   - **Reason 2**: Subsidies can reduce the financial risk for farmers, encouraging more consistent production.
7. **Market Access Facilitation (5%)**
   - **Reason 1**: Improved market access ensures farmers can sell their produce, providing them with income.
   - **Reason 2**: Facilitating market access helps reduce post-harvest losses due to delayed sales.
8. **Storage and Preservation Solutions (15%)**
   - **Reason 1**: Proper storage facilities reduce post-harvest losses and ensure food availability during off-seasons.
   - **Reason 2**: Preservation solutions help maintain the quality and longevity of food supplies.
9. **Renewable Energy for Farming (5%)**
   - **Reason 1**: Using renewable energy can reduce operating costs and make farming more sustainable.
   - **Reason 2**: Renewable energy sources ensure reliable power for farming operations, especially in remote areas.
10. **Disaster Resilience Planning (5%)**
    - **Reason 1**: Resilience planning helps mitigate the impacts of natural disasters on food supply.
    - **Reason 2**: Preparedness ensures faster recovery and continuity of farming activities after disasters.

Strategy 3: *Efficient Irrigation 20%*

Strategy 8: *Storage & Preservati Solutions 15%*

10.4.2 *Water* Supply Strategies

Compare these to the Food supply previous page!

Detailed Strategies for Water Supply Availability

1. **Local Farming Training Programs**
   - **Description: Provide education and training on efficient water use in agriculture.**
   - **CapEx: £1.5 million for setting up training centers and resources.**
   - **OpEx: £0.3 million/year for trainers, materials, and ongoing support.**
2. **Improved Seed Varieties**
   - **Description: Introduce and distribute high-yield and drought-resistant seed varieties to ensure water-efficient crops.**
   - **CapEx: £0.8 million for research, procurement, and distribution.**
   - **OpEx: £0.1 million/year for maintenance and additional seeds.**
3. **Efficient Irrigation Systems**
   - **Description: Implement drip irrigation and other water-efficient systems.**
   - **CapEx: £2.0 million for infrastructure and installation.**
   - **OpEx: £0.4 million/year for maintenance and operations.**
4. **Soil Health Improvement**
   - **Description: Use organic fertilizers and soil conditioners to enhance water retention.**
   - **CapEx: £1.2 million for initial soil treatment and organic material procurement.**
   - **OpEx: £0.2 million/year for ongoing soil health programs.**
5. **Community Cooperative Formation**
   - **Description: Establish cooperatives to manage water resources and share knowledge.**
   - **CapEx: £1.0 million for setup, legal support, and initial operations.**
   - **OpEx: £0.15 million/year for administration and support.**
6. **Agricultural Subsidies**
   - **Description: Provide financial support to farmers for purchasing water-efficient equipment.**
   - **CapEx: £1.8 million for subsidy fund establishment.**
   - **OpEx: £0.5 million/year for disbursing subsidies.**
7. **Market Access Facilitation**
   - **Description: Create channels for farmers to access markets and sell their water-efficient produce.**
   - **CapEx: £1.0 million for setting up market infrastructure and logistics.**
   - **OpEx: £0.2 million/year for market operations and support.**
8. **Storage and Preservation Solutions**
   - **Description: Build facilities for water storage and purification to ensure a steady supply.**
   - **CapEx: £2.2 million for construction and equipment.**
   - **OpEx: £0.6 million/year for maintenance and operation.**
9. **Renewable Energy for Farming**
   - **Description: Use solar and wind energy to power water purification systems.**
   - **CapEx: £1.5 million for renewable energy installations.**
   - **OpEx: £0.4 million/year for energy system maintenance.**
10. **Disaster Resilience Planning**
    - **Description: Develop plans and infrastructure to ensure water supply during natural disasters.**
    - **CapEx: £2.0 million for resilience infrastructure and training.**
    - **OpEx: £0.3 million/year for plan updates and drills.**

Table 2: *Water* Supply Availability

Priority 1 Goal: [Namibia, Drinking, Solar Water Purifiers] - 15 liters per person per day within 3 years

Objectives	Strategy 1: Local Farming Training Programs	Strategy 2: Improved Seed Varieties	Strategy 3: Efficient Irrigation Systems	Strategy 4: Soil Health Improvement	Strategy 5: Community Cooperative Formation	Strategy 6: Agricultural Subsidies	Strategy 7: Market Access Facilitation	Strategy 8: Storage and Preservation Solutions	Strategy 9: Renewable Energy for Farming	Strategy 10: Disaster Resilience Planning
Water Supply Availability	10 %	5 %	20 %	10 %	10 %	5 %	5 %	20 %	10 %	5 %
CapEx Estimate (£ million)	1.5	0.8	2.0	1.2	1.0	1.8	1.0	2.2	1.5	2.0
OpEx Estimate (£ million/year)	0.3	0.1	0.4	0.2	0.15	0.5				

Reasons for Value Percentages in Water Supply Availability

1. **Local Farming Training Programs (10%)**
   - **Reason 1**: Educating farmers on efficient water use can significantly reduce water wastage.
   - **Reason 2**: Training can introduce innovative water-saving techniques, directly impacting water availability.
2. **Improved Seed Varieties (5%)**
   - **Reason 1**: Water-efficient crops reduce overall water demand.
   - **Reason 2**: Drought-resistant seeds ensure reliable yields even with limited water.
3. **Efficient Irrigation Systems (20%)**
   - **Reason 1**: Drip and other efficient irrigation systems maximize water use efficiency.
   - **Reason 2**: These systems can significantly increase water availability for drinking purposes by reducing agricultural water consumption.
4. **Soil Health Improvement (10%)**
   - **Reason 1**: Improved soil health enhances water retention, reducing the need for frequent irrigation.
   - **Reason 2**: Healthier soils lead to more sustainable water use practices.
5. **Community Cooperative Formation (10%)**
   - **Reason 1**: Cooperatives can manage water resources more effectively, ensuring equitable distribution.
   - **Reason 2**: Collective efforts can lead to better water conservation practices.
6. **Agricultural Subsidies (5%)**
   - **Reason 1**: Financial support helps farmers invest in water-efficient technologies.
   - **Reason 2**: Subsidies can encourage the adoption of water-saving practices.
7. **Market Access Facilitation (5%)**
   - **Reason 1**: Efficient market access reduces the time produce spends in the field, conserving water.
   - **Reason 2**: Better market access can lead to optimized water usage through improved planning.
8. **Storage and Preservation Solutions (20%)**
   - **Reason 1**: Water storage solutions ensure a steady supply of drinking water.
   - **Reason 2**: Preservation solutions can maintain water quality, ensuring it remains potable.
9. **Renewable Energy for Farming (10%)**
   - **Reason 1**: Renewable energy can power water purification and distribution systems.
   - **Reason 2**: Sustainable energy sources reduce the environmental impact of water supply operations.
10. **Disaster Resilience Planning (5%)**
    - **Reason 1**: Planning for disasters ensures water supply continuity during emergencies.
    - **Reason 2**: Resilience planning helps mitigate the impacts of natural disasters on water resources.

NOTE THE STRATEGIES FOR FOOD AND WATER ARE Suspiciously similar
We will call AI to account. But maybe there is a link (Fixed in 10.4.2.2!)

10.4.2.2 More-specific *'Water Supply'* Strategies

A correction to the 'lazy' AI

AI: *Certainly! Let's focus on creating a more specific set of strategies tailored for **water** supply availability, distinct from the strategies for food supply reliability.*

*Here are 10 strategies specifically aimed at ensuring **water** supply availability in remote areas, followed by a detailed table and explanation for the ratings.*

Water Supply Availability Strategies

1. **Solar-Powered Water Purification Systems**
2. **Rainwater Harvesting Installations**
3. **Borehole Drilling and Management**
4. **Community Water Management Training Programs**
5. **Pipeline Infrastructure Development**
6. **Water Storage Tanks and Reservoirs**
7. **Water Quality Monitoring and Treatment**
8. **Water Use Efficiency Education Campaigns**
9. **Desalination Units for Coastal Areas**
10. **Emergency Water Supply Contingency Plans**

Table: Water Supply Availability Strategies
Priority 1 Goal: [Namibia, Drinking, Solar Water Purifiers] - 15 liters per person per day within 3 years

Strategies -> Objectives	Strategy 1: Solar Powered Water Purification Systems 20%	*Strategy 2 : Storage & Rainwater Harvesting Installations 15%*	Borehole Drilling and Management	Community Water Management Training Programs	Pipeline Infrastructure Development	Water Storage Tanks and Reservoirs	Water Quality Monitoring and Treatment	Water Use Efficiency Education Campaigns	Desalination Units for Coastal Areas	Emergency Water Supply Contingency Plans
Water Supply Availability	20 %	15 %	15 %	10 %	10 %	10 %	10 %	5 %	3 %	2 %
CapEx Estimate (£ million)	2.5	1.5	2.0	1.0	3.5	2.0	1.2	0.5	3.0	1.0
OpEx Estimate (£ million/year)	0.4	0.2	0.3	0.15	0.4	0.3	0.2	0.1	0.5	0.15

Reasons for Value Percentages in Water Supply Availability

1. **Solar-Powered Water Purification Systems (20%)**
   - **Reason 1**: Provides a sustainable and renewable source of clean water, crucial for remote areas with limited infrastructure.
   - **Reason 2**: Reduces dependency on traditional energy sources, making it cost-effective in the long term.
2. **Rainwater Harvesting Installations (15%)**
   - **Reason 1**: Efficiently captures and stores rainwater, a vital resource in arid regions like Namibia.
   - **Reason 2**: Reduces the strain on other water sources during rainy seasons, ensuring a continuous supply.
3. **Borehole Drilling and Management (15%)**
   - **Reason 1**: Provides access to groundwater, which is often the most reliable water source in remote and arid areas.
   - **Reason 2**: Proper management ensures the sustainability and longevity of boreholes.
4. **Community Water Management Training Programs (10%)**
   - **Reason 1**: Empowers local communities to manage their water resources effectively, leading to sustainable usage.
   - **Reason 2**: Training programs can reduce wastage and improve water conservation practices.
5. **Pipeline Infrastructure Development (10%)**
   - **Reason 1**: Facilitates the transportation of water from sources to areas where it is needed, ensuring reliable distribution.
   - **Reason 2**: Reduces the time and labor required for water collection, improving overall efficiency.
6. **Water Storage Tanks and Reservoirs (10%)**
   - **Reason 1**: Ensures that there is a steady supply of water even during dry periods or when immediate sources are unavailable.
   - **Reason 2**: Helps in managing water supply during peak demand periods.
7. **Water Quality Monitoring and Treatment (10%)**
   - **Reason 1**: Ensures that the water being supplied is safe for drinking, preventing waterborne diseases.
   - **Reason 2**: Regular monitoring helps in maintaining the overall health of the water supply system.
8. **Water Use Efficiency Education Campaigns (5%)**
   - **Reason 1**: Educating the community on efficient water use can significantly reduce wastage.
   - **Reason 2**: Promotes a culture of conservation, which is essential for sustainable water management.
9. **Desalination Units for Coastal Areas (3%)**
   - **Reason 1**: Provides an alternative water source for coastal regions, where seawater can be converted to potable water.
   - **Reason 2**: Desalination can be a reliable solution during severe droughts when other sources are unavailable.

10.4.2.3 Two different types of '*strategy decomposition*'

Evo-steps versus ***Tasks***

- A large cost strategy (it takes a month-to-years) can be decomposed in two ways:
  - **1. *Task* Decomposition**
    - This lists the many tasks which need doing, *before* we can expect the strategy to deliver value.
    - Probably *no task alone* delivers much value. Most tasks do not deliver the planned Value Objective Improvement towards its Goal.
    - Running a marathon does not deliver value until you cross the finish line, if at all.
  - **2. Sub-strategy decomposition, into Evo-steps.**
    - This decomposes a strategy into a number of **independently implementable** *sub-strategies*, which can be implemented, in *any* sequence, and will *each* deliver value.
      - They can be implemented **in parallel**, too
    - These are **Evo Steps**; *resulting in value.*
    - A strategy can *potentially* be decomposed into Evo-steps. Not always.

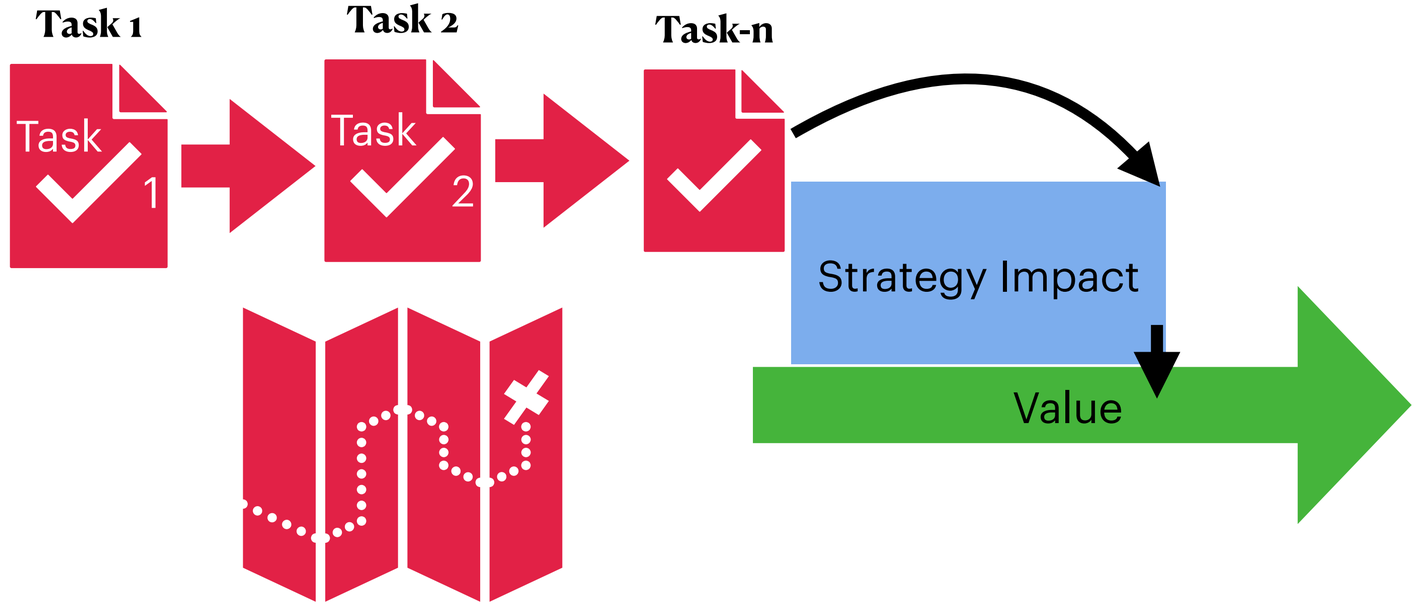


Figure 4.2.3 A. Task decomposition: Impact when last task completed well.. Strategy 'complete'

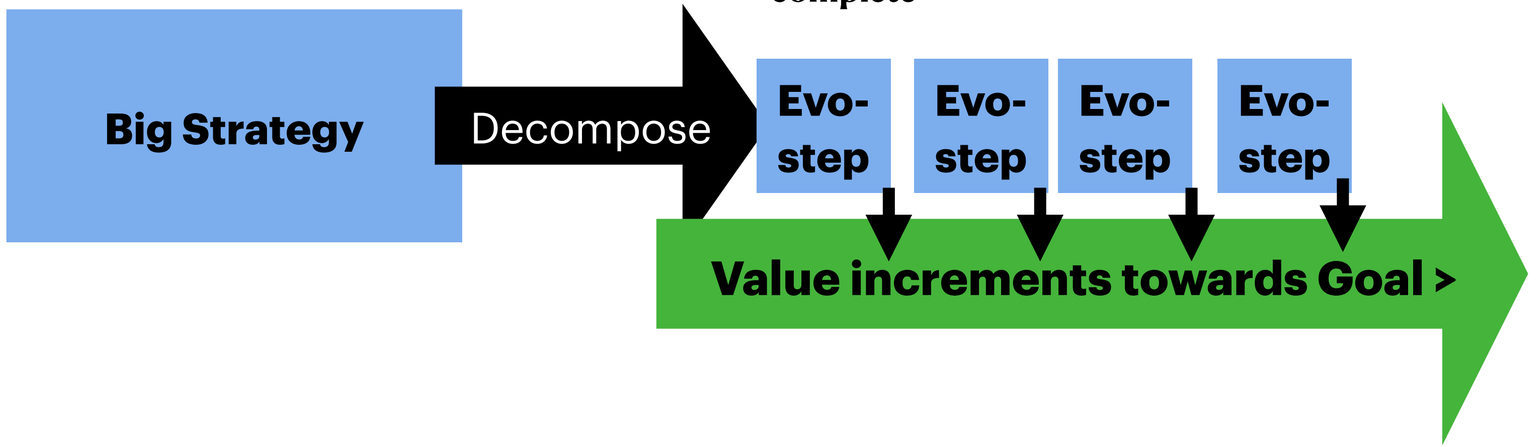


Figure 4.2.3 B. Decomposing into Evo-steps, not 'tasks', enables us to get measurable results, feedback, and planning improvement faster

Strategy 3: FOOD
Efficient Irrigation
20%



10.4.3. Specify (Strategy 3, FOOD) 'Efficient Irrigation' in Detail *Task Decomposition*, example

You ultimately need this detail, to validate effects and costs, and 'to implement', allocating tasks to people and suppliers.

PROMPT Specify the strategy *Efficient Irrigation*, in a page of practical detail about what we have to do in practice

ChatGPT
Strategy: Efficient Irrigation Systems

Objective:

Implement efficient irrigation systems to maximize water use efficiency and ensure reliable water supply for agriculture in Namibia. The aim is to increase crop yields and conserve water, contributing to the overall goal of food supply reliability.

Steps to Implement Efficient Irrigation Systems:

1. Assessment and Planning:

- Site Assessment: Conduct a thorough assessment of the agricultural land to understand the soil type, topography, crop types, and existing water sources.
- Water Resource Analysis: Analyze the available water resources, including groundwater, surface water, and rainwater harvesting potential.
- Needs Assessment: Determine the specific water needs of different crops and the most suitable irrigation methods for each type.

2. System Design:

- Irrigation Method Selection: Choose the most appropriate irrigation methods based on the site assessment. Common methods include drip irrigation, sprinkler systems, and surface irrigation.
  - Drip Irrigation: Ideal for precise water delivery directly to the root zone, minimizing evaporation and runoff.
  - Sprinkler Systems: Suitable for a wide range of crops and can cover large areas efficiently.
  - Surface Irrigation: Utilizes gravity flow to distribute water, often used for crops requiring high water volumes.
- System Layout: Design the layout of the irrigation system, including the placement of pipes, emitters, sprinklers, and control valves.
- Automation and Control: Incorporate automated systems such as timers, moisture sensors, and flow controllers to optimize water use and reduce labor.

3. Procurement of Materials:

- Pipes and Fittings: PVC or polyethylene pipes, connectors, and fittings for distributing water.
- Emitters and Sprinklers: Drip emitters, micro-sprinklers, or standard sprinklers, depending on the chosen method.
- Pumps and Filters: Pumps to ensure adequate water pressure and filters to prevent clogging of emitters.
- Control Systems: Timers, sensors, and controllers for automation.
- Installation Tools: Tools and equipment for trenching, pipe cutting, and fitting assembly.

4. Installation:

- Site Preparation: Clear the land and prepare trenches for pipe installation.
- Pipe Installation: Lay the main and lateral pipes according to the design layout.
- Emitter/Sprinkler Installation: Install emitters or sprinklers at the designated positions.
- Pump and Filter Setup: Install and connect pumps and filters to the water source.
- Control System Setup: Install and configure the automation and control systems.

5. Testing and Calibration:

- System Testing: Conduct a comprehensive test of the irrigation system to check for leaks, pressure issues, and proper operation of emitters/sprinklers.
- Calibration: Adjust the flow rates and coverage areas to ensure uniform water distribution and optimal performance.

6. Training and Capacity Building:

- Farmer Training: Conduct training sessions for local farmers on the operation and maintenance of the irrigation system.
- Maintenance Procedures: Provide guidelines on routine maintenance tasks such as cleaning filters, checking for leaks, and adjusting emitters.

7. Monitoring and Evaluation:

- Performance Monitoring: Regularly monitor the system's performance, water usage, and crop yields.
- Data Collection: Collect data on water savings, crop health, and productivity to evaluate the system's impact.
- Continuous Improvement: Make necessary adjustments and improvements based on monitoring data and feedback from farmers.

Practical Considerations:

- Cost Management: Ensure budget allocation for both capital expenses (CapEx) and operational expenses (OpEx).
- Local Adaptation: Customize the irrigation system to local conditions, considering factors such as climate, soil, and crop types.
- Sustainability: Focus on sustainable practices, including the use of renewable energy sources (e.g., solar pumps) and water conservation techniques.
- Community Involvement: Engage the local community and stakeholders throughout the process to ensure buy-in and long-term success.

Expected Outcomes:

- Increased Crop Yields: More efficient water use leads to higher and more reliable crop production.
- Water Conservation: Significant reduction in water wastage, ensuring sustainable use of water resources.
- Improved Livelihoods: Enhanced agricultural productivity and profitability for local farmers.
- Scalability: The success of the pilot project in Namibia can serve as a model for other regions with similar conditions.

By following these detailed steps, the implementation of efficient irrigation systems can contribute significantly to the goals of food supply reliability and water supply availability, driving the success of the EU-funded startup's mission in remote poor areas.

10.4.4 Decomposition of (*Strategy 3, FOOD) 'Efficient Irrigation',* into independently implementable sub-strategies: *Evo-Steps*

Strategy 3: FOOD
Efficient Irrigation
20%



Instead of one bigger task set, to implement the strategy, previous page (4.3)

Evo-steps ->	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	8	5	8	6	6	9	7	7	8	7
Cost (1-10)	3	2	4	3	3	5	4	4	5	5
Efficiency Ratio	2.67	2.50	2.00	2.00	2.00	1.80	1.75	1.75	1.60	1.40

Here are two reasons for the effectiveness on value and one reason for cost estimation for each of the 10 sub-strategies:

1. Utilizing Weather Forecast Data

- Value Effectiveness:
  1. Enhances irrigation scheduling by predicting rainfall, reducing unnecessary water usage.
  2. Helps optimize water use during dry periods, ensuring plants receive adequate moisture.
- Cost Estimation:
  1. Generally low cost as it primarily involves accessing existing weather data, potentially with some investment in weather stations or data subscriptions.

2. Mulching

- Value Effectiveness:
  1. Reduces water evaporation from soil, maintaining moisture levels.
  2. Suppresses weed growth, which can otherwise compete with crops for water.
- Cost Estimation:
  1. Low cost because it often involves using organic materials like leaves or straw, which may be locally sourced or even waste products.

3. Soil Moisture Sensors Installation

- Value Effectiveness:
  1. Provides precise measurements of soil moisture, allowing for targeted irrigation.
  2. Reduces water waste by ensuring irrigation only occurs when necessary.
- Cost Estimation:
  1. Moderate cost due to the expense of purchasing and installing the sensors, as well as potential maintenance.

4. Regular Maintenance and Inspection

- Value Effectiveness:
  1. Ensures that the irrigation system operates efficiently, reducing water loss due to leaks.
  2. Prevents breakdowns that could lead to overwatering or underwatering.
- Cost Estimation:
  1. Low to moderate cost, involving labor and potential replacement parts.

5. Proper Plant Selection

- Value Effectiveness:
  1. Choosing drought-resistant plants reduces overall water needs.
  2. Selecting plants suited to local climate and soil conditions improves growth and water efficiency.
- Cost Estimation:
  1. Moderate cost, mainly involving the purchase of specific plant varieties.

6. Drip Irrigation System Installation

- Value Effectiveness:
  1. Delivers water directly to the plant roots, minimizing evaporation and runoff.
  2. Provides a consistent moisture level, which can enhance plant health and yield.
- Cost Estimation:
  1. High cost due to the complexity of installing the system and potential for high initial investment in materials.

7. Rainwater Harvesting Implementation

- Value Effectiveness:
  1. Captures and stores rainwater for later use, reducing dependency on external water sources.
  2. Provides a sustainable and environmentally friendly water source.
- Cost Estimation:
  1. Moderate cost, involving the setup of collection systems, storage tanks, and potential filtration systems.

8. Training and Education for Farmers

- Value Effectiveness:
  1. Empowers farmers with knowledge to implement efficient irrigation practices.
  2. Enhances overall farm management, leading to better resource utilization and crop yields.
- Cost Estimation:
  1. Moderate cost, including training materials, workshops, and possibly hiring experts.

9. Smart Irrigation Controllers

- Value Effectiveness:
  1. Automates irrigation based on real-time data, optimizing water use.
  2. Reduces labor and human error in managing irrigation schedules.
- Cost Estimation:
  1. High cost due to the technology involved and potential ongoing maintenance or software updates.

10. Water Recycling and Reuse

- Value Effectiveness:
  1. Reduces water consumption by reusing treated wastewater for irrigation.
  2. Provides a reliable water source, especially in areas with limited water availability.
- Cost Estimation:
  1. High cost due to the infrastructure needed for treatment and distribution systems.

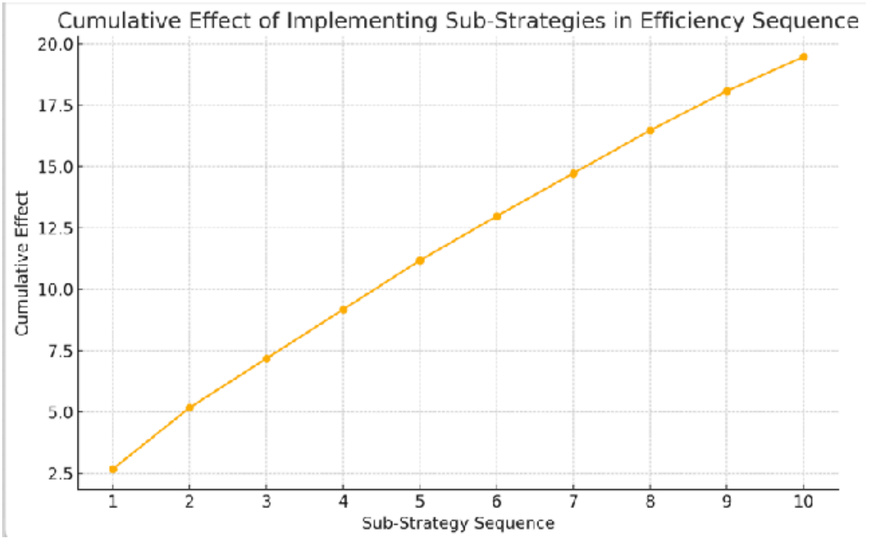
10.4.5 Local Prioritization: Most-efficient Evo-step sub-strategy, first

Here is the Impact Estimation table, with the columns sequenced in descending order of Efficiency Ratio

	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	8	5	8	6	6	9	7	7	8	7
Cost (1-10)	3	2	4	3	3	5	4	4	5	5
Efficiency Ratio	2.67	2.50	2.00	2.00	2.00	1.80	1.75	1.75	1.60	1.40

Final Sequenced Sub-Strategies by Efficiency Ratio

1. **Utilizing Weather Forecast Data (2.67)**
2. **Mulching (2.50)**
3. **Soil Moisture Sensors Installation (2.00)**
4. **Regular Maintenance and Inspection (2.00)**
5. **Proper Plant Selection (2.00)**
6. **Drip Irrigation System Installation (1.80)**
7. **Rainwater Harvesting Implementation (1.75)**
8. **Training and Education for Farmers (1.75)**
9. **Smart Irrigation Controllers (1.60)**
10. **Water Recycling and Reuse (1.40)**



10.4.6. 'Tags': can refer to different sets of specs. For the same 'entity'

Efficient Irrigation Systems:

Efficient Irrigation Systems:

- Type: Summary.
- **Description**: Implement drip irrigation and other water-efficient systems.
- **CapEx**: £2.0 million for infrastructure and installation.
- **OpEx**: £0.4 million/ year for maintenance and operations.

Efficient Irrigation Systems: Type: Task Description

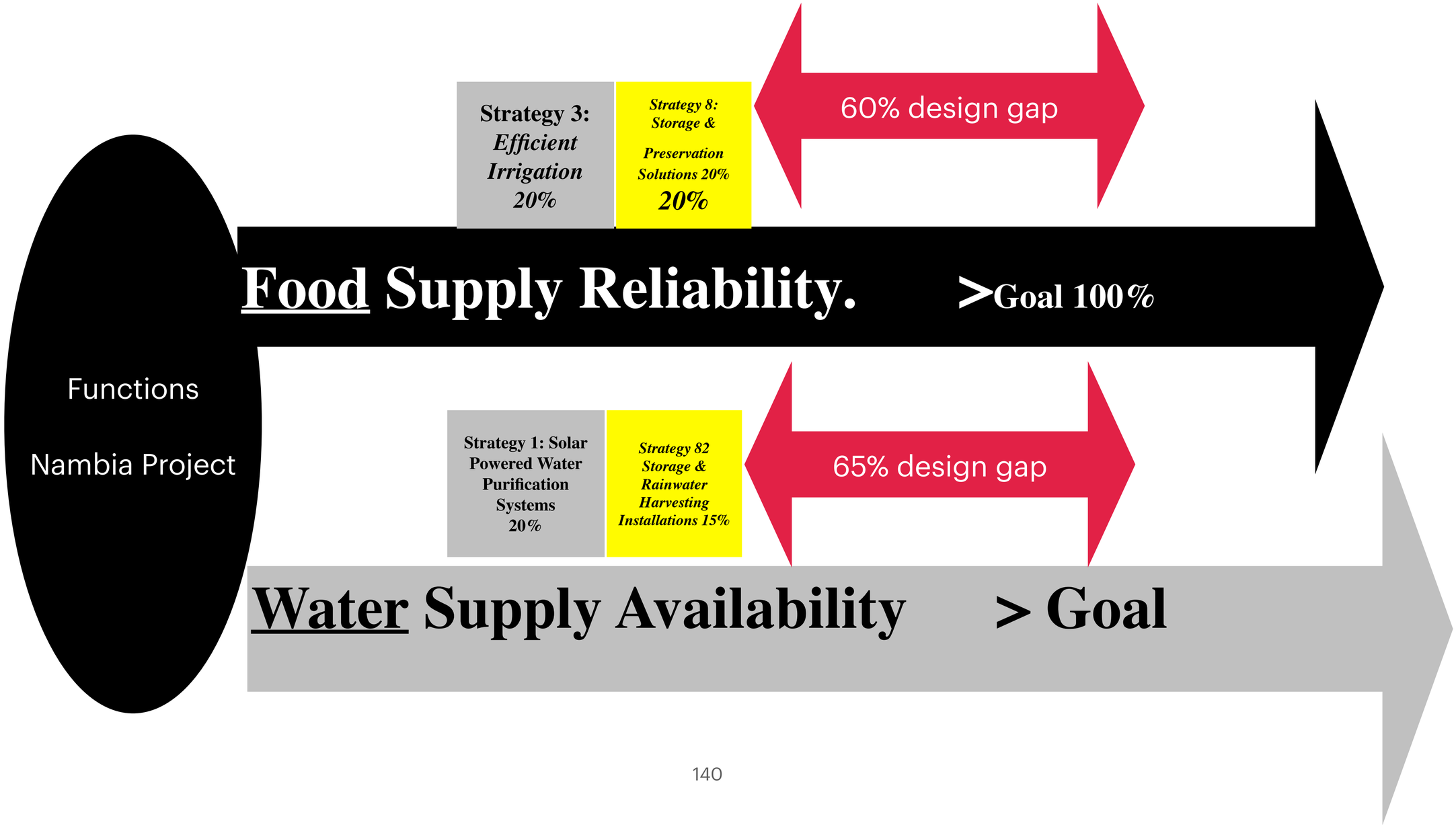
1. **Assessment and Planning:**
   - Site Assessment: Conduct a thorough assessment of the agricultural land to understand the soil type, topography, crop types, and existing water sources.
   - Water Resource Analysis: Analyze the available water resources, including groundwater, surface water, and rainwater harvesting potential.
   - Needs Assessment: Determine the specific water needs of different crops and the most suitable irrigation methods for each type.
2. **System Design:**
   - Irrigation Method Selection: Choose the most appropriate irrigation methods based on the site assessment. Common methods include drip irrigation, sprinkler systems, and surface irrigation.
     - Drip Irrigation: Ideal for precise water delivery directly to the root zone, minimizing evaporation and runoff.
     - Sprinkler Systems: Suitable for a wide range of crops and can cover large areas efficiently.
     - Surface Irrigation: Utilizes gravity flow to distribute water, often used for crops requiring high water volumes.
   - System Layout: Design the layout of the irrigation system, including the placement of pipes, emitters, sprinklers, and control valves.
   - Automation and Control: Incorporate automated systems such as timers, moisture sensors, and flow controllers to optimize water use and reduce labor.
3. **Procurement of Materials:**
   - Pipes and Fittings: PVC or polyethylene pipes, connectors, and fittings for distributing water.
   - Emitters and Sprinklers: Drip emitters, micro-sprinklers, or standard sprinklers, depending on the chosen method.
   - Pumps and Filters: Pumps to ensure adequate water pressure and filters to prevent clogging of emitters.
   - Control Systems: Timers, sensors, and controllers for automation.
   - Installation Tools: Tools and equipment for trenching, pipe cutting, and fitting assembly.
4. **Installation:**
   - Site Preparation: Clear the land and prepare trenches for pipe installation.
   - Pipe Installation: Lay the main and lateral pipes according to the design layout.
   - Emitter/Sprinkler Installation: Install emitters or sprinklers at the designated positions.
   - Pump and Filter Setup: Install and connect pumps and filters to the water source.
   - Control System Setup: Install and configure the automation and control systems.
5. **Testing and Calibration:**
   - System Testing: Conduct a comprehensive test of the irrigation system to check for leaks, pressure issues, and proper operation of emitters/sprinklers.
   - Calibration: Adjust the flow rates and coverage areas to ensure uniform water distribution and optimal performance.
6. **Training and Capacity Building:**
   - Farmer Training: Conduct training sessions for local farmers on the operation and maintenance of the irrigation system.
   - Maintenance Procedures: Provide guidelines on routine maintenance tasks such as cleaning filters, checking for leaks, and adjusting emitters.
7. **Monitoring and Evaluation:**
   - Performance Monitoring: Regularly monitor the system's performance, water usage, and crop yields.
   - Data Collection: Collect data on water savings, crop health, and productivity to evaluate the system's impact.
   - Continuous Improvement: Make necessary adjustmen

Efficient Irrigation Systems: Type: Evo Steps

1. **Utilizing Weather Forecast Data**
   - Value Effectiveness:
     1. Enhances irrigation scheduling by predicting rainfall, reducing unnecessary water usage.
     2. Helps optimize water use during dry periods, ensuring plants receive adequate moisture.
   - Cost Estimation:
     1. Generally low cost as it primarily involves accessing existing weather data, potentially with some investment in weather stations or data subscriptions.
2. **Mulching**
   - Value Effectiveness:
     1. Reduces water evaporation from soil, maintaining moisture levels.
     2. Suppresses weed growth, which can otherwise compete with crops for water.
   - Cost Estimation:
     1. Low cost because it often involves using organic materials like leaves or straw, which may be locally sourced or even waste products.
3. **Soil Moisture Sensors Installation**
   - Value Effectiveness:
     1. Provides precise measurements of soil moisture, allowing for targeted irrigation.
     2. Reduces water waste by ensuring irrigation only occurs when necessary.
   - Cost Estimation:
     1. Moderate cost due to the expense of purchasing and installing the sensors, as well as potential maintenance.
4. **Regular Maintenance and Inspection**
   - Value Effectiveness:
     1. Ensures that the irrigation system operates efficiently, reducing water loss due to leaks.
     2. Prevents breakdowns that could lead to overwatering or underwatering.
   - Cost Estimation:
     1. Low to moderate cost, involving labor and potential replacement parts.
5. **Proper Plant Selection**
   - Value Effectiveness:
     1. Choosing drought-resistant plants reduces overall water needs.
     2. Selecting plants suited to local climate and soil conditions improves growth and water efficiency.
   - Cost Estimation:
     1. Moderate cost, mainly involving the purchase of specific plant varieties.
6. **Drip Irrigation System Installation**
   - Value Effectiveness:
     1. Delivers water directly to the plant roots, minimizing evaporation and runoff.
     2. Provides a consistent moisture level, which can enhance plant health and yield.
   - Cost Estimation:
     1. High cost due to the complexity of installing the system and potential for high initial investment in materials.
7. **Rainwater Harvesting Implementation**
   - Value Effectiveness:
     1. Captures and stores rainwater for later use, reducing dependency on external water sources.
     2. Provides a sustainable and environmentally friendly water source.
   - Cost Estimation:
     1. Moderate cost, involving the setup of collection systems, storage tanks, and potential filtration systems.
8. **Training and Education for Farmers**
   - Value Effectiveness:
     1. Empowers farmers with knowledge to implement efficient irrigation practices.
     2. Enhances overall farm management, leading to better resource utilization and crop yields.
   - Cost Estimation:
     1. Moderate cost, including training materials, workshops, and possibly hiring experts.
9. **Smart Irrigation Controllers**
   - Value Effectiveness:
     1. Automates irrigation based on real-time data, optimizing water use.
     2. Reduces labor and human error in managing irrigation schedules.
   - Cost Estimation:
     1. High cost due to the technology involved and potential ongoing maintenance or software updates.
10. **Water Recycling and Reuse**

- Value Effectiveness:
  1. Reduces water consumption by reusing treated wastewater for irrigation.
  2. Provides a reliable water source, especially in areas with limited water availability.
- Cost Estimation:
  1. High cost due to the infrastructure needed for treatment and distribution systems.

10.4.7 Last page of ‘Design’ for Product and Qualities: The ‘Means’ to the ‘Values’



10.4.8 Design Principles [CE]

Principles: The Design Engineering Process

1. The Principle of 'Design Ideas are only as Good as the Requirements Satisfied'

Design ideas cannot be correctly judged or validated except with respect to all the performance and cost requirements they must satisfy.

2. The Principle of 'The Best Chess Move'
You should try with each increment of design specification or design implementation, to get the best possible satisfaction of your unsatisfied performance requirements, from your unused cost budgets.

3. The Principle of 'Results Beat Theory'
Design ideas are only as good as their real results, not their intent.

4. The Principle of 'Early Surprises'
You never know how it works, until you have actually tried out a design idea in practice. Get surprised as early as possible!

5. **The Principle of 'It's Not Just What You Do, It's How You Do It'**
Design ideas must try to exercise control over both design content and design implementation. The devil is in the details!

6. **The Principle of 'Good is Not Always Good Enough'**
A 'good' design idea might not be good enough to meet all your targets on time.

7. The Principle of 'Designs should have Good Return on their Investment'

'Good' design ideas might cost too much, sooner or later.

8. **The Principle of 'Sneaky Gremlins'**
Apparently 'good' design ideas might have subtly-hidden nasty side effects. Estimate them, know when you don't know them, measure them, and don't assume they won't hurt you! They will show you no sympathy!

9. **The Principle of 'Design Beats Test'**
Design performance 'in', and design 'to control' costs: You cannot test quality into a badly-designed system.

10. The Principle of 'Eternal Vigilance for the Butterfly Effect'

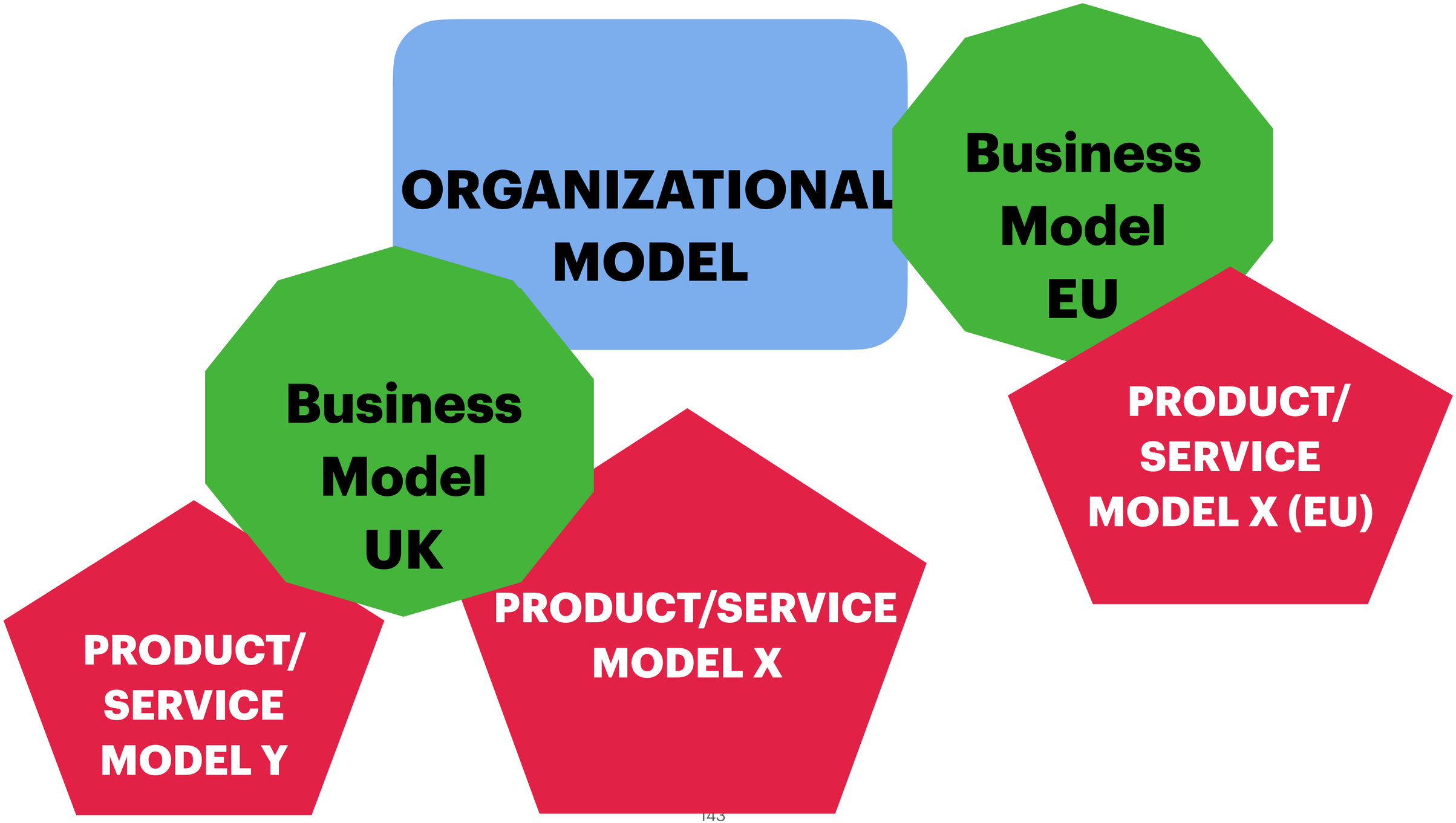
You never finally know about a design idea's effects; Tomorrow's slightest change might ruin your whole project. Even initially successful designs might have to be adjusted for growth and change.

These are examples of decomposition, and of using AI for decomposition

11. Strategy (the business model, the organization)

Business Model UK

ORGANIZATIONAL MODEL


ORGANIZATIONAL MODEL
Business Model EU
Business Model UK
PRODUCT/ SERVICE MODEL X (EU)
PRODUCT/SERVICE MODEL X
PRODUCT/ SERVICE MODEL Y

The Business Model Strategy

Business Model Value Objectives

Business Model UK

Based on the search results and current trends in social entrepreneurship and education technology, here are the top ten value objectives for a social startup business model in the UK working on a European educational tool product:

1. Social Impact: Focus on creating measurable positive change in the education sector across Europe, addressing specific challenges or inequalities in learning[1].

2. Scalability: Design the educational tool to be easily scalable across different European countries, considering various educational systems and languages[5].

3. Sustainability: Develop a business model that ensures long-term financial sustainability while maintaining the social mission[1][4].

4. Innovation: Incorporate innovative technologies or pedagogical approaches that set the product apart in the competitive edtech market[2][5].

5. Accessibility: Ensure the educational tool is accessible to diverse learners, including those from disadvantaged backgrounds or with special educational needs[1][3].

6. User-Centric Design: Prioritize user experience and engagement, tailoring the product to meet the needs of students, teachers, and educational institutions[5].

7. Data-Driven Improvement: Utilize data analytics to continuously improve the product and demonstrate its effectiveness in enhancing learning outcomes[2].

8. Collaboration: Foster partnerships with educational institutions, policymakers, and other stakeholders to increase adoption and impact[4].

9. Environmental Sustainability: Incorporate eco-friendly practices in product development and operations, aligning with growing environmental concerns[1].

10. Ethical AI and Data Use: If using artificial intelligence or collecting user data, ensure ethical practices and compliance with European data protection regulations[2].

These value objectives combine the principles of social entrepreneurship with the specific needs of the education sector and the European market. By focusing on these areas, a social startup can create a product that not only achieves commercial success but also contributes significantly to improving education across Europe.

Citations:

[1] https://www.f6s.com/companies/social-impact/united-kingdom/co

[2] https://sifted.eu/articles/uk-startup-london-investor-manchester

[3] https://www.bigissue.com/news/activism/these-are-officially-the-best-social-enterprises-in-the-uk-right-now/

[4] https://www.socialenterprise.org.uk

[5] https://www.founderoo.co/resources/the-top-44-startups-in-the-uk

The Business Model Strategy

Business Model Value Objectives, in Planguage

Business Model UK

Here are the top ten value objectives for a social startup business model in the UK working on a European educational tool product, expressed in Planguage format:

1. Social Impact

Ambition: Create measurable positive change in the European education sector

Scale: Number of students benefiting from the tool

Tolerable (1 year): 10,000 students

Goal (3 years): 500,000 students

Goal (6 years): 5 million students

Critical Stakeholders: Students, Teachers, Education Ministries

2. Scalability

Ambition: Expand tool adoption across European countries

Scale: Number of European countries where the tool is implemented

Tolerable (1 year): 3 countries

Goal (3 years): 10 countries

Goal (6 years): 20 countries

Critical Stakeholders: Education Ministries, School Administrators, EU Education Policymakers

3. Sustainability

Ambition: Achieve financial sustainability while maintaining social mission

Scale: Percentage of operational costs covered by revenue

Tolerable (1 year): 30%

Goal (3 years): 80%

Goal (6 years): 120% (profitability)

Critical Stakeholders: Investors, Board of Directors, Customers

4. Innovation

Ambition: Incorporate cutting-edge technologies and pedagogical approaches

Scale: Number of innovative features implemented

Tolerable (1 year): 2 features

Goal (3 years): 5 features

Goal (6 years): 10 features

Critical Stakeholders: R&D Team, Education Experts, Tech Partners

5. Accessibility

Ambition: Ensure tool is accessible to diverse learners

Scale: Percentage of tool features meeting WCAG 2.1 AA standards

Tolerable (1 year): 70%

Goal (3 years): 90%

Goal (6 years): 100%

Critical Stakeholders: Accessibility Experts, Special Education Teachers, Diverse Student Groups

6. User-Centric Design

Ambition: Tailor product to meet needs of students, teachers, and institutions

Scale: User satisfaction score (out of 5)

Tolerable (1 year): 3.5

Goal (3 years): 4.2

Goal (6 years): 4.8

Critical Stakeholders: UX Designers, Students, Teachers

7. Data-Driven Improvement

Ambition: Utilize analytics to enhance product effectiveness

Scale: Percentage improvement in learning outcomes

Tolerable (1 year): 5%

Goal (3 years): 15%

Goal (6 years): 30%

Critical Stakeholders: Data Scientists, Educators, Education Researchers

8. Collaboration

Ambition: Foster partnerships to increase adoption and impact

Scale: Number of strategic partnerships

Tolerable (1 year): 5 partnerships

Goal (3 years): 20 partnerships

Goal (6 years): 50 partnerships

Critical Stakeholders: Education Institutions, EdTech Companies, Policymakers

9. Environmental Sustainability

Ambition: Incorporate eco-friendly practices in operations

Scale: Percentage reduction in carbon footprint

Tolerable (1 year): 10%

Goal (3 years): 30%

Goal (6 years): 60%

Critical Stakeholders: Environmental Consultants, Suppliers, Employees

10. Ethical AI and Data Use

Ambition: Ensure ethical AI practices and GDPR compliance

Scale: Percentage of data processing activities fully compliant

Tolerable (1 year): 95%

Goal (3 years): 99%

Goal (6 years): 100%

Critical Stakeholders: Legal Team, Data Protection Officers, EU Regulators

Citations:

[1] https://single-market-economy.ec.europa.eu/

The ‘Organizational Strategy’ Design

Business Model UK

Business Model

Based on the research and insights from social entrepreneurship and business model innovation, here are the top ten most effective business model strategies for reaching the 6-year goals, along with an Impact Estimation table rating their effectiveness:

1. Multi-stakeholder Collaboration
2. Scalable Technology Platform
3. Hybrid Revenue Model
4. Impact Measurement and Reporting
5. User-Centric Design and Co-creation
6. Strategic Partnerships
7. Circular Economy Integration
8. Freemium Model for Accessibility
9. Social Franchise Model
10. Blockchain for Transparency

This Impact Estimation table provides a quantitative assessment of how each business model strategy contributes to reaching the 6-year goals across key objectives. The percentages represent the estimated effectiveness of each strategy in achieving the specific goal.

The "Total" column shows the overall effectiveness of each strategy across all objectives. This allows for a quick comparison of which strategies might be most impactful overall and which objectives are most strongly supported by the collective strategies.

Based on this analysis, the Scalable Technology Platform appears to be the most effective overall strategy, with User-Centric Design and Co-creation, and Strategic Partnerships following closely behind.
However, it's important to note that a combination of these strategies would likely yield the best results, as they complement each other in addressing different aspects of the business model goals.

STRATEGIES / Objectives	Multi-stakeholder Collaboration	Scalable Technology Platform	Hybrid Revenue Model	Impact Measurement and Reporting	User-Centric Design and Co-creation	Strategic Partnerships	Circular Economy Integration	Freemium Model for Accessibility	Social Franchise Model	Blockchain for Transparency
Social Impact	80 %	70 %	60 %	85 %	75 %	80 %	85 %	70 %	75 %	65 %
Scal-ability	70 %	90 %	75 %	65 %	70 %	85 %	60 %	80 %	85 %	70 %
Sustain-ability	75 %	80 %	90 %	70 %	65 %	75 %	80 %	65 %	70 %	60 %
Inno-vation	60 %	85 %	65 %	60 %	80 %	70 %	75 %	60 %	65 %	75 %
Access-ibility	65 %	75 %	70 %	55 %	85 %	70 %	65 %	90 %	75 %	60 %
Ethical AI & Data Use	55 %	70 %	50 %	80 %	60 %	65 %	55 %	55 %	60 %	90 %
Total	405 %	470 %	410 %	415 %	435 %	445 %	420 %	420 %	430 %	

The organization Model

ORGANIZATIONAL MODEL

Organizational *Values*

Top 10 Organizational Values in Planguage

1. **Agility**
   - Ambition: Maintain flexibility and rapid responsiveness to market changes.
   - Scale: Number of significant pivots or adaptations per year.
   - Goal (6 years): 10 major pivots/adaptations.
   - Stakeholders: Leadership Team, Employees, Investors.

2. **Innovation Culture**
   - Ambition: Foster a culture of continuous innovation.
   - Scale: Number of new ideas implemented annually.
   - Goal (6 years): 50 new ideas implemented.
   - Stakeholders: R&D Team, Employees, Customers.

3. **Diversity and Inclusion**
   - Ambition: Ensure a diverse and inclusive workplace.
   - Scale: Percentage of workforce from diverse backgrounds.
   - Goal (6 years): 50% diverse workforce.
   - Stakeholders: HR Department, Employees, Community.

4. **Sustainability**
   - Ambition: Operate in an environmentally sustainable manner.
   - Scale: Reduction in carbon footprint.
   - Goal (6 years): 60% reduction in carbon footprint.
   - Stakeholders: Environmental Consultants, Suppliers, Employees.

5. **Employee Well-being**
   - Ambition: Prioritize the well-being and development of employees.
   - Scale: Employee satisfaction score (out of 5).
   - Goal (6 years): 4.8 satisfaction score.
   - Stakeholders: HR Department, Employees, Leadership Team.

6. **Customer Centricity**
   - ***Ambition***: **Place customers at the core of all decisions.**
   - ***Scale***: **Customer satisfaction score (out of 5).**
   - ***Goal*** **(6 years): 4.8 satisfaction score.**
   - ***Stakeholders***: **Customer Support, Sales Team, Customers.**

7. **Transparency**
   - Ambition: Maintain high levels of transparency in operations.
   - Scale: Number of transparency reports published annually.
   - Goal (6 years): 6 reports per year.
   - Stakeholders: Legal Team, Investors, Customers.

8. **Collaboration**
   - Ambition: Foster a collaborative working environment.
   - Scale: Number of cross-functional projects.
   - Goal (6 years): 30 cross-functional projects.
   - Stakeholders: All Departments, Employees, Partners.

9. **Social Responsibility**
   - Ambition: Engage in socially responsible practices.
   - Scale: Number of community initiatives supported annually.
   - Goal (6 years): 20 initiatives supported.
   - Stakeholders: Community, Employees, Customers.

10. **Scalability**
    - Ambition: Ensure the organization can scale efficiently.
    - Scale: Growth in organizational size and market reach.
    - Goal (6 years): 200% growth in size and reach.
    - Stakeholders: Leadership Team, Investors, Employees.

The organization Model

ORGANIZATIONAL MODEL

Organizational Strategies

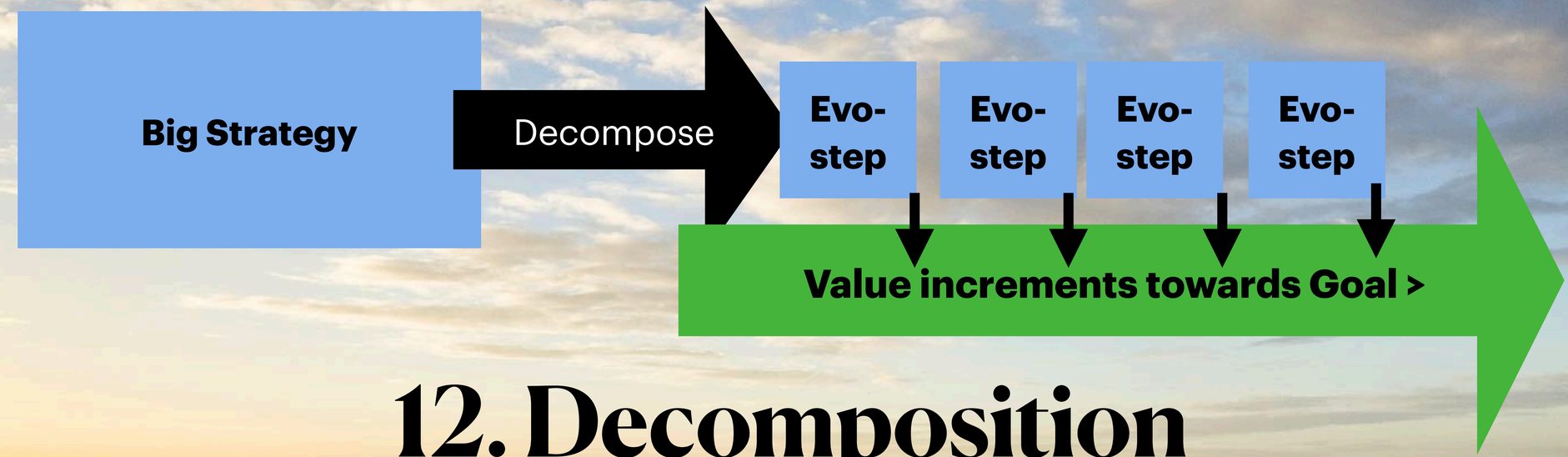
Top 10 Effective Strategies for Delivering Organizational Values

1. **Agile Management Practices**
2. **Innovation Labs and Hackathons**
3. **Diversity Hiring Programs**
4. **Sustainable Operations Initiatives**
5. **Employee Wellness Programs**
6. **Customer Feedback Loops**
7. **Transparent Reporting Mechanisms**
8. **Cross-Functional Teams**
9. **Corporate Social Responsibility (CSR) Programs**
10. **Scalable Infrastructure Development**

Insights

- **Scalable Infrastructure Development is the most effective strategy overall, with a total effectiveness of 690%.**
- Agile Management Practices and Cross-Functional Teams also show strong effectiveness, indicating the importance of flexibility and collaboration in achieving organizational goals.
- Each strategy contributes uniquely to various objectives, suggesting that a combination of these strategies would be beneficial for maximizing impact.

Objectives	Agile Management Practices	Innovation Labs and Hackathons	Diversity Hiring Programs	Sustainable Operations Initiatives	Employee Wellness Programs	Customer Feedback Loops	Transparent Reporting Mechanisms	Cross-Functional Teams	CSR Programs	Scalable Infrastructure Development	Total
Agility	90 %	70 %	60 %	50 %	60 %	70 %	50 %	70 %	50 %	80 %	650 %
Innovation Culture	70 %	90 %	60 %	50 %	60 %	70 %	50 %	70 %	50 %	70 %	640 %
Diversity and Inclusion	60 %	50 %	90 %	50 %	60 %	50 %	50 %	60 %	70 %	60 %	600 %
Sustainability	50 %	50 %	50 %	90 %	50 %	50 %	60 %	50 %	60 %	70 %	580 %
Employee Well-being	60 %	60 %	60 %	50 %	90 %	60 %	50 %	60 %	60 %	60 %	610 %
Customer Centricity	60 %	60 %	50 %	50 %	60 %	90 %	50 %	60 %	50 %	70 %	610 %
Transparency	50 %	50 %	50 %	60 %	50 %	50 %	90 %	50 %	60 %	60 %	570 %
Collaboration	70 %	70 %	60 %	50 %	60 %	60 %	50 %	90 %	60 %	70 %	650 %
Social Responsibility	50 %	50 %	70 %	60 %	50 %	50 %	60 %	60 %	90 %	60 %	600 %
Scalability	80 %	70 %	60 %	70 %	60 %	70 %	60 %	70 %	60 %	90 %	690 %
Total	640 %	620 %	610 %	580 %	600 %	630 %	570 %	640 %	630 %	690 %	6210



12. Decomposition

The art of dividing up systems strategies, and requirements, so that we get small Evo-step delivery cycles, and the benefits of low risks, feedback and re-tuning, as well as visible good value results early

12.10.4.2.3 Two different types of '*strategy decomposition*'

Evo-steps versus ***Tasks***

- A *large-cost* strategy (it takes a month-to-years) can be decomposed in two ways:
  - **1. *Task* Decomposition**
    - This lists the many tasks which need doing, *before* we can expect the strategy to deliver value.
    - Probably *no task alone* delivers much value. Most tasks do not deliver the planned Value Objective Improvement towards its Goal.
    - Running a marathon does not deliver value until you cross the finish line, if you finish at all.
  - **2. Sub-strategy decomposition, into Evo-steps.**
    - This decomposes a strategy into a number of **independently implementable** *sub-strategies*, which can be implemented, in *any* sequence, and will *each* deliver value.
      - They can be implemented **in parallel**, too
    - These are **Evo Steps**; *resulting in value.*
    - A strategy can *potentially* be decomposed into Evo-steps. Not always. But far more often that people believe, who are not trained in how to decompose.

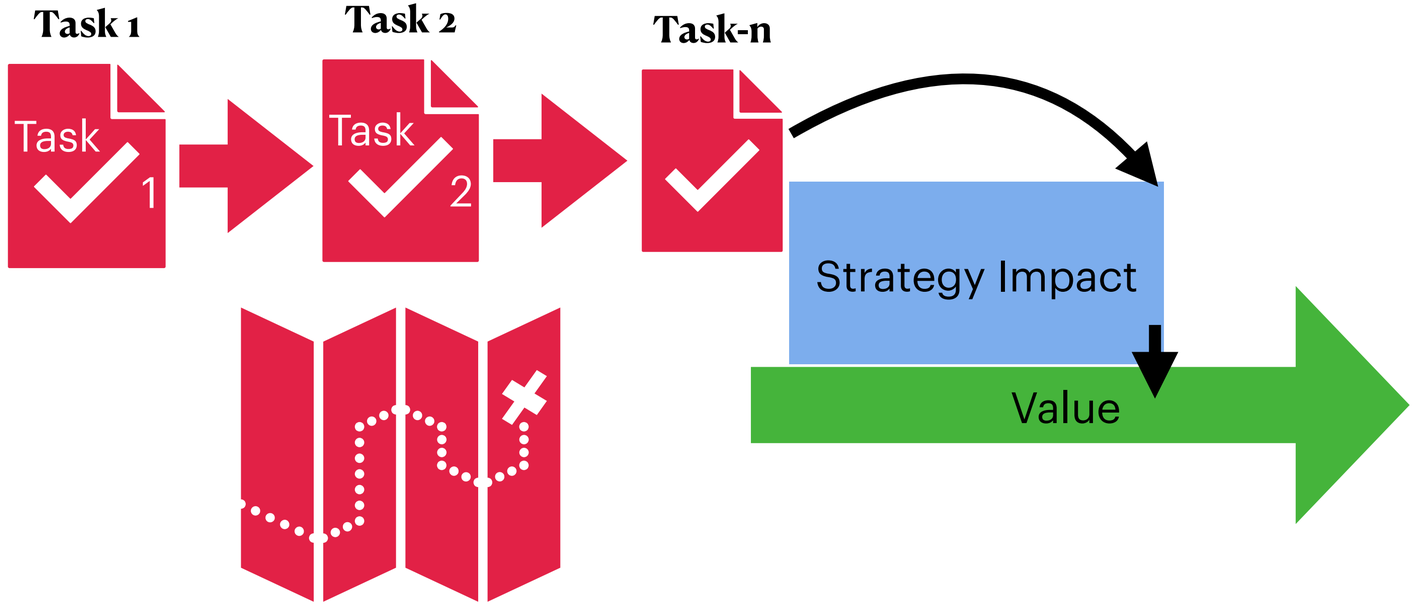


Figure 4.2.3 A. Task decomposition: Impact when last task completed well. Strategy 'complete'

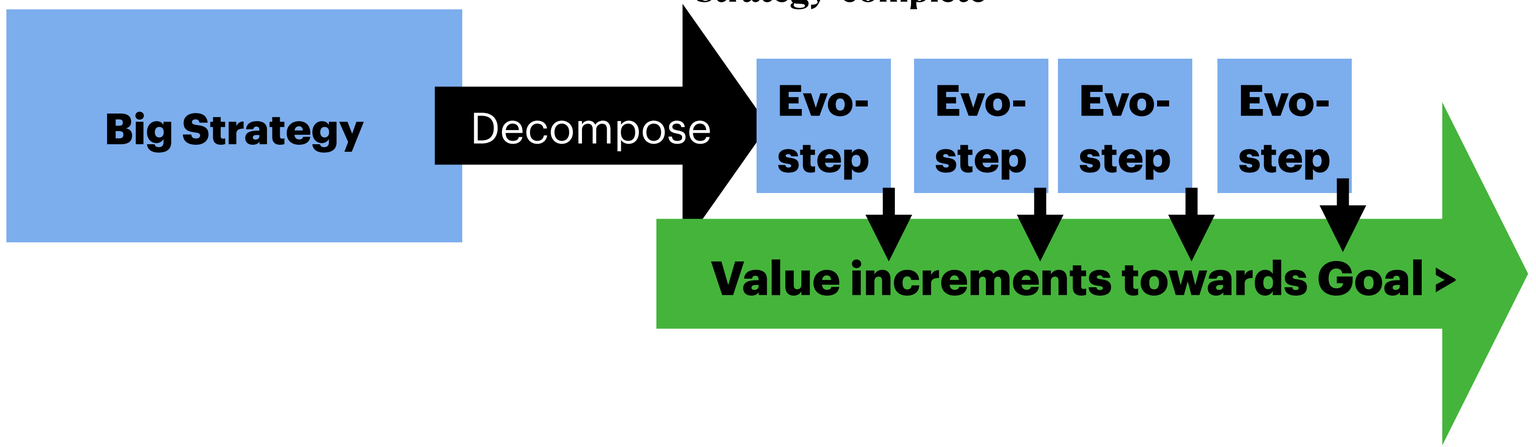


Figure 4.2.3 B. Decomposing into Evo-steps, not 'tasks', enables us to get measurable results, feedback, and planning improvement faster

Strategy 3: FOOD
Efficient Irrigation
20%



12.10.4.3. Specify (Strategy 3, FOOD) 'Efficient Irrigation' in Detail *Task Decomposition*, example

You ultimately need this level of process-detail, to validate effects and costs, and 'to implement', allocating tasks to people and suppliers.

PROMPT Specify the strategy *Efficient Irrigation*, in a page of practical detail about what we have to do in practice

ChatGPT
Strategy: Efficient Irrigation Systems

Objective:

Implement efficient irrigation systems to maximize water use efficiency and ensure reliable water supply for agriculture in Namibia. The aim is to increase crop yields and conserve water, contributing to the overall goal of food supply reliability.

Steps to Implement Efficient Irrigation Systems:

1. Assessment and Planning:

- Site Assessment: Conduct a thorough assessment of the agricultural land to understand the soil type, topography, crop types, and existing water sources.
- Water Resource Analysis: Analyze the available water resources, including groundwater, surface water, and rainwater harvesting potential.
- Needs Assessment: Determine the specific water needs of different crops and the most suitable irrigation methods for each type.

2. System Design:

- Irrigation Method Selection: Choose the most appropriate irrigation methods based on the site assessment. Common methods include drip irrigation, sprinkler systems, and surface irrigation.
  - Drip Irrigation: Ideal for precise water delivery directly to the root zone, minimizing evaporation and runoff.
  - Sprinkler Systems: Suitable for a wide range of crops and can cover large areas efficiently.
  - Surface Irrigation: Utilizes gravity flow to distribute water, often used for crops requiring high water volumes.
- System Layout: Design the layout of the irrigation system, including the placement of pipes, emitters, sprinklers, and control valves.
- Automation and Control: Incorporate automated systems such as timers, moisture sensors, and flow controllers to optimize water use and reduce labor.

3. Procurement of Materials:

- Pipes and Fittings: PVC or polyethylene pipes, connectors, and fittings for distributing water.
- Emitters and Sprinklers: Drip emitters, micro-sprinklers, or standard sprinklers, depending on the chosen method.
- Pumps and Filters: Pumps to ensure adequate water pressure and filters to prevent clogging of emitters.
- Control Systems: Timers, sensors, and controllers for automation.
- Installation Tools: Tools and equipment for trenching, pipe cutting, and fitting assembly.

4. Installation:

- Site Preparation: Clear the land and prepare trenches for pipe installation.
- Pipe Installation: Lay the main and lateral pipes according to the design layout.
- Emitter/Sprinkler Installation: Install emitters or sprinklers at the designated positions.
- Pump and Filter Setup: Install and connect pumps and filters to the water source.
- Control System Setup: Install and configure the automation and control systems.

5. Testing and Calibration:

- System Testing: Conduct a comprehensive test of the irrigation system to check for leaks, pressure issues, and proper operation of emitters/sprinklers.
- Calibration: Adjust the flow rates and coverage areas to ensure uniform water distribution and optimal performance.

6. Training and Capacity Building:

- Farmer Training: Conduct training sessions for local farmers on the operation and maintenance of the irrigation system.
- Maintenance Procedures: Provide guidelines on routine maintenance tasks such as cleaning filters, checking for leaks, and adjusting emitters.

7. Monitoring and Evaluation:

- Performance Monitoring: Regularly monitor the system's performance, water usage, and crop yields.
- Data Collection: Collect data on water savings, crop health, and productivity to evaluate the system's impact.
- Continuous Improvement: Make necessary adjustments and improvements based on monitoring data and feedback from farmers.

Practical Considerations:

- Cost Management: Ensure budget allocation for both capital expenses (CapEx) and operational expenses (OpEx).
- Local Adaptation: Customize the irrigation system to local conditions, considering factors such as climate, soil, and crop types.
- Sustainability: Focus on sustainable practices, including the use of renewable energy sources (e.g., solar pumps) and water conservation techniques.
- Community Involvement: Engage the local community and stakeholders throughout the process to ensure buy-in and long-term success.

Expected Outcomes:

- Increased Crop Yields: More efficient water use leads to higher and more reliable crop production.
- Water Conservation: Significant reduction in water wastage, ensuring sustainable use of water resources.
- Improved Livelihoods: Enhanced agricultural productivity and profitability for local farmers.
- Scalability: The success of the pilot project in Namibia can serve as a model for other regions with similar conditions.

By following these detailed steps, the implementation of efficient irrigation systems can contribute significantly to the goals of food supply reliability and water supply availability, driving the success of the EU-funded startup's mission in remote poor areas.

12.10.4.4 Decomposition of (*Strategy 3, FOOD) 'Efficient Irrigation',* into independently implementable sub-strategies: *Evo-Steps*

Strategy 3: FOOD
Efficient Irrigation
20%



Instead of one bigger task set, to implement the strategy, previous page (4.3)

Evo-steps ->	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	8	5	8	6	6	9	7	7	8	7
Cost (1-10)	3	2	4	3	3	5	4	4	5	5
Efficiency Ratio	2.67	2.50	2.00	2.00	2.00	1.80	1.75	1.75	1.60	1.40



Here are two reasons for the effectiveness on value and one reason for cost estimation for each of the 10 sub-strategies:

1. Utilizing Weather Forecast Data

- Value Effectiveness:
  1. Enhances irrigation scheduling by predicting rainfall, reducing unnecessary water usage.
  2. Helps optimize water use during dry periods, ensuring plants receive adequate moisture.
- Cost Estimation:
  1. Generally low cost as it primarily involves accessing existing weather data, potentially with some investment in weather stations or data subscriptions.

2. Mulching

- Value Effectiveness:
  1. Reduces water evaporation from soil, maintaining moisture levels.
  2. Suppresses weed growth, which can otherwise compete with crops for water.
- Cost Estimation:
  1. Low cost because it often involves using organic materials like leaves or straw, which may be locally sourced or even waste products.

3. Soil Moisture Sensors Installation

- Value Effectiveness:
  1. Provides precise measurements of soil moisture, allowing for targeted irrigation.
  2. Reduces water waste by ensuring irrigation only occurs when necessary.
- Cost Estimation:
  1. Moderate cost due to the expense of purchasing and installing the sensors, as well as potential maintenance.

4. Regular Maintenance and Inspection

- Value Effectiveness:
  1. Ensures that the irrigation system operates efficiently, reducing water loss due to leaks.
  2. Prevents breakdowns that could lead to overwatering or underwatering.
- Cost Estimation:
  1. Low to moderate cost, involving labor and potential replacement parts.

5. Proper Plant Selection

- Value Effectiveness:
  1. Choosing drought-resistant plants reduces overall water needs.
  2. Selecting plants suited to local climate and soil conditions improves growth and water efficiency.
- Cost Estimation:
  1. Moderate cost, mainly involving the purchase of specific plant varieties.

6. Drip Irrigation System Installation

- Value Effectiveness:
  1. Delivers water directly to the plant roots, minimizing evaporation and runoff.
  2. Provides a consistent moisture level, which can enhance plant health and yield.
- Cost Estimation:
  1. High cost due to the complexity of installing the system and potential for high initial investment in materials.

7. Rainwater Harvesting Implementation

- Value Effectiveness:
  1. Captures and stores rainwater for later use, reducing dependency on external water sources.
  2. Provides a sustainable and environmentally friendly water source.
- Cost Estimation:
  1. Moderate cost, involving the setup of collection systems, storage tanks, and potential filtration systems.

8. Training and Education for Farmers

- Value Effectiveness:
  1. Empowers farmers with knowledge to implement efficient irrigation practices.
  2. Enhances overall farm management, leading to better resource utilization and crop yields.
- Cost Estimation:
  1. Moderate cost, including training materials, workshops, and possibly hiring experts.

9. Smart Irrigation Controllers

- Value Effectiveness:
  1. Automates irrigation based on real-time data, optimizing water use.
  2. Reduces labor and human error in managing irrigation schedules.
- Cost Estimation:
  1. High cost due to the technology involved and potential ongoing maintenance or software updates.

10. Water Recycling and Reuse

- Value Effectiveness:
  1. Reduces water consumption by reusing treated wastewater for irrigation.
  2. Provides a reliable water source, especially in areas with limited water availability.
- Cost Estimation:
  1. High cost due to the infrastructure needed for treatment and distribution systems.

12.11 Decomposition Methods

The basic need and basic idea of an Evo-step decomposition

- When a strategy is bigger than your Evo-step length (for example, a week step).
- You need to decompose strategies into **sub-strategies**, no larger than your Evo-step length.
- The *main* decomposition rule is that Evo-steps must deliver '*planned value*-requirement increases', not be mere 'tasks'.
- A *secondary* rule, though not so critical, is that the sub-strategies can be *delivered in any sequence, independently of each other*. Useful, for optimization, but not ctitical.
- AI apps can do this decomposition if you ask (as in 12.10.4.4)
- Value Evo-steps can more-easily be found, if you know the domain well. *Someone* knows what is practical, in real life.

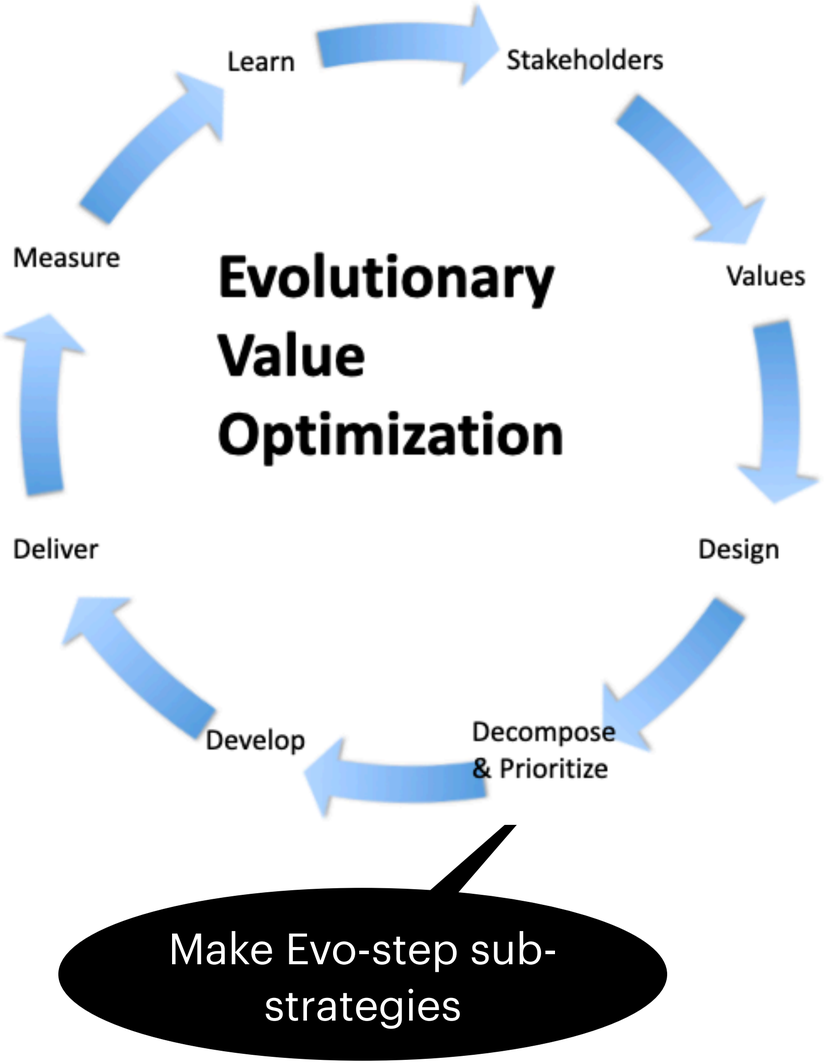


Decompose and then Prioritize

12.11 Decomposition Methods

Various Decomposition Tactics

- PRIORITY VALUE LEVELS: do steps necessary for reaching scalar constraint levels, Tolerable, before working to reach Goal levels.
- DEADLINE: do steps necessary for reaching earlier deadlines, first.
- [CONDITIONS], SCALE QUALIFIERS
  - WHERE: do steps for some locations before others.
  - WHO: do steps for particular stakeholders before others.
  - HOW: do steps for defined 'HOW' conditions (example, application, frequency, skilled people, environmental conditions)
  - WHY: do steps for defined use purposes, for example emergency, annual, children, automotive, vacations.
- BASICS: do basic elements of strategies, before adding advanced features.
- Do *not* get 'hung up' by your 'fears about *sub-optimization*', of not seeing and doing the 'whole picture'.
  - The advantage of *small steps, value streams, and feedback*, is far greater than any such problem, generally, we believe.
- Decades of experience has shown that you can *always* decompose, if you have imagination and competence. See 'backroom'. Decompose into Deliver step each week.
  - And that you can always decompose large strategies, into arbitrarily small ones (like a week or a day).
  - This of course is a prerequisite for agility of learning and choosing better decisions, and strategies.



12.12.4.3 AI Decomposition of a prioritized most-effective Strategy

So that we get things that can deliver results soon and frequently: ‘Evo’

Based on the search results and the given query, I'll decompose System Modernization into 10 effective, independently implementable sub-strategies and provide a brief description for each. Then, I'll construct an Impact Evaluation Table (IET) using these sub-strategies and the provided values.

1. API Integration: Implement APIs to connect legacy systems with modern applications, enabling better data flow and interoperability. This approach allows for gradual modernization without completely replacing existing systems[1].

2. Cloud Migration (Rehosting): Move existing applications to cloud infrastructure without significant changes to the code, often called "lift and shift." This strategy provides quick wins in terms of scalability and cost savings[2].

3. Refactoring: Optimize existing code for better performance in the cloud environment without changing its external behavior. This improves application efficiency and maintainability while preserving core functionality[2][3].

4. Microservices Adoption: Break down monolithic applications into smaller, independently deployable services. This architecture enhances scalability, flexibility, and ease of maintenance[3].

5. Container Adoption: Implement containerization technologies like Docker to package applications and their dependencies, improving portability and consistency across different environments[2].

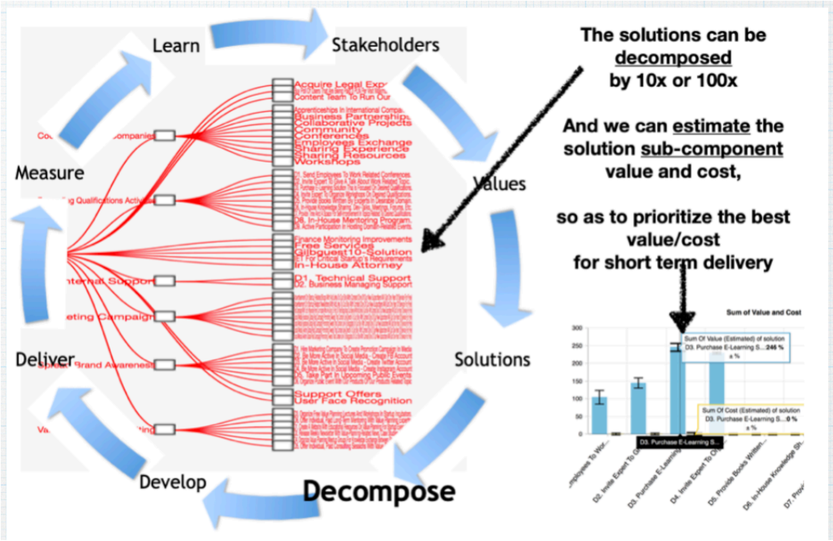
6. Serverless Computing: Transition suitable components to serverless architectures, reducing infrastructure management overhead and improving scalability[3].

7. Data Modernization: Upgrade databases and data storage systems to modern, cloud-native solutions, enhancing data accessibility, security, and analytics capabilities[4].

8. DevOps Implementation: Adopt DevOps practices and tools to streamline development, testing, and deployment processes, improving collaboration and reducing time-to-market[3].

9. Legacy Application Replacement: Identify and replace outdated applications with modern, off-the-shelf solutions or custom-built alternatives that better meet current business needs[4][5].

10. Continuous Integration/Continuous Deployment (CI/CD): Implement automated CI/CD pipelines to facilitate frequent, reliable software updates and reduce manual intervention in the deployment process[3].



Citations:
[1] https://www.linkedin.com/pulse/6-key-strategies-legacy-systems-modernization-koukiali-mba
[2] https://stratusgrid.com/blog/rs-of-application-modernization
[3] https://www.redhat.com/en/blog/modenization-strategies-modernizing-existing-code
[4] https://www.trianz.com/insights/application-modernization-strategies-7-rs-of-transformation
[5] https://guidehouse.com/insights/advanced-solutions/2024/setting-a-strategy-for-successful-technology-modernization

12.15 Second Level Decomposition

Source [KEN]

https://tinyurl.com/KENGilb

3.2 Second-Level Decomposition ('Specification Object' Decomposition)

One specification object (in this case a Value spec) can have many 'Statements' which give another opportunity for decomposition. In this case the Tolerable, Goal and Wish statements all refer to different points in time. So we can naturally exploit this ready-made decomposition, and plan strategies for one of these 3 at a time, probably the earliest first. The 'Tolerable' level.

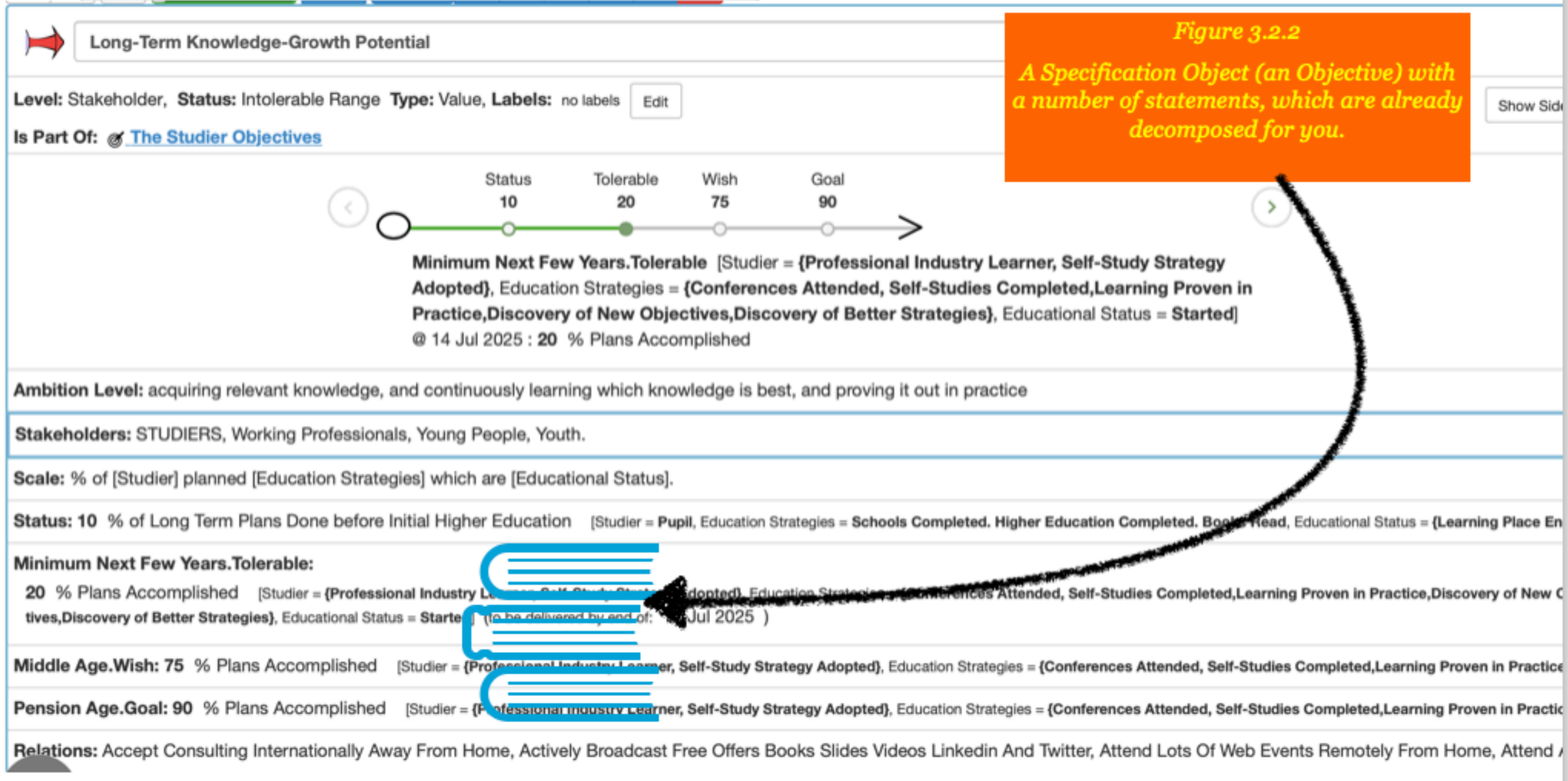


Figure 3.2.2

A Specification Object (an Objective) with a number of statements, which are already decomposed for you.

12.13 Third Level Decomposition

Source [KEN]

https://tinyurl.com/KENGilb

3.2 Third-Level Decomposition ('Statements into Parameters Decomposition': 'Slices of the system'

This Third level of decomposition allows you to select slices of a much larger complex long-term reality, for short term action, like 'next week', and done by 'the end of the week'. This is very practical for getting quick results for critical slices of your reality. It also sets you up for an agile process of incrementing results and learning what works or not.

Stakeholders: STUDIERS, Working Professionals, Young People, Youth.

Scale:

% of **[Studier]** planned **[Education Strategies]** which are **[Educational Status]**.

Target Time Units: Calendar Date

Education Strategies: defined as:

Schools Completed. Higher Education Completed. Books Read, Conferences Attended, Courses Completed, Self-Studies Completed, Learning Proven in Practice, Discovery of New Objectives, Discovery of Better Strategies,

Educational Status: defined as:

Completed, Started, Dropped as Irrelevant, Exams Passed, Thesis Accepted, Paper Accepted, Talk Accepted, Learning Place Entrance Granted, Educational Funding Secured, Practice Situation Secured and Started,

Studier: defined as:

Pupil, Undergraduate, Graduate, Professional Industry Learner, Self-Study Strategy Adopted,

Source:

tg quick draft

Figure 3.2.3

The Scale statement can usually be decomposed into a number of [Scale Parameters]; different aspects or dimensions of a system: who, when, what.

The definition of each [Scale Parameter] isa further decomposition, into Conditions, which can be used to decompose into 'slices of the system'.

Figure 3.2.4. Selecting a particularly critical or valuable slice of a system, and setting a Goal for it.

The 'Pension Age' Goal for a level of 90% by year 2065,

applies to

The conditions specified in the 3 boxes below.

Pension Age.Goal:

90 | 14/07/2065 (dd/mm/yyyy) | % Plans Accomplished

Qualifiers: Copy from... + Add additional qualifier

[Studier] =

× Professional Industry Learner × Self-Study Strategy Adopted

[Education Strategies] =

× Conferences Attended × Self-Studies Completed × Learning Proven in Practice

× Discovery of New Objectives × Discovery of Better Strategies

[Educational Status] =

× Started

Source: by tomgilb - Jul 14th 2020, 22:56

12.16 Decomposition Principles for Knowledge

Figure 3.2.5 Stakeholder hierarchy. Knowledge `consumers.



KEN 3.2 Decomposition Principles: for education and knowledge.

1. **System Specification Decomposition can be done at many levels, for many different purposes.**

2. **Decomposition is a basic tactic, to describe a complex system, so as to allow you to focus on *critical* things *early*, to separate *cause and effect* of your actions better, to allow *incremental progress* towards long-term larger changes, and to *limit damage* from bad decisions, to small reversible losses.**

3. **Most everything can be usefully decomposed when planning for education and knowledge. One central idea is to get to a detailed level, which is simple to *act* on, and to *understand*.**

4. **There are many different methods of decomposition, some more useful that others. My favorite decomposition rules are that the decompositions are 1. *Independent* of the others (can be delivered in any sequence), and 2. That when they are delivered, they will give a *measurable result*, in the direction of our value objectives.**

5. **You can decompose almost any idea, Objectives, Time Sequences, Budgets, Strategies, Stakeholders, Actions, Constraints, Functions.**

6. **You really do not have to worry much about sub-optimalizing, of 'painting yourself in a corner', with early small steps. But there are methods for avoiding such problems, such as having an *open-ended architecture*: an environment where *change* is easy. For example, by *not* making irrevocable large payment or contracts.**

7. **You do not have to decompose the *entire* system you are dealing with, up front. You can decompose it *gradually*, and focus on *critical* things you want to *prioritize* early.**

8. **Ready-Made decomposition *templates* can be useful guides for many kinds of decomposition, for example Stakeholder hierarchies for an Educational Domain.**

9. **It is possible to *store decomposition templates in apps* and use them to start and then *tailor* to your current circumstances (example ValPlan.net)**

10. **Decomposition of *ideas*, designs and specifications is a precursor to *physical* real world *decomposition*, *prioritization* and *sequencing*.**

Source [KEN]

12.17 The Backroom and Frontroom

Here is one of the 'tricks' of decomposition: when it *seems* impossible.

Frequently, we encounter a strategy or architecture that just refuses to divide up into our desired smaller Evo-steps of a week or so. For example a supply chain delay.

I do like to say that 'everything can be decomposed, into Evo Steps'. But a birth cycle does take about 9 months, even if you try to use 9 women in parallel.

The solution is to accept defeat, gracefully. With a wry smile. Some things cannot *really* be decomposed into weekly incremental delivery steps.

But what we *can* do is to start the **Develop** stage (Evo Cycle), and the, 'wait' until it is *complete enough* to 'Deliver'.

We put the Developing Strategy in the 'Backroom' until it is ready to **Deliver**.

And we find *something else* which is actually ready to deliver, that week. Maybe even something, a 'ready to Deliver step' strategy, that has been hibernating in the Backroom until now.

So, in order to more boldly make a claim to decomposition powers, it needs 'clarification'.

You can generally expect to find frequent Evo-step Deliveries, for every 'week' or cycle you choose, but a few of these Delivery Steps will need to reside in your Backroom, until they are ready to bloom.

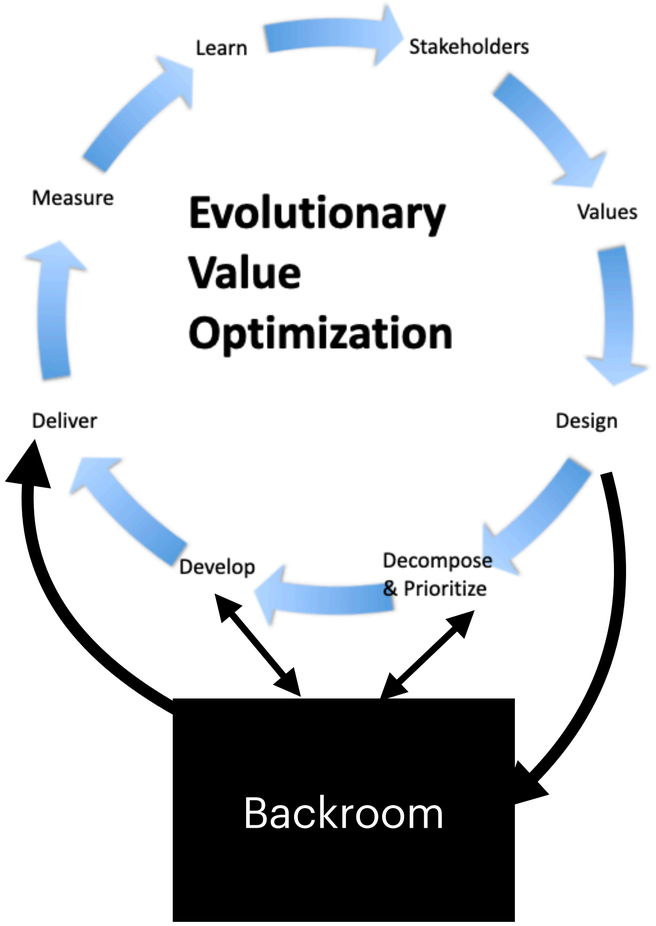
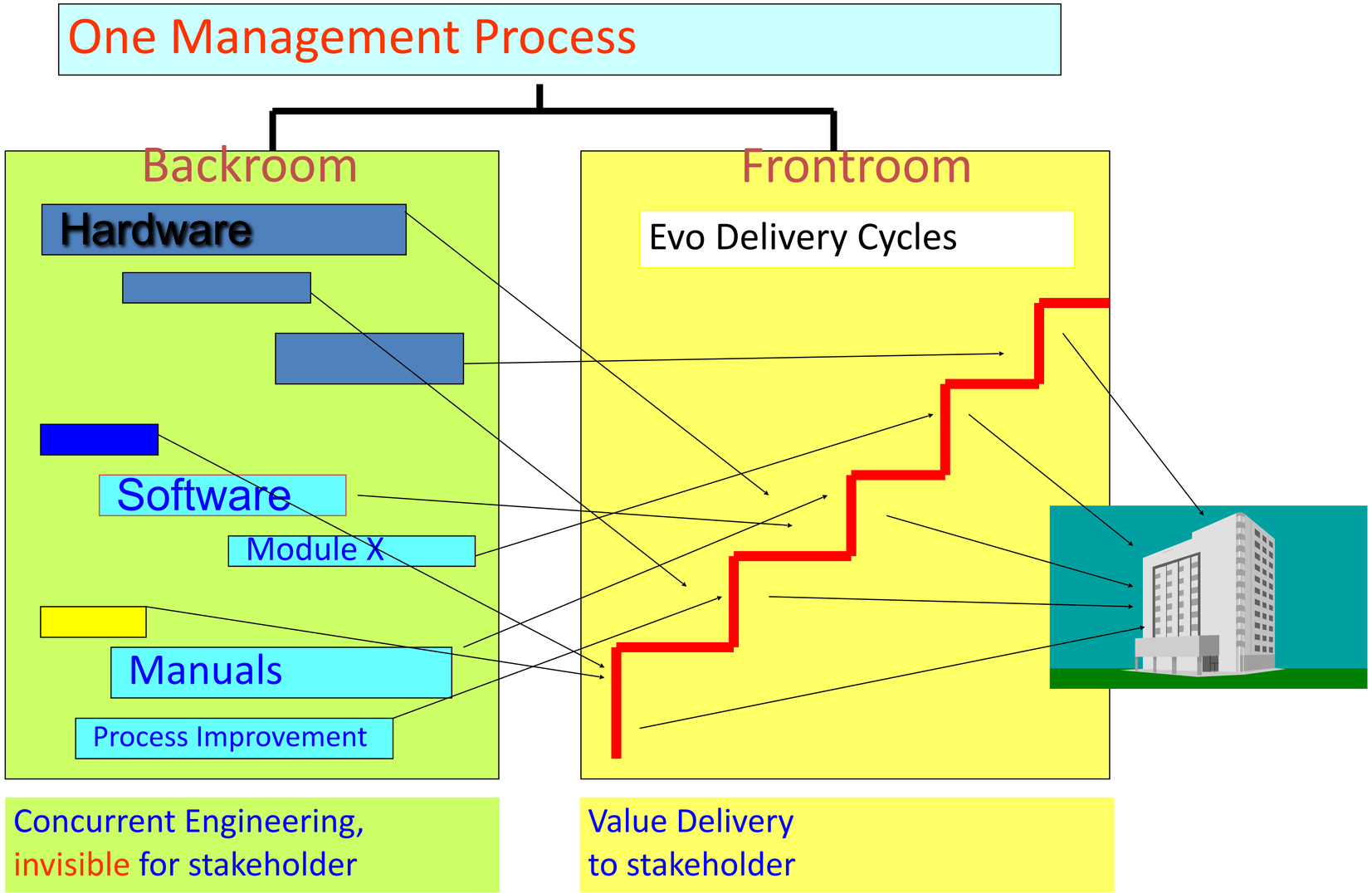


Figure 12.17 When a strategy is placed in the backroom, to 'mature', it is in a Develop phase, but could be just 'waiting' for an event. It might be decomposed and the sub-strategies might be prioritized. But not delivered. When it is ready for Delivery, it still might've to wait some cycles to be prioritized for delivery (usually on Value/Cost basis)

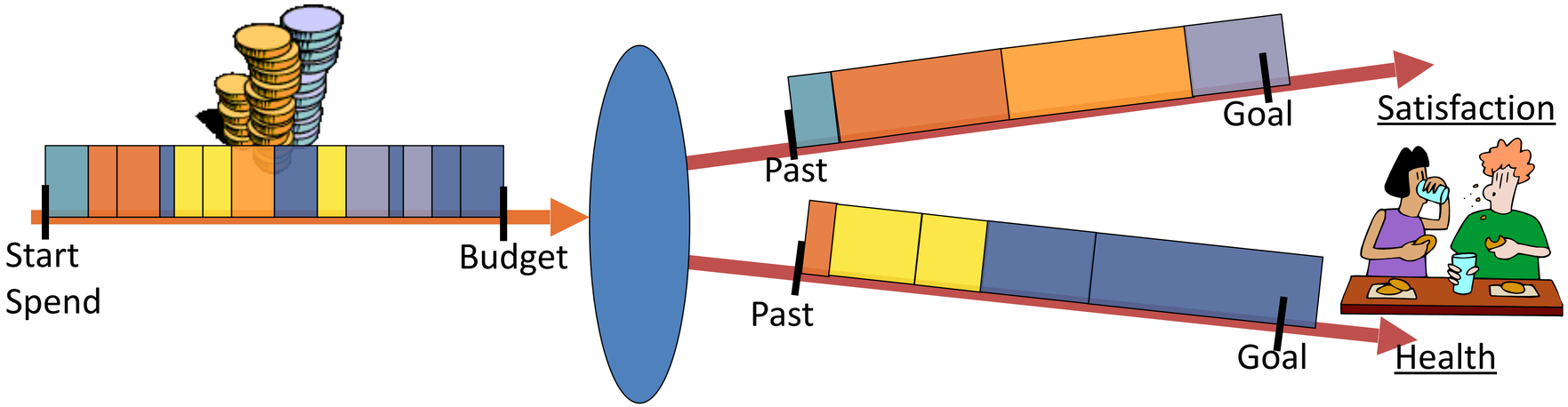
12.18 EVO Management

Backroom and Frontroom: Stakeholder Visibility

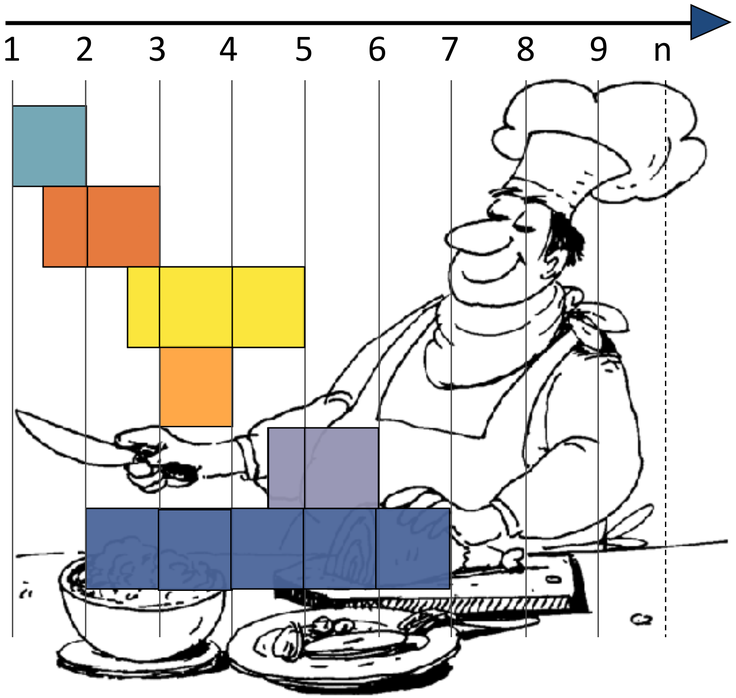


COURTESY SIMON PORRO WWW.SPIPARTNERS.NL 22 MAY 2001

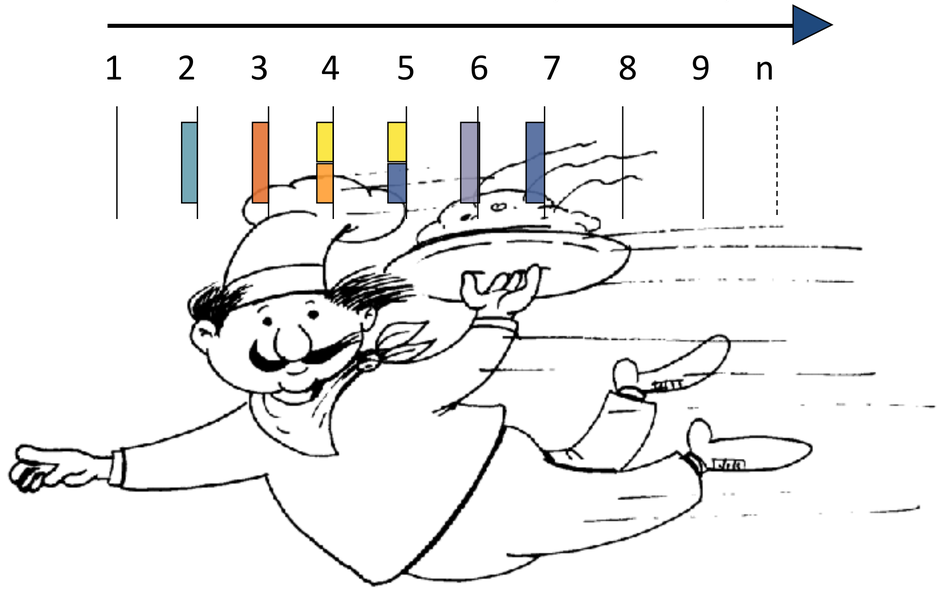
12.19 Backroom/Frontroom (Courtesy Kai Gilb)



Back-room Design Development

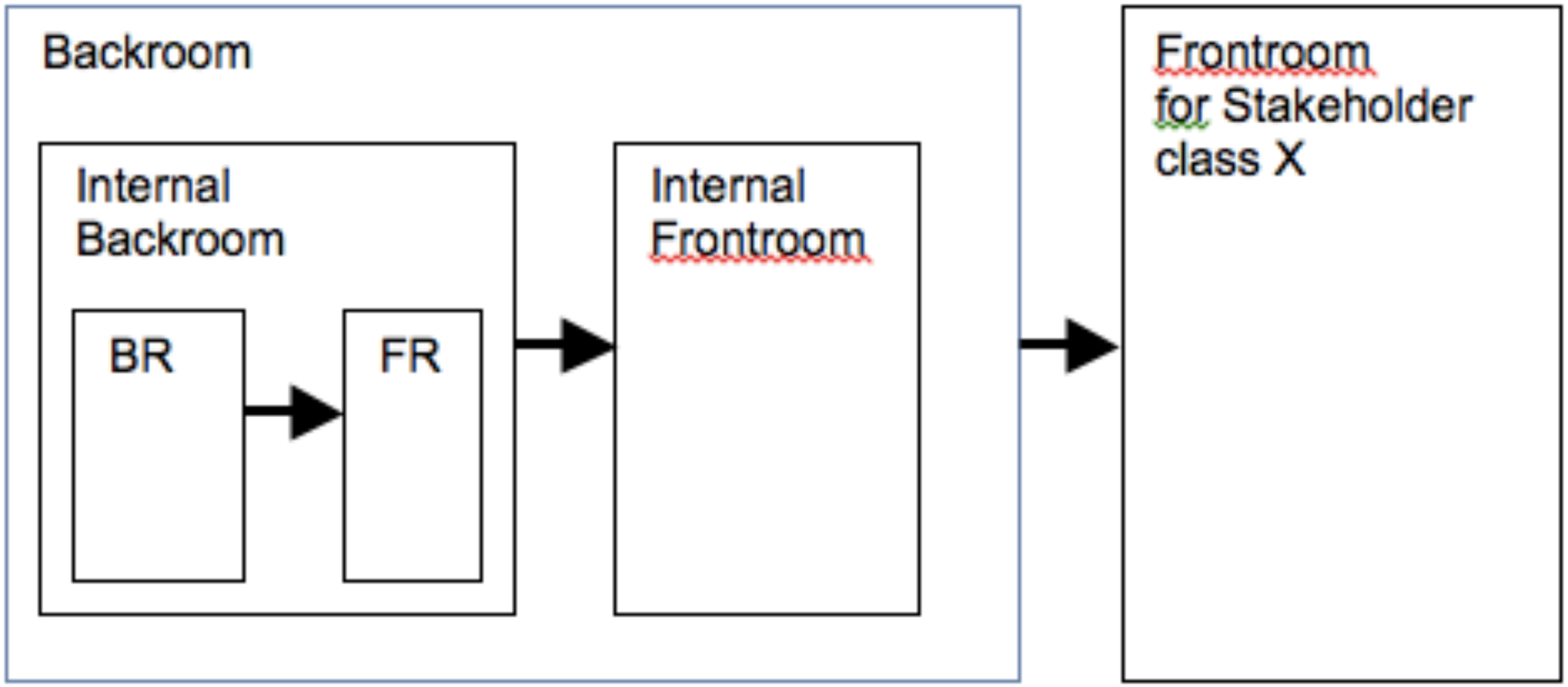


Front-room Evolutionary Delivery



12.20. Nested Backroom Model

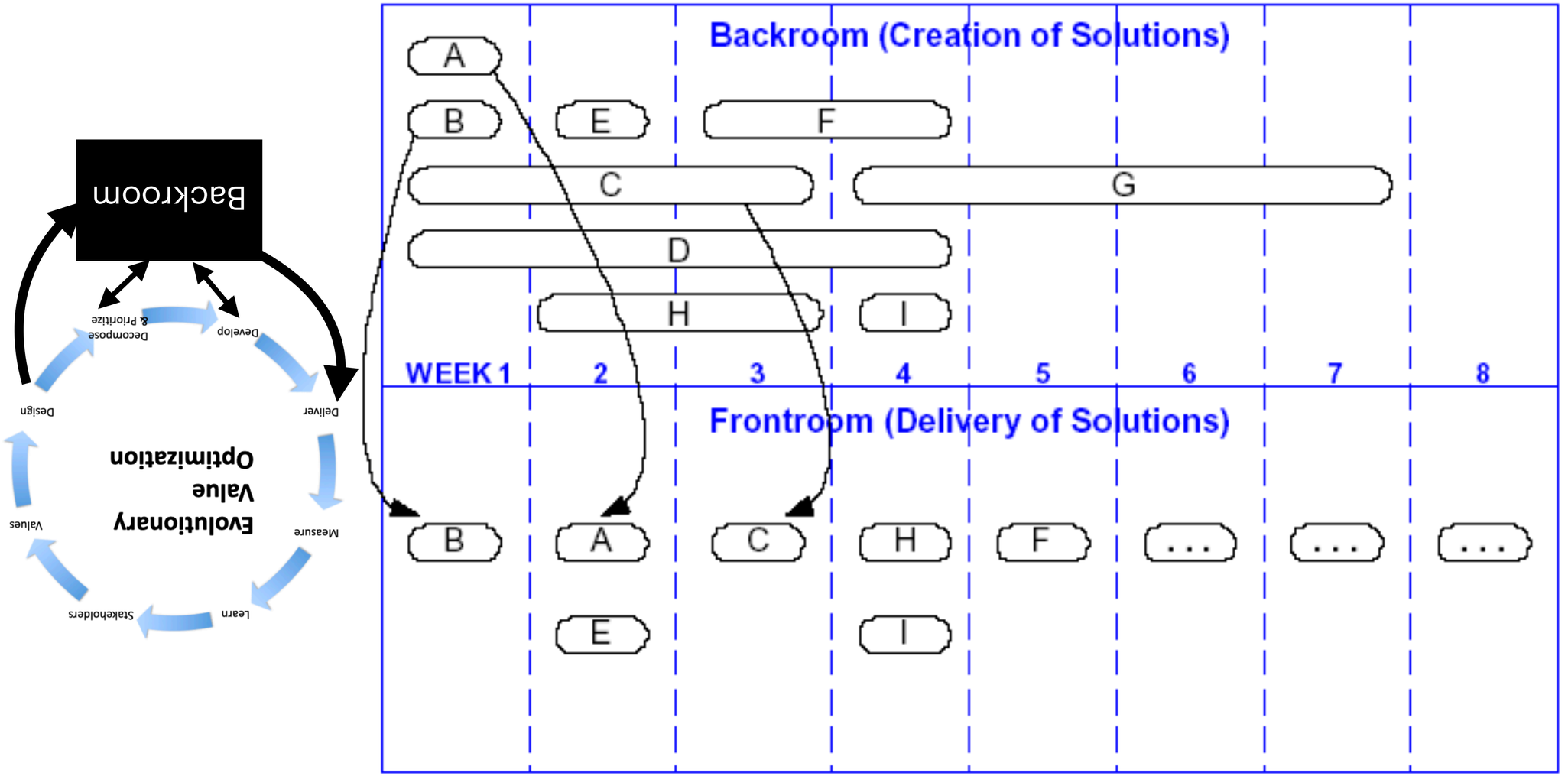
Simon Porro, 2001 Philips NL
Sometimes the Backroom has it's own Frontroom
Philips discovered this in its use of the Backroom method.



See Notes or more detail from Porro.

Note add case Philips production timing

12.21. Backroom - Frontroom (Craig Larman 2003)



Another way of visualizing this is to think of the Backroom as a **Develop** stage, and the Frontroom as a **Deliver** stage. TsG 2024

12.22 The size of an Evo-step

How big should an Evo-step be?

I have long found that a weekly delivery cycle works well. Like since 1960, for me. That weekend buffer, works wonders!

The most important thing is that the step size is relatively small. For example 1% to 5% of time to big deadline (project deadline)

At one extreme we could observe that in Lean Startup (Ries, Blank) with DevOps, they could be A/B testing about 60 different parallel changes. These are more-likely small functional or interface changes, not really blue strategies, decomposed. But, that is a lot.

I find Tesla's practices [MM] with a 10 hour cycle of Mobbing, where they literally aim for a change to the production car that day, to be an amazing industrial feat. Enabled by the car architecture (highly modular, half software) and *'reporting changes to authorities methods same day'.*

What matters in your determination of step size is, you real technical, cultural, and governance environment. We just want to get something done quick. Get a series of increments done (a value stream) [Productivity Stream]. Reduce risk of large irrecoverable failures. Force people to 'get real' about the complexities o their environment. Learn easy, learn quickly. Adjust quickly. Be able to adjust to changing eternal stakeholders and values, and constraints. Force architects to design a highly adaptable architecture, both modularity, and in contracts that do not lock you in! We also want to be able to sequence, prioritize (next chapter) delivery steps that have a large value to cost ratio, to guarantee fast profitability,, and project success visibility (motivation9.

Now, Is that asking too much?

12.23 Some Evo Cases

[CE] page 223

"The evolutionary development methodology has become a significant asset for Hewlett-Packard software developers. Its most salient, consistent benefits have been the ability to get early, accurate, well-formed feedback from users and the ability to respond to that feedback."

Elaine May and Barbara Zimmer, Hewlett-Packard (May and Zimmer 1996, Page 44). [HP]

Microsoft has also been documented as using Evo extensively (MacCormack 2001; Cusumano and Selby 1995). The Open Source Methods (like Linux) (Maier and Rechtin 2002) and Agile Software Development methods (Cockburn 2002; Abrahamsson etal. 2002) have also clearly demonstrated the power of Evo in delivering good software rapidly.

"This book makes it clear that the evolutionary method is a dominant method at several levels of technology and marketing at Microsoft. Most dramatic is the daily 5 PM software build cycle, but it extends to much larger cycles too." Cusumano and Selby 1995

The use of Evo is also proven within engineering processes.

For example in 1988, this author consulted with over 25 projects (about 120 aircraft design engineers) at Douglas Aircraft. Of course, new aircraft did not fly the next week (most of the projects were modifications and upgrades, or integrating new components). But, real results capable of giving useful feedback, were delivered to real stakeholders, in weekly increments. Management approved each Evo step in advance. They found the method was practical, low risk and they could not resist seeing results fast.

12.24. Philips Evo Pilot May 2001

Frank's director said this 'Evo Idea' was impossible, but Frank persisted. A project stuck in Limbo for months Was suddenly a Success.

# Jobs	Week	[- 5%,+10%]	[-10%,+20%]	[-15%,+30%]	out of range
6	wk 8	1	5		
11	wk 9	3	1	7	
19	wk 10	6	3	7	3
25	wk 11	6	4	6	9
25	wk 12	17	3	5	
42	wk 13	31	3	2	6
55	wk 14	37	11	1	6
55	wk 15	39	9	1	6
55	wk 16	48	4	1	2
55	wk 17	50	4		1

Frank van Latum,
The Manager

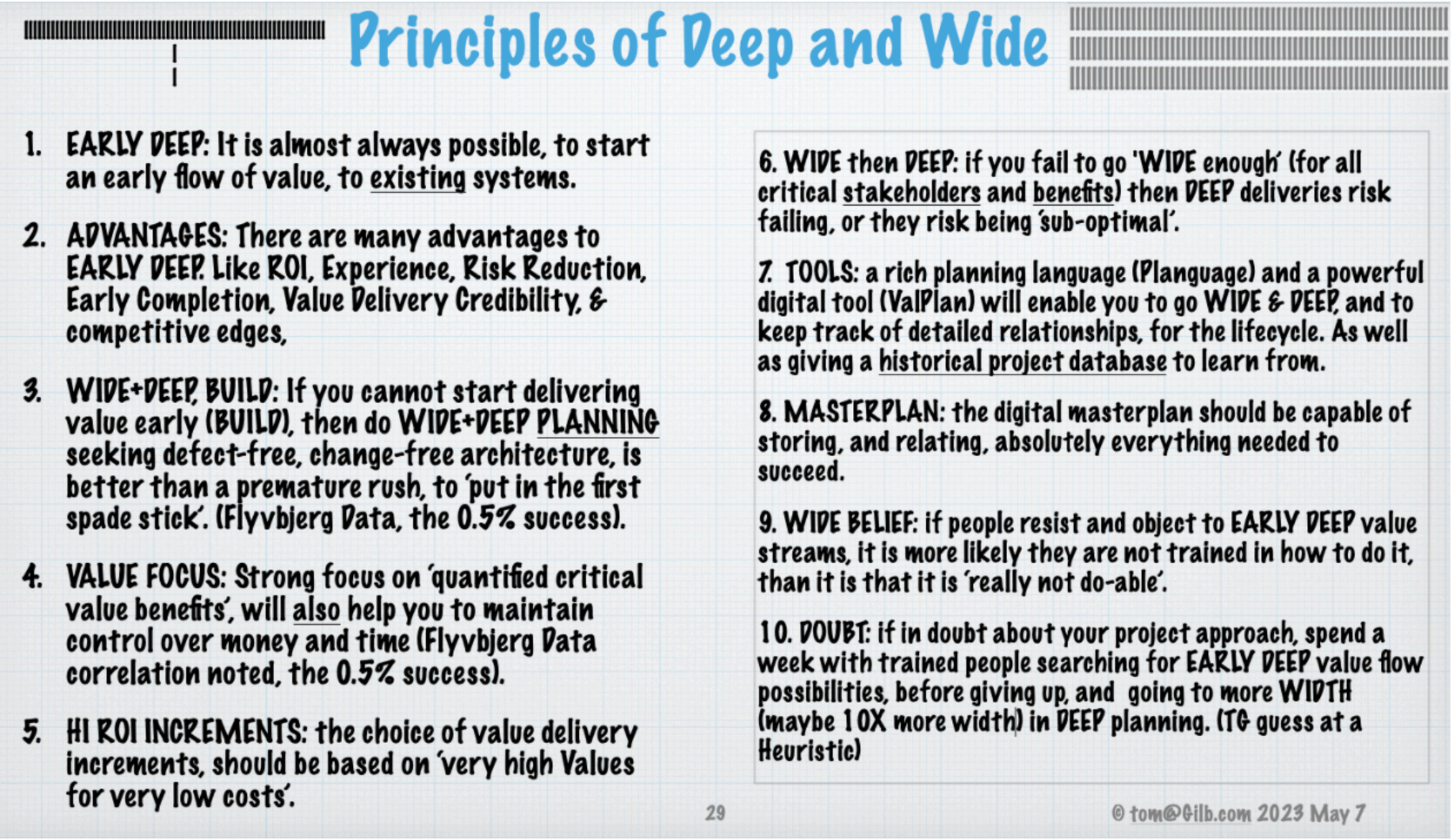


The GxxLine PXX Optimizer EVO team proudly presents the success of the Timing Prediction Improvement EVO steps.
Shown are the results of the test set used to monitor the improvement process.
The size of the test set has grown, as can be seen in the first column. (In the second column the week number is shown.)
We measured the quality of the timing prediction in percentages, in which –5% means that the prediction by the optimizer is 5% too optimistic.
Excellent quality (–5% to +10%) is given the color green, very good quality quality is yellow, good quality is orange, & the rest is red.
The results are for the ToXXXz X(i) and EXXX X(i), and are accomplished by thorough analysis of the machines, and appropriate adaptation of the software.

The GXXline Optimiser Team presented the word document below to the Business Creation Process review team.
The results were received with great applause. The graphics are based on the timing accuracy scale of measure that was defined with Jan verbakel. Classification: Unclassified

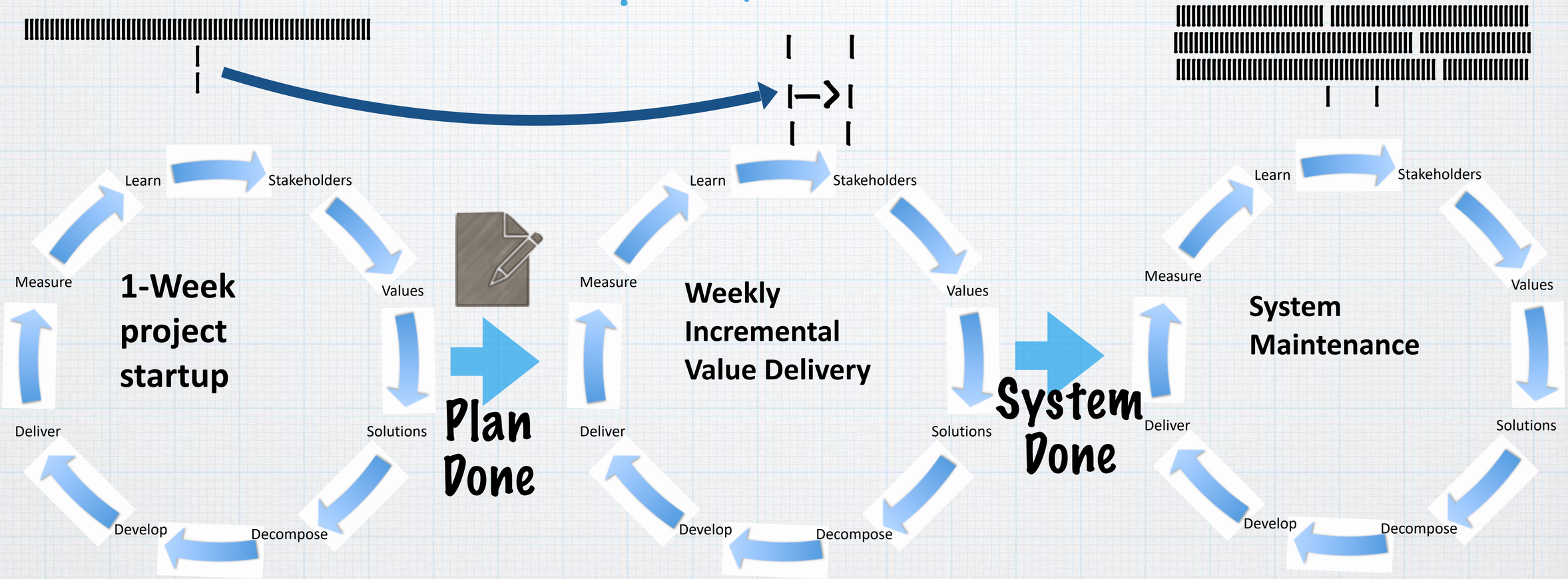
12.25 Principles of Deep and Wide Decomposition

See illustration 12.26



Different planning environments will determine whether you will plan deeply before delivering an Evo-step: or jump into it and get feedback which might be better, faster, more accurate than theoretical planning.

12.26 'Evo' Development Process
Deep Start, Incremental Build



Wide View
(system-wide, top-10 stuff)
With selection of a deep slice
Of action.
Monday-Wednesday (Wide)
Thursday (Deep slice)

Many '2%' cycles
of a slice of 'value delivery'
until
'Requirements Met'

Keeping the system in
Good shape

12.27 Deep Start, or Wide Start? When should we choose which option?

Deep Start Evo

(Depth plan for a series of quick delivery steps)

* Existing Systems (IT, organizations, Education)
* RESULT COSTS: Useful incremental change costs CAN BE relatively small (1% of a capital budget, or within existing operational budgets)
* MOTIVATION: rapid measurable change, stream of value improvements, control over risks of failure (financial overrun, benefits failure)
* ENVIRONMENT: target system exists now, and needs improvement

Wide Start Evo

(Planning everything in depth)

* New Physical Systems (Buildings, Dams)
* RESULT COSTS: High Capital Build Cost (Entertainment, Movies), Example Billions
* MOTIVATION: low build costs, final new system on time and under budget. Ability to deal with very complex system, in advance planning and NOT by fail and retry while building.
* ENVIRONMENT: target system does not exist, and most all detail needs to be constructed from scratch.



12.28 Gilb's Design Decomposition Principles

Inspired by CE book -2005, p. 314, and much-improved text detail (2019), Source [Value Design] slides

Based on many decades of decomposition practice.

1. **PERSIST:** do not give up in 5 minutes, because you cannot think of useful decompositions.
   1. Assume there are many possibilities
   2. Think simple and small
   3. Don't eat drink or rest until you have solved the problem
   4. Ask some friends for good ideas, challenge them
   5. Ask what is the simplest thing we could do next week to improve that value level?
2. **RULES:** use these primary rules A. The decomposed design increment MUST deliver measurable planned value. B. The decomposed design increment must be implementable independently of all other potential increments (so that most value-effective increments can be done early, and not wait for others)
   1. These rules apply to all the top ten designs
   2. This work can be done in parallel
   3. First teams to get small increments can immediately put ideas into action
   4. If necessary, decompose more than once, to get to small 2% increments
   5. When you have several options, do the best value-to-cost options, first
3. **OBSTACLES:** it is OK to name and list potential obstacles and problems.
   1. But NOT OK to use problems as an excuse for inaction, for not finding good solutions.
   2. *Use* these problem lists, and challenge the creative powers amongst you, to find solutions, to get rid of the obstacle 'by smart design'.
   3. Really good designers, can design their way past almost any obstacle
4. **111111 DECOMPOSITION:** be quite satisfied with *some* real progress in value delivery, rather than *none* (procrastination and excuses). Maybe just '1 function, 1 stakeholder, 1 value, 1 design, 1 week delivery cycle, 1% is better than 0%'.
5. **MOTIVATION NUDGES:** Do not divert your focus to the design ideas themselves, or the technical architecture. *re-focus on the values*.
   1. Sometimes you only need a small nudge, like a motivation, or a reward, to move the value to a better level.
   2. Nudges are simple and cheap, and you can experiment more easily, to find what works.
6. **TEMPORARY SCAFFOLDING:** Don't get stuck in the *new* system design-increments
   1. Sometimes it is worth doing *temporary scaffolding* to make progress.
   2. For example use old systems, temporary agreements, *borrowed* systems or services.
7. **ELEGANCE IS NOT THE POINT:** Do not worry about elegance of design, it is only the value and cost results, that really matter.
8. **THEY WANT RESULTS NOT DESIGN:** Do not be afraid that stakeholders will not like your design itself:
   1. If it delivers the *value* to them, they will like it.
   2. If no value, then even 'elegance' alone will not be of interest.

A. **DON'T WORRY GIVE VALUE:** Do not procrastinate by worrying about 'what *might* happen after we release this increment'.
   A. You are not in full control of all potential future problems, anyway.
   B. Just focus on delivering value.
   C. Problems can be fixed, by design, as they occur.
   D. There may be a thousand things that *might* go wrong, but only a few will really happen, so it pays off to wait, to sense them early, and deal with them quickly.

B. **DON'T GET *ANALYSIS PARALYSIS*;** You cannot foresee everything that will happen when you release a design increment: don't try.
   A. But make sure you can sense negative feedback quickly (design it)
   B. And make sure you can reverse the change as quickly, to limit damage (design it)
   C. Declare the negative feedback to be a scientific success in learning what NOT to do on a larger scale.
   D. Thank the participants profusely for their contribution, and ASK THEM what would be better!
   E. Tell them it is called 'Participatory Design Feedback Experiments'. And they are a vital part of the scientific process of making their own systems better. Be humble. They have really useful insights.

C. **REAL VALUE BREEDS COOPERATION:** If you really constantly focus on delivering value to your stakeholders, then they will be quite forgiving of an occasional slip up.
   A. They are not used to getting so much, real improvement, so fast.
   B. Make sure you seriously ask them, when you slip up, what you *should* have done to improve *their* value
   C. Make a big point of doing *something* quite soon, to show you listened, and *can* deliver
   D. That will teach the stakeholders that you are worth giving advice about value to.

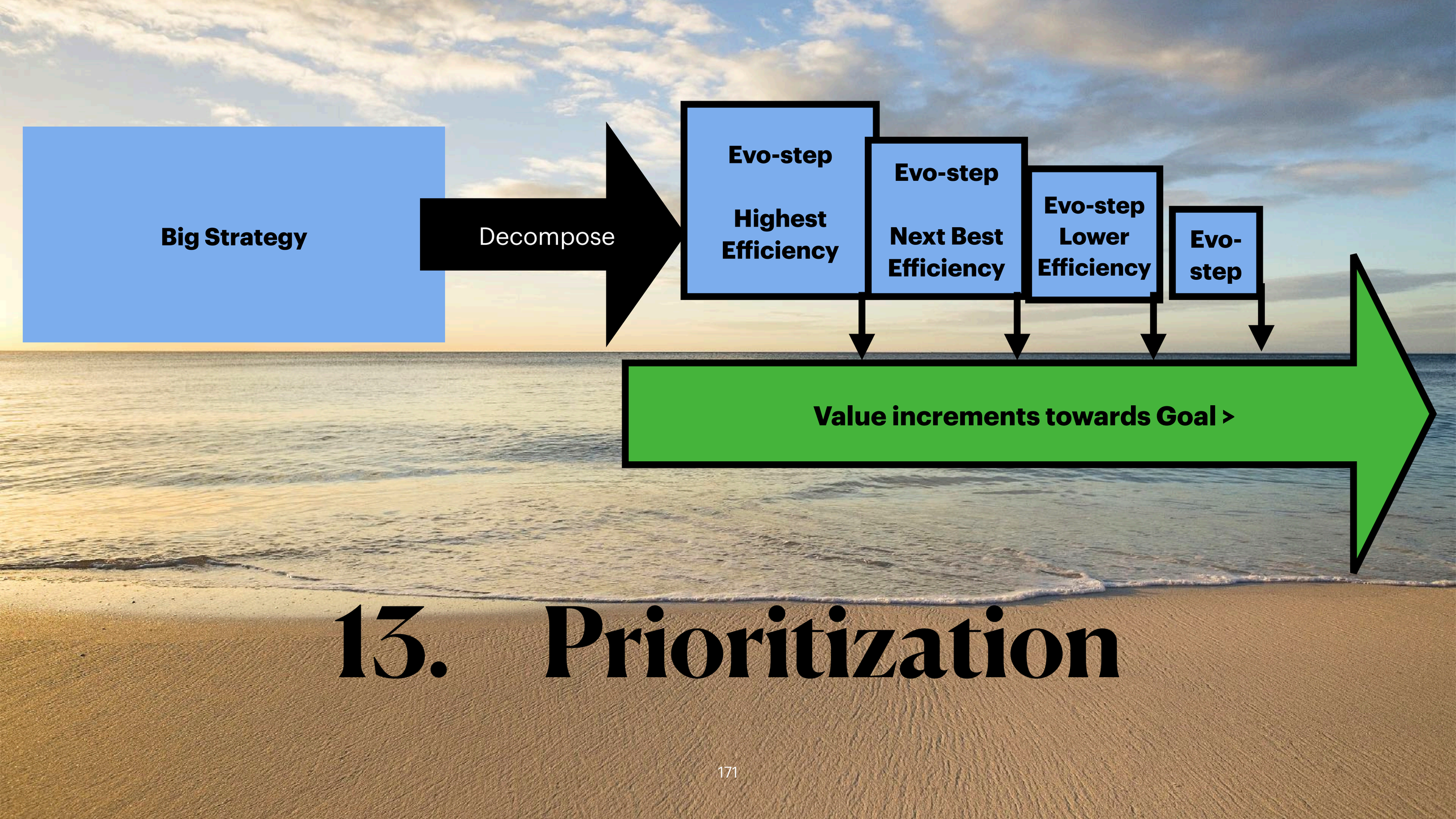
D. **ACTION GIVES FACTS:** If you just dive in, and do real stuff, you will more quickly learn what gives value, than if you discuss and plan in an office.
   A. No 'modelling' beats the real thing, for trustworthy and useful insights in several dimensions
   B. Arguments based on observed real facts and experiences are more convincing

E. **HARNESS THE POSITIVE:** Do early steps of a new design, before scaling up, with *positive, mature and supportive* stakeholders.
   A. That will help make things work better, after you scale up.
   B. Keep *local* if you can, with insiders and friends.
   C. Let early adopters know that you really need their help and advice, to make early increments better, before you scale up.
   D. Credit their ideas in writing. In plans, and in 'the real thing'

F. **THE TROOPS KNOW WHERE THE BOOT PINCHES:** If you have problems thinking of some valuable early small delivery steps, then maybe it is time to observe directly, and talk directly to your stakeholders.
   A. Ask them for ideas. Their design ideas.
   B. Allocate time and energy to make some of those ideas come true; build *your* credibility
   C. Build in (design in) simple feedback devices where stakeholders can at least display discomfort, and maybe suggest improvements in 'our' design
   D. This applies to get started early, but is also a continuous long-term design process

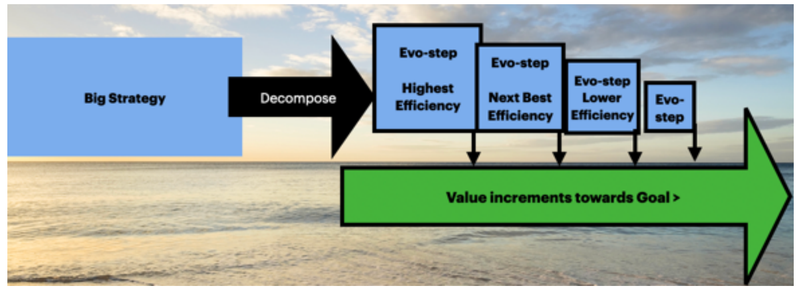
G. **DON'T REINVENT YOUR WHEEL:** Do not be afraid to make use of the 'awful old system' initially, to get some real improvements: you can move the proven successful design ideas to the 'new system' when new components are incrementally in place.

H. **OPEN-ENDED ARCHITECTURE:** your basic architecture skeleton needs to be 'open ended'.
   A. That means you need to plan specific architecture ideas which make it easy to change designs, when early design don't work well, into design which work better.
      1. There are many devices such as plug-and-play modularization, centralized architecture control , self-programming AI, and flexible contracts (not locked in, not over-specified in advance, pay-as-you-deliver?Value)



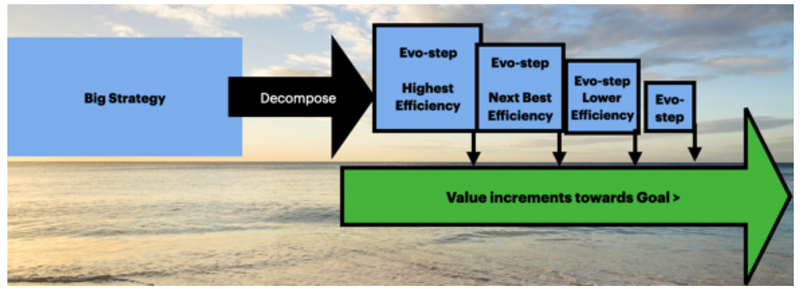
13. Prioritization

13.1 Prioritization Purposes



- ***Prioritization***, meaning ‘putting decomposed sub-strategy Evo-steps , into a better than random sequence’, is *potentially* useful; but *not* generally a ‘critical’ necessity
  - If in doubt, do *something*, ***anything***, any sub-strategy, and any Evo-step. Get *results*, get **value**, get feedback, get *payback*, get ***credibility***.
  - But, you can improve values, costs, efficiency, and stakeholder appreciation, by more-carefully selecting, sub-strategies, with such ‘optimization’ purposes in mind.

13.2 Prioritization Methods



- **PRIORITIZATION POLICY:**
  - **Write down a formal Evo Prioritization policy. 13.5.6 Prioritization Policies**
  - **Generally, it should be 'Efficiency First': highest Value/Costs first.**
  - **Use an Impact Estimation Table to bring out the numbers, indicating your priorities (vales, costs, risks).**
  - **This could include the advanced notion of considering the risks, of Uncertainty, and Credibility [VIET]**
- **SPECIFIC POLICY EXCEPTIONS:**
  - **There are often occasions when the general long-term policy is not the smartest ting to do. For example:**
    - **Do a step -to impress a key stakeholder (executive, customer), with *value,* and early real results.**
    - **Do a step because you have enough of one key resource, at the moment (like cash) and lack another key resource, for the moment (like expertise).**
    - **Do at step for a very high risk, unfamiliar, new strategy, early; just to get the *experience, knowledge, and confidence* to pursue that strategy or to dump or modify it. The value or costs are not too relevant, compared to the *experience*, and *facts* about a new strategy, that will be scaled up, if successful, or too costly if not good.**
    - **Do Evo-steps in *parallel*, irrespective of their value/cost efficiency, in order to exploit your capacity to do so, and to win *time to deadline*.**
- **LEAVE IT TO EMPOWERED TEAMS:**
  - **Delegate the sequencing to smaller empowered local teams, *because* they are local. Do not make a sequencing decision at the planner or executive levels.**
    - **They know more. They are closer to the action. They need to feel empowered, not second-guessed in advance, too early. They need to feel responsible.**
- **INSTRUCT AI APPS TO HELP YOU, decompose, rate values and costs, and to prioritize. But ask for the basis for scoring (13.3.10.4.4) and review it skeptically. AI is rarely your local expert. It is not locally expert, but if you ask wisely, it is quick, cheap, and possibly as good as any human sequencing. Just do it, if you cannot prove it wrong quickly. Something *will* get done.**

13.3.10.4.4 Decomposition of (*Strategy 3, FOOD) 'Efficient Irrigation',* into independently implementable sub-strategies: *Evo-Steps*

Strategy 3: FOOD
Efficient Irrigation
20%



Instead of one bigger *task* set (12.10.4.3), to implement the strategy

Evo-steps ->	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	8	5	8	6	6	9	7	7	8	7
Cost (1-10)	3	2	4	3	3	5	4	4	5	5
Efficiency Ratio	2.67	2.50	2.00	2.00	2.00	1.80	1.75	1.75	1.60	1.40

Here are two reasons for the effectiveness on value and one reason for cost estimation for each of the 10 sub-strategies:

1. Utilizing Weather Forecast Data

- Value Effectiveness:
  1. Enhances irrigation scheduling by predicting rainfall, reducing unnecessary water usage.
  2. Helps optimize water use during dry periods, ensuring plants receive adequate moisture.
- Cost Estimation:
  1. Generally low cost as it primarily involves accessing existing weather data, potentially with some investment in weather stations or data subscriptions.

2. Mulching

- Value Effectiveness:
  1. Reduces water evaporation from soil, maintaining moisture levels.
  2. Suppresses weed growth, which can otherwise compete with crops for water.
- Cost Estimation:
  1. Low cost because it often involves using organic materials like leaves or straw, which may be locally sourced or even waste products.

3. Soil Moisture Sensors Installation

- Value Effectiveness:
  1. Provides precise measurements of soil moisture, allowing for targeted irrigation.
  2. Reduces water waste by ensuring irrigation only occurs when necessary.
- Cost Estimation:
  1. Moderate cost due to the expense of purchasing and installing the sensors, as well as potential maintenance.

4. Regular Maintenance and Inspection

- Value Effectiveness:
  1. Ensures that the irrigation system operates efficiently, reducing water loss due to leaks.
  2. Prevents breakdowns that could lead to overwatering or underwatering.
- Cost Estimation:
  1. Low to moderate cost, involving labor and potential replacement parts.

5. Proper Plant Selection

- Value Effectiveness:
  1. Choosing drought-resistant plants reduces overall water needs.
  2. Selecting plants suited to local climate and soil conditions improves growth and water efficiency.
- Cost Estimation:
  1. Moderate cost, mainly involving the purchase of specific plant varieties.

6. Drip Irrigation System Installation

- Value Effectiveness:
  1. Delivers water directly to the plant roots, minimizing evaporation and runoff.
  2. Provides a consistent moisture level, which can enhance plant health and yield.
- Cost Estimation:
  1. High cost due to the complexity of installing the system and potential for high initial investment in materials.

7. Rainwater Harvesting Implementation

- Value Effectiveness:
  1. Captures and stores rainwater for later use, reducing dependency on external water sources.
  2. Provides a sustainable and environmentally friendly water source.
- Cost Estimation:
  1. Moderate cost, involving the setup of collection systems, storage tanks, and potential filtration systems.

8. Training and Education for Farmers

- Value Effectiveness:
  1. Empowers farmers with knowledge to implement efficient irrigation practices.
  2. Enhances overall farm management, leading to better resource utilization and crop yields.
- Cost Estimation:
  1. Moderate cost, including training materials, workshops, and possibly hiring experts.

9. Smart Irrigation Controllers

- Value Effectiveness:
  1. Automates irrigation based on real-time data, optimizing water use.
  2. Reduces labor and human error in managing irrigation schedules.
- Cost Estimation:
  1. High cost due to the technology involved and potential ongoing maintenance or software updates.

10. Water Recycling and Reuse

- Value Effectiveness:
  1. Reduces water consumption by reusing treated wastewater for irrigation.
  2. Provides a reliable water source, especially in areas with limited water availability.
- Cost Estimation:
  1. High cost due to the infrastructure needed for treatment and distribution systems.
-



13.5.1 Priority - introduction

A dramatically different view of prioritization: a multi-factor engineering view

Priority

Concept *112

A ‘priority’ is the determination of a relative claim on limited resources.

Priority is the relative right of a competing requirement to the budgeted resources.

If resources were unlimited, there would be no need to prioritize things.

You could ‘have it all.’

An explicit Priority parameter can be used to specify any direct priority relationships.

Prioritization is one of the main themes and purposes of this book.

I believe Prioritization, in serious large-scale complex real-life decision-making is extremely poorly understood, taught and practiced. I cannot believe that grown people, and academics persist in teaching and using such poor illogical prioritization ideas.

There is a long list of priority ‘sins’. And I would prefer to present a positive a contructive set of ideas, and let you decide what you believe, about prioritization.

A common sin is using fixed weights to prioritize (as in Balanced Scorecard and QFD). I argued in detail against this in 2005 [Priority].

It is *when you decompose* into Evo Steps that the notion of ‘prioritization’ pops up. Which one of the *independently implementable strategies* is it smartest to do first or early?

Here is a partial summary of the bad priority practices.

1. Lack of considering stakeholders
2. Lack of considering non-financial values, quantified
3. Lack of considering multiple financial costs, especially long term maintenance and operational costs.
4. Lack of considering suitable large sets of values (like 30) and costs (like 5 to 10) simultaneously
5. Lack of serious description of strategies, including evidence and sources of evidence for belief in their virtues.
6. Lack of ability to change priorities dynamically, as evidence factors change. Fixed ‘weights’
7. Lack of ability to document, and to question, *constraints* and *assumptions*.

Here is a partial summary of our suggested priority practices.

1. Considering stakeholders
2. Considering non-financial values, quantified
3. Considering multiple financial costs, especially long term maintenance and operational costs.
4. Considering suitable large sets of values (like 30) and costs (like 5 to 10) simultaneously
5. Serious description of strategies, including evidence and sources of evidence for belief in their virtues.
6. Ability to change priorities dynamically, as evidence factors change. Evidence not weights [Priority]
7. Ability to document, and to question, *constraints* and *assumptions*.

13.5.2. Our Priority Practices

Strategy Prioritization for best *results flow*

- **Simple trivial systems:** I do not care as long as participants are happy. It is perhaps enough to 'learn from hard (personal) experience'.
- **Systems of consequence:** large, many-people impacts, and many *involved*, dangerous, economic consequences
  - This is the area I am talking about. Intelligent Prioritization really matters. Learning, about poor prioritization practices, at the level of learning from the individual planner's mistakes is *not* an intelligent option for the group.
  - We need a strong, intelligent *prioritization culture*, which has learned, and does learn, from all relevant sources, as soon as possible.
  - A paper with might be a sample of advanced thinking is [Anthes]. Multi-dimensional and Space.
  - Looking at real systems. Like my Norway Health Service 1-patient-one-journal catastrophe, I sometimes suspect there is a plot to prioritize the *highest cost* (8 years, 8 billion NOK) and *lowest value* (Doctors protesting in public, in fear of the system's dangers and lack of benefits) prioritization.
    - There may even be clear reasons for this: greed and stupidity. I tried, health minister etc. to do something about the stupidity, but was told, 'no problems'. One of [Flyvbjerg]'s 99.5% failures, I assume. [Akson]
  - The 'decompose' process, in the Evo Cycle, is necessary, only when a given strategy, is larger than your desired Evo step, length (a day a week, for example). But the moment we decompose into 2 or more *independently imlementable* Evo steps, we have an interesting choice. *Which sequence* shall we implement them in, to get the best results 'flow' (early values, early profit, early feedback)? This process is '***prioritization***'.

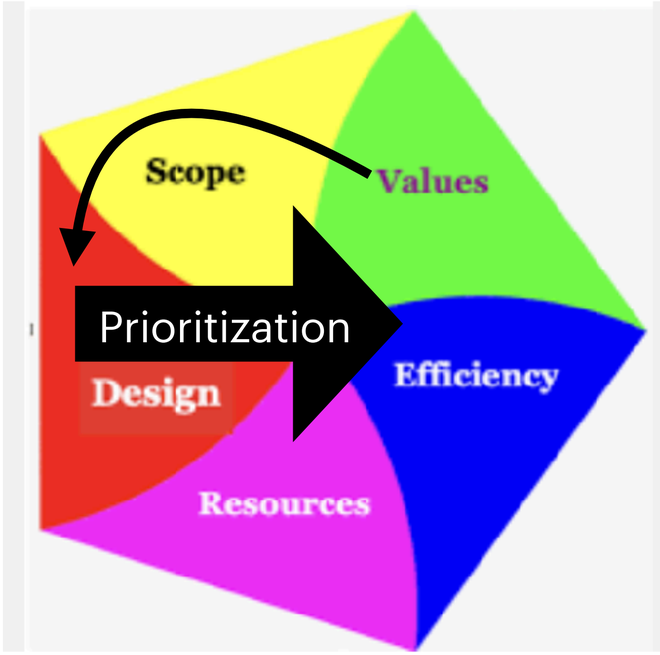


The Gilb's Planguage Evo Cycle [CE]
Synonyms: Means, Solution, Design, Strategy

13.5.3 The Dangers of Poor Prioritization

Of doing the wrong Evo-step, sub-strategies, too early

- **Poor prioritization,** is not as catastrophic as
  - Forgetting *critical* stakeholders
  - Forgetting *critical* stakeholder values
  - Forgetting *critical* constraints
  - Making poor-to-catastrophic, static, decisions
    - For these alone, even *one*, can totally destroy your system.
- Assuming, all these above are done well, the subject earlier in the book, then
  - prioritizing the top-ten strategies, in the *absolutely reverse order of* 'good',
    - and even less dangerous, wrongly prioritizing the top strategies, or even one set of their 10 sub-strategies,
    - are **not likely to have catastrophic consequences.**
  - That will be admittedly *sub-optimal*. Things *might* cost a bit more, and take longer time than necessary, and feedback experiences might not be the best. But, we will probably get by. Maybe no-one will even notice.
    - Because, if you have identified the critical stakeholders, and their critical values, and found 'great' value/cost strategies; *any* sequence of implementing sub-strategies, and top strategies, will *seem* pretty damn good. Because measurable results will flow earlier than usual, more than usual, and total failure (the [Flyvbjerg] 99.5% failure of his 16,000 project database, is *unlikely* to happen, at least not for the 'benefits', and not *so* severely for the resources.
- In summary: if you do all the *'stakeholder, value, strategy'* things - *right*, then - *bad priority implementation of strategies* is not going to lose you your job.
- Until further notice, the main threat is that the *'stakeholder, value, strategy'* practices need to be your main concern. Prioritization is sort of 'icing on the cake'.
- But, prioritizaton is still capable of saving *billions,* and *years*..... My main worry now is the 99.5% |Flyvbjerg] failure rates.



'[Penta]' Model of system attributes.

Figure 13.5.3 Values *drive* 'design of strategies' (thin arrow).

Decomposition of big strategies, into *independently implementable* small ones (**Evo-Steps**)

enables us to optimize **Efficency**,

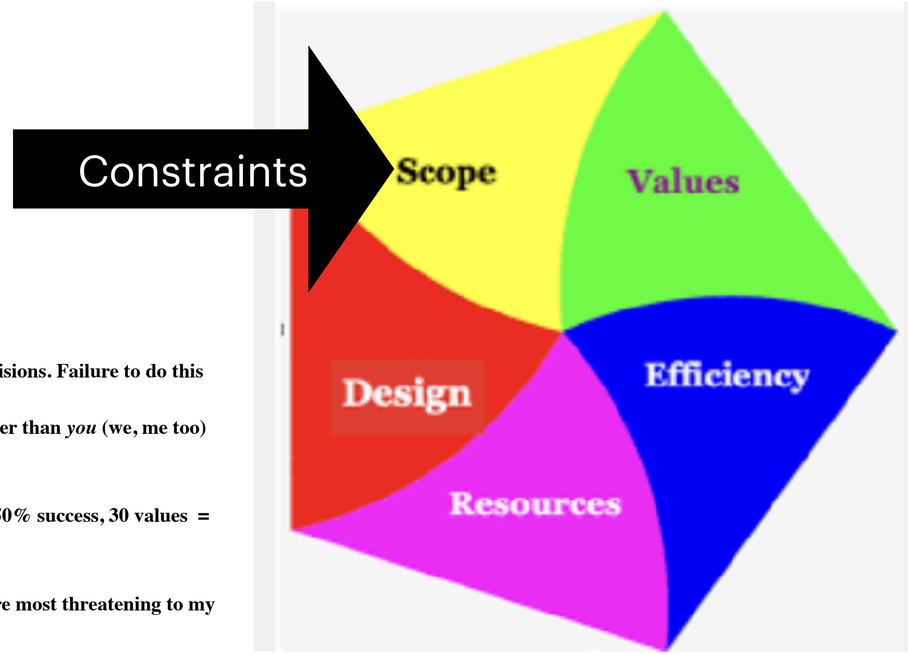
by smart *prioritization* (Fat arrow)

13.5.4 Intelligent Prioritization (IP), of stakeholders, values, constraints, and strategies.

What I recommend based on capabilities in Planguage [CE]

1. Considering stakeholders
2. Considering non-financial values, quantified
3. Considering multiple financial costs, especially long term maintenance and operational costs.
4. Considering suitable large sets of values (like 30) and costs (like 5 to 10) simultaneously
5. Serious description of strategies, including evidence and sources of evidence for belief in their virtues.
6. Ability to change priorities dynamically, as evidence factors change.
7. Ability to document, and to question, *constraints* and *assumptions*.

1. Considering stakeholders
    1. Be thorough in identifying critical stakeholders, because stakeholder *needs and power* is an important basis for prioritization decisions. Failure to do this thoroughly will directly lead to wrong priorities.
    2. Make use of AI to suggest stakeholders, and their values. It is at least 10x 'better' (comprehensive, knowledgeable) and 100x faster than *you* (we, me too) are.
2. Considering non-financial values, quantified.
    1. Identify at least 30 most-critical values, and work initially on <u>at least the most critical</u> 10 values. Think of it this way: 6 values = 50% success, 30 values = 95% success. [Plan-gineering, Coutts]
    2. Specify critical values <u>extremely clearly</u>, and <u>especially</u> the 'Scale, Tolerable, Goal' parameters.
    3. Get help from an AI tool, to draft this quickly, then refine it manually, or with further AI questions (like: Which critical values are most threatening to my project?)
3. Considering multiple financial costs, especially long term maintenance and operational costs.
    1. Plan resource budgets for 5 to 10 resources, especially long-term resources
    2. Use AI to suggest the resources you should define, in your type of project. 'Where do my types of projects go badly with regard to resources?'
4. Considering suitable large sets of values (like 30) and costs (like 5 to 10) simultaneously.
    1. Your probably need to define, and prioritize, using about 30 *critical* values, and at least 10 *most critical* values, for the greater body (80%) of stakeholders
5. Serious description of strategies, including *evidence* and *sources* of evidence, for belief in their virtues (values and costs).
    1. Specifying strategies, finally, with a few buzzwords, or on a yellow stickie/Miro Board, is ridiculous. Not enough detail to understand. Guaranteed failure.
    2. Learn to describe strategies, in enough detail, that your priority decisions for strategies, and sub-strategies are well-founded and smart
    3. AI will help your quickly, cheaply, and intelligently to flesh out detail for strategies . Use AI, refine it. You cannot beat it.
6. Ability to change priorities dynamically, as evidence factors change.
    1. Feedback(measurement, opinions, complaints) from Evo-step implementations, can be fed back into Impact Estimation, to update your priority decision data base [ValPlan]
    2. Feedback, technical and economic updates, experts can update earlier data, and enrich it, and using Impact Estimation tables (with source, evidence, credibility) which *then* help you prioritize smarter.
    3. Reviews of all plans, especially older ones (1 year or more), to find bad or misleading assumptions and data, is necessary, in this fast moving world.
7. Ability to document, and to question, *constraints* and *assumptions*.
    1. All critical constraints (binary and scalar [CE]) need to be made in the 'formal planning model' (like in [ValPlan] app), written, revised, approved, and questioned.
    2. Most of the constraints are wrong, or unnecessary to some degree, with imaginative planning and design, andre need to try to relax or remove them, as it can be useful to do so. They have a profound effect on your prioritization, so do not let unnecessary assumptions about constraints [Guides] dwell too long.



Basic 'Penta' Model of system attributes.

Figure 13.5.4 Binary Constraints are part of the Scope definition.
Like 'Legal', 'Privacy Compliant', Ethical', 'Contractual'

13.5.5 The 'Impact Estimation Table. 'IET', as a *prioritization tool*

- Even simple and rough prioritization, will probably do some good, optimizing your resources and results. Better than 'nothing'.
- So it really boils down to having a sense, for your project, of how much *benefits* you can expect to get from prioritizing more-or-less well, and the *costs* of doing the prioritization.
- If we can reduce the **effort** (training, doing)we need, to use *advanced* prioritization, down to a trivial level, then we can afford to use more-advanced prioritization methods.
- *Human* effort to do a reasonable IET draft, as a parallel team, takes one calendar day. Not too bad. But it is usually very incomplete in many respects. For example sources of beliefs about impacts-
  - AI effort is much better, for humans. Drafts of large tables can be done, within the hour. And they can be re-done in 5 minutes. They will also be more broad and insightful.
    - But they will lack *local* project ideas and data, that can 'make all the difference'. So you will still need humans to *review*, and *tune* models drafted by AI (like the one at the right on this pages)
    - AI can do silly things, as we know, so we have to ask for its reasoning, its sources (*read* the (URLs) suggested citations!), and check them, as thoroughly as we would for humans!
    - Humans are worse than AI, but both sources are *suspicious until verified*.
      - But, that problem does not make them useless or dangerous, just not 'perfectly trustworthy'. *Both* are a good combination..

Simple Prioritization

	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	**8**	**5**	**8**	**6**	**6**	**9**	**7**	**7**	**8**	**7**
Cost (1-10)	**3**	**2**	**4**	**3**	**3**	**5**	**4**	**4**	**5**	**5**
Efficiency Ratio	**2.67**	**2.50**	**2.00**	**2.00**	**2.00**	**1.80**	**1.75**	**1.75**	**1.60**	**1.40**

Figure 13.5.5 A simple AI-generated Impact Estimation Table (from 13.3.10.4.4)

13.5.5.2 The Impact Estimation Table as a prioritization tool

Simple Prioritization, methods using the simple table below

1. The overall idea is to use the Impact Estimation Table (IET, [VIET]) to sequence (prioritize) the 'best strategy 'sub-strategies', in terms of value for money.
2. I do not think that the optimal sequencing, of the best priority strategy, makes much real difference, at all. Just do them.
3. Certainly do not delay doing them, to argue what should go first!
4. I have asked the AI to explain (give an *argument*, if not factual evidence) for the value ratings, and the cost ratings. They seemed better than I could get from a human. But I have not researched them deeply. I would need citations, and Chat GPT isn't offering them. Use Perplexity AI if you want citations.
5. I have an implied, common sense, default prioritization policy. Do profitable stuff earliest. So the Value/Cost ratio is used, to suggest a sequence, and why not. Little harm can be done. Get going.
6. Note that in Planguage and Evo, that very first Evo step (*Utilizing Weather Forecast Data*) is supposed to measure some form of Planned Objectives movement, along a Scale or more, along with some kind of cost feedback, against some of the Budgets.
7. So, if there is some *obvious deviation* from the 8 and the 3, like 'no measurable value against any Goal was measurable', or the 'one week Evo-step took 4 weeks'; then we have useful feedback, to make our hypothesis (8 Value / 3 Cost) seem to have failed. And we can sit down and analyze what happened (Learn).
8. There is an argument that if we start with extreme V/C like 8/3 then it will be easier to roughly sense low value and high costs. Just a thought!
9. So, no matter what, we get into contact with *'system reality'* next week, and every weekly Evo-step. Btw this is a similar idea to 'Lean Startup' (get out on the street day one, and ask passersby if they will buy your conceptual product). [SteveBlank.com]
10. But, I would argue that this, prioritization method, is a superior and more focussed approach to the multidimensional value management problem we have. [Plan-gineering, 3.2, [Coutts]]. [CE] seems to have a serious (220 project) track record of improving 3-year startup-survival from 40% to 80% in Europe [Coutts].
11. So, to summarize, something like this simple table, might be more than enough to teach, and apply, intelligent engineering-like prioritization decisions.

	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	**8**	**5**	**8**	**6**	**6**	**9**	**7**	**7**	**8**	**7**
Cost (1-10)	**3**	**2**	**4**	**3**	**3**	**5**	**4**	**4**	**5**	**5**
Efficiency Ratio	**2.67**	**2.50**	**2.00**	**2.00**	**2.00**	**1.80**	**1.75**	**1.75**	**1.60**	**1.40**

A simple AI-generated Impact Estimation Table.

Sequenced by Efficiency Ratio (V/C)

13.5.5.2 Advanced IET Prioritization

Adding parameters and data to manage risk

- The Impact Estimation Table can be used for a wide variety of purposes. It is general purpose tool, like a Swiss Army knife.
- The most general purpose is to compare, see the relationship, between two levels of planning, one of which impacts the other. Means impact Ends.
- We are all aware that estimations are difficult, 'especially about the future'. [Cost Engineering]. So when a 10 x 10 values and strategies matrix IET gives you 100 estimates, you are suspicious, but checking each one in depth is something you are unlikely to do. You might be tempted to accept it and move on. Can't blame you.
- But, for some purposes, it is necessary that we know how much we can trust the IET numbers, before making big decisions. So I have designed some options to give us more data about the estimates.
- There are two primary sub-methods, which lead to other options
  - Plus/Minus **Uncertainty**: we can add the range of uncertainty to any estimate: 90±8
  - **Credibility** a is expressed on a Scale of 0 (none) to 10 (perfect), and it is based on written specification of the 'evidence', and the source of the Evidence (URL, person, research, case).
- Some of the consequent IET-method options, are that we can use these data *cumulatively* to get a 'figure of merit' for Uncertainty, for a strategy. And a figure-of-merit for the 'cumulative Credibility' for a Strategy. [VIET, CE]
  - We can then look at 'worst possible' cases, and maybe choose options that are relatively good, i.e. the 'worst possible case'. Or, maybe you are *'feeling lucky today, Punk?'*



An ancient Multi-tool I found at the Kistefoss Art Museum, Norway.

13.5.5.3 Specific Uses of IET

Here is a list of different uses of the IET process. From my own practice.

There are many more, and hopefully this book will point them out. Like 'Stakeholder Management', 'prioritization', and 'decomposition'.

I think of IET as a Swiss Army Knife tool.



An ancient Multi-tool I found at the Kistefoss Art Museum, Norway.

VIE can be used for a wide variety of purposes including:

1. **Evaluating a single design idea**. How good is the idea for us?

2. **Comparing two or more design ideas to find a winner,** or set of winners. Hint: Use . IET, if you want to set up an argument against a prevailing popular, but weak design idea!

3. Gaining an **architectural overview of the impact of all the design ideas on all the objectives and budgets.** Are there any negative side effects? What is the cumulative effect?

4. Obtaining **systems engineering views** of specific components, or specific performance aspects. *For example: Are we going to achieve the reliability levels?*

5. **Analyzing risk**: evaluating a design with regard to 'worst case' uncertainty and minimum credibility.

6. **Planning evolutionary project delivery steps with regard to performance, value, benefits and cost.**

7. **Monitoring**, for project management accounting purposes, the **progress** of individual evolutionary project delivery steps and, the progress to date compared to the requirement specification or management objectives.

8. **Predicting** future costs, project timescales and performance levels.

9. **Understanding organizational responsibility** in terms of performance and budgets by organizational function.

10. Achieving **rigorous quality contro**l of a design specification prior to management reviews and approval.

11. **Presenting ideas** to committees, management boards, senior managers, review boards and customers for approval.

12. **Identifying** which parts of the **design** are the **weakest** (risk analysis). *Hint: If there are no obvious alternative design ideas, any 'weak links' should be tried out earliest, in case they do not work well (risk management). This impacts scheduling*.

13. Enabling **configuration management** of design, design changes, and change consequences.

14. Permitting **delegation of decision-making to teams.** People can achieve better internal progress control using VIE, than they can from repeatedly making progress reports to others, and acting on others' feedback.

15. Presenting **overviews of very large, complex projects and systems** by using hierarchical VIE tables. Aim for a one page top-level VIE view for senior management.

16. **Enabling cross-organizational co-operation** by presenting overviews of how the design ideas of different projects contribute towards corporate objectives. Any common and conflicting design ideas can be identified. *Hint: This is important from a customer view-point; different projects might well be delivering to the same customer interface*.

17. **Controlling the design process.** You can see what you need, and see if your idea has it by using an IE table. For example, which design idea contributes best to achieving usability?
Which one costs too much?

18. **Strengthening design.** You can see where your design ideas are failing to impact sufficiently on the objectives; and this can provoke thought to discover new design ideas or modify existing ones.

19. Helping informal reasoning and discussion of ideas by **providing a framework model** in our minds of how the design is connected to the requirements.

20. **Strengthening the specified requirements.** *Sometimes, you can identify a design idea, which has a great deal of popular support, but doesn't appear to impact your requirements. You should investigate the likely impacts of the design idea with a view to identifying additional stakeholder requirements. This may provide the underlying reason for the popular support. You might also identify additional types of stakeholders*.

Source: Competitive Engineering, 2005 [CE]

13.5.6 IET Rating the Credibility of an estimate

- An estimation cell is selected (1)
- This opens a detail window (2) for that cell, at the right, where we can enter data about Credibility [3] and Uncertainty.
- Data about the evidence for the estimate is entered, along with a source [3]
- The Credibility slider (4) is moved to the corresponding credibility (AI needs to do this for us, and it can, but not in this app just now)
- The ± uncertainty range is entered [5]. Default ±0
- The cumulative Worst Case and Credibility are calculated [6]
- This is something you can do on any spreadsheet.
- The yellow and red colors in the cells are tracking incremental progress towards Goals and Budgets, with safety margins. And we need not go into that here. [CE, VIET].

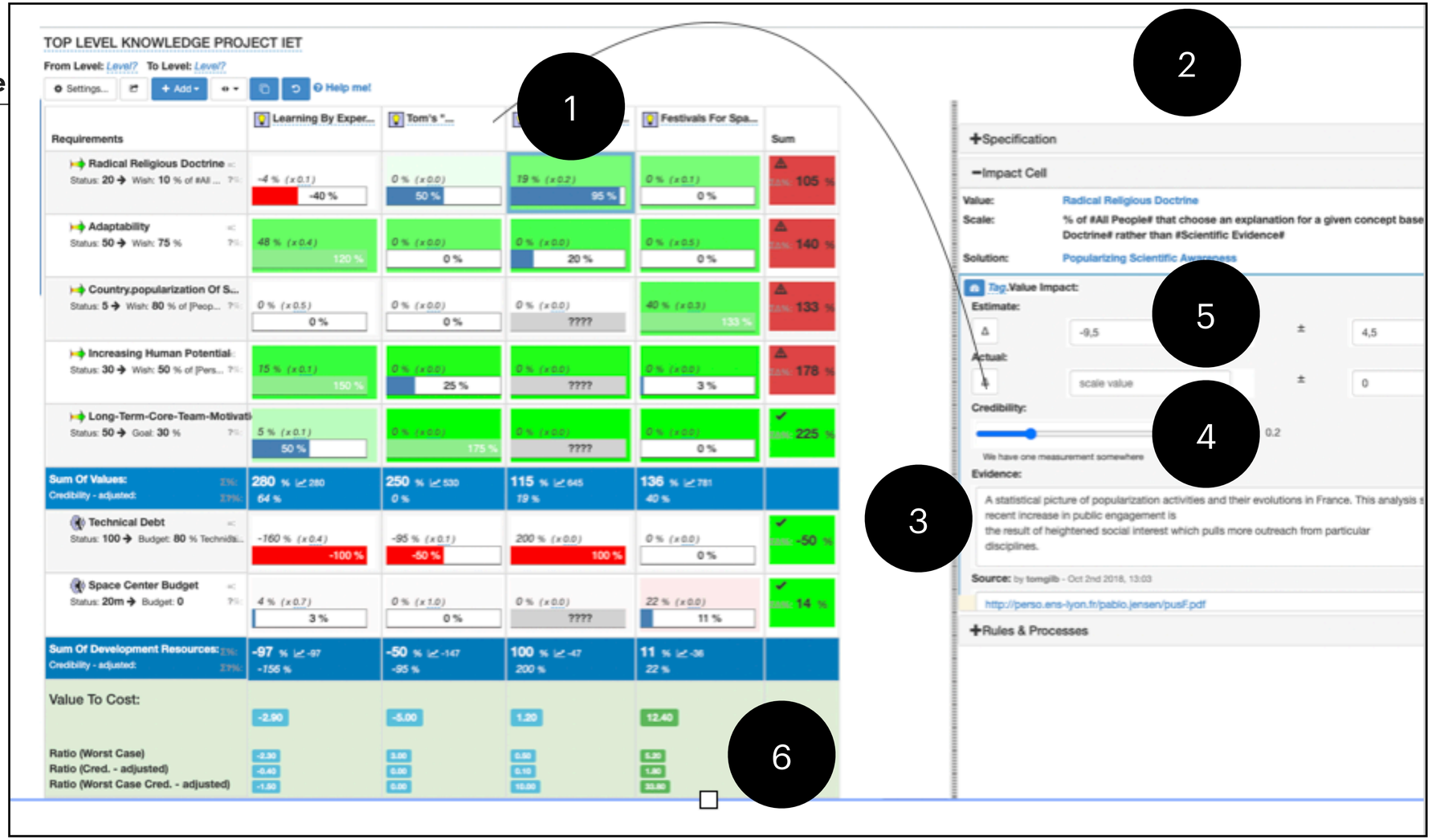


Figure 5.6. IET with Credibility ratings, using [ValPlan] app.

13.5.7 IET Table. Another example, previous page. + Bar Chart with Uncertainty and Credibility (upper right)

Source [KEN]

Requirements		GraphMetrix Products	Oracle Products	Decentralized Net...	Sum
Knowledge Accuracy	Δ:	**30** ± 15	**-40** ± 30	???? ± 0	⚠
Status: **80** → Wish: **20** % differen...	=:	110 % diffe...	40 % diffe...	0 % diffe...	ΣΔ%: **17** ± 75 %
% difference between Correct Value and G	?%:	*-5 % (x 0.1)*	*34 % (x 0.5)*	*0 % (x 0.0)*	
[Knowledge Type = **Historical E...**]		-50 ± 25 %	67 ± 50 %	????	
13 Jul 2030					
Knowledge Automatability	Δ:	**85** ± 15	**30** ± 10	**42** ± 0	⚠
Status: **5** → Wish: **95** % of [Know...	=:	90 % of [K...	35 % of [K...	47 % of [K...	ΣΔ%: **174** ± 28 %
% of [Knowledge Type] for a [Knowledge S	?%:	*28 % (x 0.3)*	*0 % (x 0.0)*	*0 % (x 0.0)*	
[Knowledge Type = **Internet Of ...**]		94 ± 17 %	33 ± 11 %	47 ± 0 %	
13 Jul 2025					
Sum Of Values:	Σ%:	**44** ± 42 %	**100** ± 61 %	**47** ± 0 %	
Worst Case:	Σ±%:	*2 %*	*39 %*	*47 %*	
Credibility - adjusted:	Σ?%:	*23 %*	*34 %*	*0 %*	
Worst Case Cred. - adjusted:	Σ±?%:	*16 %*	*9 %*	*0 %*	
Knowledge Distribution Costs	Δ:	**10** ± 4	**40** ± 20	???? ± 0	✔
Status: **0** → Budget: **100** % of Budge...	=:	10 % of Bu...	40 % of Bu...	0 % of Bu...	ΣΔ%: **50** ± 24 %
% of Budget for [Knowledge Package] of [	?%:	*17 % (x 0.3)*	*68 % (x 0.3)*	*0 % (x 0.0)*	
[Knowledge Package = **Construct...**]		10 ± 4 %	40 ± 20 %	????	
13 Jul 2025					
Sum Of Development Resources:	Σ%:	**10** ± 4 %	**40** ± 20 %	**0** ± 0 %	
Worst Case:	Σ±%:	*14 %*	*60 %*	*0 %*	
Credibility - adjusted:	Σ?%:	*17 %*	*68 %*	*0 %*	
Worst Case Cred. - adjusted:	Σ±?%:	*4 %*	*18 %*	*0 %*	
Value To Cost:		4.40	2.50	Infinity	
Ratio (Worst Case)		0.10	0.70	Infinity	
Ratio (Cred. - adjusted)		1.40	0.50	0.00	
Ratio (Worst Case Cred. - adjusted)		3.70	0.50	0.00	

We KNOW we don't know

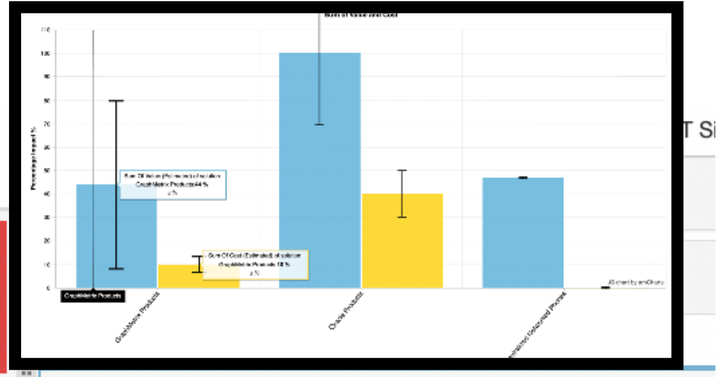


Figure 4.21 B
A bar chart based on the VIE Table, a summary.

Tag.Value Impact:

Estimate:

Δ | 85 | ± | 15

Actual:

Δ | scale va | ± | 0

Credibility:

0.3

There are several measurements in the estimated range

Evidence:

current measures of GM vs Competitor A and B

Source: by **tomgilb** - Jul 13th 2020, 15:00

GraphMetrix slides 2020

Templates

Add Comment...

+Rules & Processes

13.5.6 Prioritization Policies

Policies = strong, top level default guidelines, aka wisdom. 230724 TG

1. EFFICIENCY: Prioritize things (stakeholders, values, strategies) which estimably give a best set of values, for lowest set of costs.
2. POWER: Prioritize stakeholders with power and influence.
3. THE MANY: Prioritize Values which serve many stakeholders, and are ethical.
4. RISK: Prioritize Strategies, and sub-strategies (Evo-steps), which give high values for costs, with respect to uncertainty and credibility (low risk)
5. LEARN FAST: Prioritize quickly, up front, but do quickly, and learn quickly, whether the prioritization was such a good idea.
6. DO, DON'T TALK: Prioritize the quick experiment over the lengthy discussion.
7. LONG-TERM: Prioritize long-term operational results, over short-term thinking.
8. USER: Prioritize exceptional user-and-customer experiences, the best. Worry about costs of it later [Cost Engineering]
9. FREEDOM: Prioritize giving freedom and resources to your staff, to constantly, incrementally, improve results.
10. PROCESS: Prioritize the mechanisms, like organization, motivation, and funding which continuously, and in the long term, help give better results.

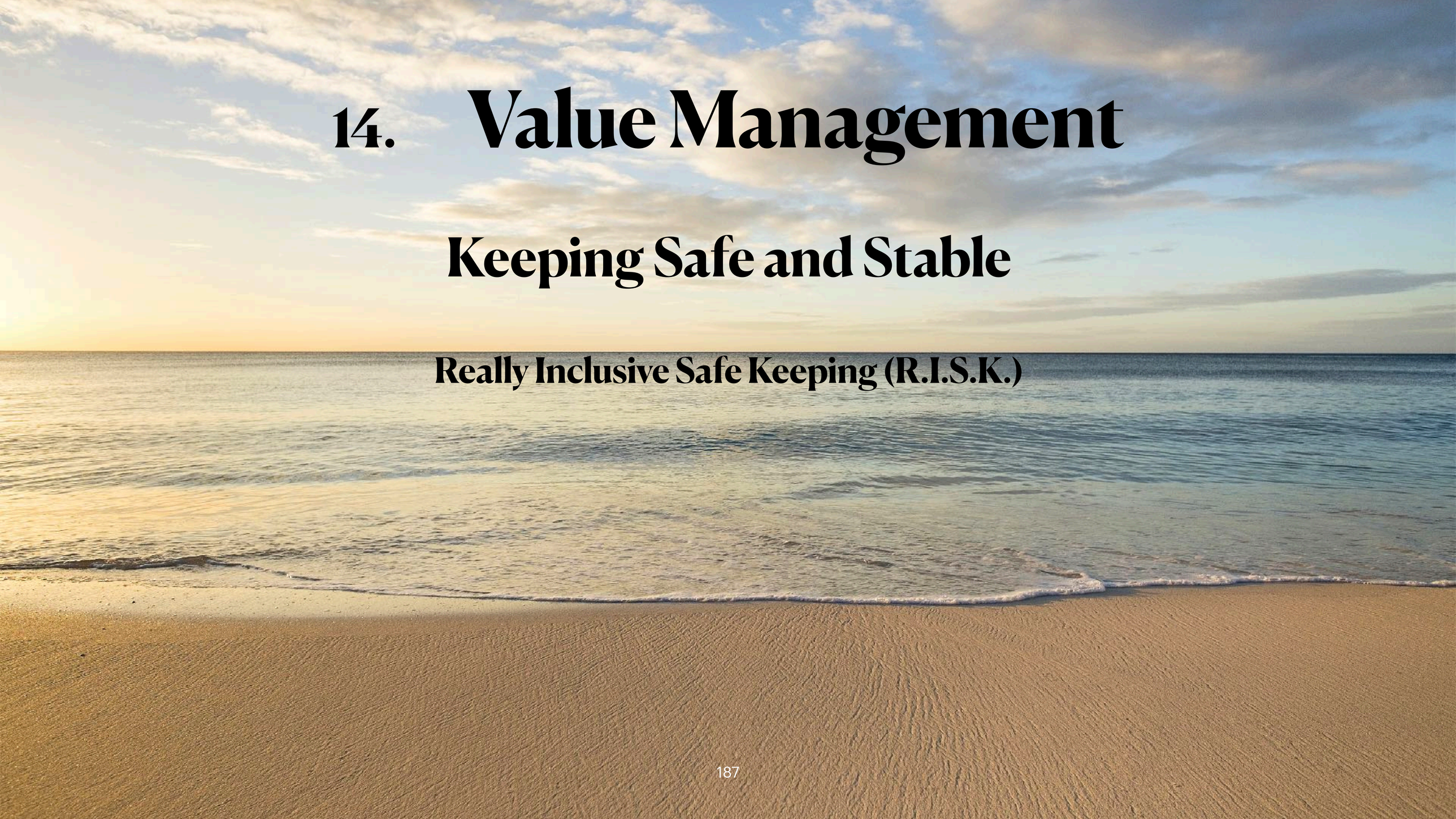
13.5.7 Prioritization Principles

© 220724 Tom Gilb, 050824 added 11

5.5	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	8	5	8	6	6	9	7	7	8	7
Cost (1-10)	3	2	4	3	3	5	4	4	5	5
Efficiency Ratio	2.67	2.50	2.00	2.00	2.00	1.80	1.75	1.75	1.60	1.40

Figure (see earlier)

1. OPTIMIZATION: Prioritization *sequencing*, of service to stakeholders, values and strategies, is about optimization, but probably not survival.
2. CRITICAL SET: Selecting a sufficiently large critical set of stakeholders, and values is necessary for survival, avoidance of failure, by definition of 'critical'.
3. CRITICAL VALUE SCOPE: The quantity of critical stakeholder-value objectives you choose to manage initially, will be corrected to your success probability: 6 = 40%, 25 = 90% roughly.
4. STAKEHOLDER POWER: The priority of a single critical stakeholder is proportional to their powers regarding your project.
5. STAKEHOLDER SATISFACTION: The priority of a single critical stakeholder-value objective is proportional to the % of critical stakeholders which it can satisfy. (Figure 3.5)
6. EVO-STEP EFFICIENCY: The priority of a single sub-strategy Evo-step, is generally in relation to the values it can deliver, for the resources it needs to deliver the values.
7. POLICY EXCEPTIONS: A single prioritization decision, can be made by default, using an agreed prioritization Policy; but any single prioritization decision, can consciously violate a general policy, for specific urgent reasons, for example *'expected budget not yet available',* or *'the CEO will be impressed if we do this before year end'.*
8. EVO FLOW EFFICIENCY: Smart streams of'Evo-step' prioritization decisions, will optimize efficiency; which is useful *after* the 'survival prioritizations' have been correctly made.
9. PRIORITY LOGIC: Prioritization decisions can generally be computed, dynamically based on quantified data, using a spreadsheet or AI, or Impact Estimation Table.
10. ADVANCED PRIORITIZATION: More-advanced prioritization decisions can be made to the degree that impact estimates are backed up by IET tools such as [CE, VIET] such as quantified, qualified values, detailed strategy specification, varied time horizon evaluation, ± uncertainty specification, impact evidence and evidence sources.
11. EFFICIENCY DECREASE: The efficiency of a prioritization sequence will not continue to decrease: it may improve as we Learn from experience, and as new technologies change the values/costs ratios.



14. Value Management

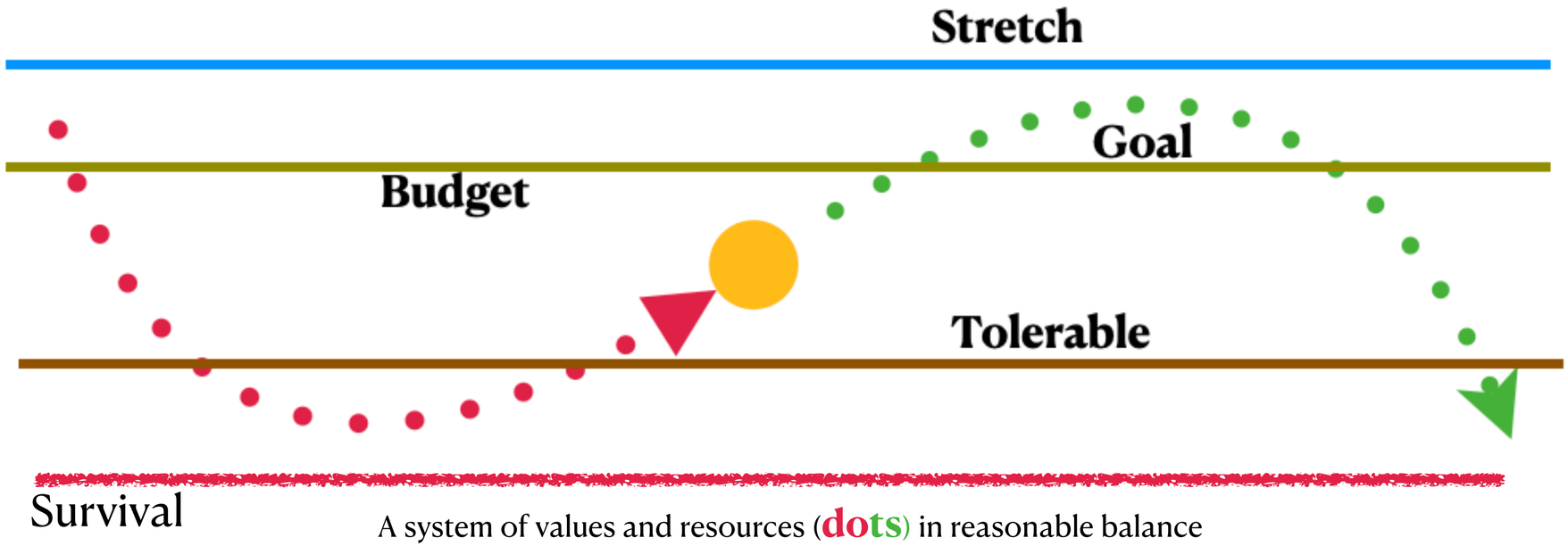
Keeping Safe and Stable

Really Inclusive Safe Keeping (R.I.S.K.)

14.1. Keeping Safe

Keeping Critical Values within acceptable borders

- I am going to try to do a chapter, 14. on a subject that many people would call ‘risk’.
- But I am also going to accept a challenge from a friend (CD) to discuss the subject, while avoiding the term and the concept. [Dowie].
- Maybe I can come up with ideas about the subject, that fit the problem, using Planguage, as a design, which is better than conventional ‘risk’ thinking? Let us see.



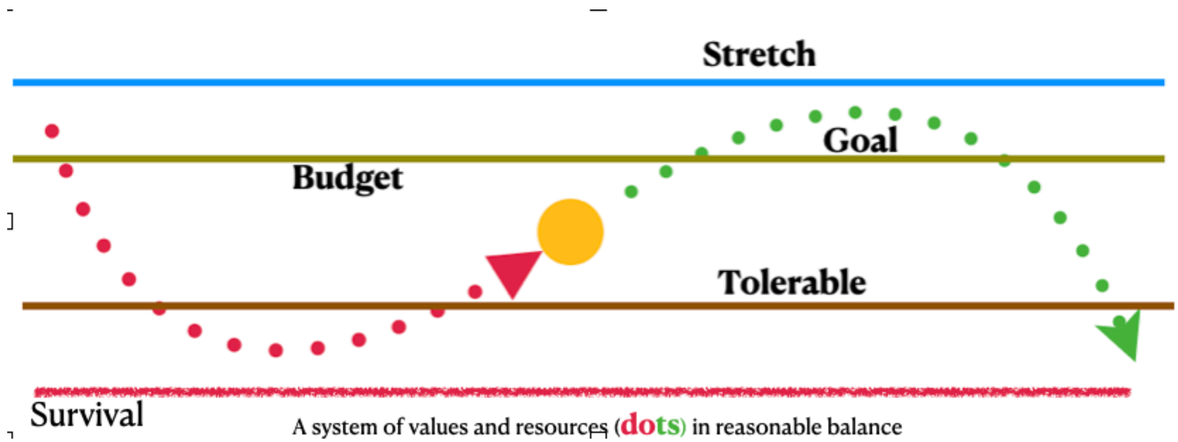
A system of values and resources (dots) in reasonable balance

[Optima]

14.2. Keeping Safe: 'Value Management'

Keeping Critical Values within acceptable borders

- I think I have, just come upon, the key ideas, and have managed to capture them in the diagram. (At right)
  - and also in the comparative Table (Next page), which Perplexity made for me, based on 2 of my books [Optima, CE] and some of my papers.
- The *key idea, I have to offer,* is that we focus our 'management' on the critical stakeholder Values (green dotted arrow) , including critical resources (red dotted arrow).
- These values are all quantified, in terms of the levels we want (require).
- We manage a number, '*n*' of these Values (for example, *from 5 to 25*).
  - Each value is symbolized by each dot, in the arrows, in the diagram.
- Our *value management* process aims to drive values to at least a Tolerable (Constraint) level, initially for all *critical* values (defined as having a specified 'Survival' level).
- Then, secondarily, our management process needs to *drive values up to 'Target' levels, for example 'Goal' levels* (success levels). [Success]
- Then *resource* values, need to be reduced, for *efficiency* of the system, good *values/costs ratios* [Cost Engineering]
- All the while, we keep *managing* so that no 'critical value' can cause 'system disaster', 'failure'), I.e. cannot fall below a defined 'Survival' level.



14.3. Comparing *[Value Management] using Planguage,* with 'Risk' Management Conventions

'Value Management'

- *Perplexity*: "Based on the information provided in your [Optima] book and the [CE] book, I'll construct a comparative table, focusing on Gilb's Planguage approach, versus conventional Risk Management, specifically addressing the management of stakeholder value levels and costs over a *startup product's 10-year lifetime*:

- Gilb's Planguage approach offers a more comprehensive, quantitative, and value-focused method for managing stakeholder value levels and costs throughout a product's lifetime.

- It explicitly defines survival and target levels, continuously measures progress, and adapts to changing priorities.

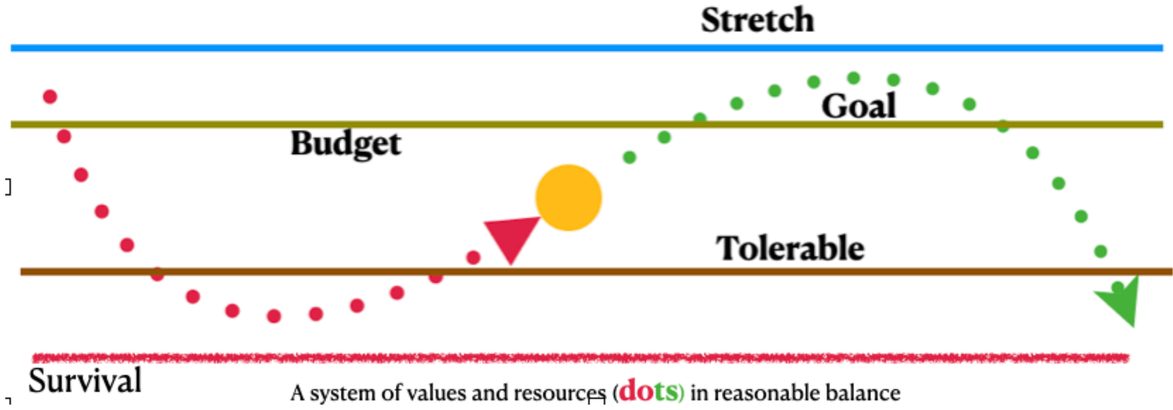
- This approach is particularly well-suited for managing the complex, long-term value delivery required in a startup product's 10-year lifecycle, ensuring that stakeholder values are maintained above survival levels and driven towards target levels over time."

Aspect	*[Value Management], [Optima] Gilb's Planguage Approach*	Conventional Risk Management
Focus	*Stakeholder values and quantifiable objectives*	Primarily potential threats and their mitigation
Value Definition	*Precise, quantitative metrics for stakeholder values*	Often qualitative or semi-quantitative risk assessments
Survival Levels	*Explicitly defined as quantitative thresholds*	May not be explicitly defined or quantified
Target Levels	*Clearly specified and quantified goals*	Often general or qualitative goals
Time Horizon	*Considers entire product lifetime (e.g., 10 years)*	Often focuses on short-term project risks
Measurement	*Continuous measurement of actual value delivery against objectives*	Typically measures risk reduction or avoidance
Adaptability	*Highly flexible to changing requirements and priorities*	May struggle with rapidly changing environments
Decision Tools	*Uses Value Decision Tables and Impact Estimation tables*	Often relies on risk matrices or subjective assessments
Cost Management	*Integrated with value delivery, focusing on efficiency*	Often treated separately from value considerations
Stakeholder Involvement	*Comprehensive stakeholder analysis and value quantification*	May have limited stakeholder input beyond risk identification
Value Delivery Approach	*Iterative, incremental, focused on delivering measurable value*	Often linear, focused on completing predefined tasks
Success Criteria	*Defined in terms of stakeholder value satisfaction over time*	Often defined as absence of negative outcomes or project completion
Planning Language	*Uses specific Planguage for clarity and precision*	Varied terminology, often organization-specific
Value-Cost Balance	*Explicitly balances multiple stakeholder values against costs*	May focus more on cost avoidance than value optimization
Long-term Value Tracking	*Designed to track and adjust for value delivery throughout product*	Often focuses on immediate project success, not long-term product value

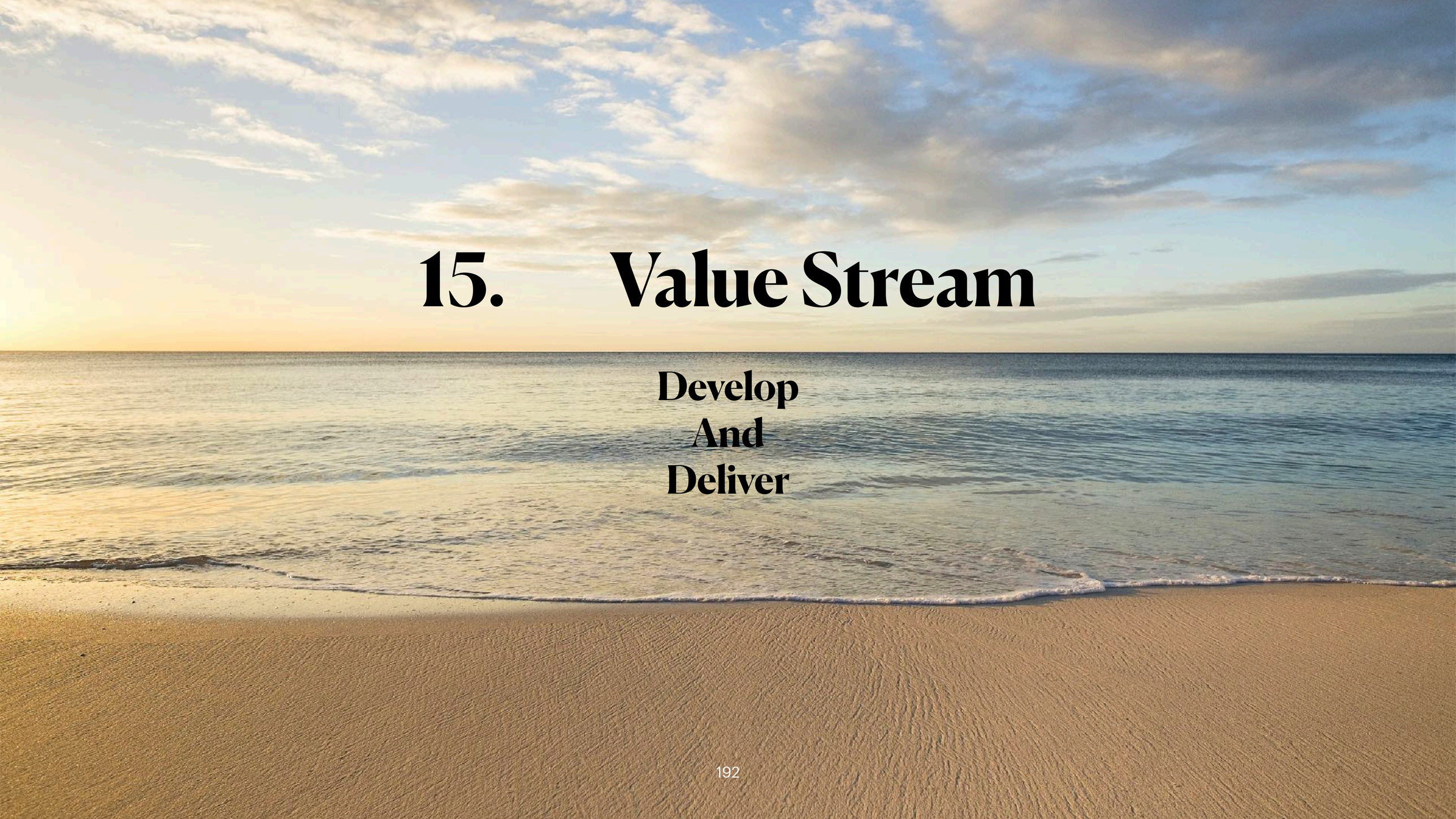
14.4. Keeping Safe : Really Inclusive Safe Keeping (R.I.S.K.)

Keeping Critical Values within acceptable borders. Secure and Future Efficiency. (S.A.F.E.)

- This '[**Value Management**]' process involves the following *disciplines*.
  - **Stakeholder** Analysis
  - **Value** Analysis and value level specifications
  - **Architecture** and design to meet the required levels
  - **Measurement** of all critical value levels, for system lifetime, in time to prevent them getting out of control.
  - **Interventions**, long after initial delivery, to re-install specified levels, and to *improve levels,* based on new competitive requirements.
  - This implies a **life-cycle long Evo-Cycle** for managing the value levels.
- We have shifted the focus from a *'set of things that could go wrong' ('risks'),* to ***'Managing the overall stakeholder values'***
- We seem to have made a case that conventional 'risk management' practices can be superseded by 'Value Management'. (I plan my next book to be on that subject, so let us see how that larger effort goes!)







15. Value Stream

Develop And Deliver

15.1 Value (s) Stream

You startup Values are many, and parallel

- The word 'value' is often, impulsively, narrowly interpreted, as some kind of *financial* value. Maybe the 'interpreter's focus is not on 'human values' (Search for that!)
- By now, in this Startup book, we know 'Value' is *many different* critical stakeholder values. **Not** *mainly* 'financial', but, mainly, other stakeholder-valued 'qualities'.
- Up to this point in the book, and in the Evo-cycle, the Value ideas have been theoretical, and hypothetical, during the *Stakeholder to Values to strategies to Decompose and Prioritize* Evo-cycle stages.
- NOW: we are ready to turn *theory* into **practice**.
  - '**Develop**': acquire, buy, build, the sub-strategy, serious money spent, perhaps.
  - Then, '**Deliver**' it to the system: *integrating it into real life with real stakeholders.*
- Upon 'Delivery' it should begin to display some Values
  - We *can* experience bad results. In spite of trying to estimate these in IET (15.2).
    - For example bringing *any,* or *several,* values <u>down</u>, from an *already-achieved* Goal level
    - or, bringing value levels <u>below</u> an already-achieved *Tolerable* level.
  - Hopefully we can get *some* degree, <u>any improvement is better than none</u>, of <u>'primary *critical* value'</u>, for which the Sub-strategy Evo-step was consciously designed. The side-effects are largely 'unconscious' but uncontrollable set of effects.
  - Secondarily, IF other 'side effect' Values, are hopefully not 'negative', or at least not *destructively* negative, we can achieve some *useful* critical Value level. And move on in the Evo-cycle process.

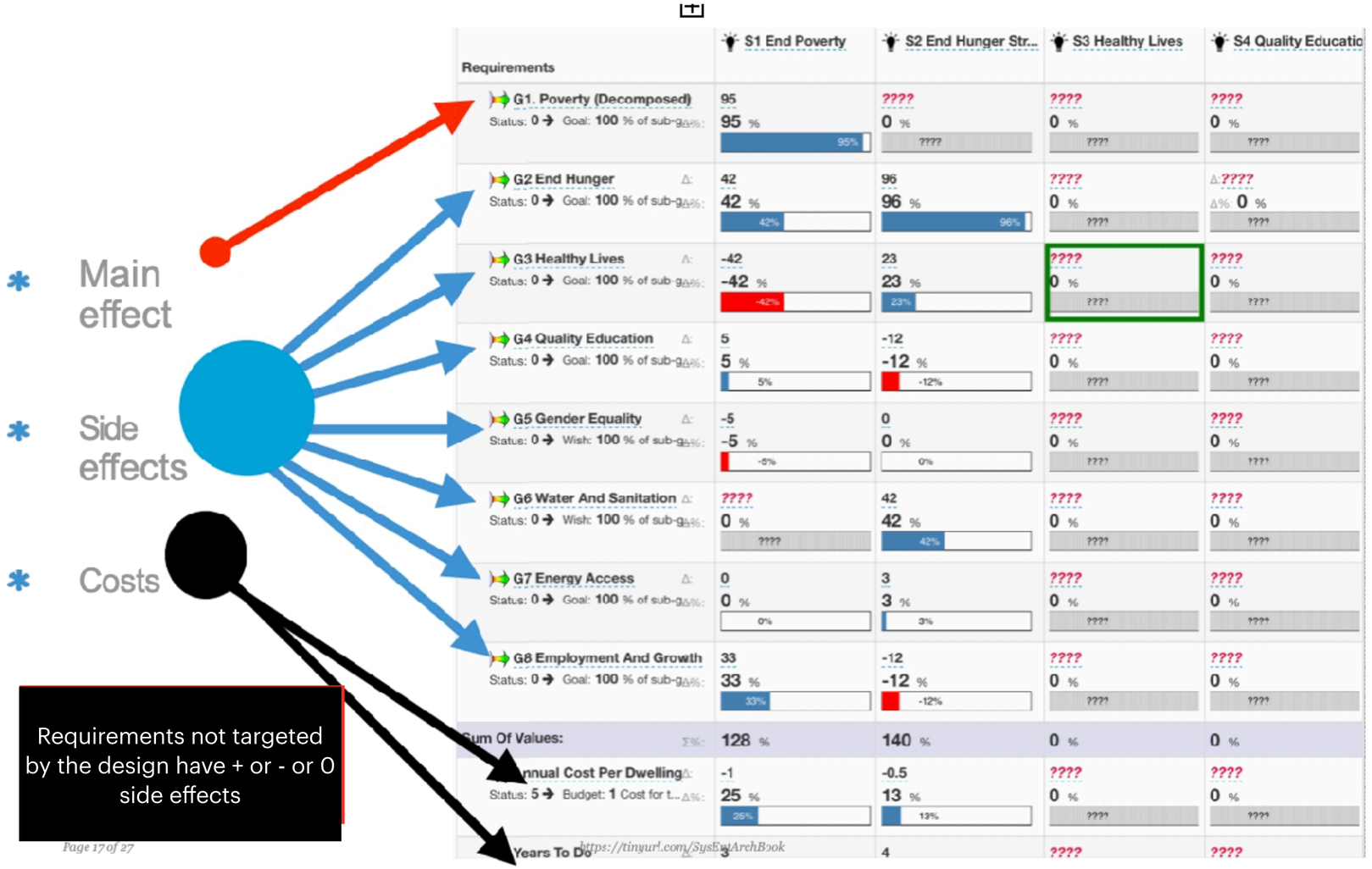


The Gilb's Planguage Evo Cycle [CE]

15.2 Side Effects, their nature

Almost a law of nature: things have multiple value attributes and multiple costs: like it or not.
And you cannot simply adjust the single attributes you don't like: you need a new design, and the resulting set of attributes are out of your control. They just happen.

- If you design to get a specific effect, in values or costs, then any effects in all *other* values and costs, are, by definition, *side effects*.
- The only reason to keep a strategy which is intended for some 'primary value effect', is if that effect is sufficient large, and the side effects are 'acceptable' overall. Balance.
- The only way to change side effects is to change the *design*.
- If you are lucky, you will with *other* designs, get the same Main Effect, but also a *better set* of side effects.
- There are infinite possible designs, so you can iterate designs, until you get a good enough, set of side effects.
  - But it is probably better to experiment with real short Evo-steps than to estimate infinite new strategies.
- The 'multi-effect totality' is intimately built in to the design itself, including its implementation and operation.
- Some of the unintended side-effects are *positive*, they contribute to necessary value, and to lower costs.
- Some side effects are *negative*, the opposite of desired.

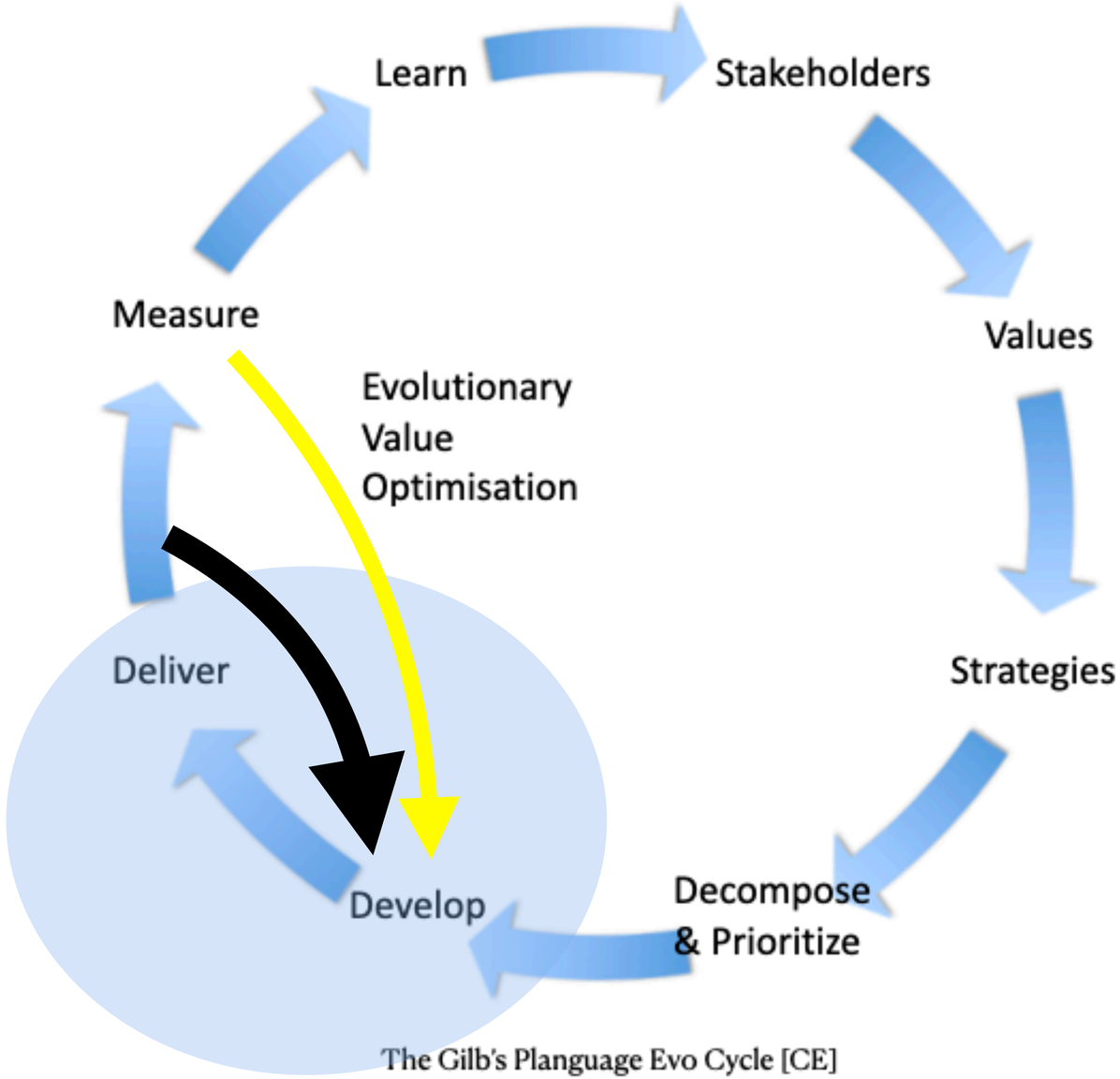


Source [SEA]

15.3 Value (s) Stream

Your startup's Values, are many, and concurrent

- A '**Deliver**' Evo-step can, if you are unlucky, result in some. *'immediate bad surprise'* side-effect,
  - *bad enough* to remove the sub-strategy Step, and to try to get back to normal (15.4 Retreat)
  - The black arrow (Figure 15.3), *before* 'Measure', indicates reaction to an *immediate bad effect* of the Evo-step.
    - Hasty retreat is necessary! No need to go to the 'Measure' step. Failure was **painfully obvious**.
    - Another possible system reaction, is that we do not get a painfully obvious reaction, or if '**Measure**' (the 'Meter') of the Evo-step, is *built-in automatic, immediate*; and the main measure is disappointing; of 'little value'.
    - Or, the primary Step-targeted Value does not meet expectations, and is *not* worth keeping in the system, so we retract that step too. (**Lemon arrow**)
    - The same is true (Lemon-colored arrow) if we consciously progress to the '**Measure**' Evo-step, maybe a Measure-step using human feedback, over several days, and discover , after a while, that the 'main Value' result' is disappointing.
  - We need, then, to find a *better* step. Either a *new* one, or a *modification* to the failed one, which we *hope* solves the problem.
  - Because the Value(s) stream incremental levels, resulting from a single sub-strategy step, are quite difficult to predict, almost anything can happen. Including 'very good' unexpected impacts from side-effect values. Which might result in us keeping the Step. And working on a new or modified step, to deliver the primary critical value, that eluded us first time.



The Gilb's Planguage Evo Cycle [CE]

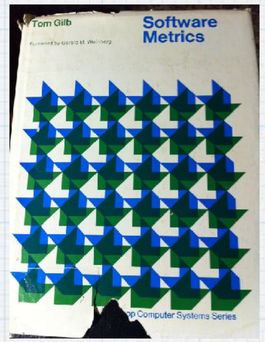
15.4. Agile, defined in 1976, with ‘retreat’ if failure at step

This is same info as at beginning of book

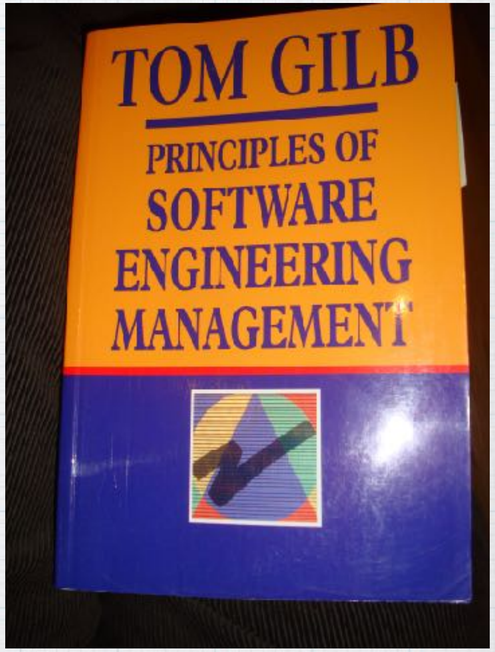
On step-results measurement, and retreat (=‘agile’) possibility

'A complex system will be most successful
if it is implemented in small steps
and if each step has a clear measure of
successful achievement
as well as a "retreat" possibility to a
previous successful step upon failure.'

(p. 214), Software Metrics 1976-7.



1976 (S, EU), 1977 (USA)



See Slide Presenter Note for more detail, or ask me tom@Gilb.com

I was 35 years old when this was published. I had 16 years **‘agile’** (or ‘Evo’, ‘deliver value to stakeholder in small increments, measure value and retreat if necessary’) experience by then (from 1960 Dobloug Case, and UiOslo Publisher/Admin (1968) etc).

Chapter 15, Deeper Perspectives, 15.10.1 page 14 for digital SM (1976) quotes on Evolutionary Delivery https://www.researchgate.net/publication/380874956_Ch_15_Deeper_perspectives_on_Evolutionary_Delivery_later_2001_known_as_Agile_in_Gilb_Principles_of_Software_Engineering_Management

15.5. The Fork Productivity-Stream Model.

A simple visualization of the Productivity Stream [Productivity Stream]

The Fork Productivity-Stream Model.

- A simple visualization of the Productivity Stream. It is also a simple “Supply Chain” .
- A set of Stakeholders (Globe) provide a set of resources (Costs, **red** fork) For a System (F.1, the system’s **Functions**, *what* it *does*).
- These resources *generate* a set of Values (**Green** Fork), which are delivered to a set of *Stakeholders* (Globe).
- These Value-Receiver Stakeholders can utilize the Values, to create resources, or to *be* resources: which they *can* choose to re-invest in another system (F.2).
- The main visual point I am trying to make, in this instance is that
  - (A.) both the Value Set, and the Costs Set are *multiple* (any number).
  - And (B.) the Sets are *different*, and are *tailor-made* for the critical priorities, of the *systems*, and for *their* Stakeholders.

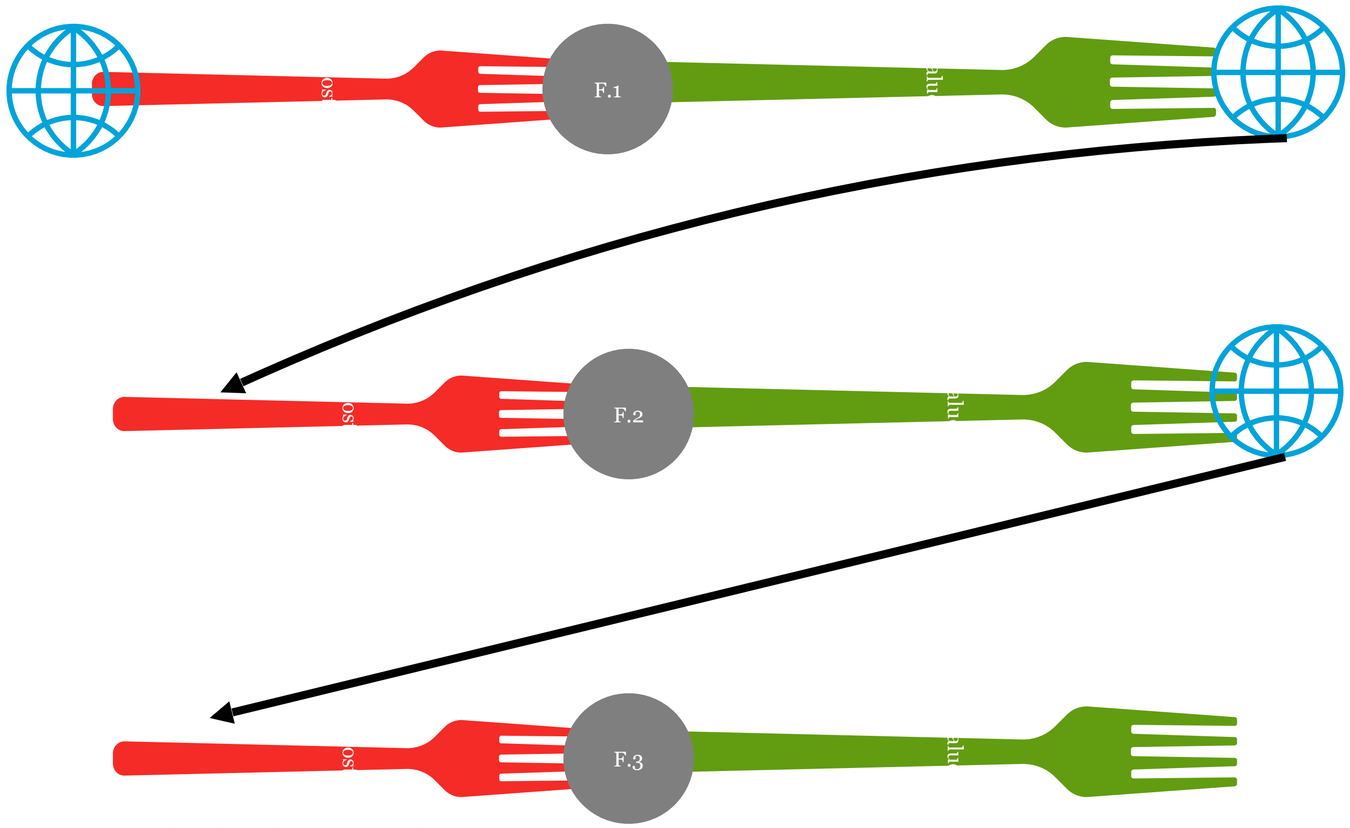


Figure 15.5. **The Fork Productivity-Stream Model.**

15.6 Startup System Productivity

System Productivity is the ratio of the Value Set and The Cost Set. And it is likely to be 'constantly varying', over time.

- **In fact, not only are the individual Values and individual Costs going to vary over time, but the *quantity* of Values and Costs which are being *consciously managed*, will <u>also</u> vary over time.**
- **And the finite 'desired and actual' levels-of-Productivity, will also vary over time (t).**
- **Some of the Values and Costs will be reasonably managed ('good' values, and 'tolerable' costs).**
- **But some Values, and some Costs might be difficult-to-impossible to manage well, or at all.**

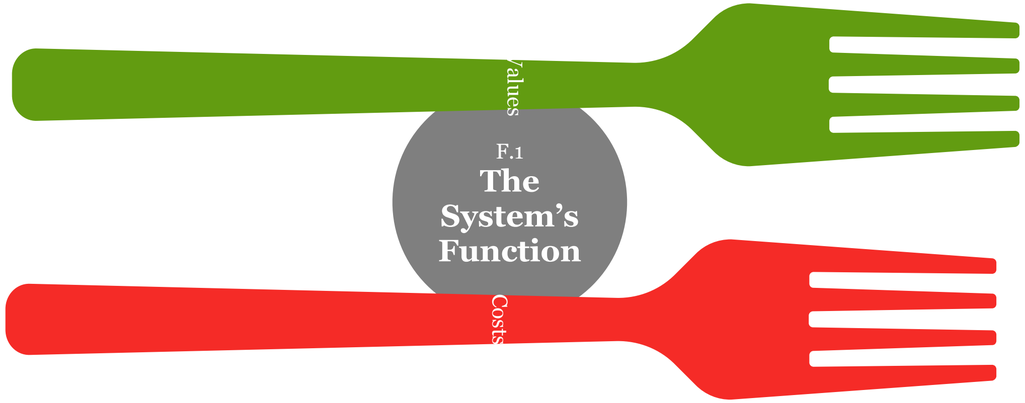


Figure 15.6.A **The Fork Productivity-Ratio Model.**

F.1 Values:F1 Costs [t]
=
F.1 Productivity [t]

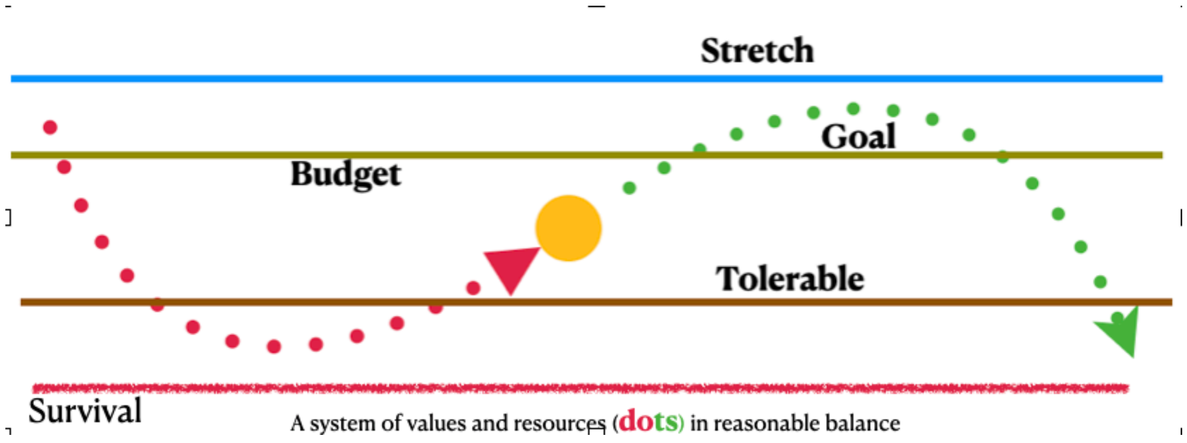


Figure 15.6.B The values and costs constantly varying over time: a static snapshot.

15.7 Efficiency: *'Values: Costs'*, are impacted by choice of 'Design'

The common *poor practices*, that threaten real 'systems efficiency (Productivity)'

1. **ISOLATED VALUES**: Discussing (*presenting, evaluating, approving*) one single value idea (*like: comfort, addictiveness, usability, security*) without simultaneously documenting the *other* critical values in the same set of values, for the project, or improvement process.

2. **FUZZY VALUES**: discussing values with nice words, rather than clear metrics. *For example: 'Market Leading Safety'*

3. **NO DESIGN**: discussing Values, without any well-specified satisfactory design (*strategy, architecture, means*) being *on the table* (in the background, but explicitly present), so that the 'viability of the Value' can be evaluated, in terms of costs and value side-effects.

   - If there is no specific 'design' then it is *logically impossible* to know anything about any values and costs, which *would* be 'known', to some degree, for a *specific* design option.

4. **NARROW 'COST THINKING'**: thinking of costs, only in *financial* terms (not *time*, *people*, *reputation*).

   - And then, thinking only in terms of Capital Costs, but not *recurrent* costs, and not *maintenance* costs, not *implementation* costs, and not *decommissioning* costs.

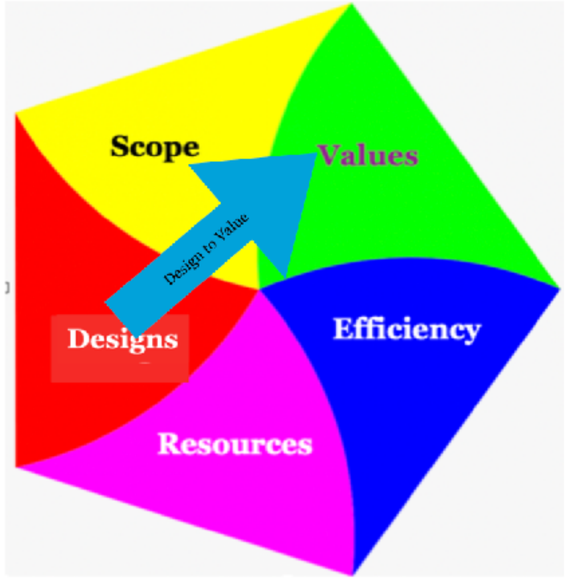


Figure 15.7.1
Designs carry a set of *Values*, *like them or not [Penta]*

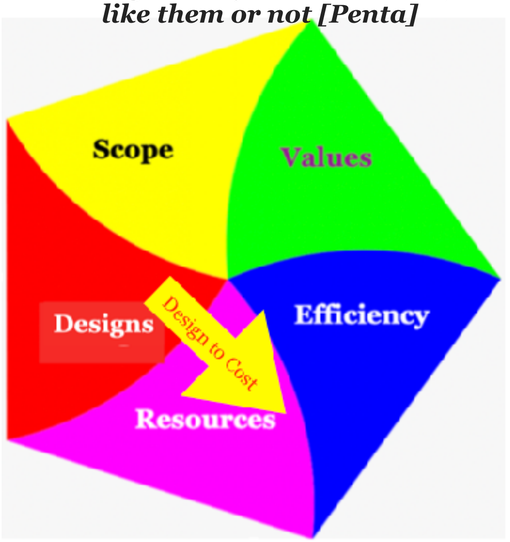
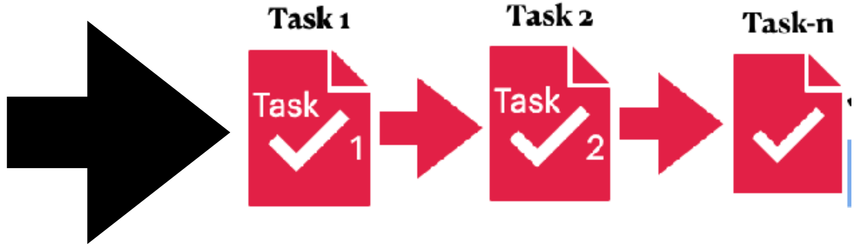


Figure 15.7.
Designs carry a set of *resource costs*, *like them or not*

Strategy 3: FOOD
Efficient
Irrigation
20%



15.8 Develop

Next page (15.9): Copied from earlier Decomposition phase, 12.10.4.3, is a set of Tasks, planned so that the sub-strategy, '**Efficient Irrigation**' can be delivered.

Notice that this Strategy '3' (not the task elements themselves, they were discarded, too big a step) was subsequently decomposed *(my request for re-planning, when I realized AI did not understand my initial request for 'value-delivery' steps)* into smaller sub strategies (see 15.11). So, we are then never going to do this big task, we are going to do the smaller sub-strategy Evo-steps (15.11)

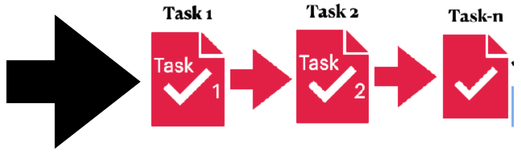
These smaller sub-strategies will each require a 'Develop' plan sort of like this, but a *different* one depending on their nature.

The single tasks, 15.9, do not normally give us *our* planned Values. But, you could say they provide value to one stakeholder, the 'Developer'. But these are not the main point of the startup!

Note that we can use AI to remind us of tasks we probably should do; and might forget, if we do not use our AI 'reminder'.



Strategy 3: FOOD
Efficient Irrigation
20%



15.9. Specify (Strategy 3, FOOD) 'Efficient Irrigation' in Detail *Task Decomposition*, example

You ultimately need this detail, to validate effects and costs, and 'to implement', allocating tasks to people and suppliers.

PROMPT Specify the strategy *Efficient Irrigation*, in a page of practical detail about what we have to do in practice

ChatGPT
Strategy: Efficient Irrigation Systems

Objective:

Implement efficient irrigation systems to maximize water use efficiency and ensure reliable water supply for agriculture in Namibia. The aim is to increase crop yields and conserve water, contributing to the overall goal of food supply reliability.

Steps to Implement Efficient Irrigation Systems:

1. Assessment and Planning:

- Site Assessment: Conduct a thorough assessment of the agricultural land to understand the soil type, topography, crop types, and existing water sources.
- Water Resource Analysis: Analyze the available water resources, including groundwater, surface water, and rainwater harvesting potential.
- Needs Assessment: Determine the specific water needs of different crops and the most suitable irrigation methods for each type.

2. System Design:

- Irrigation Method Selection: Choose the most appropriate irrigation methods based on the site assessment. Common methods include drip irrigation, sprinkler systems, and surface irrigation.
  - Drip Irrigation: Ideal for precise water delivery directly to the root zone, minimizing evaporation and runoff.
  - Sprinkler Systems: Suitable for a wide range of crops and can cover large areas efficiently.
  - Surface Irrigation: Utilizes gravity flow to distribute water, often used for crops requiring high water volumes.
- System Layout: Design the layout of the irrigation system, including the placement of pipes, emitters, sprinklers, and control valves.
- Automation and Control: Incorporate automated systems such as timers, moisture sensors, and flow controllers to optimize water use and reduce labor.

3. Procurement of Materials:

- Pipes and Fittings: PVC or polyethylene pipes, connectors, and fittings for distributing water.
- Emitters and Sprinklers: Drip emitters, micro-sprinklers, or standard sprinklers, depending on the chosen method.
- Pumps and Filters: Pumps to ensure adequate water pressure and filters to prevent clogging of emitters.
- Control Systems: Timers, sensors, and controllers for automation.
- Installation Tools: Tools and equipment for trenching, pipe cutting, and fitting assembly.

4. Installation:

- Site Preparation: Clear the land and prepare trenches for pipe installation.
- Pipe Installation: Lay the main and lateral pipes according to the design layout.
- Emitter/Sprinkler Installation: Install emitters or sprinklers at the designated positions.
- Pump and Filter Setup: Install and connect pumps and filters to the water source.
- Control System Setup: Install and configure the automation and control systems.

5. Testing and Calibration:

- System Testing: Conduct a comprehensive test of the irrigation system to check for leaks, pressure issues, and proper operation of emitters/sprinklers.
- Calibration: Adjust the flow rates and coverage areas to ensure uniform water distribution and optimal performance.

6. Training and Capacity Building:

- Farmer Training: Conduct training sessions for local farmers on the operation and maintenance of the irrigation system.
- Maintenance Procedures: Provide guidelines on routine maintenance tasks such as cleaning filters, checking for leaks, and adjusting emitters.

7. Monitoring and Evaluation:

- Performance Monitoring: Regularly monitor the system's performance, water usage, and crop yields.
- Data Collection: Collect data on water savings, crop health, and productivity to evaluate the system's impact.
- Continuous Improvement: Make necessary adjustments and improvements based on monitoring data and feedback from farmers.

Practical Considerations:

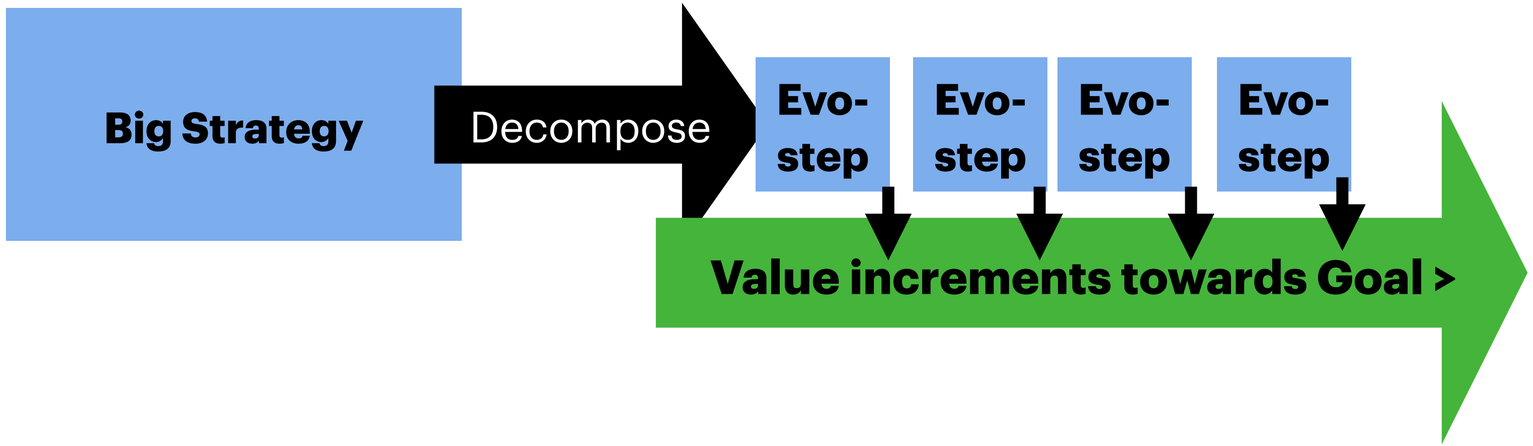
- Cost Management: Ensure budget allocation for both capital expenses (CapEx) and operational expenses (OpEx).
- Local Adaptation: Customize the irrigation system to local conditions, considering factors such as climate, soil, and crop types.
- Sustainability: Focus on sustainable practices, including the use of renewable energy sources (e.g., solar pumps) and water conservation techniques.
- Community Involvement: Engage the local community and stakeholders throughout the process to ensure buy-in and long-term success.

Expected Outcomes:

- Increased Crop Yields: More efficient water use leads to higher and more reliable crop production.
- Water Conservation: Significant reduction in water wastage, ensuring sustainable use of water resources.
- Improved Livelihoods: Enhanced agricultural productivity and profitability for local farmers.
- Scalability: The success of the pilot project in Namibia can serve as a model for other regions with similar conditions.

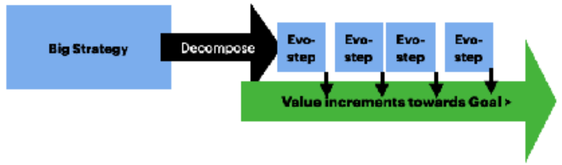
By following these detailed steps, the implementation of efficient irrigation systems can contribute significantly to the goals of food supply reliability and water supply availability, driving the success of the EU-funded startup's mission in remote poor areas.

15.10 This following ‘Step decomposition’ (15.11) is an earlier plan 12.10.4.4, where I asked AI to decompose a larger strategy into smaller Evo-steps.
So now, we need to plan the development process for each one of these steps, and we can ask AI to draft a plan for us. We should, as usual, modify the plan to our real environment.



(Earlier) Figure 4.2.3 B. Decomposing into Evo-steps, not ‘tasks’, enables us to get measurable results, feedback, and planning improvement faster

15.11. (earlier12.10.4.4) Decomposition of (*Strategy 3, FOOD) 'Efficient Irrigation',* into independently implementable sub-strategies: *Evo-Steps*



Instead of one bigger task set, to implement the strategy, previous page (4.3)

Evo-steps ->	Utilizing Weather Forecast Data	Mulching	Soil Moisture Sensors Installation	Regular Maintenance and Inspection	Proper Plant Selection	Drip Irrigation System Installation	Rainwater Harvesting Implementation	Training and Education for Farmers	Smart Irrigation Controllers	Water Recycling and Reuse
Value (1-10)	8	5	8	6	6	9	7	7	8	7
Cost (1-10)	3	2	4	3	3	5	4	4	5	5
Efficiency Ratio	2.67	2.50	2.00	2.00	2.00	1.80	1.75	1.75	1.60	1.40

Here are two reasons for the effectiveness on value and one reason for cost estimation for each of the 10 sub-strategies:

1. Utilizing Weather Forecast Data

- Value Effectiveness:
  1. Enhances irrigation scheduling by predicting rainfall, reducing unnecessary water usage.
  2. Helps optimize water use during dry periods, ensuring plants receive adequate moisture.
- Cost Estimation:
  1. Generally low cost as it primarily involves accessing existing weather data, potentially with some investment in weather stations or data subscriptions.

2. Mulching

- Value Effectiveness:
  1. Reduces water evaporation from soil, maintaining moisture levels.
  2. Suppresses weed growth, which can otherwise compete with crops for water.
- Cost Estimation:
  1. Low cost because it often involves using organic materials like leaves or straw, which may be locally sourced or even waste products.

3. Soil Moisture Sensors Installation

- Value Effectiveness:
  1. Provides precise measurements of soil moisture, allowing for targeted irrigation.
  2. Reduces water waste by ensuring irrigation only occurs when necessary.
- Cost Estimation:
  1. Moderate cost due to the expense of purchasing and installing the sensors, as well as potential maintenance.

4. Regular Maintenance and Inspection

- Value Effectiveness:
  1. Ensures that the irrigation system operates efficiently, reducing water loss due to leaks.
  2. Prevents breakdowns that could lead to overwatering or underwatering.
- Cost Estimation:
  1. Low to moderate cost, involving labor and potential replacement parts.

5. Proper Plant Selection

- Value Effectiveness:
  1. Choosing drought-resistant plants reduces overall water needs.
  2. Selecting plants suited to local climate and soil conditions improves growth and water efficiency.
- Cost Estimation:
  1. Moderate cost, mainly involving the purchase of specific plant varieties.

6. Drip Irrigation System Installation

- Value Effectiveness:
  1. Delivers water directly to the plant roots, minimizing evaporation and runoff.
  2. Provides a consistent moisture level, which can enhance plant health and yield.
- Cost Estimation:
  1. High cost due to the complexity of installing the system and potential for high initial investment in materials.

7. Rainwater Harvesting Implementation

- Value Effectiveness:
  1. Captures and stores rainwater for later use, reducing dependency on external water sources.
  2. Provides a sustainable and environmentally friendly water source.
- Cost Estimation:
  1. Moderate cost, involving the setup of collection systems, storage tanks, and potential filtration systems.

8. Training and Education for Farmers

- Value Effectiveness:
  1. Empowers farmers with knowledge to implement efficient irrigation practices.
  2. Enhances overall farm management, leading to better resource utilization and crop yields.
- Cost Estimation:
  1. Moderate cost, including training materials, workshops, and possibly hiring experts.

9. Smart Irrigation Controllers

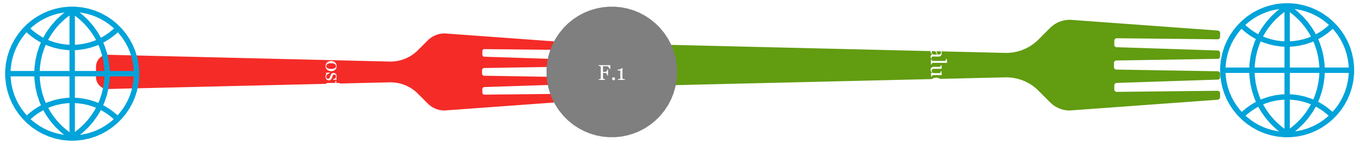
- Value Effectiveness:
  1. Automates irrigation based on real-time data, optimizing water use.
  2. Reduces labor and human error in managing irrigation schedules.
- Cost Estimation:
  1. High cost due to the technology involved and potential ongoing maintenance or software updates.

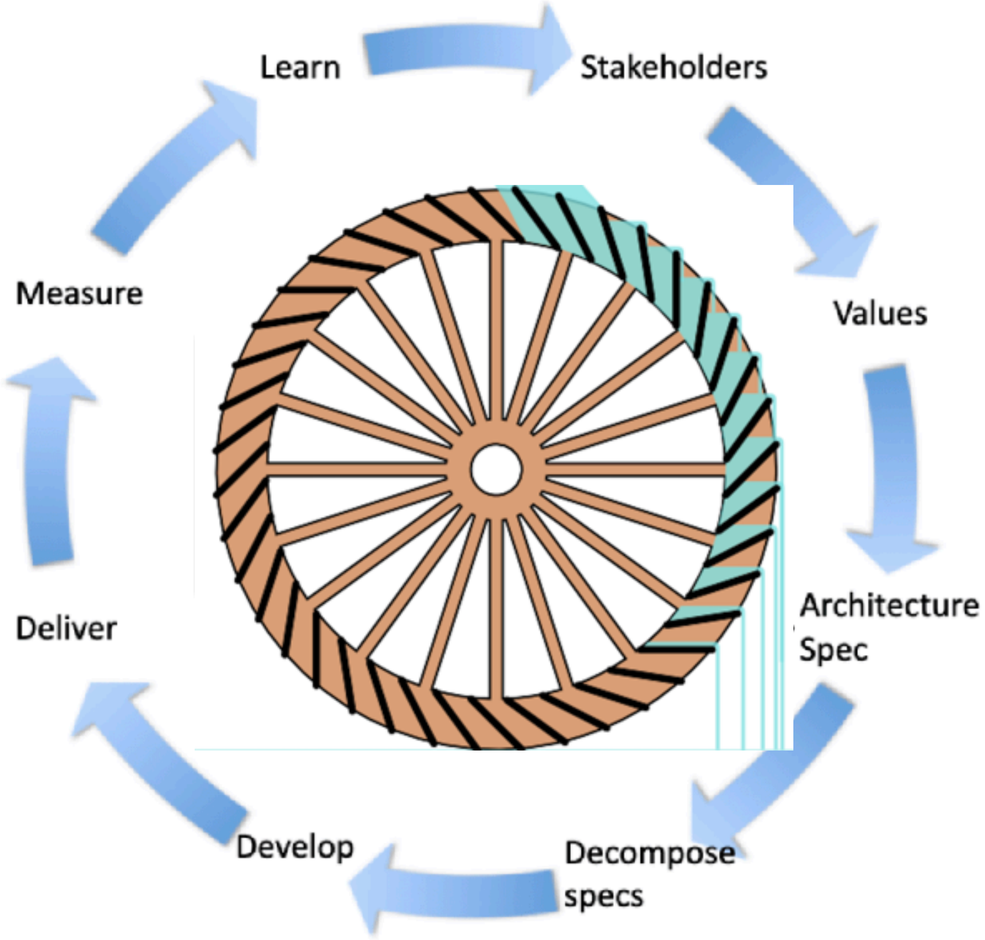
10. Water Recycling and Reuse

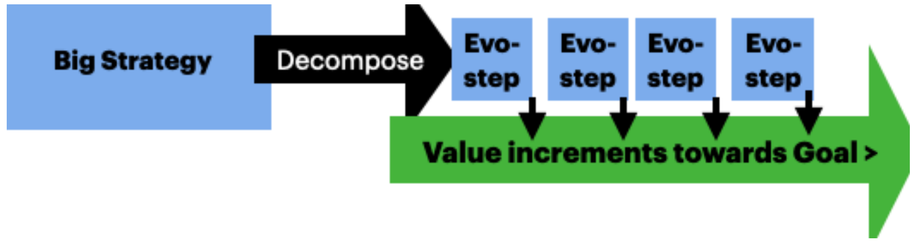
- Value Effectiveness:
  1. Reduces water consumption by reusing treated wastewater for irrigation.
  2. Provides a reliable water source, especially in areas with limited water availability.
- Cost Estimation:
  1. High cost due to the infrastructure needed for treatment and distribution systems.

15.12 The Mill Wheel of the Value Streams, and Cost Streams

The mill wheel 'energy and motion' moves *values*, and consumes *costs*







15.13 Bent's 'Iron Law of Megaprojects'

Bent Flyvbjerg



"Over time, over budget, under benefits, over and over again. (2017)

* He further argues that in order to matter the social sciences must inform practical reason instead of abstract rationality. This is best done by a focus on values and power in actual human decision making, according to Flyvbjerg.

* Flyvbjerg shows that a root cause is that competition between megaprojects and their sponsors creates biases rooted in political and organizational pressures that lead to the consistent overestimating of project benefits and the underestimating of project costs.

* The best megaprojects do not get implemented, but rather the ones that look best on paper.

* Flyvbjerg argues that the ones that look best on paper are the ones for which costs and benefits have been misrepresented the most, either deliberately through strategic misrepresentation (political bias), or non-deliberately through optimism bias (psychological bias) or, typically, through a combination of both.

* He identifies three antidotes to the Iron Law:

  * (a) Realistic planning, including de-biasing of all cost, schedule, and benefit estimates;

  * (b) High-quality delivery teams; and

  * (c) Governance structures with incentives for realistic plans and for delivering those plans to time, budget, and benefits; and with ***early-warning-sign systems to immediately capture and act on things that go wrong,*** which they invariably do in megaprojects, due to their sheer size and complexity, according to Flyvbjerg

"..early-warning-sign systems to immediately capture and act on things that go wrong,#



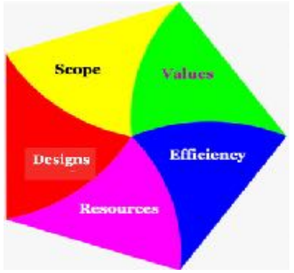
* from wikipedia https://en.wikipedia.org/wiki/Bent_Flyvbjerg

* https://hbr.org/2023/01/how-frank-gehry-delivers-on-time-and-on-budget

* HIS PROJECT DATABASE

(16,000 Projects) Only 8.5% of them were delivered on time and on budget, while a nearly invisible 0.5% of projects were completed on time and on budget and produced the expected benefits. To put that more bluntly, 99.5% of large projects failed to deliver on time, under budget, and expected benefits.





Tom's Note:
It is about
'efficiency'

15.14 *It's the Benefits, Stupid!*

Bent Flyvbjerg



·A project leader pointed out that project teams and owners focus too little on the benefits of their projects and too much on costs and schedule.

·It's not that cost and schedule are not important, emphasized this leader.

·But the ultimate reason for doing projects is their benefits.

·Cost and schedule are means to an end -- the end being benefits -- not ends in themselves.

·We must therefore keep our eyes on the benefits, or we lose sight of why we do what we do, the leader concluded. --

·Again, the cohort was sympathetic.

·And again, our research supports the heuristic. (I assume the 16,000 project database, TG)

·First, we have found that most projects don't even measure benefits, making their study difficult.

·Second, project managers who do measure and manage benefits perform better than managers who do not.

·Not only do these managers perform better in delivering benefits, but also in delivering on budget and on time.

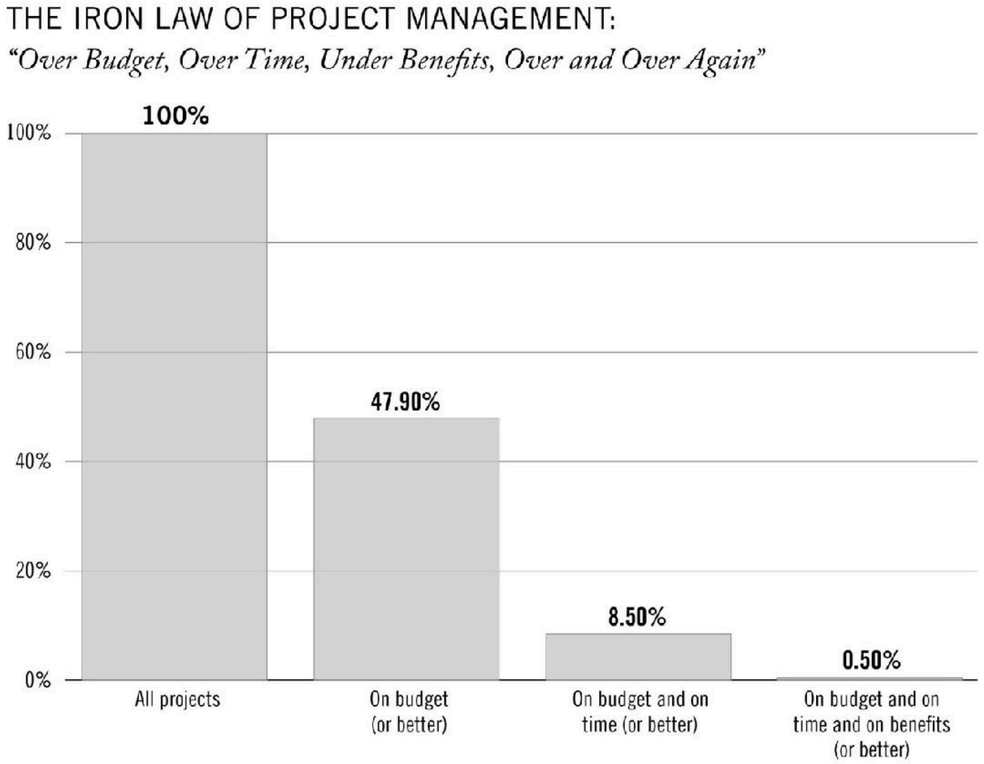
·It appears that once project managers know how to get benefits right, they know how to get everything right.

·They have graduated to the level of the mature and effective project leader.

[Value First]

·Therefore, if you don't already focus on benefits in your projects, now is a good time to start.

·You will not truly master project management until you do.



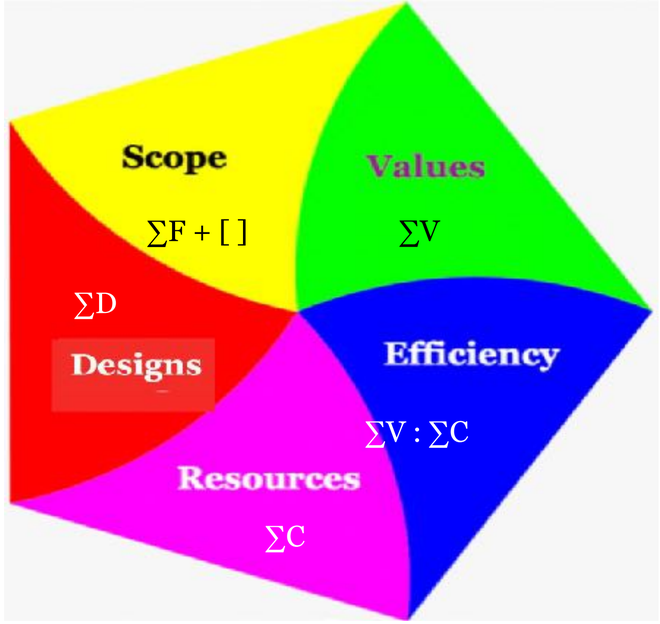
16,000 project base, (Flyvbjerg)



16. Income Stream: managing costs

Based initially on [Cost Engineering] book 2023-4

16.1 LIMITS TO COST REDUCTION



The Penta is a high level visualization to remind us of 5 critical and related system elements. Designs, necessary to give values; also give Value side-effects, and a set of costs.
But the designs must not give bad efficiency, nor give negative profitability, nor give a bad ∑V : ∑C. Ratio.

Note: ∑R is the total set of allocated or budgeted Resources. While ∑C is the set of costs those resources, which are incurred by design or implementation, at any given time (t)

The Penta Model [Penta]

As hinted earlier, we cannot simply drive costs down. There are cost-reduction limits we need to respect. Cost reduction is, generally speaking, *secondary* to 'stakeholder value delivery'. To put it simply, we need, above all, to deliver critical stakeholder value levels: but of course in reasonable time (not 2 minutes) and at a reasonable cost (not $2). There is something called delivering value 'too early', before stakeholders can appreciate it. There is something called 'reducing costs too much', when the cost of doing so, the cost-reduction-development process, exceeds the cost reduction value itself, of a product or service.

16.2. Level 2 Penta
∑C(the set of costs)

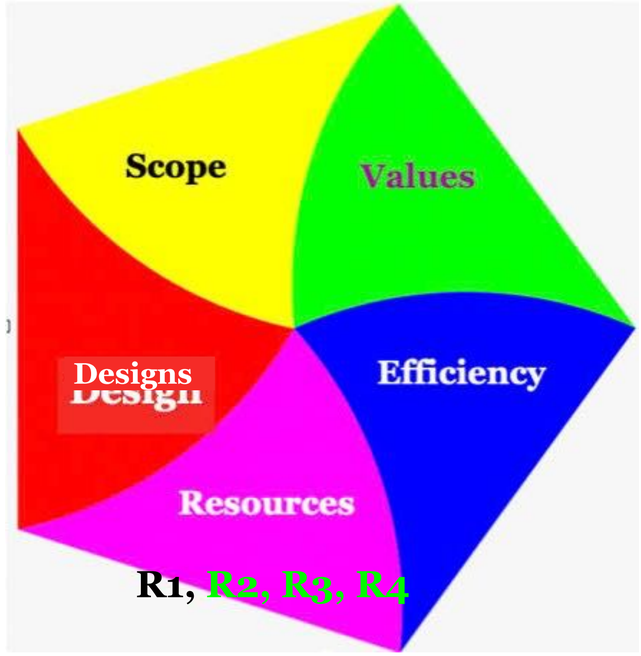
R1: **Financial** Resources. Capital Expenditure, Operating Expenditure, De-commissioning Expenditure

R2. **Human** Resources: Manual labour, Intellectual Labour, Training work.

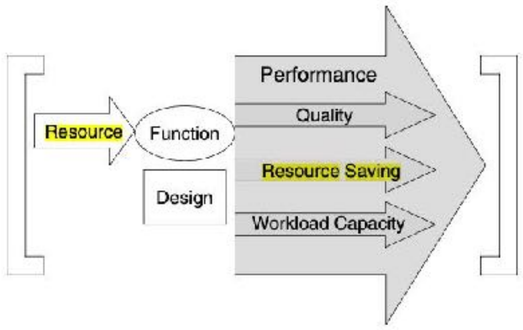
R3: **Time** Resources: *calendar time* to complete, *cycle time* for a process.

R4: **Other Resources**, l*ike space, sustainability, trust, credibility, loyalty.*

The examples give, hint at a third level Penta model R1.1, R1.2, etc)



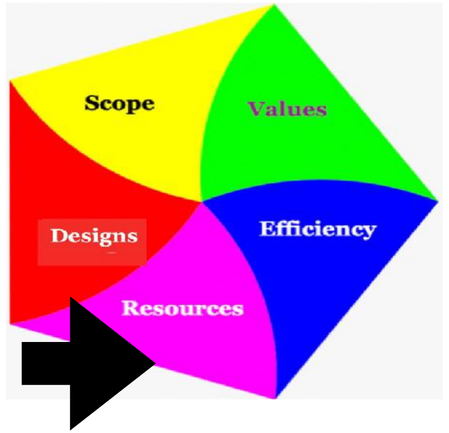
Resources are various limited fuels to create and maintain a system.



Source: [CE] Fig. G12. Notice the 'Resources' feeding the Function/ Design/Values

16.3. Exploring Limits to Costs.

SOME COST MANAGEMENT LOGICAL RULES.
NATURE IS NOT SIMPLE!



1. **COST-REDUCTION DESIGN: We can drive down costs in the direction of future, required and specified, cost levels (Constraint levels, Target levels).**

2. **DESIGNS CAN REDUCE COSTS: But, every cost-reduction, must, logically, use a *specific Design* (or a set of designs)**

3. **SIDE-EFFECTS: Designs, like it or not, by their nature, come with *more attributes*, than the specific cost reduction, for which we have chosen a design, to deliver to us.**

4. **VALUE AND COST EFFECTS: The designs come with side-effects, good and bad, in both Values, and other types of Costs, than our primary Cost-reduction target.**

5. **KNOWN UNKNOWNS: The degree of side-effects is initially unknown, and unknowable (until you analyze a specific design), but it is then *estimable* and *measurable*. The only way to change undesired side-effects, is to change the *design itself*.**

6. **FUTURE UNKNOWNS: The Design side-effects, for both Values and Costs, interact (they change system-level attributes) with the current existing set of implemented designs. But *future* incremental design changes, can modify the earlier side-effect results, in *unpredictable*, but ultimately, when implemented, *measurable* ways.**

7. **UNACCEPTABLE SIDE-EFFECTS: We cannot use, accept, *any design,* to drive down the cost, which we are trying to reduce, if that *design increment*, results in *unacceptable* levels, of any Values or Costs.**

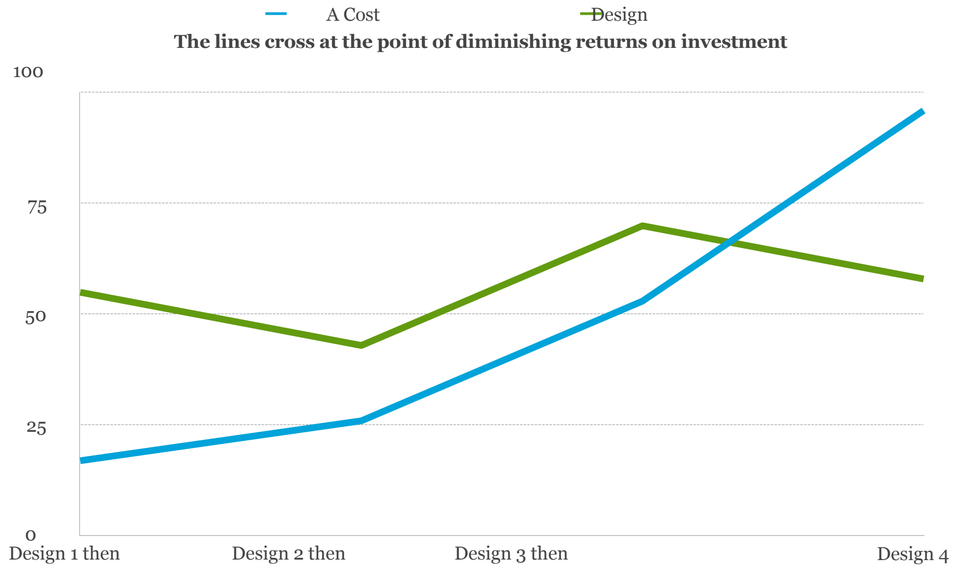
8. **COST VIOLATION: A design is 'unacceptable' if any of its costs, violate a Constraint level (e.g., a cost goes above a 'Tolerable' level). Designs that do not allow us to stay within a Budget level, when other necessary designs are added to reach Value Targets, are also unacceptable.**

9. **VALUE THREAT: 'Cost-reduction designs' that result in preventing critical values from reaching minimum Constraint Levels, are not acceptable. Furthermore, if these designs move any Value below our 'Target levels', they are also unacceptable.**

10. **IF THE COST IS OK: A design is acceptable *as long as* the 'costs we are trying to reduce' are within Budget and/or within any Constraint levels; and that none of the *Value* requirements are *unacceptably modified* as a result.**

16.4. Some remarks on previous page,

SOME ‘COST MANAGEMENT’ LOGICAL RULES



A conceptual incremental series of designs for cost reduction.

The ‘Exploring Limits to Costs’ Rules, page above, is ’10 deep insights into cost management'. We can, and do, simplify this system development process, but then we increase the risks of unwanted costs, and disappointing values, later. We can ignore all of these Rules, and experience the problems we then incur, at *later* stages. But then - the consequential cost damage is, with risks exploding, about 10X and 100X or more, downstream. This is also the nature of the complex systems we have, and of scaling-up to large systems. I believe these rules are logically true in a multi-attribute world we actually live in.

But I just drafted them based on my common sense observation. At least we might consider learning about these rule’s existence. Read them slowly, and think logically. Maybe we only need to apply them, with analysis of ‘big decisions’. We can certainly model each incremental design fully, using Impact Estimation tables, to make these decisions clear to ourselves, and to others. It is called ‘Cost *Engineering*’ for a reason. Some people prefer Wild West, but they are the 99.5% that fail one way or another [Flyvbjerg]. Maybe you would like to avoid failure? Or, just not get blamed for it?

16.5. Another reason why we need to stop or start - a process of cost reduction

1. If a cost requirement specification says *the Budget level is 'X'*, then there is no objective reason to optimize costs, or to reduce costs, when the Budget level is *not* exceeded.

2. It is probably possible to continue improving designs, and reducing selected costs, but someone (a documented responsible specification Owner, with a Rationale for the Budget levels, as Musk prefers) thought they had a good reason for the Budget level. The specified level is declared *acceptable*. Leave it alone, and focus your attention on the 'always' Main Event, **improving Value levels** (to *their* specifications: Targets, and Constraints).
3. We have two additional Planguage [CE] devices to try to regulate cost-reduction activity.

    1. The Safety Margin.

    2. The Cost Constraint level.

4. **The Safety Margin: A** well-known engineering practice, is a **level** that your project or organization can stipulate, as needed, for different classes of project, with regard to risk acceptability, size, complexity, and legal requirements. A conservative generic rule, if you cannot think of a better one, is to only design, so that you reach all your Target levels of Values, while using *only half* of the Budgeted resources. This 50% -of-budget level is automatically the default in the ValPlan app [ValPlan] Impact Estimation table, and warnings are displayed when you use more than 50% of Budget in your set of designs. You can also do this on IET spreadsheets, or just keep track of it as informal notes. Experience over time, will inform you, whether you need different Safety Margins for Costs. Hint, many people experience, that 'actual costs' *exceed* 'planned costs' by 3.14X, even though 'everybody' said this project *'would be a piece of cake', it* turned out to be 'pi' (π). (Tom's Dad Joke)

5. **Cost Constraints**. A specific Cost Constraint level can be specified in the form of a 'Tolerable level' (or other Planguage scalar constraint [2]). For example: '**Capex**: **Budget** 1 Million, **Tolerable**: 2 million', is a specification that means *'we expect and hope to not pay more than 1 million'. 'But we can tolerate spending 2 million'*. But after that, we have a problem: like, *run out of money (no more budget, unprofitable, illegal);* and design-steps should be taken, to reduce expenditure, until we are within these limits. Hopefully you have not irretrievably spent over '2 million', before you discover this problem. Use an IE Table, for early warning signals, about this problem: backed up by incremental measurement, as designs are applied to the real system.

16.6. An analysis of conventional cost-management ideas

Much of the specific methods commented on below, are copied from my Value Requirements book Chapter 15 [VR]

16.7. Conventional Cost Management Methods

The main purpose of *the* book [Cost Eng] is to introduce more-advanced multi-attribute ideas, for cost management. So in order to clarify this even more, I feel the need to 'analyze' and criticize, conventional cost-management ideas, and *related ideas* such as prioritization, Value, and Architecture methods, so the reader can better understand the 'method-weaknesses', which we are trying to address in this book. Here are the main points.

Conventional cost management lacks:

1. QUALITIES AND VALUES (as numeric engineering-specifications, *not* as the usual, nice platitudes)
2. ENGINEERING (digitization, modelling, logic, numeracy, experiment)
3. MULTI-DIMENSIONAL THINKING (keeping more-than-one critical factor in mind, at a time)
4. TRADEOFFS (intelligent (data driven) and dynamic prioritization)
5. MULTIPLE COST TYPES AND DIMENSIONS (it is not all about *money* and CapEx)
6. SPECIFICATION CLARITY - AS BASIS FOR LOGIC (fuzzy ideas lead to bad results)
7. A SYSTEMS ENGINEERING VIEW (as opposed to 'coding', or 'finance' views)
8. A STAKEHOLDER VIEW [SE], (instead of narrower *user, customer, shareholder* view).

16.8. Well-known Cost Management Methods

Here are some of the ‘well-known’ cost management methods’ I am referring to

1. Financial Accounting
2. Cost and Time Estimation
3. Balanced Scorecard (Harvard Business School)
4. Quality Function Deployment (Toyota, Lean)
5. Prioritization (using fixed weights, subjective judgement, BSC, QFD)
6. Requirement Specification, (User Story, Design as requirement)
7. Enterprise Systems Architecture (TOGAF etc)
8. Iron Triangle ideas



The Cost Management Process

16.9. Well-Known Cost Management Methods : with a *hint of their weaknesses*

Here are some of the typical weaknesses we suffer from these 'popular methods'

1. Financial Accounting: *pure backwards-in-time financial data, Annual & Quarterly Cycles*

2. Cost and Time Estimation: *Estimates <u>not</u> based on quantified qualities, but on <u>volume</u>.*

3. Balanced Scorecard: *Failure to quantify some values and qualities, failure to give evidence.*

4. Quality Function Deployment: *failure to define many value-terms clearly, failure to give basis for estimates.*

5. Prioritization: *static decisions for sequencing, rather than dynamic step-by-step. Not based on Values, and on a broad set of stakeholders.*

6. Requirement Specification, example a User Story.See 16.10. *Total lack of quantification of Values. Inclusion of 'solutions in the requirement' itself, <u>prematurely</u>. Unsuitable for large-scale systems.*

7. Enterprise Systems Architecture (TOGAF etc): T*otal lack of economic thinking. Total lack of quantified quality, value specification, and clarification. Lack of <u>efficiency</u> ('values-for-costs') <u>thinking</u>.*

8. Iron Triangle: *complete falsehoods, ignoring the role, the powerful potential, of a multi-attribute design. [Penta]. See 16.11*



The Cost Management Process

16.9.2 SOME CRITICAL ASPECTS OF COST MANAGEMENT

How do methods here [CE] compare on 10 'Cost Management Dimensions'/Qualities ?

To define the ten most critical stakeholder value objectives for cost management using Planguage, we can identify objectives that align with common cost management goals.

Planguage is a planning language that uses specific parameters such as Ambition, Scale, Meter, Tolerable, Goal, and Stakeholders. Here are the objectives:

1. Budget Adherence
   - Ambition: Complete the project within the approved budget.
   - Scale: Project cost.
   - Meter: Cost variance analysis.
   - Tolerable: 5% over budget.
   - Goal: 0% over budget.
   - Stakeholders: Project sponsors, financial managers.
2. Resource Optimization
   - Ambition: Optimize resource allocation and utilization.
   - Scale: Resource usage efficiency.
   - Meter: Resource leveling index.
   - Tolerable: 10% inefficiency.
   - Goal: 0% inefficiency.
   - Stakeholders: Project managers, resource planners.
3. Cost Control
   - Ambition: Maintain control over project costs.
   - Scale: Cost control effectiveness.
   - Meter: Cost performance index (CPI).
   - Tolerable: CPI ≥ 0.9.
   - Goal: CPI = 1.0.
   - Stakeholders: Project controllers, auditors.
4. Cost-Benefit Balance
   - Ambition: Balance cost against benefits.
   - Scale: Net benefit.
   - Meter: Cost-benefit ratio.
   - Tolerable: Ratio ≥ 1.0.
   - Goal: Ratio ≥ 1.5.
   - Stakeholders: Project sponsors, business analysts.
5. Cost Avoidance
   - Ambition: Avoid unnecessary costs.
   - Scale: Cost avoidance.
   - Meter: Cost avoidance index.
   - Tolerable: 5% unnecessary costs.
   - Goal: 0% unnecessary costs.
   - Stakeholders: Project managers, quality assurance.
6. Risk Mitigation
   - Ambition: Mitigate financial risks.
   - Scale: Risk exposure.
   - Meter: Risk mitigation index.
   - Tolerable: 10% risk exposure.
   - Goal: 0% risk exposure.
   - Stakeholders: Risk managers, project sponsors.
7. Stakeholder Confidence
   - Ambition: Build stakeholder confidence.
   - Scale: Confidence level.
   - Meter: Stakeholder satisfaction survey.
   - Tolerable: 80% satisfaction.
   - Goal: 95% satisfaction.
   - Stakeholders: Clients, investors.
8. Competitive Advantage
   - Ambition: Achieve cost-effective project delivery.
   - Scale: Competitive edge.
   - Meter: Market competitiveness index.
   - Tolerable: 10% less competitive.
   - Goal: 0% less competitive.
   - Stakeholders: Business strategists, marketing.
9. Quality Assurance
   - Ambition: Maintain quality standards within budget.
   - Scale: Quality compliance.
   - Meter: Quality index.
   - Tolerable: 5% quality deviation.
   - Goal: 0% quality deviation.
   - Stakeholders: Quality managers, clients.
10. Timely Reporting
    - Ambition: Provide timely cost reports.
    - Scale: Reporting timeliness.
    - Meter: Reporting delay.
    - Tolerable: 10% delay.
    - Goal: 0% delay.
    - Stakeholders: Project managers, financial analysts.



The Cost Management Process

Citations:

[1] https://fastercapital.com/content/Cost-Management-Objectives--The-Goals-and-Outcomes-of-Cost-Management.html

[2] https://www.hadroninternational.com/insights/cost-management-what-role-do-stakeholders-play

[3] https://fastercapital.com/content/Cost-Management-Stakeholders--How-to-Engage-and-Manage-Your-Cost-Management-Stakeholders-and-Expectations.html

[4] https://www.linkedin.com/advice/1/how-do-you-deal-stakeholder-resistance-dissatisfaction

[5] https://bakkah.com/knowledge-center/project-cost-management

16.9.3 Comparison Planguage: Conventional Cost management

So it seems a side-app of Planguage is not only ‘risk management’ but also ‘cost management’

To define the ten most critical stakeholder value objectives for cost management using Planguage, we can identify objectives that align with common cost management goals.

Planguage is a planning language that uses specific parameters such as Ambition, Scale, Meter, Tolerable, Goal, and Stakeholders. Here are the objectives:

1. Budget Adherence
   - Ambition: Complete the project within the approved budget.
   - Scale: Project cost.
   - Meter: Cost variance analysis.
   - Tolerable: 5% over budget.
   - Goal: 0% over budget.
   - Stakeholders: Project sponsors, financial managers.
2. Resource Optimization
   - Ambition: Optimize resource allocation and utilization.
   - Scale: Resource usage efficiency.
   - Meter: Resource leveling index.
   - Tolerable: 10% inefficiency.
   - Goal: 0% inefficiency.
   - Stakeholders: Project managers, resource planners.
3. Cost Control
   - Ambition: Maintain control over project costs.
   - Scale: Cost control effectiveness.
   - Meter: Cost performance index (CPI).
   - Tolerable: CPI ≥ 0.9.
   - Goal: CPI = 1.0.
   - Stakeholders: Project controllers, auditors.
4. Cost-Benefit Balance
   - Ambition: Balance cost against benefits.
   - Scale: Net benefit.
   - Meter: Cost-benefit ratio.
   - Tolerable: Ratio ≥ 1.0.
   - Goal: Ratio ≥ 1.5.
   - Stakeholders: Project sponsors, business analysts.
5. Cost Avoidance
   - Ambition: Avoid unnecessary costs.
   - Scale: Cost avoidance.
   - Meter: Cost avoidance index.
   - Tolerable: 5% unnecessary costs.
   - Goal: 0% unnecessary costs.
   - Stakeholders: Project managers, quality assurance.
6. Risk Mitigation
   - Ambition: Mitigate financial risks.
   - Scale: Risk exposure.
   - Meter: Risk mitigation index.
   - Tolerable: 10% risk exposure.
   - Goal: 0% risk exposure.
   - Stakeholders: Risk managers, project sponsors.
7. Stakeholder Confidence
   - Ambition: Build stakeholder confidence.
   - Scale: Confidence level.
   - Meter: Stakeholder satisfaction survey.
   - Tolerable: 80% satisfaction.
   - Goal: 95% satisfaction.
   - Stakeholders: Clients, investors.
8. Competitive Advantage
   - Ambition: Achieve cost-effective project delivery.
   - Scale: Competitive edge.
   - Meter: Market competitiveness index.
   - Tolerable: 10% less competitive.
   - Goal: 0% less competitive.
   - Stakeholders: Business strategists, marketing.
9. Quality Assurance
   - Ambition: Maintain quality standards within budget.
   - Scale: Quality compliance.
   - Meter: Quality index.
   - Tolerable: 5% quality deviation.
   - Goal: 0% quality deviation.
   - Stakeholders: Quality managers, clients.
10. Timely Reporting
    - Ambition: Provide timely cost reports.
    - Scale: Reporting timeliness.
    - Meter: Reporting delay.
    - Tolerable: 10% delay.
    - Goal: 0% delay.
    - Stakeholders: Project managers, financial analysts.

Objective	Gilb Planguage Strategy	Conventional Cost Management
Budget Adherence	90 %	70 %
Resource Optimization	85 %	65 %
Cost Control	95 %	75 %
Cost-Benefit Balance	80 %	60 %
Cost Avoidance	88 %	68 %
Risk Mitigation	92 %	72 %
Stakeholder Confidence	90 %	70 %
Competitive Advantage	87 %	67 %
Quality Assurance	93 %	73 %
Timely Reporting	89 %	69 %
Total Effectiveness	889 %	689 %



16.9.4 Justification for the Cost Management methods ratings (16.9.3)

Certainly! Here are two reasons for each score assigned to the Gilb Planguage strategy and Conventional Cost Management in the Impact Estimation Table (IET):

Budget Adherence

- Gilb Planguage Strategy (90%)
  1. Clear Objectives: Planguage provides precise definitions and metrics, allowing for better tracking and adherence to budget constraints.
  2. Continuous Feedback: Regular feedback loops help identify budget deviations early, enabling corrective actions.
- Conventional Cost Management (70%)
  1. Traditional Methods: Often relies on historical data and less dynamic methods, which may not adapt quickly to changes.
  2. Limited Flexibility: Less focus on real-time adjustments can lead to budget overruns.

Resource Optimization

- Gilb Planguage Strategy (85%)
  3. Detailed Planning: Emphasizes detailed resource allocation plans, improving efficiency.
  4. Dynamic Adjustments: Allows for ongoing adjustments based on real-time data.
- Conventional Cost Management (65%)
  1. Static Allocation: Often uses static resource allocation, leading to inefficiencies.
  2. Reactive Approach: Tends to react to problems rather than proactively optimize resources.

Cost Control

- Gilb Planguage Strategy (95%)
  3. Precision in Metrics: Uses precise metrics to monitor cost performance closely.
  4. Proactive Management: Encourages proactive cost management through continuous assessment.
- Conventional Cost Management (75%)
  1. Periodic Reviews: Relies on periodic reviews, which may miss interim cost issues.
  2. Lagging Indicators: Often uses lagging indicators, delaying corrective actions.

Cost-Benefit Balance

- Gilb Planguage Strategy (80%)
  3. Holistic View: Considers both costs and benefits in decision-making, ensuring balanced outcomes.
  4. Value Focus: Prioritizes value delivery over mere cost-cutting.
- Conventional Cost Management (60%)
  1. Cost-Centric: Focuses primarily on cost reduction, sometimes at the expense of benefits.
  2. Limited Benefit Analysis: Often lacks thorough benefit analysis, leading to imbalances.

Cost Avoidance

- Gilb Planguage Strategy (88%)
  3. Preventive Measures: Emphasizes preventive measures to avoid unnecessary costs.
  4. Risk-Based Approach: Uses risk assessment to identify potential cost drivers early.
- Conventional Cost Management (68%)
  1. Reactive Nature: Tends to address costs after they occur rather than preventing them.
  2. Limited Forecasting: Often lacks sophisticated forecasting tools to anticipate costs.

Risk Mitigation

- Gilb Planguage Strategy (92%)
  3. Comprehensive Risk Analysis: Incorporates detailed risk analysis to mitigate financial risks.
  4. Adaptive Strategies: Allows for adaptive strategies to manage emerging risks.
- Conventional Cost Management (72%)
  1. Standard Risk Protocols: Uses standard risk protocols that may not cover all contingencies.
  2. Delayed Response: Often has a delayed response to risk, increasing exposure.

Stakeholder Confidence

- Gilb Planguage Strategy (90%)
  3. Transparent Communication: Promotes transparency and clear communication with stakeholders.
  4. Stakeholder Involvement: Actively involves stakeholders in planning and monitoring.
- Conventional Cost Management (70%)
  1. Periodic Updates: Provides periodic updates, which may not fully engage stakeholders.
  2. Limited Interaction: Often limits stakeholder interaction to formal reviews.

Competitive Advantage

- Gilb Planguage Strategy (87%)
  3. Innovative Practices: Encourages innovative practices that enhance competitive positioning.
  4. Value-Oriented: Focuses on delivering value, which strengthens competitive advantage.
- Conventional Cost Management (67%)
  1. Conventional Methods: Relies on conventional methods that may not differentiate from competitors.
  2. Cost Focused: Primarily focuses on cost, potentially neglecting competitive factors.

Quality Assurance

- Gilb Planguage Strategy (93%)
  3. Integrated Quality Metrics: Integrates quality metrics into cost management, ensuring standards are met.
  4. Continuous Improvement: Encourages continuous improvement in quality processes.
- Conventional Cost Management (73%)
  1. Quality Checks: Conducts quality checks, but may treat them as separate from cost management.
  2. Reactive Quality Control: Often reacts to quality issues rather than preventing them.

Timely Reporting

- Gilb Planguage Strategy (89%)
  3. Real-Time Data: Utilizes real-time data for timely and accurate reporting.
  4. Automated Processes: Incorporates automated processes to reduce reporting delays.
- Conventional Cost Management (69%)
  1. Periodic Reporting: Relies on periodic reporting cycles, which can introduce delays.
  2. Manual Processes: Often uses manual processes that are prone to errors and delays.

These reasons highlight the strengths and limitations of each strategy in achieving the defined stakeholder value objectives

Objective	Gilb Planguage Strategy	Conventional Cost Management
Budget Adherence	90 %	70 %
Resource Optimization	85 %	65 %
Cost Control	95 %	75 %
Cost-Benefit Balance	80 %	60 %
Cost Avoidance	88 %	68 %
Risk Mitigation	92 %	72 %
Stakeholder Confidence	90 %	70 %
Competitive Advantage	87 %	67 %
Quality Assurance	93 %	73 %
Timely Reporting	89 %	69 %
Total Effectiveness	889 %	689 %



16.10 User Stories [User Stories]

User Stories are at the level of a Planguage 'Ambition Level' (aka Management 'BS'], and we can use User Stories to *start* the process of deeper understanding of the implied Value requirement. But User Stories do not pretend to go into enough depth - themselves.

My good friend Mike Cohn (Mr. User Stories) specifically referred to our Planguage methods, when asked on his website what to do about qualities and quantification.

I like the fact that the User Story does not merely have a 'requirement' idea, but that it specifically includes information about the 'stakeholder' (ok, 'User' only), and the justification ('because')

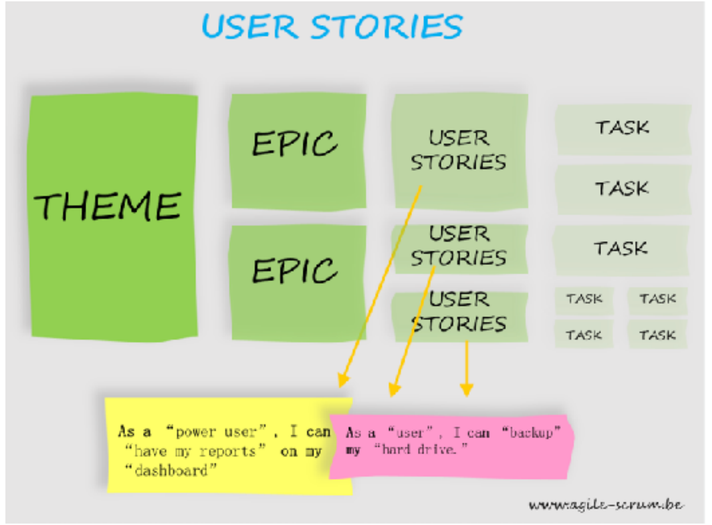
As a method for 'generating ideas about requirements', and possible values and qualities, for small and less-critical systems (no state-of-the-art competition levels, no huge national health systems) - user stories are quite OK.

My problem is, that I see user stories being used way beyond their 'level of competence', and I think user stories, as a 'primary requirements culture', are probably one initial cause of project failure, for large systems.

Success [Success] and failure are not defined by user stories; user stories are more of a detail. But as we have pointed out earlier, the Value level 'Tolerable level' defines a failure border, and Goal level defines success [CE].

User stories just do not deal clearly with values and qualities (a 'buzzword' is not 'clearly'), so we need something more; operating at at a higher level of *controlling the system stakeholder results, values, qualities and costs .*

My advice, if you are committed to using user stories, is to use them as an Ambition Level, a simplified departure point, and then analyze what the real, but implied-only, 'value level' has to be (derive a Scale and a Wish for them, for example). Do more work to clarify them [Quanteer], and make them ready for an engineering process.



16.11. Iron Triangles (IΔ)

Why the Iron Triangle is a misleading '*cost* teaching'.

1. The Iron Triangle teaching (IΔ) does not contain any ideas about *multiple* Qualities, and *multiple* costs.
2. IΔ does not contain any ideas about quantification of the *quality* ideas. It is likely to be a one-dimensional concept of 'quality', and may be (implied, not defined), something like 'product defects', even that - badly defined.

Figure 1 – The Iron Triangle of Project Management



The widely spread falsehood that you have to trade these off in some zero-sum environment, does not correspond to an engineering reality. It ignores the nature of the many attributes of any design.

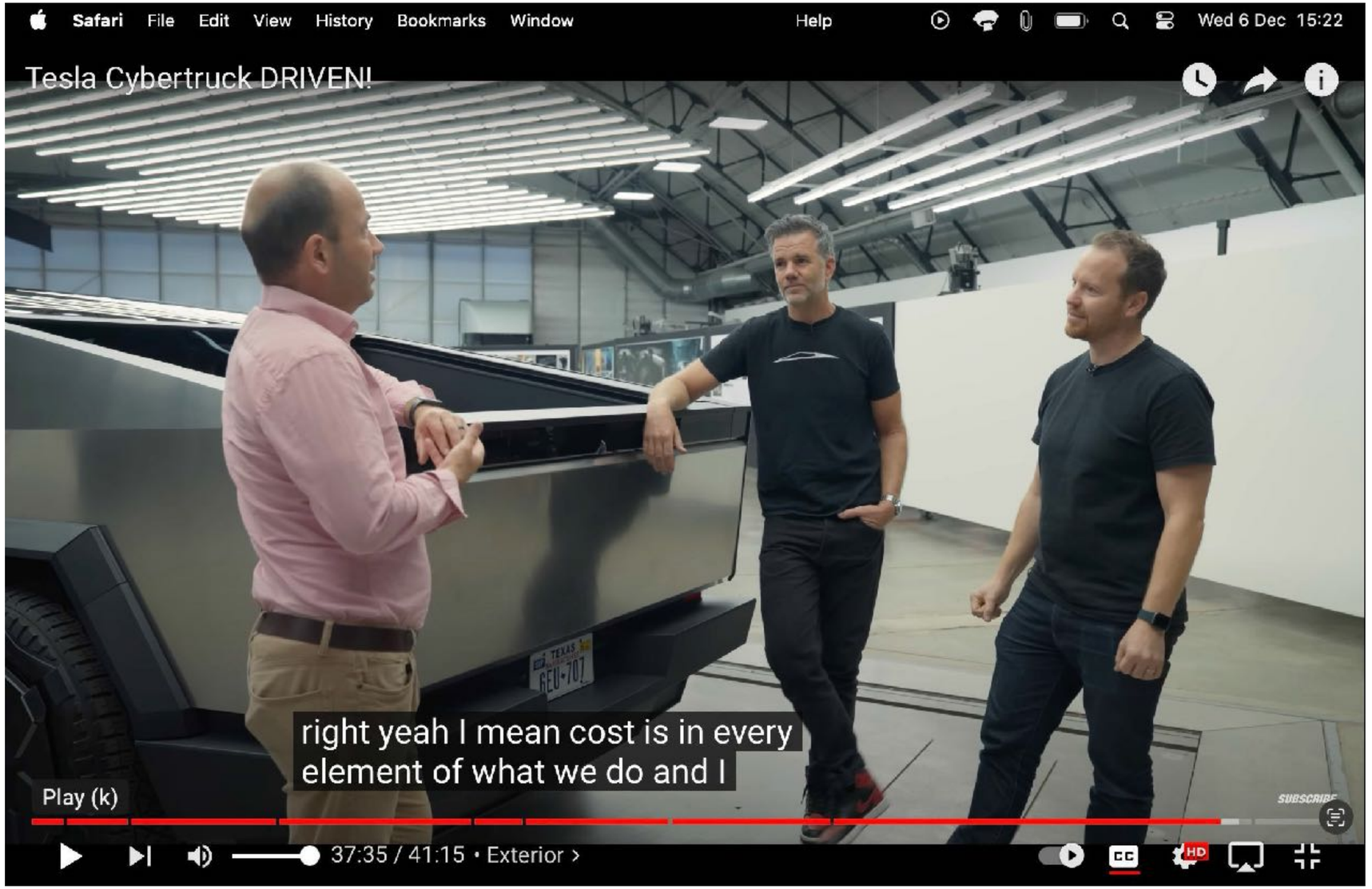
3. IΔ does not recognize the idea of a 'design', (by any synonym, like strategy, architecture), and the ignores fact that <u>*designs have multiple qualities, and costs aspects*</u>. Like *food* for example. And that a *design process* can improve ***all factors simultaneously***.

4. IΔ does not recognize the idea of cost-effectiveness, or efficiency (E = ∑V : ∑C). It seems to assume something like the one-and-only-quality should be 'high enough', and you have to choose to meet a deadline (Schedule) or a Budget. And if that is not possible, your <u>*only*</u> tactic is to reduce Scope (Functionality, 'what the system *does*').



5. Our counter suggestion is the, hopefully, more-realistic Penta Model [Penta].

16.12 Cost ideas from a successful entrepreneur



https://www.youtube.com/watch?v=uefydJUbRhc
The 2 chief Tesla Designers at left, TopGear interviewer at right

16.13. Musk's Cost Improvement methods simplified and summarized. [MM]

1. Musk is a strong believer in reducing all unnecessary costs, by **everybody**, *every* day.
2. Musk **empowers** his engineers and factory workers to find ways to drive down costs on a daily, real time basis, as a team (Mobs). Literally 'on the production line'.



https://www.youtube.com/watch?v=t705r8ICkRw&t=804s

3. Musk gives them time, the permission to fail, and money to spend: to reduce costs.
4. He uses as a *'first principle' that the cost is limited by the basic costs* of materials, not the current inflated cost, by a Space Agency, or Government, or car-supplier bureaucracy.
5. Musk encourages his staff to attack the costs at the ***systems*** *engineering* level: not just one physical or software component, but applying the *big picture*, including government and laws (like car dealerships, and cost-plus contracts).
6. Musk strongly encourages ***daily incremental improvements*** everywhere, to costs, over the long term. NO 'waiting around for the new model in 5 years'.
7. Musk has amazing **vertical supply-chain control, over the costs**, not least the costs of transport, supply and delay. A Tesla car has about 20,000 parts (and many fewer, by design, all the time).
8. Musk is willing to take some **risks of being wrong** *(Lidar on a Tesla, full automation on Model 3 assembly line)* and tries to make a 'component cost' *finally*, nearly justify itself, more clearly. *(Lidar good in bad weather! and Drop many manufacturing bots, initially installed).*
9. Musk wants **incredible levels of quantified qualities** (like Safety - Tesla had 3 car models topping the 50-safest cars rated), or reusable rockets) and he does ***not*** **compromise on qualities**. ***Costs*** **are optimized secondly**, after critical quality levels are reached and improved.

16.14 Musk's Engineering Philosophy:

Musk overviewed his **five step engineering process,** which *must be completed in order:*

1. **Make the requirements less dumb.** The requirements are definitely dumb; it does not matter who gave them to you. He notes that it's particularly dangerous if an intelligent person gives you the requirements, as you may not question the **requirements** enough. "Everyone's wrong. No matter who you are, everyone is wrong some of the time." He further notes that "all **designs** are wrong, it's just a matter of how wrong."

   Additionally, each *required* part and process must come from a **name**, not a department, as a department cannot be asked why a requirement exists, but a ***person*** can.

2. **Try very hard to delete the part or process.** If parts are not being added back into the design at least 10% of the time, not enough parts are being deleted. Musk noted that the bias tends to be very strongly toward "let's add this part or process step in case we need it."

3. **Simplify and optimize the design.** This is step three as the most common error of a smart engineer is to optimize something that should not exist.

4. **Accelerate cycle time**. Musk states "you're moving too slowly, go faster! But don't go faster until you've worked on the **other three things first."**

5. **Automate**. An important part of this is to remove in-process testing after the problems have been diagnosed; if a product is reaching the end of a production line with a high acceptance rate, there is no need for in-process testing.



Additionally, Musk restated that he believes everyone should be a **chief engineer**. Engineers need to understand the system at a high level to understand when they are making a bad optimization. As an example, Musk noted that an order-of-magnitude more time, has been spent reducing engine mass, than reducing residual propellant, despite both being equally as important.

Source: https://everydayastronaut.com/starbase-tour-and-interview-with-elon-musk/

16.15 Relevant Requirements, and Designs

Elon Musk, and his many companies, no matter what you think of his politics or personality, have clearly found a set of methods for improving industrial and human **productivity**. These methods are not unlike Haier's Methods [Productivity Stream], and not unlike my own Planguage Methods (see comparative Table in [Productivity Stream]).

I have tried to collect Musk's Methods [MM] and I will here highlight some of them, as they pertain to productivity: values and costs. I will focus on 'Musk's Engineering Philosophy'. And with his '*First*' (Requirements and Designs) *Methods*.

1. **Make the requirements less dumb.** "The requirements are definitely dumb; it does not matter who gave them to you". *Additionally, 'each required part and process must come from a name, not a department, as a department cannot be asked why a requirement exists, but a person can'."*

- He notes that it's "particularly dangerous if an intelligent person gives you the requirements, as you may not question the requirements enough".

- "Everyone's wrong. No matter who you are, everyone is wrong some of the time."

- He further notes that "**all designs are wrong**, it's just a matter of how wrong."

Requirements define the costs level, quality level and productivity levels you want.

Musk and his teams describe themselves as **'Fanatic About Metrics'** (2023 Investor Day Video). So we do not have to go deeper into *that* aspect.



But in this statement "**Make the requirements less dumb"** Musk is not focussing on 'exact and clear' specification of requirements. He is focussing on '**relevance**' to the system, the market, and the stakeholders. The 'right stuff' as Space-people put it.

I have to agree that **requirements *relevance*** (the ones aligned with stakeholders) is a far higher priority than precision. Especially for costs, and values, which is defined in terms of stakeholders' cost-needs and their values. *Precise but irrelevant* requirements, do not sound like a productive idea to me.

Notice the basic assumption. The initial alignment of the requirements, to a complex and dynamic set of stakeholders, and dealing with beyond state-of-the-art industrial technology; is so difficult to get right initially, that the only safe assumption is that the requirements are not quite right, yet. We have some hard incremental work, to make them competitive. Including work long after major operations, like car production, has started.

16.16. Relevant Requirements, and Designs from a *person* source

Step One: (make an individual responsible).

1. **Make the requirements less dumb.** “The requirements are definitely dumb; it does not matter who gave them to you”.

Additionally, ‘each required part and process must come from a ‘name’, not a department; as a department cannot be asked why a requirement exists, but a person can’.

- *Musk* also does something very interesting, which I also do in Planguage. He assigns a named person as the specification ‘Owner’.

- ‘Personal responsibility. Personal consciousness, that if I bloat the specification, my own name is on it. Nowhere to hide! A very practical method-detail indeed!

- I go one step further:

  - Instead of waiting to ask the Spec Owner why they did silly things, Planguage encourages **explicit immediate specification** (Assumptions, Justification, Rationale, Source, Evidence, Impact Estimates [VIET).

  - We call this ‘Background’ specification [VR, SEA, CE]

  - Useful when the person who did it, is no longer there!

  - People leave.



* *This phrase was initially recorded as part of Step 2. But a comment pointed out it was (obviously) intended as a remark to Step 1 (Requirements) in Comments to . So I treat it here as part of Step 1*

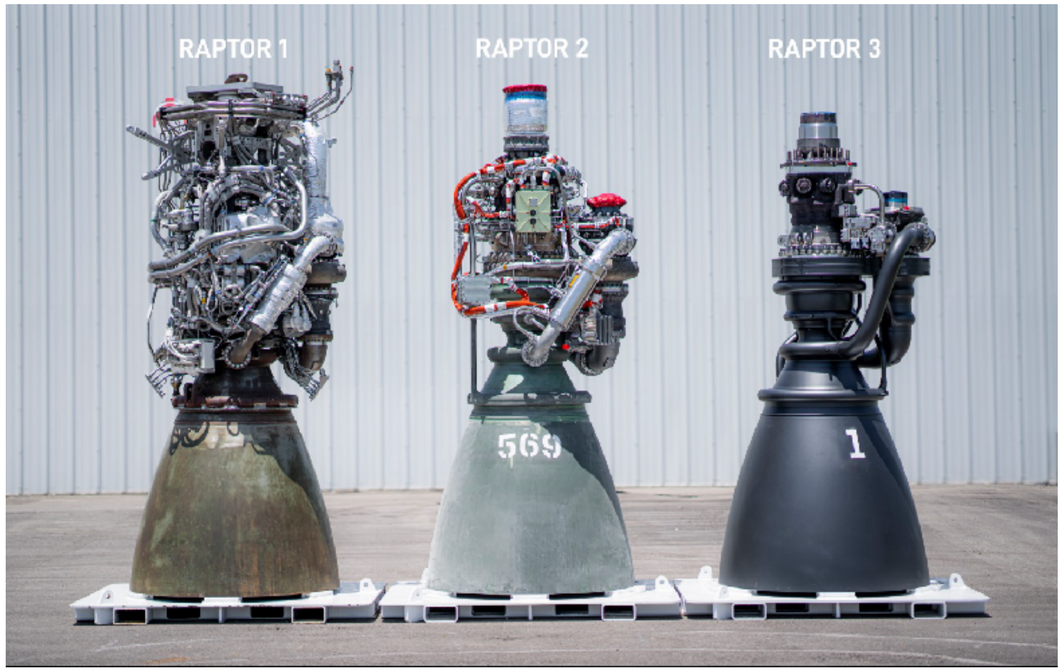
* *PS User Stories, do contain an element of source of the requirement, but usually just a class (‘a User’), but not a name. They also contain an element of Justification (‘in order to speed things up’). Good format or structure. But for large systems, not enough detail.*

16.17 Get leaner: eliminate stuff.

Step Two:

"2. ***Try very hard to delete the part or process.***
'If parts are not being added back into the design at least 10% of the time, not enough parts are being deleted.'

Musk noted that *the bias tends to be very strongly toward - 'let's add this part or process step, in case we need it.'*



2024 Raptor 3 , Even simpler, https://x.com/SpaceX/status/1819772716339339664



https://www.youtube.com/watch?v=3N7L8Xhkzqo, Thursday, February 10 2022 SpaceX CEO

Did you notice that Musk recommends these 5 methods be **done in sequence.** So: '1. **Make the requirements and design less dumb'** is *first priority*. *Then*, we can simplify (the *right* requirement or design, not a dumb one).

So Musk is saying, assume you have required, or designed, too much.

Force yourself to simplify, remove the fat, brutally. Be lean until it hurts.

Notice he is speaking about the entire system: all work processes, all hardware, software, production-line-ware. He is reducing costs, and increasing values; **productivity**.

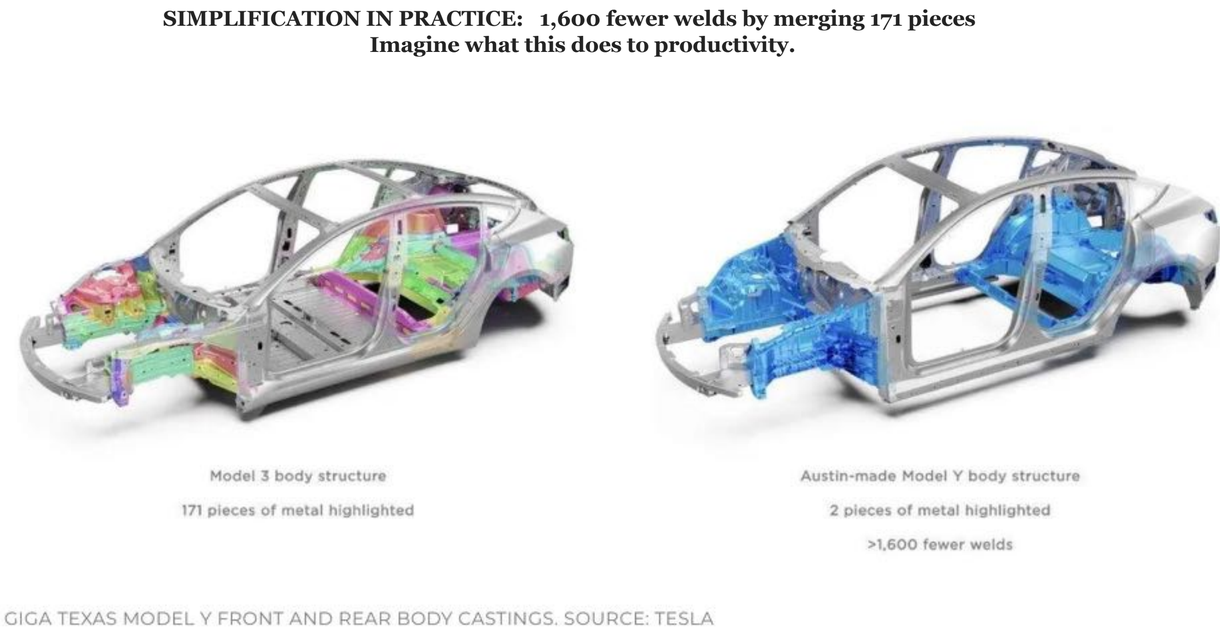
In Mid-2023 Tesla Model 3 Upgrade, you will notice Musk **added back** the Radar! And in Full Self Driving Version 12 (August 2023), Tesla eliminated about 300,000 lines of code, by changing the Full Self Driving software, driving architecture, from "pre- programmed' to 'learned' (Neural Net trained on 1-million-plus Videos).

16.18. Simplify and Optimize

Musk's Step 3. "***Simplify and optimize the design.*** *This is step three, as the most common error of a smart engineer is to optimize something that should not exist."*

Notice Musk's emphasis on doing steps 1 and 2 first: Get good requirements and design, *then* eliminate the superfluous. THEN simplify and reduce costs for what remains.

Get the 'right thing' *working; then find a simpler design, for the requirement. For example, one that is faster to produce, or maintain, like his Model-Y cast. See ill.*



GIGA TEXAS MODEL Y FRONT AND REAR BODY CASTINGS. SOURCE: TESLA

Source: *https://thedriven.io/2023/03/01/toyota-admits-tesla-model-y-is-truly-a-work-of-art-as-it-rethinks-ev-strategy/*

'First principles thinking' does not remove the need for continuous improvement, but it does ***alter the direction of improvement.*** *Without reasoning by first principles, you spend your time making small improvements to a bicycle, rather than a snowmobile. First principles thinking sets you on a* ***different trajectory.*** *What are you trying to accomplish?*

Source: https://jamesclear.com/first-principles

My own 'Planguage' method interpretation [CE] is that we need to get out of the bad habit of specifying our objectives, in terms of 'specific known designs' or strategies. This is what Musk calls an 'analogy'. We need to define the *improvement* dimensions (values, qualities, costs) we want. **Then** *design* to meet them [SEA].

What are called 'First Principles' here, are viewed by my Planguage methods as '*incremental design components'. What Musk is saying is 'do not specify your design ideas in terms of analogy to known complicated solutions'. Instead, build your design solutions, one basic component at a time. And, identify the most cost-effective, new technology, components, as you go. Don't get locked into old thinking, but 'think fresh' to be competitive.*

16.19. Improve cycle time.

Step 4: **Accelerate cycle time**. Musk states ***"you're moving too slowly, go faster! But don't go faster until you've worked on the other three things first."***

Here is a very Musk-personal view of one type of 'cycle time' tactic.

Musk divides his working time into 5 minutes slots.

"This increases his attention and effectiveness. And he completes his tasks much quicker". This apparently applies to his lunch break. Meals are brought to him at meetings, to save time.

His first 30 minutes after waking at 7am, after 6 hours sleep, is to review critical emails (meetings and go-aheads for research milestones).

There should only be 4-6 people in a meeting . People who can no longer contribute to a meeting, are expected to leave, to save their own time.

Meetings should not be excessive. When the meeting purpose is resolved, the meeting should stop.



Source Joe Justice

Weekends are exclusively for the family, unless there is a work emergency.

At Tesla, the **daily Mob** has a one-day, 12-hour cycle, for reaching a clear goal, set at beginning of day. The Mob rotates positions every 7 minutes (cross training).

They are not having a 'meeting'. They are building prototypes, and trying to get incremental ideas into real production, by end of day. Compare to many-year cycles, for other auto factories!

A Day in the life of Elon Musk, 2021



Source https://www.youtube.com/watch?v=GkEyCrqDvVo

16.20. Then.... Automate

Step 5, **Automate**. *“An important part of this is to remove in-process testing after the problems have been diagnosed; if a product is reaching the end of a production line, with a high acceptance rate, there is no need for in-process testing”.*

Musk admitted he made a big mistake by too much automation in the initial Tesla Model 3 production line (*“Production Hell”*, he called it). He succeeded, by getting rid of many of the robots,.
Productivity was better, in some cases *without* automation!

Step 5 refers to the fact that there were many automated tests in the Model 3 production line, which were necessary *initially*, to sense problems in the process. But which were superfluous, when things were r*unning more smoothly*, and which could be productively replaced, by only measuring *final* results.

The main point he is making, by positioning automation as the 5th step, is that we should only automate, ‘the right stuff’. And, to do that we need to do the *other* steps to a mature degree, first.



https://www.bloomberg.com/news/features/2018-06-08/tesla-model-3-photos-of-elon-musk-s-factory-in-fremont?embedded-checkout=true

Here is a summary I made of this step:

AUTOMATE: Automate processes with high automation payoff, and avoid bad automation.

A. **Automate incrementally (not all at once), and make sure they are successful; before moving on.**

B. ***Design* processes, and *design* ‘product components', so that they are *easy to automate*. To ‘manufacture’.**

D. **Remove early-step built-in testing and measurement when the downstream quality is stable and good. <- E Musk.**

16.20. Systems Thinking: 'Holism'

So that, above, was Musk's off-the-cuff 5 steps, while wandering around Space-X . But, he added another principle, which is *not* a sequential step, but which has to be integrated into all the other steps. It is about **'systems engineering'**. Seeing the whole system. That means, not merely the part or the 'car', or the rocket ship. But it refers to *everything surrounding it, over its* ***life-cycle.***

He put it this way:

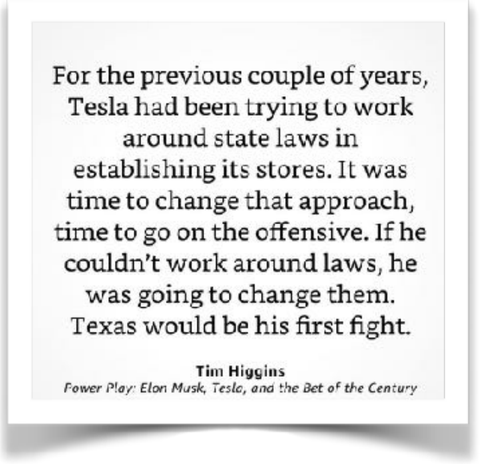
"everyone should be a chief engineer. Engineers need to understand the system at a high level, to understand when they are making a bad optimization".

As an example, Musk noted that *"an order of magnitude more time, has been spent reducing engine mass, than reducing residual propellant, despite both being equally as important."*

Musk does not want his engineers, and other planners, of other corporate processes, to merely optimize the component they are primarily expert in, and responsible for. He wants them to do like he always does, and try to get even better solutions, by factoring in other elements, of the related system, in all related dimensions, and over life-cycle time.

He is asking his team to avoid sub-optimization, and to seek overall optimization.

One method to promote this thinking is the 'Mob Team' [MM]. Another is the great freedom, teams have, to use money, and to cross organizational 'boundaries'.



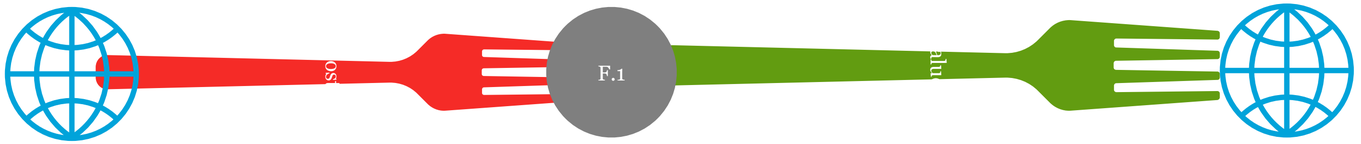


This 'systems thinking' opens up possibilities of order-of-magnitude productivity improvements; rather than just a few percentage points, here and there, re-using already well-known technologies.

This is not merely 'engineering'. This is also **politics**, (see Texas case above) Well, let me modify that. He is a **total systems thinker.** He is willing to take on *anything* standing-in-the-way of his visions. No excuses, deliver the vision. Deal with constraints [Guides] ! Run around them, or *removes* them. Make no mistake, this is a *method*. Maybe he would call it a **First Principles** method, or philosophy. Do not get stuck in the current analogy (like - 'car dealer franchises', 'NASA Space development methods'). Not if you want to conquer established markets.

The 'trick' is simple. **He keeps his long-range vision steady**, and he and his troops, find ways to get there.

16.21 Cost Stream Definition



What does 'cost **stream**' mean in the context of the book [Productivity Stream] ?

The *cost stream* is the knock-on effect path, that any and all types of costs traverse, during a system life-cycle. From resources in, to Values Out to Stakeholders, then From these stakeholders to the next, supply chin Resources In, Values Out, and so on.

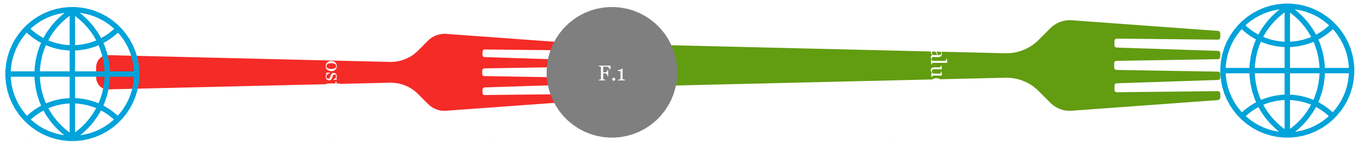
This is interesting for

A. Understanding how one level, or type, of cost decision ('reduce it') has future good or bad effects on future costs, of the same, or different types. 'Knock-On Effects'.
B. Managing the *whole system set of costs* for the system lifecycle.

What does cost **flow** mean in the context of this book?

Flow refers to the potential future areas of responsibility and opportunity, for managing both costs and other stakeholder values, as a consequence of earlier cost management decisions, or changes.

16.22.1 Cost Stream Management Principles



1. DYNAMIC COSTS DESIGN: We can *dynamically* manage, a critical set of costs, in a sequence of related processes.

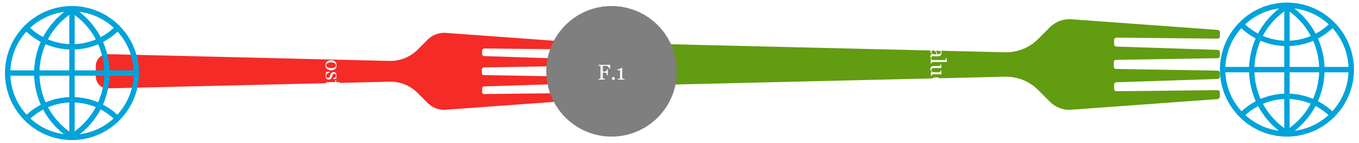
2. BALANCE SETS: Multi-cost management tries to *balance* the current set of costs, both in *individual* processes, and for a *set* of processes.

3. CONSTRAINTS RULE: The cost management process is guided by a set of requirements, values, budgets, and constraints, which themselves are probably changing. *For example a* quality *'Tolerable Level' (a constraint) cannot be made impossible to reach by some cost-reduction effort or decision.*

4. PRIORITY SACRIFICE: One process might *sacrifice costs*, in order to re-invest in 'costs reduction' of related processes.

5. DELIBERATE DESIGN ALTERNATIVES: The cost management can be applied by *technical* or *strategic* '*design of the process*', or by some 'real-time mechanism' for changing the costs.

16.22.2. Cost Stream Management Principles



6. ALL COST TYPES: Costs are not just financial, they are *all limited resources*, including time, people, space, reputation etc.
7. A concurrent flow-stage cost, of any type, can be *designed to improve* any other concurrent cost type. For example most financial *capital costs*, can be used to reduce *building time*.

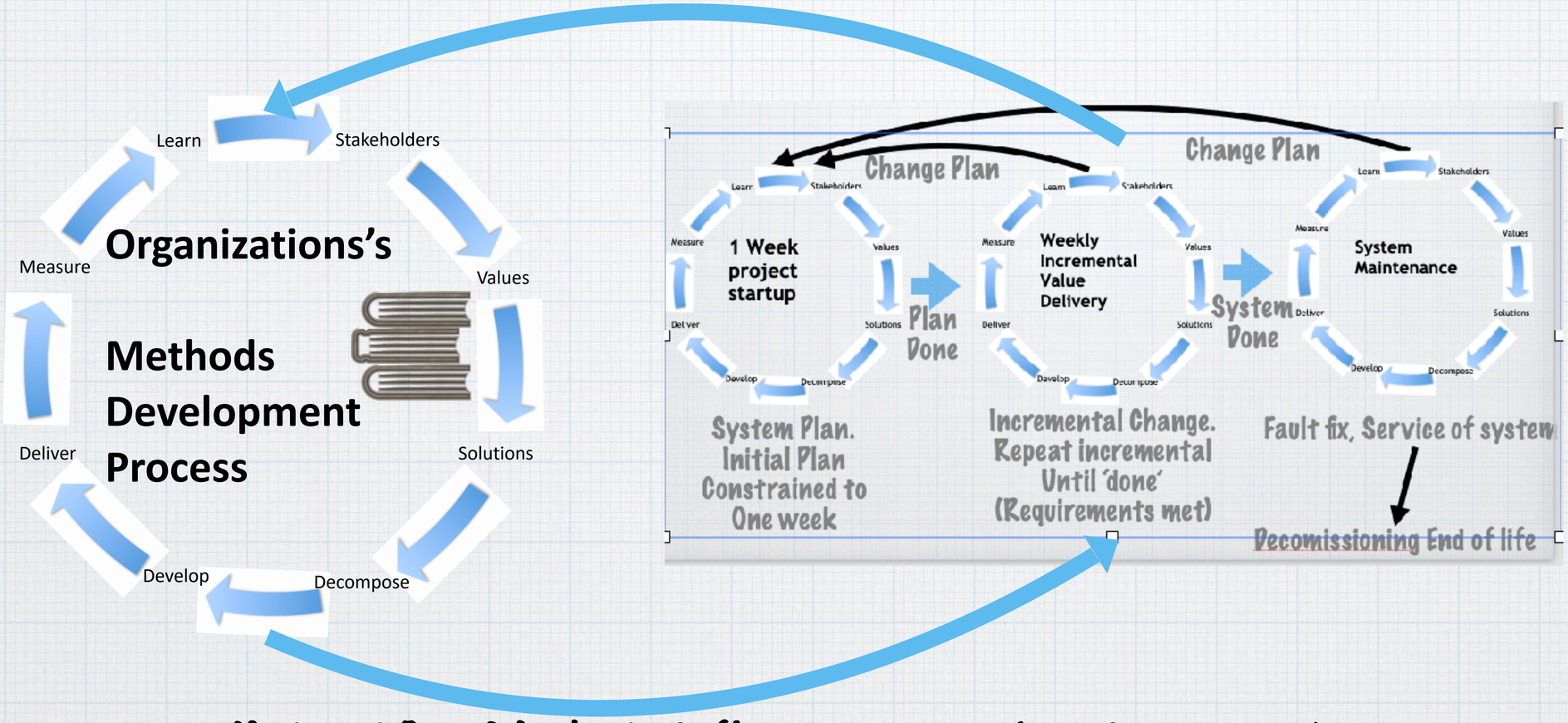
8. LOOPBACK IMPROVEMENT: Assuming we are (Evo-Cycle) cycling back to previous processes, we can impact previous process' future costs by designing current processes better [Evo]. See 16.23.

9. GLOBAL COST OPTIMIZATION: *Local* cost optimization potentially threatens *global* cost optimization. (JM)

10. COST TYPE TRADEOFF: it is possible to re-design, to reduce *one* type of cost, by sacrificing other types or stages of costs, or other value levels.

16.23 Feedback to/from Organization

Org. Methods updated, as result of a current project [Evo] (for ex. using 'DPP' **)



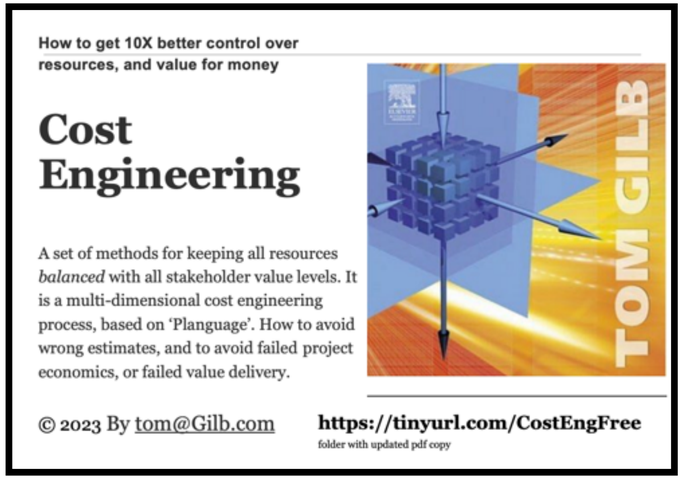
Updated Org. Methods, influence even an 'ongoing project'; which itself, earlier, changed those same generic methods

**DPP see and presenter notes :original IBM SJ, Robert May paper (also in Gilb, Sw Inspection book Chapter 17)
http://agileconsortium.pbworks.com/w/file/fetch/1527643/Mays1990ExperiencesDefectPreventionIBMSysJ.pdf

16.24.3.0 Cost Engineering

A tactic for handling uncertainty, threats, attacks and damages

- [Cost Engineering] is a discipline we can learn and apply to the book's main theme, or dealing with uncertainty, threats, attacks, and damage to our system. 'Cost Engineering' is my book about this discipline in detail.
- This is a discipline using engineering design methods, to manage any types of cost (money, people, time, space, resources) at any time, by '***design***'. 'Design' meaning 'by changing the specific architecture, design, means - as well as changing the many value and cost requirements', so that we get better 'balance of values and costs. For example reducing an availability requirement from 99.998 % to 99.90 might reduce costs by at least 10x.
- This *cost engineering* is something we should use throughout the system lifecycle. Awe should use it as a tool for dealing with the many uncertainties about resources, at all times.
- The most powerful single idea, and one not widely appreciated, is that, there is almost always a possibility, that there is some design, probably many designs, which will enable you to reduce any type of cost, significantly.
  - You just have to find that design somehow, and then you have to be willing to invest in the design implementation, and to tolerate the side effects it has, in all other areas of costs and values.
  - But the *big idea* is that this is generally possible. This 'universe of design ideas' is infinite, and there are systematic ways to identify those cost-reduction designs.
  - Sometimes it helps if your architect is skilled, persistent, well-trained [SEA], imaginative, rigorous, and brilliant. Like Frank (Gehry, or Lloyd-Wright), or Elon Musk [MM].
  - Initially, I am going to copy some pages from [Optima] from the Resources Chapter, to give you the theory of Planguage Cost Engineering. ([Cost Engineering] sections 3.1 to 3.39) .



[Cost Engineering) https://leanpub.com/costengineering

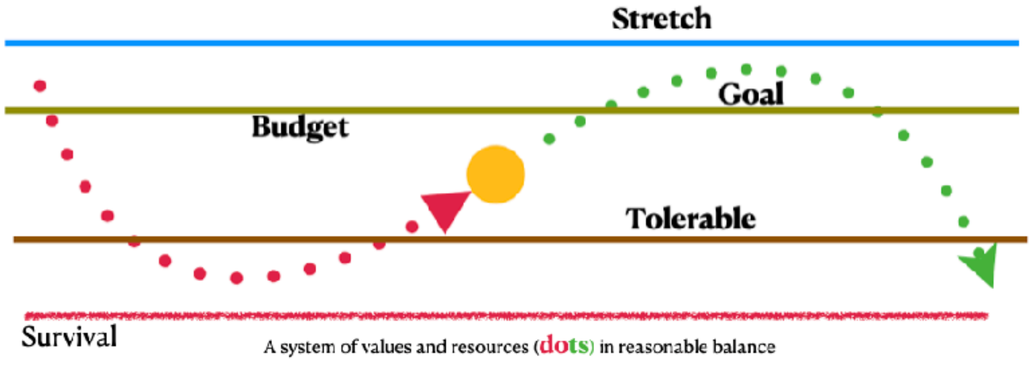


Figure 16.24.3.0 Visualizing the balance of Values and costs, where 'Cost Engineering' is a process to help us balance these things, and to maintain reasonable efficiency.

16.24.3.1 Resources

The limited fuel to pay for value results

Resource **Concept *199 May 19, 2003**

A resource is any potential system input.

A resource is any kind of input ‘fuel’ necessary for building, operating or maintaining a given system. A resource is an asset or a supply that can be used to produce the functionality, or performance levels, of a system.

A resource can be defined using a scale of measure. A requirement, for a resource, can be specified by a target level, or a constraint level. Previous levels of resource utilization (incurred costs) can be specified by benchmark levels, like Past.

We can distinguish between budgeted and unbudgeted resources. Resource budgets are found in our formal plans.

Target Level : Wish, Stretch, Budget

Constraint Level : Survival, Fail, Tolerable

Benchmark Level: Past, Status, Trend

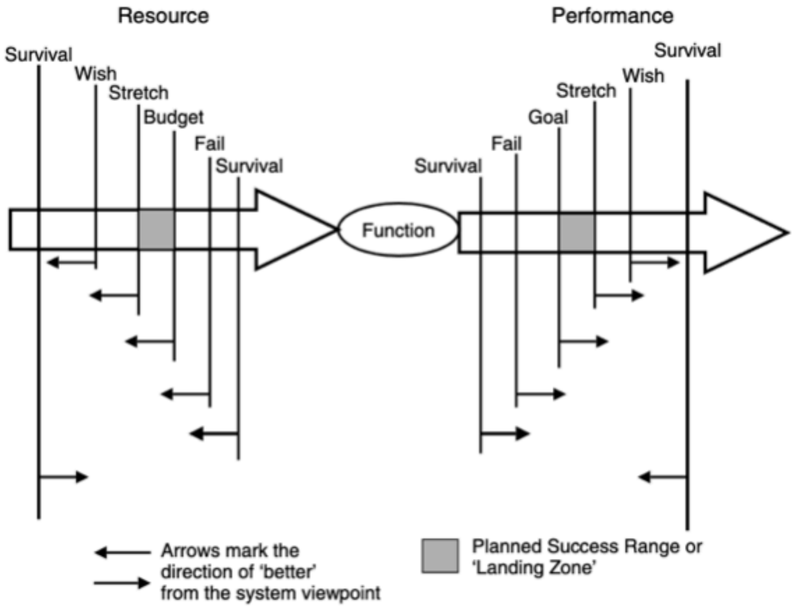


Figure 3.1 B [CE] ‘Resources(left arrow) are the ‘fuel’ that enables us to deliver strategies, which result in stakeholder values (right arrow)

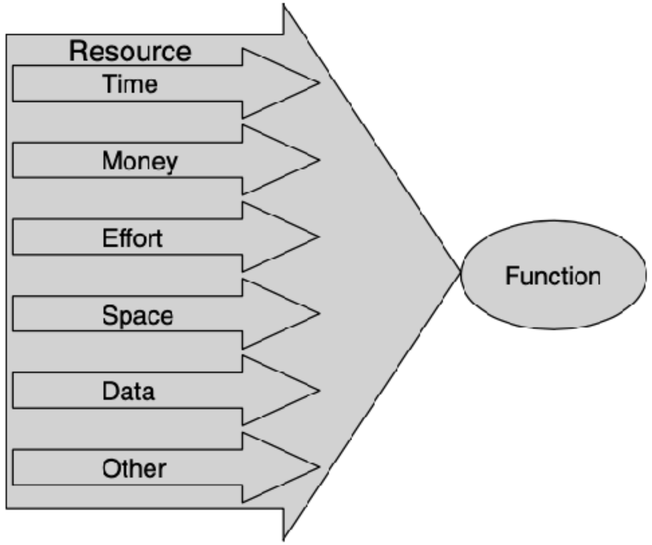
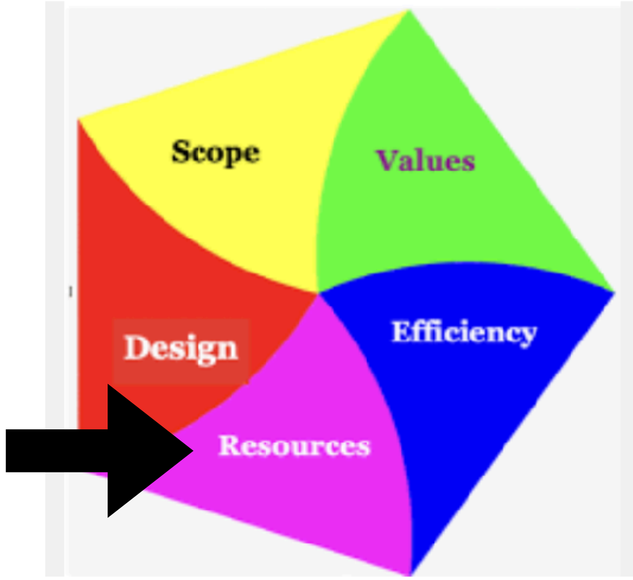


Figure 3.1 B Types of resources

16.24.3.2 Resources:

The problem with Resources



Basic ‘Penta’ Model of system attributes

- Limited
  - Resources are limited quantitatively, by decision, by contract, do not exist
- Suddenly Constrained
  - Previously available, budgeted resources disappear. Removed, stolen, never materialized.
- Used more than planned
  - Completion to Tolerable or Goal level of a Value, actually cost more than estimated and allocated
- Balanced: Long and Short term
  - Over focus on short term resources, CapEx, and little or no focus on long term resources (like OpEx, Maintainability (3.3.2), will probably lead to strategies which are out of control (nobody knows, nobody is responsible) regarding recurrent, operational and long term costs (like dismantling, cleaning up, damage payments)
- Balanced: amongst other resources
  - How can we prioritize all resource types, so that the total lifetime costs are minimized?
- Future Resources, Current Resources
  - How can we balance current spending of resources in reason to future spending, especially considering different resource supply at different times?
- Wrong type of resource available
  - What happens if we run out of time to deadline but have lots of investment money available? For example.

16.24.3.3 Resources:

The opportunity with Resources: for more depth see [Cost Engineerng]

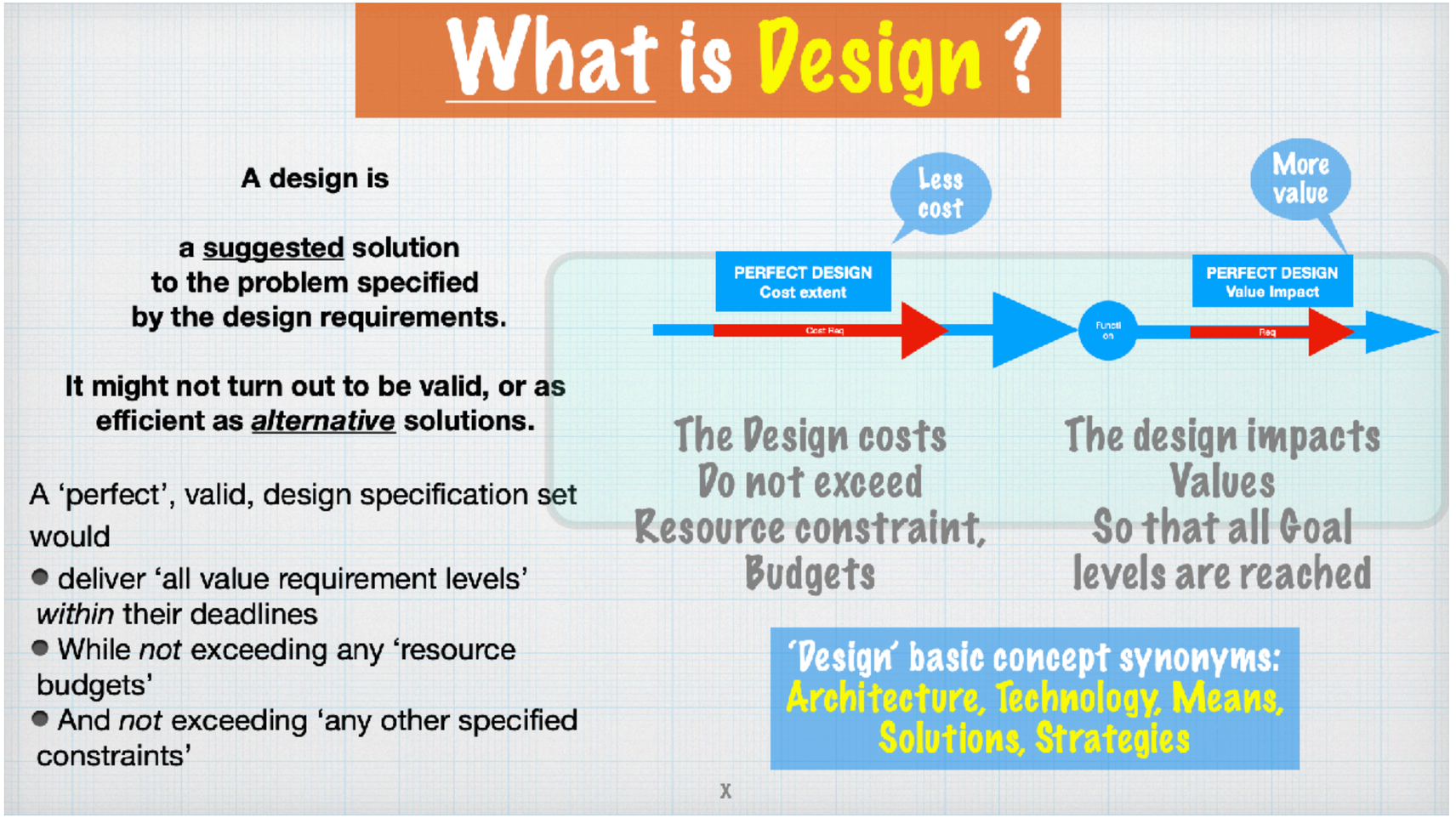
- Limited
  - Resources are limited quantitatively, by decision, by contract, do not exist
  - OPPORTUNITY: design to cost [Cost Engineering], dynamic design to cost, prioritize cheaper values
- Suddenly Constrained
  - Previously available, budgeted resources disappear. Removed, stolen, never materialized.
  - OPPORTUNITY: argue for more resources by clarifying the quantitative value levels, and there stakeholders, and ROI
- Used more than planned
  - Completion to Tolerable or Goal level of a Value, actually cost more than estimated and allocated
  - OPPORTUNITY: improve the incremental dynamic design to cost process for the future or this and other projects
- Balanced: Long and Short term
  - Over focus on short term resources, CapEx, and little or no focus on long term resources (like OpEx, Maintainability (3.3.2), will probably lead to strategies which are out of control (nobody knows, nobody is responsible) regarding recurrent, operational and long term costs (like dismantling, cleaning up, damage payments)
  - OPPORTUNITY: make a clear policy to prioritize the long term and recurrent resources, at early strategy design stages, use IET to evaluate
- Balanced: amongst other resources
  - How can we prioritize all resource types, so that the total lifetime costs are minimized?
  - OPPORTUNITY: adjust ‘too high’ value Goals down, so as to allow less costly strategies
- Future Resources, Current Resources
  - How can we balance current spending of resources in reason to future spending, especially considering different resource supply at different times?
  - OPPORTUNITY: use Impact Tables (IET, [VIET]) with conservative levels of multiple resources, especially threatened resources, and design to costs
- Wrong type of resource available
  - What happens if we run out of time to deadline, but have lots of investment money available? For example.
  - OPPORTUNITY: re-design strategies using time, so new strategies use other resources like people and money

16.24.3.4 What is a 'design' ? It is the key thing we can choose for meeting requirements.

Design strategies determine all your costs

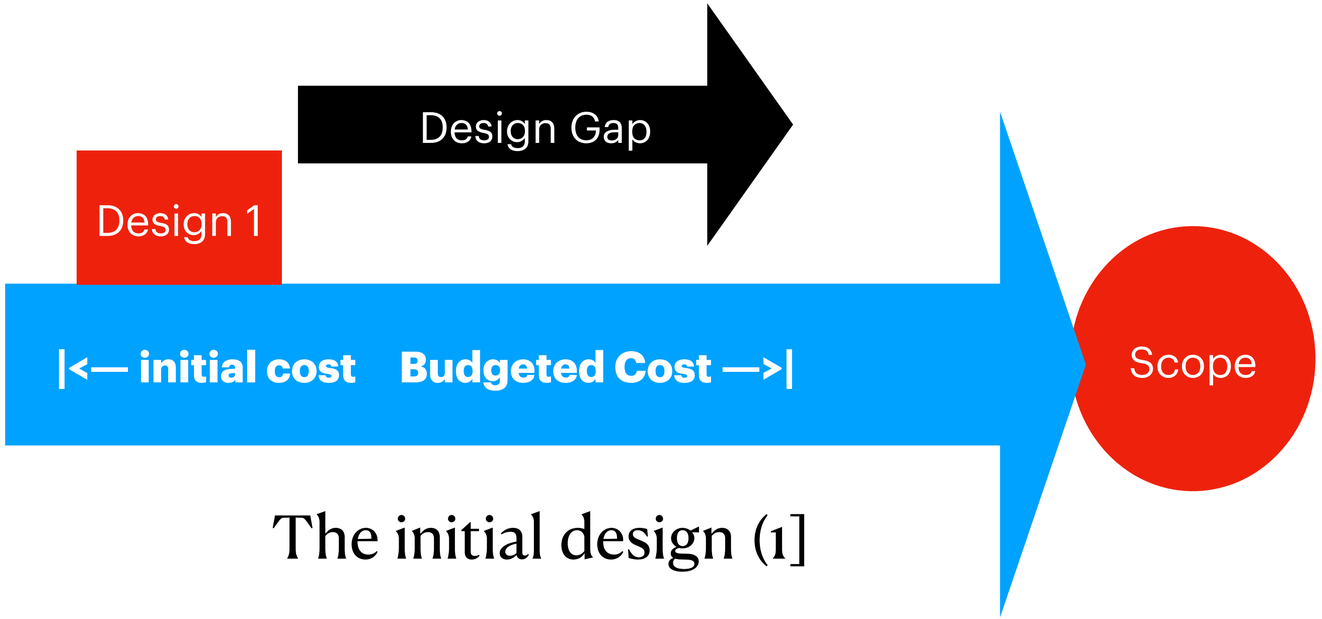
- Each specific detailed design, as implemented, carries with it a *set of costs*.
- You cannot, normally, delete or adjust the costs you do not like.
- They are hardwired into that specific design, *at the moment it is integrated into your system.*
- All you can do is to change the design, to another design, with a better set of costs.
- You can analyze, present, and view, designs of alternative and similar nature, but each with different cost sets, using an Impact Estimation Table, and select the best option you have estimated.

A 'design' is chosen because it gives good Value impacts and low Costs



Source: Value Project Management slides [Value Management]

16.24.3.45 Visualization of ‘Design to Cost’



The initial design (1]

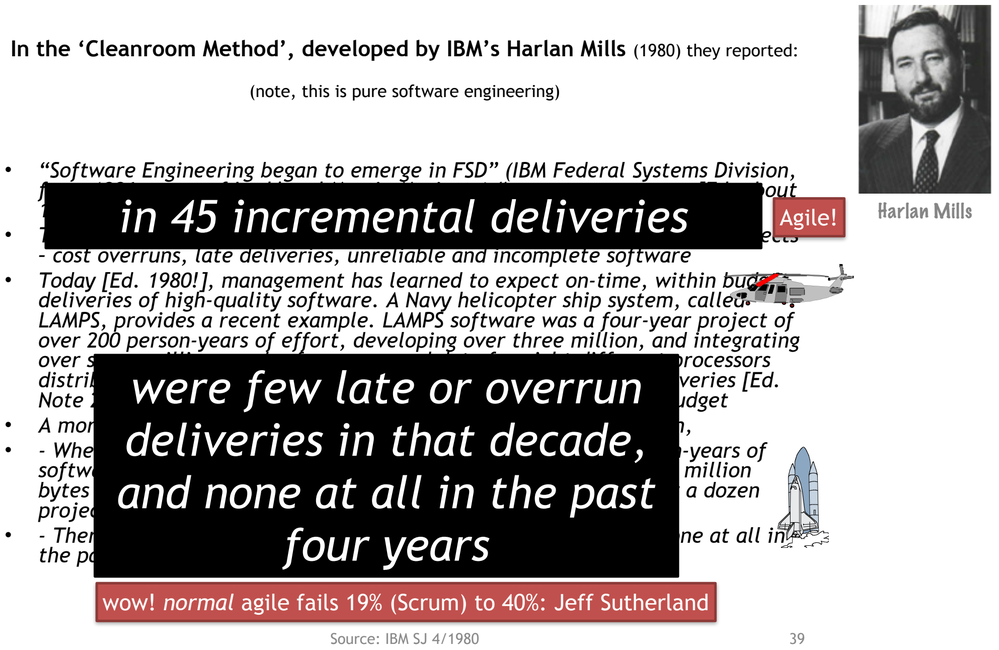


The design (2) that reduces the cost to desired level, like a Budget level.

16.24.3.5 The best method for delivering high-tech software on time, under budget, every time.

A ten year, 26 project, published case history. ‘Agile Engineering’

DDtC, Dynamic Design to Cost, not only works on Gehry skyscrapers, and Pixar films. It works on advanced state-of-the-art software projects for IBM, DoD, and NASA. This was done in the 1970’s for 10 years. And resulted in a perfect record (qualities, deadline, costs) for 4 years running. This is much better than any other software projects or organizations I know about. This is ‘agile as it was, but ‘is not’ today’. Lots of numeric feedback and correction, and about 45 cycles of homing in on the more-perfect design. Not unlike Pixar and Gehry (Flyvbjerg] ! I knew these IBM FSD guys and traded ideas with them. I did not just ‘read about it’. I reported it in my 1988 book [PoSEM] which many Agile Manifesto signers cite as inspiration. This DDtC method is inspiring, but the agile guys did *not* do



[Mills]

16.24.3.6 IBM Cleanroom, initial cost estimates

The competitive bid basis, in accepted contracts

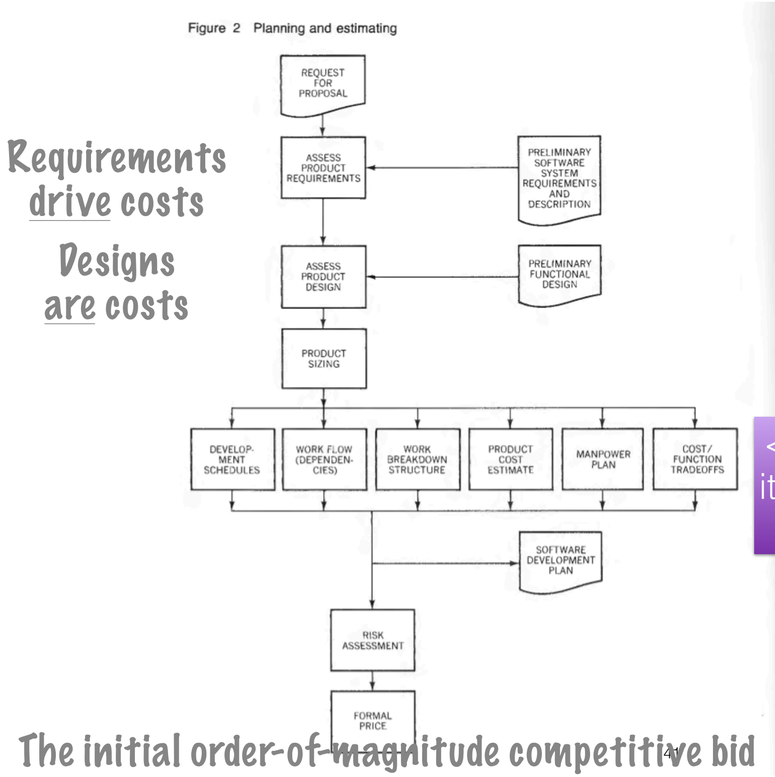
IBM Federal Systems Division had the usual problem.

When they won a low-bidder government space-or-military contract, it had a fixed deadline with penalties, and very well-defined quality levels (like 99.98% availability in flight).

They initially lost money on all these, and assigned Dr. Mills to find a profitable process. It took 6 years to perfect it.

Here is one part of the Cleanroom [Mills] method answer, the bid basis.

A rigorous estimation process, up front. They knew that was an inaccurate estimate, but it was a rigorous engineering start point.



Requirements drive costs

Designs are costs

The initial order-of-magnitude competitive bid

'Cleanroom'
Planning and Estimating.
For making a fixed price bid

"developing a design, estimating its cost"

<— making sure it is cost effective (static)



Robert Quinnan.
The software architect
For Cleanroom

Source: Quinnan, IBM SJ, page 472
http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan

16.24.3.8 Basing Cost Estimates on dozens of data parameters from dozens of projects

They used insights from their own project histories- captured rigorously

One impressive engineering tactic, they used, was the Walston and Felix project database of historical IBM FSD projects, and their cost-driving parameters, on about 20 pages each project. Local (IBM FSD, Space, Military) knowledge.

This allowed them to tune their bid estimates more realistically than competition, who were probably *without* such systematic local cost and attribute data.

Example of attributes data collected: Use of Software Inspection [Gilb, Software Inspection 1993] gave a project '50% less cost' than informal review/walkthrough methods!

Good to know to price low. IBM invented the Inspections. Competitive advantage.

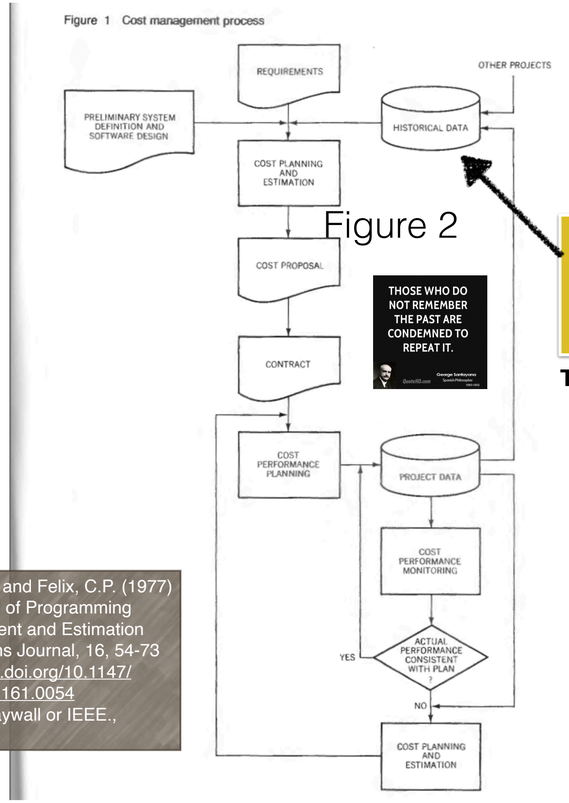


Figure 2

THOSE WHO DO NOT REMEMBER THE PAST ARE CONDEMNED TO REPEAT IT.

Walston, C.E. and Felix, C.P. (1977) A Method of Programming Measurement and Estimation . IBM Systems Journal, 16, 54-73 . http://dx.doi.org/10.1147/sj.161.0054 $33 Paywall or IEEE.,

'Cleanroom' Cost Management Process

IBM FSD had a very advanced detailed collection of historical data from previous projects. Published in IBM SJ, Walston and Felix About 20 pages of data per project were collected

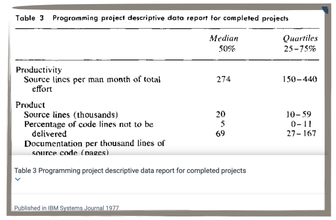


Table 3 Programming project descriptive data report for completed projects

	Median 50%	Quartiles 25–75%
Productivity		
Source lines per man month of total effort	274	150–440
Product		
Source lines (thousands)	20	10–59
Percentage of code lines not to be delivered	5 69	0–11 27–167
Documentation per thousand lines of source code (pages)		

Table 3 Programming project descriptive data report for completed projects

Published in IBM Systems Journal 1977

Think: Fighting Covid-19 Virus, by using data collection

"ensuring that the design is cost effective"



Compare to Bent Flyvbjerg's 16,000 project database

Source: Quinnan, IBM SJ, page 471
http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan

42

16.24.3.9 Dynamic Design to Cost in practice



Their Evo steps are about 2% of total project. They do the 'Evo Cycle', and measure and learn. That is 50 opportunities for the architect to improve the design, and the costs

So here is the Dynamic Design to Cost (DDtC). At each increment (think 45 monthly increments, as for LAMPS project) which showed any negative results for cost, time or qualities, Robert Quinnan, the architect, to the rescue.

These were not mere *build* increments, but were put into the real environment (on the ship) as far as possible.

Change the systems design to get back on track, NOW!

Their rigorous engineering gave better than normal initial estimates. Say ±40%.

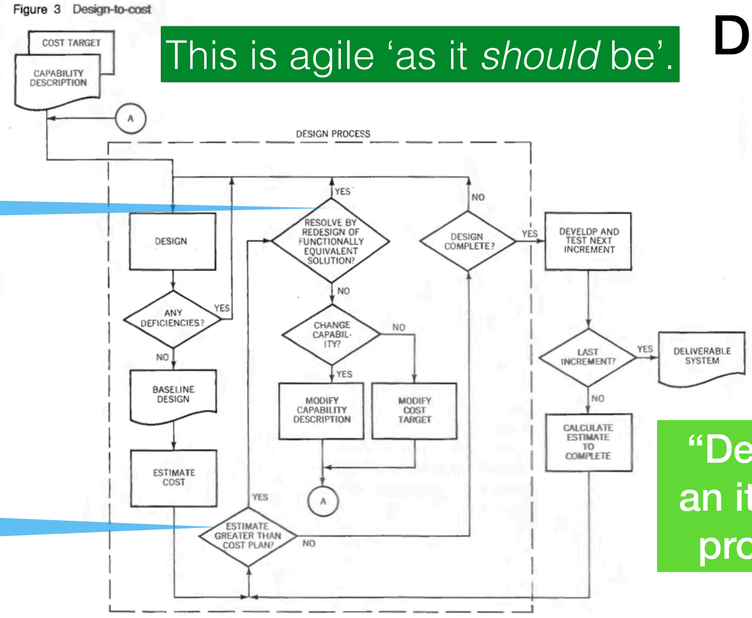
This incremental feedback and re-design brought it to something like ±10% in the first 6 years, and finally ±0% or better (more profit) the next 4 years.

This adjusts the 'right order of magnitude', the last 10% to the bid level, the deadline, the value levels

At Tesla, every team is expected to do this every day, forever. https://tinyurl.com/MusksMethods

Redesign to align with costs, time, value levels
Fix design Bob, now!

Increment Result Deviation (oops!)



This is agile 'as it *should* be'.

Design to 'Cost'

The 'values' are a fixed quantified contract-specified set-of-qualities like 'availability' 99.98%

"Design is an iterative process"



Robert Quinnan: The dynamic Incremental 'architect'

43

Source: Value Agile
https://www.dropbox.com/sh/o2g7ib3z2g2uzfw/AAAypXIN0yA2WS4obwlDzZR3a?dl=0

Source: Quinnan, IBM SJ, page 473
http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan
And PoSEM (Book, 1988) page 105, Figure 7.10

Source [Mills]

16.24.3.10 Principles of Dynamic Design to Requirements

One way to get your mind around design to cost

I love to summarize 'sets of deep technology' into 10 principles. I think it forces me to summarize knowledge.

Notice I have taken a 'generalization' step here, to not only apply design to *costs*, but also to *any requirements*, especially stakeholder *values*.

And it does not hurt, that principles are short and memorable, even teachable.

I have written some long books, but they are summarized by short principles.

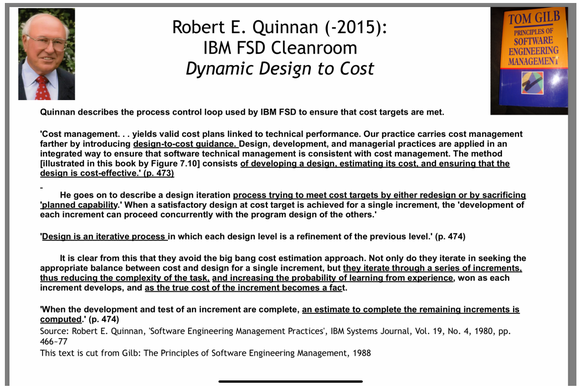
We need both extremes. I *believe* that *these*, and my other principles, are timeless, scalable, and generally true for *any* system.

But, time will tell! A principle is not a 'law', but hopefully a useful guide.

Principles of 'Dynamic Design to Requirements'

1. **MEASURE: It is easier to *measure* accurately than to *predict* accurately.**
2. **ADJUST: It is faster to *adjust* a bad design, than to figure out the *right* design to begin with.**
3. **SMALL: Even small systems can be complex, so frequent incremental measurement of progress is smart.**
4. **INTERACTIONS: Everything can interact with everything else, now and in future, so incremental measurement of critical factors is the only safe way to keep score.**
5. **DtC: It is better to 'design to a cost' you require, than to 'estimate a design cost' you can't afford.**
6. **DtV: It is easier to 'design to a stakeholder *value* level' than to just *hope* your design will meet that level.**
7. **ESTIMATE EARLY: Estimation of design impacts, helps sort out 'probably good design' early.**
8. **CONFIRM: Only *measurement* of design value-delivery and design-costs will confirm you got it right.**
9. **DESIGN CONFLICT: Designs interact with all other system designs, past and future, compounded by changing stakeholder needs - and consequent re-design.**
10. **DESIGN STEPWISE: Small incremental steps of design, greatly simplify understanding of complex design.**

Copyright gilb.com 2019-2023.



Robert E. Quinnan (-2015):
IBM FSD Cleanroom
Dynamic Design to Cost

Quinnan describes the process control loop used by IBM FSD to ensure that cost targets are met.

'Cost management. . . yields valid cost plans linked to technical performance. Our practice carries cost management farther by introducing design-to-cost guidance. Design, development, and managerial practices are applied in an integrated way to ensure that software technical management is consistent with cost management. The method [illustrated in this book by Figure 7.10] consists of developing a design, estimating its cost, and ensuring that the design is cost-effective.' (p. 473)

"

He goes on to describe a design iteration process trying to meet cost targets by either redesign or by sacrificing 'planned capability.' When a satisfactory design at cost target is achieved for a single increment, the 'development of each increment can proceed concurrently with the program design of the others.'

'Design is an iterative process in which each design level is a refinement of the previous level.' (p. 474)

It is clear from this that they avoid the big bang cost estimation approach. Not only do they iterate in seeking the appropriate balance between cost and design for a single increment, but they iterate through a series of increments, thus reducing the complexity of the task, and increasing the probability of learning from experience, won as each increment develops, and as the true cost of the increment becomes a fact.

'When the development and test of an increment are complete, an estimate to complete the remaining increments is computed.' (p. 474)

Source: Robert E. Quinnan, 'Software Engineering Management Practices', IBM Systems Journal, Vol. 19, No. 4, 1980, pp. 466-77

This text is cut from Gilb: The Principles of Software Engineering Management, 1988

16.24.3.11 ‘Design to’ methods

Don’t estimate, create your own future

The ‘power of design’, and better - ‘Dynamic Design’- to change costs, and also to the ‘value-to-cost ratio’ (i.e. efficiency, profit) is applicable to more than simple ‘capital financial costs’; which many people are initially interested in controlling (aka the budget).

Notice that we have not totally given up on ‘estimation’ as a tool.

We are going to *enrich* estimation’s role, so that is works much better, to help us manage costs.

An initial estimate will not rule us; we shall be the masters of our final costs.

Dynamic Design to Requirements

(which includes some priority stuff)

Here are some basic ‘Design to …’ definitions:

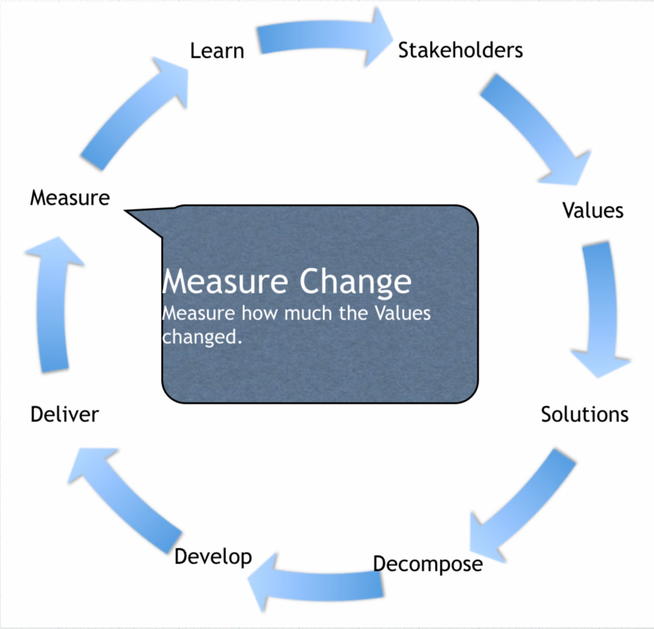
Design To Cost (D->C): a well-established engineering concept. You can find designs to meet a given cost requirement.

Design To Value (D->V): the same concept as Design to Cost, except the design process is directed towards meeting a *Value* (including any *quality*) Requirement Level of Performance.

Design To Requirements (D->R): the combination, perhaps simultaneously in a single delivery cycle, of attempting to design to any set of both ‘Value Requirement Levels’, and ‘Cost Requirement Levels’.

Dynamic Design to Requirements (DD->R): A cyclical Design process, to meet any set of Value and/or Cost requirements, by using measurement, after incremental design-implementation, comparing with requirements, predicting future cost and value levels, and re-designing if necessary, to better reach the requirements.

45



16.24.3.12. Principles of Multiple Designs to Cost

Principles of Multiple Designs to Cost (©Gilb 101023), page 1 of 2

1. **ONE MIGHT NOT BE ENOUGH: It may be necessary to apply more than one distinct design, in order to reach a specified cost target on time, and meeting all other requirement conditions (who, where, how, ...).**

2. **DESIGN DEPENDENCIES: Initially good-enough designs, might have unacceptable conflicts with necessary additional designs, so these initial designs might require deletion, replacement or adjustment.**

3. **THE COMPLETE DESIGN SET AGENDA: You cannot be sure of the appropriate set of cost designs, until the complete set is known. You cannot 'finalize' initially good designs, until the complete set has: 1. Been estimated to meet cost targets, 2. been proven in early increments, one design at a time, to meet expected estimations, and 3. Finally, in a real system, over a reasonable time, the set meets expectations, and 4. There are not unacceptable side-effects on other critical requirements.**

4. **LIMITS TO MULTIPLE COST DESIGNS' UTILITY: we must continue evaluating 'useful designs' (which improve meeting cost and value requirements) until cost targets are met, or until the new-design return-on-investment, is too-low - or negative.**

5. **PROBING COST LIMITS: We can consider the result of adding additional designs, even when we exceed current specified cost targets, and this information might lead to adjustment of our specified ambition levels for costs.**

16.24.3.13 Principles of Multiple Designs to Cost

Principles of Multiple Designs to Cost (©Gilb 101023), continued.

6. **COMPETITIVE THREATS: If many distinct designs, are estimated or shown, to give efficient cost reduction, or to improve a Values/Costs ratio, then you have to assume competitors can do this, and you have to consider meeting or beating that threat.**

7. **SIDE-EFFECTS CAUTION: any new design, independently of your desired purpose (a cost reduction) will have a large number of side-effects, good and bad, on your other critical value-objectives, and other cost budgets, and even other designs, past and future may conflict. So you need to sense this early by side-effect estimation [VIET], and incremental measurement and feedback.**

8. **INCREMENTAL IMPLEMENTATION: if you have identified more than one design, to reach your cost targets, and you are not even sure if you need them all; or need more designs, it is a good idea to implement them incrementally, and see what happens.**

9. **PRIORITIZING DESIGN IMPLEMENTATION: you can prioritize *which* designs to implement early, by several criteria; one of which is *estimated cost-reduction potential.* An impact estimation table will help us see other options [VIET].**

10. **DECOMPOSING ONE, OR MANY, DESIGNS: Even if a cost-reduction design, seems fairly small, like *'a month to implement incrementally in a real system'*, there are good reasons for decomposing the basic idea, so as to be implemented in much smaller increments, *like a day or a week.* (Musk Mobs use a 12 hour day). Decomposition**



17. Feedback

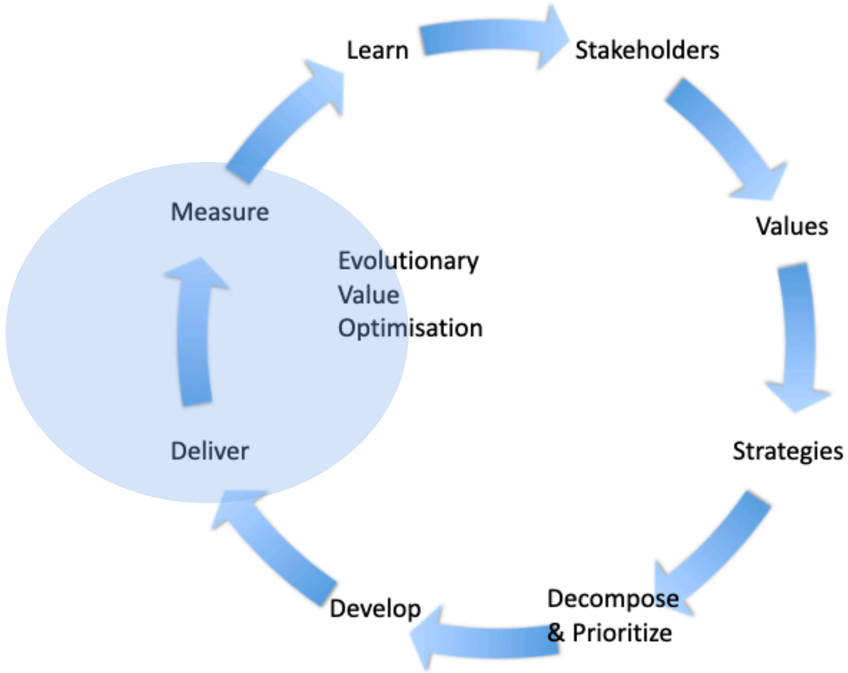
17.1 'Feedback'



My own definition. I am not happy, with conventional definitions.

- '**Feedback**' Defined
  - Any *form of information*, about a defined event, which could potentially be used by any stakeholder, for any set of defined purposes. (V 090824, TsG)

- **Form of Information**: defined: any sound, picture, text, event, etc., related to the 'Defined Feedback Causal Event', which can be *captured*, *analyzed*, and subsequently *used*, for any purpose, by any stakeholder, or witness.
  - **Captured**: in *any form of 'memory'* which allows it to be analyzed.
  - **Analyzed**: 'made sense of' for any *Defined Purpose*.
  - **Used**: applied in a Defined Application, by a Stakeholder, for a *Defined Purpose*.



The Gilb's Planguage Evo Cycle [CE]

17.2 'Feedback'

On Hold

My own definition. I am not happy, with conventional definitions.

- '**Feedback**' Defined Concept *693
- Any form of information, about a defined event, which could potentially be used by any stakeholder, for any set of defined purposes. (V 090824, TsG)

- **Form of Information**: defined: any sound, picture, text, event, etc, related to the 'Defined Feedback Causal Event', which can be captured, analyzed, and subsequently used, for any purpose, by any stakeholder, or witness.
  - **Captured**: in any form of 'memory' which allows it to be analyzed.
  - **Analyzed**: 'made sense of' for any Defined Purpose.
  - **Used**: applied in a Defined Application, by a Stakeholder, for a Defined Purpose.

- So let us home in on **Startup Product Feedback**, *as opposed to* feedback from Business Model, or Organizational Model

- **Startup Product Feedback**
- Feedback Sources: Any Planned Values and Costs
- Especially, those **targeted, for improvement** in an Evo-step
- By **Delivery Step Responsible** team
- With the intent of ,

  1. Determining if the Step is **successful for primary Value Purposes,**

  2. Determining if the step has produced **unwanted side-effects**.

  3. Determining **if the step needs re-design and retry**, or if we can proceed with another Evo-step.

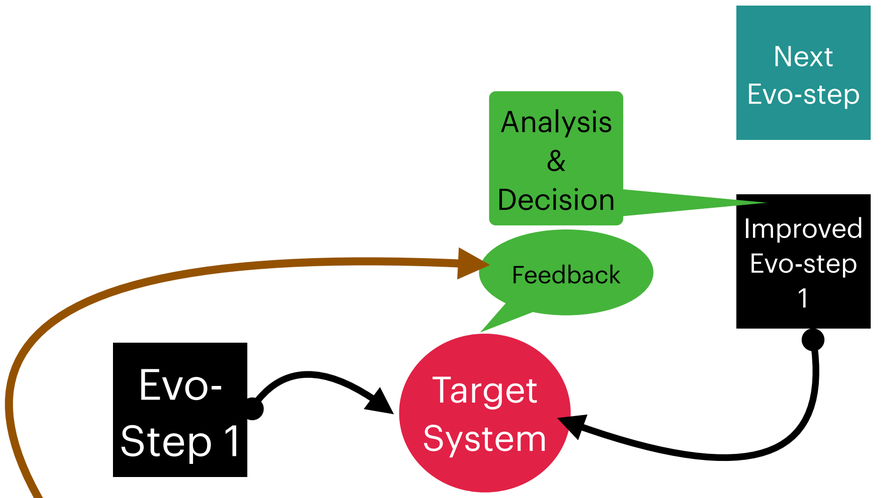


Figure 17.2.1 Basic Feedback from Evo-step Delivery

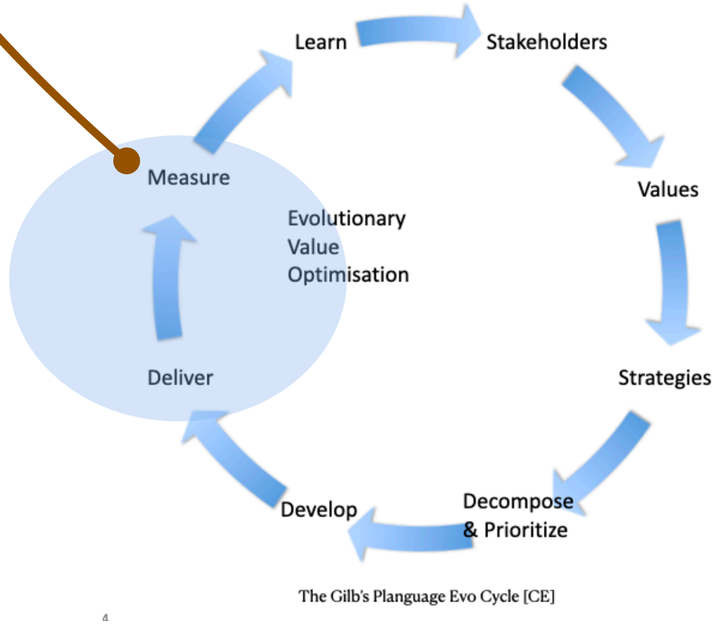
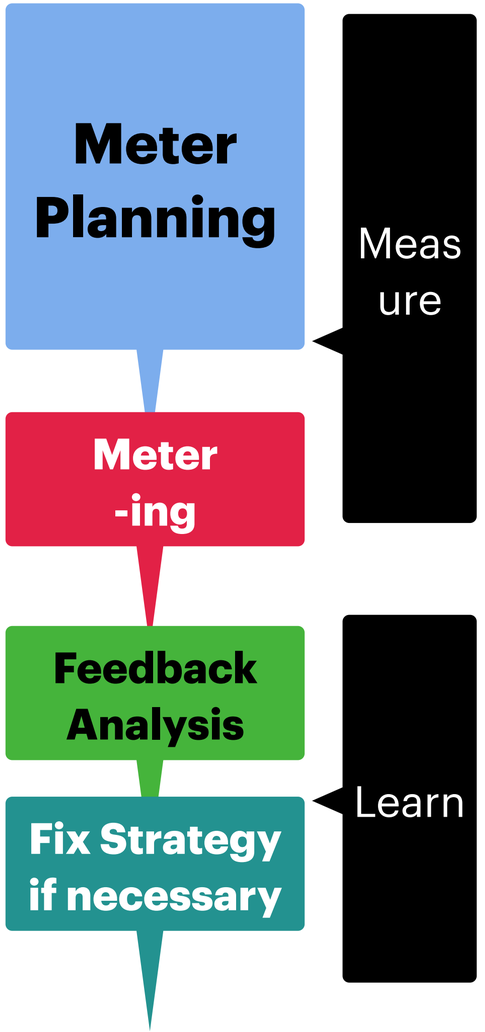


Figure 17.2.2 The Evo-cycle 'Deliver' triggers opportunity for the 'Learn' process, which is based on 'Feedback'

17.3. Feedback Evo simple Model

Normal Feedback Pattern, Measure-Learn Process

- After an Evo-step is Delivered to the system
- There is a note of the Primary Value we are trying to deliver
- There is an estimate, in an Impact Estimation Table, of the degree we will deliver with this evo step
- There is a planned way to measure the Value Effect
- And a Plan to pick up other signals, and any unwanted side effects
- The Meter, ideally is a ‘same day’ measure, but some Meters will take days to a week, for initial feedback
- When sufficient data is collected, maybe an early stage (same day) and a later stage (after a week) , an Analysis Process can take place (say an hour),
- followed by an Action Decision
- And Action Implementation



On Hold

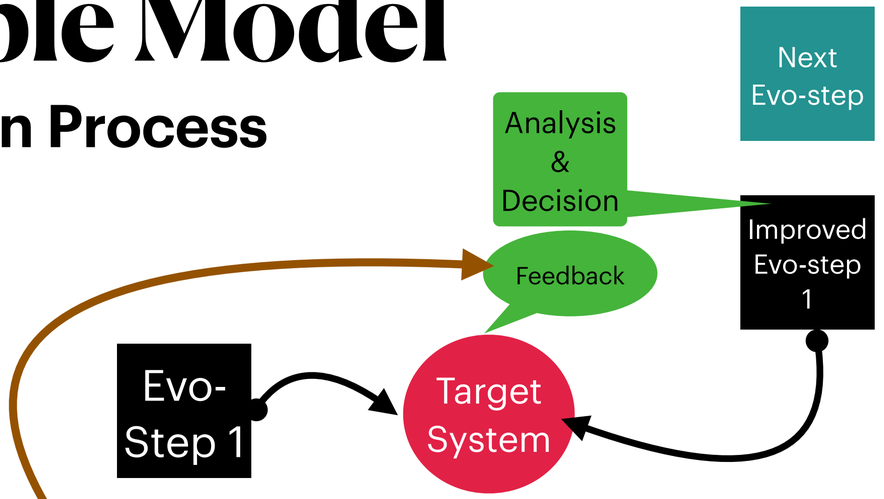


Figure 17.2.1 Basic Feedback from Evo-step Delivery

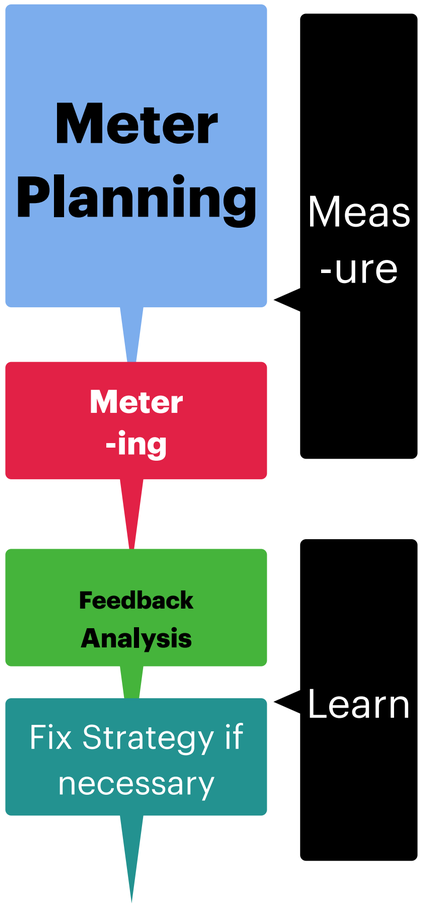


The Gilb's Planguage Evo Cycle [CE]

17.4. Feedback Evo simple Model: Examples

Normal Feedback Pattern, Measure-Learn Process

- **DELIVER: After an Evo-step is Delivered to the system**
  - It has to be delivered such that it will *get used*, and the *use can be measured*
  - We also assume there is a *possibility of retreat*, of switching off the increment, if it is sensed as a disaster
- **VALUE PLAN: There is a note of the Primary Value we are trying to deliver**
  - For example *Usability.Errors*
- **VALUE INCREMENT: There is an estimate, in an Impact Estimation Table, of the values impact, which we will deliver, with this Evo step**
  - *For example: 'about 20% reduction in user error rate, same day, as the new increment is delivered'.*
- **<u>EVO MEASURE PROCESS:</u>**
  - **VALUE METER: There is a planned way to measure the Value Effect**
    - Example: *'Meter: 100 transactions, 10 users, STAGE 1: the automatically detected Error Rate, Stage 2: Manual audit to find error NOT caught by automation'.*
  - **SIDE EFFECTS**: and a Plan to pick up *other* signals, and any *unwanted* side-effects.
    - *'Meter (Side Effects]: User Interview feedback, Transaction Performance'.*
  - **METER SPEED: The Meter, ideally is a 'same day' measure, but some Meters will take days to a week, for initial feedback**
    - If possible, and economic, use 'local meters' for this Value, and for the short term
    - But, be able to turn these meters off, later; when value levels are as planned. *(Musk advised this [MM])*
- **<u>EVO LEARN PROCESS</u>**
  - **ANALYSIS: When sufficient data is collected, maybe an early stage (same day), and at a later stage (after a week) , an Analysis Process can take place (say an hour).**
    - Who should do this? The **Strategy Designers**, of course; they need *feedback* on their design, and can correct it.
  - **ACTION DECISION: followed by an Action Decision**
    - If Values are as planned, and there are no bad side-effects, then we can document an 'OK' decision, and work on next Evo-step.
- **IMPROVED EVO STRATEGIES: DEVELOP strategies, and DELIVER them**
  - **DO IT: And Action Implementation**
  - If necessary, or opportune, re-design the Evo-step (both the strategy design, and its implementation design, and maybe the Meters,
  - When the improved Evo-step has priority, repeat the Deliver process; and then the Measure/Learn process .



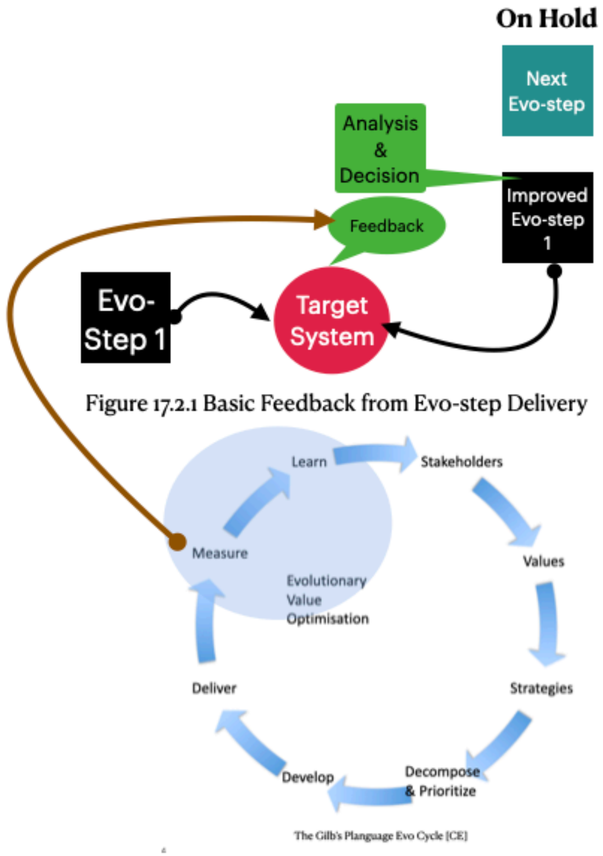


Figure 17.2.1 Basic Feedback from Evo-step Delivery



18. Specialist Roles

18.1 What are ‘Roles’

- ‘Roles’ are NOT a ‘job’, a ‘title’
- They are something you ‘do’, hopefully on a small self-managed team, like at Haier
- You can learn on the job from more-experienced guides

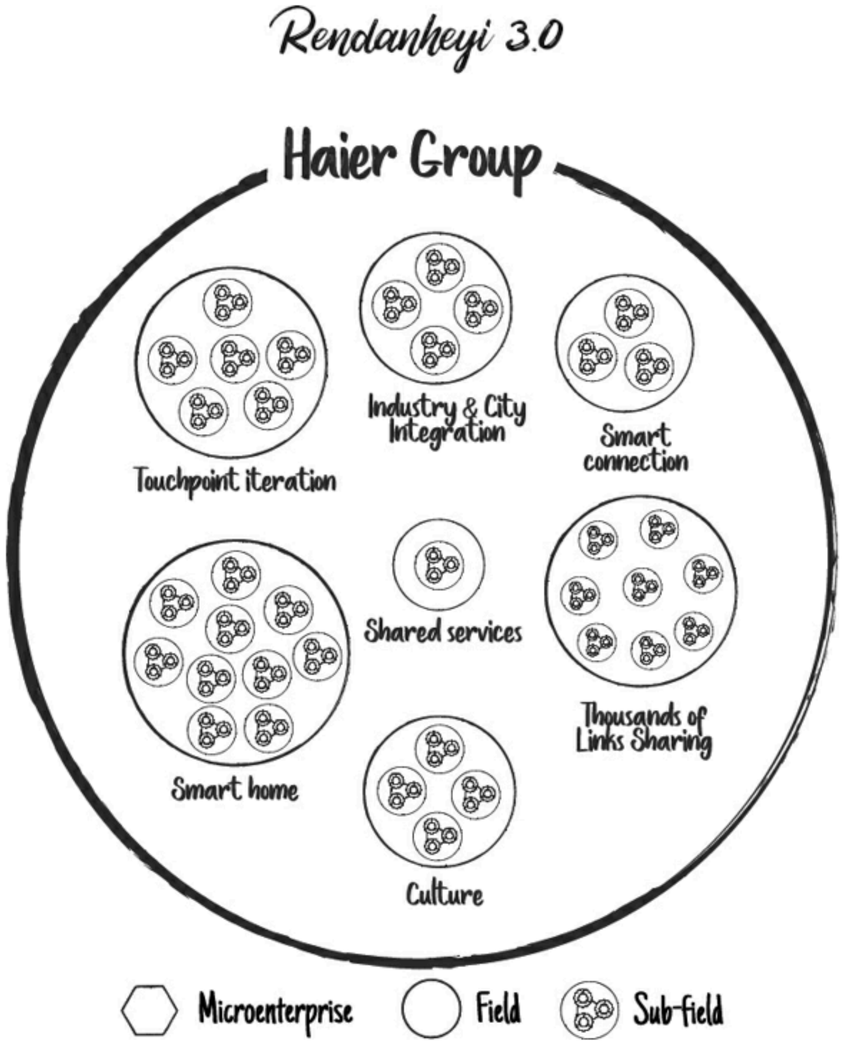


Figure 1.5, Source [44]

‘Dynamic Entrepreneurship’ Version 3.0

From [Productivity Stream]

18.2 Roles, re-defined

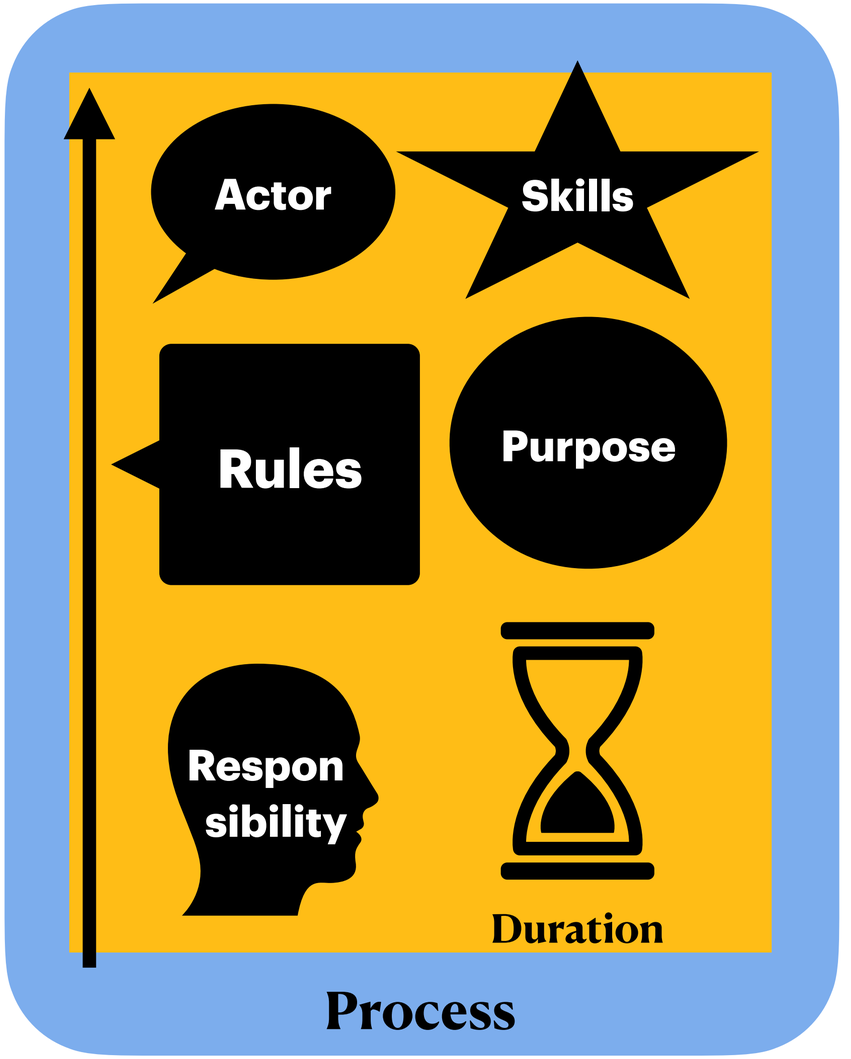
Role Concept *253 February 27, 2003

A role is a defined responsibility, interest or scope for people.

Related Concepts [Role *253]:
• Role [SQC] *411: Role as used in Specification Quality Control (SQC).
• Stakeholder *233

Type [Role *253]: Organizational Concept.

- Let me try to enrich the 'role' definition, from [CE, 2003] in 2024.
- A '**role**' is
  - A defined **Process**, carried out by,
    - Defined **Actors**, with
    - Defined **Skills**, acting according to
    - Defined **Rules**, for
    - Defined **Purposes**, with a
    - Defined **Responsibility**, and a
    - Defined **Duration**, leading to a
    - Defined Role **Result**
- Synonyms: (sort of)
  - Process Responsible
  - Task Person
  - Process Implementer
- NOT Synonyms
  - Activity
  - Job
  - Assignment
  - Position
  - Stakeholders



Role

18.3 Process

A work activity, defined sequence

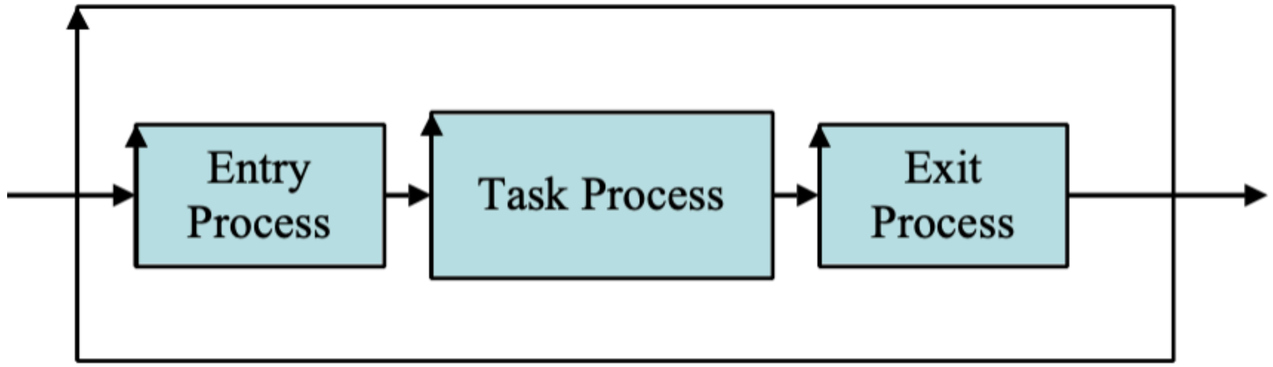
Process Concept *113 February 23, 2003tg

A process is a work activity consisting of:

- an entry process, which examines entry conditions
- a task process, which follows a procedure defining the task. There might also be an associated verification process, such as test or quality control
- an exit process, which examines exit conditions

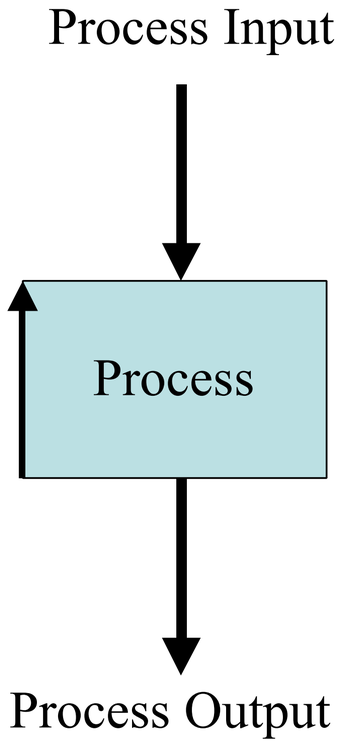
Processes transform inputs to outputs, using resources, and display their own performance and resource (cost) characteristics.

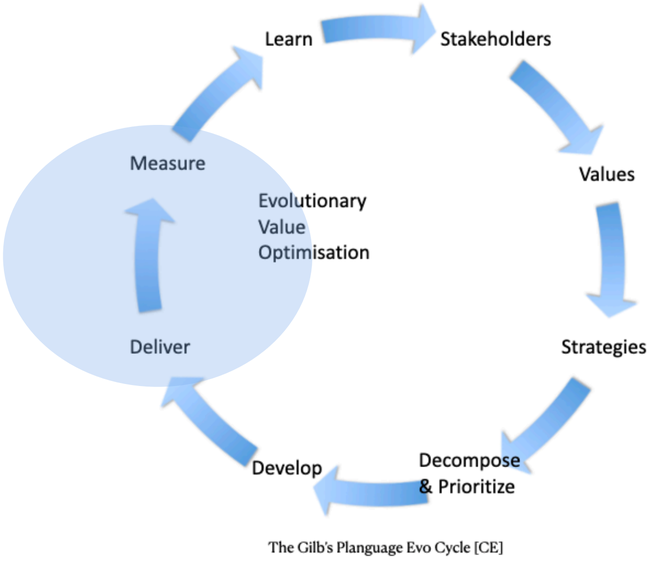
Processes, like 'Rules', capture wisdom, logic, culture, and well-intended advice, from the experienced, to the less-experienced.



A process with its three main component processes.

See many examples of Process in [CE] and later in this book.





The Evo Cycle Process

18.4 Rule

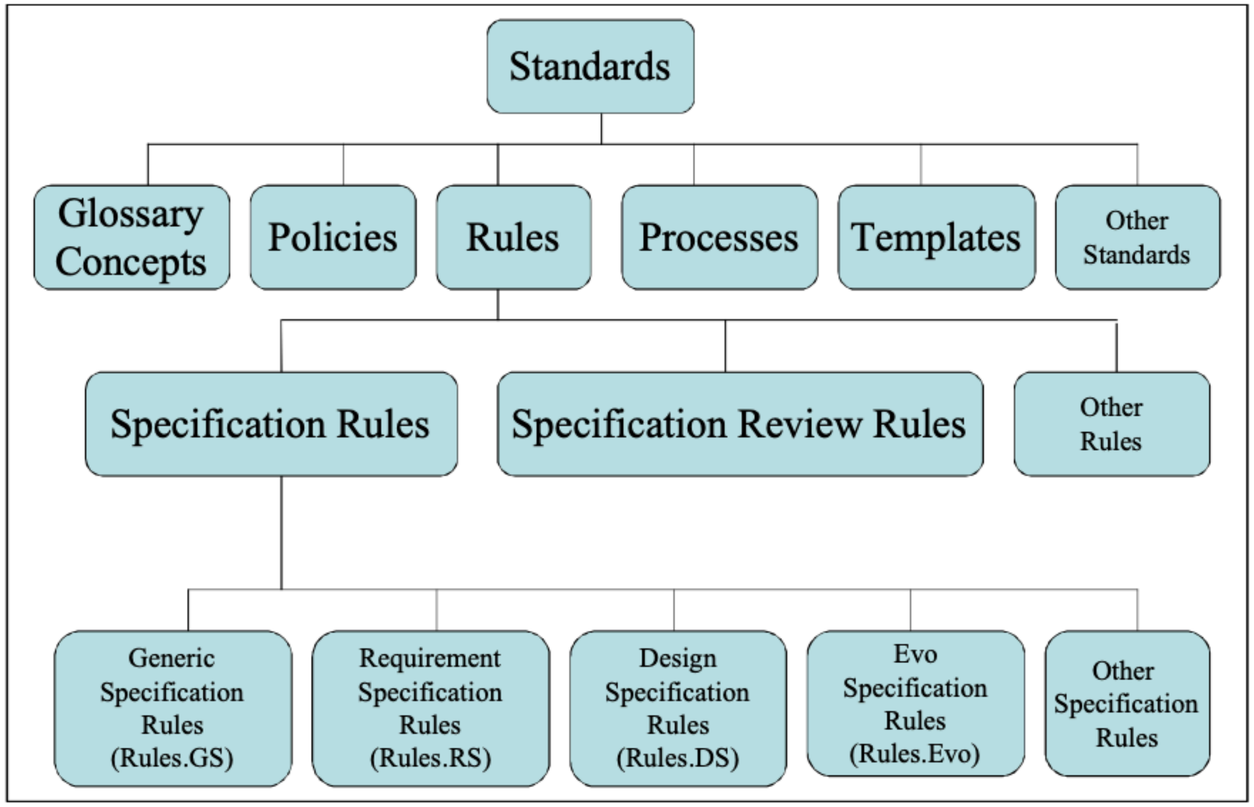
'How'

Rule **Concept *333 May 25, 2003**

A rule is any statement of a standard on how to write or carry out some part of a systems engineering or business process.

Rules are intended to capture experience, advice, wisdom: good guides to getting things right.

They make up part of our our 'culture'.



Rules, in the larger context of other 'Standards'.

See many examples of Rules in [CE] and later in this book.

18.5 Some startup Role Examples

- : PROMPT TO PERPLEXITY:
  - build me a table based on these Role ideas.
- A defined Process, carried out by,
- Defined Actors, with
- Defined Skills, acting according to
- Defined Rules, for
- Defined Purposes, with a
- Defined Responsibility, and a
- Defined Duration, leading to a
- Defined Role Result
- and add to the table 10 Roles for a social startup, using languages, which include Founders, Software Developers, Planners, Designers, and Quality Controllers, Fill the intersecting cells with reasonable examples for a small european based educational field startup.



Role

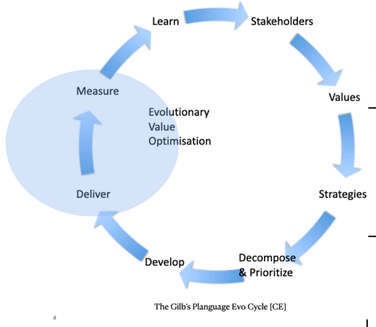
Role	Defined Process	Defined Actors	Defined Skills	Defined Rules	Defined Purposes	Defined Responsibility	Defined Duration	Defined Role Result
Founder	Vision and strategy development	Founder(s)	Leadership, strategic planning	Business ethics, compliance	Establishing the startup's vision	Overall leadership and decision-making	Long-term	Successful startup launch and growth
Software Developer	Software design and coding	Developers	Programming, problem-solving	Coding standards, agile	Creating educational software	Developing and maintaining software	Project-based	Functional and user-friendly software
Planner	Project planning and management	Planners	Organization, time management	Project management frameworks	Efficient project execution	Coordinating project timelines	Short to medium-term	Timely project completion
Designer	User interface and experience design	Designers	Creativity, design software	Design guidelines, usability	Enhancing user engagement	Designing intuitive interfaces	Project-based	Visually appealing and functional designs
Quality Controller	Testing and quality assurance	Quality controllers	Attention to detail, analytical	Quality standards, testing	Ensuring product quality	Conducting tests and reporting issues	Ongoing	High-quality, reliable products
Marketing Specialist	Marketing strategy and execution	Marketing team	Communication, digital marketing	Brand guidelines, regulations	Increasing brand awareness	Developing marketing campaigns	Ongoing	Increased market presence and engagement
Community Manager	Community engagement and support	Community managers	Communication, empathy	Community guidelines	Building a supportive community	Engaging with users and stakeholders	Ongoing	Strong community relationships
Operations Manager	Operational planning and execution	Operations managers	Leadership, logistics	Operational policies	Streamlining operations	Overseeing daily operations	Long-term	Efficient and effective operations
Finance Officer	Financial planning and management	Finance officers	Financial analysis, budgeting	Financial regulations	Ensuring financial health	Managing budgets and financial reports	Long-term	Financially sustainable startup
Educational Specialist	Curriculum development and implementation	Educational experts	Pedagogy, curriculum design	Educational standards	Delivering quality education	Creating and updating educational content	Ongoing	Effective and engaging educational programs

18.5.2 Evo Cycle Roles



- I fed *this* book pdf into Perplexity, and asked

- ***"Good, now do the same table as just before, but for Roles, base it on the Evo cycle (Stakeholder Analysis, around to Learn)"***



The Evo Cycle Process

Role	Defined Process	Defined Actors	Defined Skills	Defined Rules	Defined Purposes	Defined Responsi-bility	Defined Duration	Defined Role Result
Stakeholder Analyst	Identifying and analyzing stakeholders	Analysts	Analytical, communication	Stakeholder engagement policies	Understanding stakeholder needs	Mapping stakeholder values and needs	Project-based	Comprehensive stakeholder map
Value Planner	Quantifying and clarifying values	Planners	Strategic planning, data analysis	Value management guidelines	Defining critical stakeholder values	Setting clear value objectives	Short-term	Clear and measurable value goals
Solution Architect	Designing solutions to deliver values	Architects	Problem-solving, design thinking	Design and architecture standards	Creating feasible solutions	Developing top-level architecture	Project-based	Effective solution designs
Decom-poser	Breaking down solutions into components	Decomposers	Analytical, detail-oriented	Decomposition procedures	Simplifying complex solutions	Detailing design components	Short-term	Detailed component specifications
Developer	Developing design components	Developers	Technical, coding	Development standards	Building functional components	Coding and integrating components	Iterative	Working software components
Deliverer	Delivering design components to system	Delivery team	Project management, logistics	Delivery protocols	Implementing solutions	Coordinating delivery and integration	Short-term	Successful component delivery
Evaluator	Measuring results of design increments	Evaluators	Analytical, evaluation	Measurement procedures	Assessing value delivery	Conducting tests and analysis	Iterative	Accurate performance data
Dynamic Designer	Improving designs based on feedback	Designers	Creative, adaptive	Feedback integration rules	Enhancing design effectiveness	Adjusting designs to improve outcomes	Iterative	Optimized design solutions
Cycle Manager	Managing the Evo cycle process	Managers	Leadership, process management	Evo cycle guidelines	Ensuring continuous improvement	Overseeing the entire Evo cycle	Ongoing	Efficient and effective cycle management
Learner	Capturing and applying lessons learned	Learning team	Reflective, knowledge management	Learning and development policies	Enhancing future cycles	Documenting and sharing insights	Ongoing	Improved processes and outcomes

18.6 Some Value Planning Roles to Consider, [Value Design]

Organizing the Value Design Planning process.
Who are they players to make value delivery really happen? What responsibilities and skills do they need?
Here are some suggestions for specialist roles. These are mainly responsibilities, rather than full time jobs. But they do require training and knowledge.

Value Analyst: analyzes stakeholder needs, and priorities, and selects critical, or possibly critical, needs and specified them as requirements, at least at the 'Wish' level (potential Goal requirement).

Specification Owner: a person (or group) which has undertaken responsibility, by name, for the update and maintenance of a specification object, such as a requirement, a design, or a table.

Implementation Responsible: a person (or group) which has taken named responsibility, as specified in the specification object, for actual practical implementation of a design object. This can be for a requirement level (reach the requirement Goal), or for a design (deliver the design and try to get the maximum value from it).

Value Designer: a generic (all possible design areas) designer (or team) who undertakes to identify possible design components to reach a Value Requirement level, on time. To research them as to all side-effects and costs, documenting such facts in the design object and corresponding Value Tables. The Value Designer might hand over exploration of a design idea to a Specialist.

Value Engineering Specialist: a designer with a narrow speciality (usability, security, performance, organizational improvement, AI) who is updated on the state of the art, and has a good international network of people and sources to find good specialist designs.

Value Architect: A person or team, who sits at the Apex of the system, and synchronizes all ongoing efforts in order to get maximum necessary value for available resources. Manages the top critical values, and the top level design architecture.

Stakeholders: the dozens or more sources of requirements which the designers need to know about, and make sure their design fits them.

Value Process Manager: a person or team responsible for getting a best possible value stream flowing from the other people involved. Sort of like old project manager, except they are focussed on the Values/Costs numbers, not building stuff. They allocate resources (money, time), and assign people to specialist tasks.

Value Quality Control: these people carry out Specification Quality Control of specifications, to make sure the Defects Per Page is economically low enough before Exit to any other process. They are also responsible for measuring value levels and costs after incremental implementation. They will check that designs are in fact implemented as specified by suppliers for Exiting to integration delivery.

Value Policy: this is the written policy that gives clear guidance to the Value process, from organizational management. Perhaps Chief Technical Officer level.

Value Contracting: this contracting can be done at both internal levels and external suppliers, and basically motivate suppliers to deliver value and get rewarded for it.

Value Suppliers/Sub-contractors: internal and external to the organization people or organizations who undertake a specific responsibility, for construction, implementation, organizing, designing, or any other activity needed to deliver value.

Value Design Builders: people and organizations who build, physically, logically or organizationally, any design component or related activity.

Value Director: the person, or group, responsible for focusing on the Value Delivery, and reporting to a steering committee or Board about the plans and accomplishments to date in Value



[Value Design] Book

18.6.3 Value Role Table

By ‘Role’ definition

Based on the roles outlined in the book manuscript "Startup: Social Values Success - By Engineering Logic," here is a table for the actors involved in organizing the value design planning process, using the provided definition of a role:

This table is structured to ensure that each role is clearly defined with specific processes, actors, skills, rules, purposes, responsibilities, durations, and expected results, facilitating efficient value delivery in a social purpose startup.



[Value Design] Book

Role	Defined Process	Defined Actors	Defined Skills	Defined Rules	Defined Purposes	Defined Responsibility	Defined Duration	Defined Role Result
Value Analyst	Analyzes stakeholder needs and priorities	Analysts	Stakeholder analysis, requirement specification	Requirement analysis	Identify critical stakeholder needs	Specify requirements at the 'Wish' level	Project-based	Comprehensive stakeholder requirement map
Specification Owner	Updates and maintains specification objects	Owners	Documentation, specification management	Specification guidelines	Maintain accurate specifications	Ensure up-to-date specifications	Ongoing	Accurate and current specifications
Implementation Responsible	Implements design objects and ensures requirement goals	Implementers	Project management, implementation	Implementation standards	Achieve practical implementation	Ensure design meets requirements	Project-based	Successful implementation of design objects
Value Designer	Identifies and researches design components	Designers	Design thinking, analytical skills	Design research protocols	Meet value requirements	Document design components and costs	Project-based	Effective design solutions
Value Engineering Specialist	Specializes in specific design areas	Specialists	Domain expertise, networking skills	Specialist design standards	Provide expert design solutions	Research and apply state-of-the-art designs	Ongoing	Innovative and specialized design solutions
Value Architect	Synchronizes efforts to maximize value delivery	Architects	Strategic planning, leadership	Architectural guidelines	Manage top-level design architecture	Oversee value delivery synchronization	Ongoing	Optimized value delivery
Value Process Manager	Manages the value stream and allocates resources	Managers	Value management, resource allocation	Process management policies	Ensure efficient value stream	Assign tasks and allocate resources	Ongoing	Efficient and effective value stream
Value Quality Control	Ensures specifications meet quality standards	Quality controllers	Quality assurance, analytical skills	Quality control standards	Maintain low defect levels	Measure value levels post-implementation	Ongoing	High-quality specifications and implementations
Value Policy	Provides guidance for the value process	Policy makers	Policy development, strategic alignment	Policy guidelines	Guide value process	Develop and enforce value policies	Ongoing	Aligned and effective value process
Value Contracting	Manages contracts to motivate suppliers	Contractors	Contract management, negotiation	Contractual agreements	Ensure value delivery by suppliers	Manage supplier contracts	Project-based	Successful supplier value delivery
Value Suppliers/ Sub-contractors	Undertakes specific responsibilities	Suppliers	Project execution, collaboration	Supplier agreements	Deliver construction or design tasks	Fulfill specific responsibilities	Project-based	Completed construction or design tasks
Value Design Builders	Builds design components or related activities	Builders	Construction, design implementation	Building standards	Construct design components	Implement design activities	Project-based	Completed design components
Value Director	Focuses on value delivery and reports progress	Directors	Leadership, communication	Reporting protocols	Oversee value delivery	Report to stakeholders on progress	Ongoing	Successful value delivery and reporting

18.6.5 List of Roles in [Startup] Book

PROMPT: Based on the provided text from the book manuscript "Startup: Social Values Success - By Engineering Logic," here are the roles identified, along with their definitions and references to the chapters and subsections where they are mentioned:

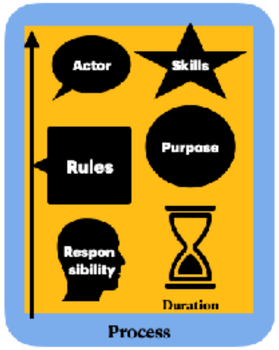
These roles are structured to support a social startup in achieving its goals efficiently by focusing on stakeholder values and continuous improvement through the Evo cycle

Based on the provided text from the book manuscript "Startup: Social Values Success - By Engineering Logic," here are the roles identified, along with their definitions and references to the chapters and subsections where they are mentioned:

1. **Founder: Responsible for establishing the startup's vision and overall leadership (Chapter 1).**
2. **Stakeholder Analyst: Identifies and analyzes critical stakeholder values (Chapter 6).**
3. **Value Planner: Quantifies and clarifies stakeholder values, setting clear objectives (Chapter 3).**
4. **Solution Architect: Designs solutions to deliver stakeholder values within constraints (Chapter 10).**
5. **Decomposer: Breaks down top-level architecture into detailed design components (Chapter 12).**
6. **Developer: Develops and integrates design components into the system (Chapter 10).**
7. **Deliverer: Coordinates the delivery and integration of design components (Chapter 9).**
8. **Evaluator: Measures the results of design increments against value requirements (Chapter 17).**
9. **Dynamic Designer: Improves designs based on feedback and measurement results (Chapter 10).**
10. **Cycle Manager: Manages the Evo cycle process to ensure continuous improvement (Chapter 13).**
11. **Learner: Captures and applies lessons learned to enhance future cycles (Chapter 17).**
12. **Marketing Specialist: Develops and executes marketing strategies to increase brand awareness (Chapter 19).**
13. **Community Manager: Engages with users and stakeholders to build a supportive community (Chapter 19).**
14. **Operations Manager: Oversees daily operations to streamline processes (Chapter 21).**
15. **Finance Officer: Manages financial planning and budgeting to ensure sustainability (Chapter 20).**
16. **Educational Specialist: Develops and implements curriculum to deliver quality education (Chapter 18).**
17. **Quality Controller: Conducts testing and quality assurance to ensure product reliability (Chapter 18).**
18. **Planner: Coordinates project timelines and resources for efficient execution (Chapter 2).**
19. **Designer: Creates user interfaces and experiences to enhance engagement (Chapter 10).**
20. **Software Developer: Codes and maintains software to meet design specifications (Chapter 10).**

Additional roles mentioned without detailed definitions include:

- **Evaluator (Chapter 17)**
- **Dynamic Designer (Chapter 10)**
- **Cycle Manager (Chapter 13)**
- **Learner (Chapter 17)**
- **Marketing Specialist (Chapter 19)**
- **Community Manager (Chapter 19)**
- **Operations Manager (Chapter 21)**
- **Finance Officer (Chapter 20)**
- **Educational Specialist (Chapter 18)**
- **Quality Controller (Chapter 18)**



These roles are structured to support a social startup in achieving its goals efficiently by focusing on stakeholder values and continuous improvement through the Evo cycle

18.6.6 Additional Startup Role Ideas from AI

PROMPT: Can you suggest, and define 30 additional roles, to the ones you listed in previous question, based on social purpose startups in UK, in the technical sector, which would help the startup succeed in delivering 25 competitive values at low costs and times? Please refer to any outside literature to support this.

Based on the concept of social purpose startups in the UK, particularly in the technical sector, here are 30 additional roles that could help such a startup succeed in delivering competitive values efficiently:

1. Impact Analyst: Evaluates the social and environmental impact of projects to ensure alignment with organizational values.
2. Sustainability Officer: Develops and implements strategies to minimize environmental impact.
3. Data Scientist: Analyzes complex datasets to drive decision-making and optimize operations.
4. User Experience Researcher: Conducts research to understand user needs and improve product usability.
5. Partnership Manager: Builds and maintains relationships with key partners and stakeholders.
6. Legal Advisor: Provides legal guidance to ensure compliance with regulations and protect intellectual property.
7. Ethical Hacker: Tests systems for vulnerabilities to enhance cybersecurity.
8. Diversity and Inclusion Officer: Promotes diversity and inclusion within the organization.
9. Customer Success Manager: Ensures customer satisfaction and retention by addressing their needs and concerns.
10. Social Media Strategist: Develops strategies to enhance the organization's online presence and engagement.
11. Grant Writer: Secures funding by writing proposals for grants and other funding opportunities.
12. Innovation Manager: Drives innovation by identifying new opportunities and fostering a culture of creativity.
13. Product Manager: Oversees product development to ensure alignment with market needs and organizational goals.
14. Quality Assurance Tester: Conducts testing to ensure products meet quality standards before release.
15. Community Outreach Coordinator: Engages with the community to promote the organization's mission and initiatives.
16. Volunteer Coordinator: Manages volunteer recruitment, training, and retention.
17. Training and Development Specialist: Designs and implements training programs to enhance employee skills.
18. Procurement Specialist: Manages the acquisition of goods and services to ensure cost-effectiveness.
19. Risk Manager: Identifies and mitigates potential risks to the organization.
20. Financial Analyst: Analyzes financial data to support budgeting and forecasting.
21. Supply Chain Manager: Oversees the supply chain to ensure efficient and timely delivery of products.
22. Crisis Manager: Develops and implements plans to handle emergencies and minimize disruptions.
23. Tech Support Specialist: Provides technical support to resolve IT-related issues.
24. Content Creator: Produces engaging content to communicate the organization's message and values.
25. Cultural Liaison: Bridges cultural gaps to ensure effective communicatio collaboration.
26. Health and Safety Officer: Ensures a safe working environment by impler health and safety protocols.
27. Policy Advisor: Advises on policy development to align with social and or goals.
28. Event Coordinator: Plans and executes events to promote the organizati and engage stakeholders.
29. Brand Manager: Manages the organization's brand identity and reputatio
30. Mentorship Program Coordinator: Develops and manages mentorship pr support employee growth and development.

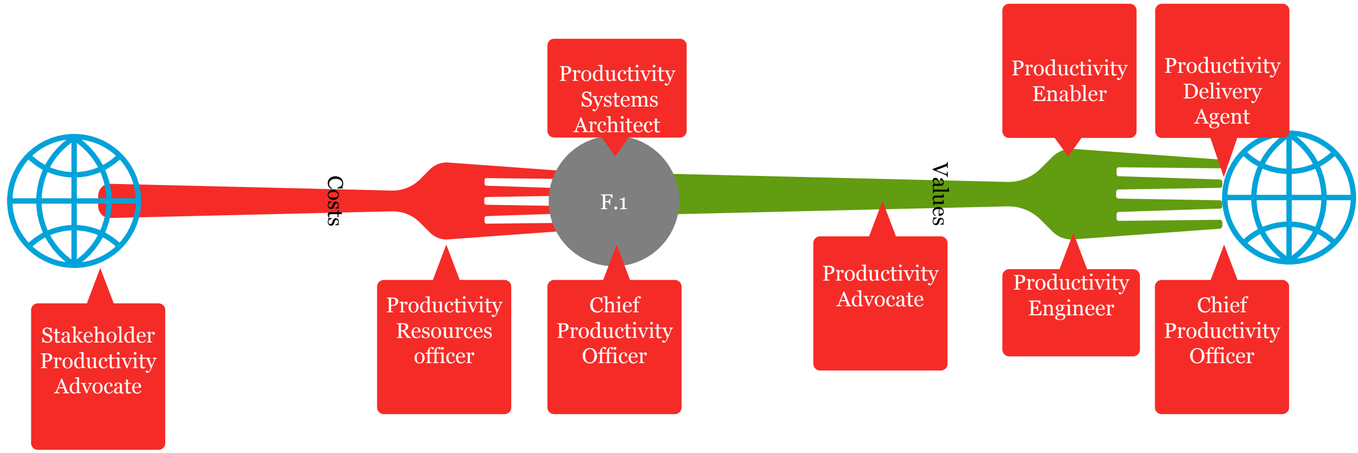


These roles are designed to support a social purpose startup in the technical sector by focusing on efficiency, innovation, and stakeholder engagement to deliver value effectively.

18.7 Productivity Roles

From [Productivity Stream]

- Another view of specialist roles is a productivity focus.



Chief Productivity Officer, Productivity Project Manager, Productivity Engineer, Productivity Systems Architect, Productivity Auditor.



19. Presentations

19.1 Principles of Powerful Presentations

© gilb.com 2024

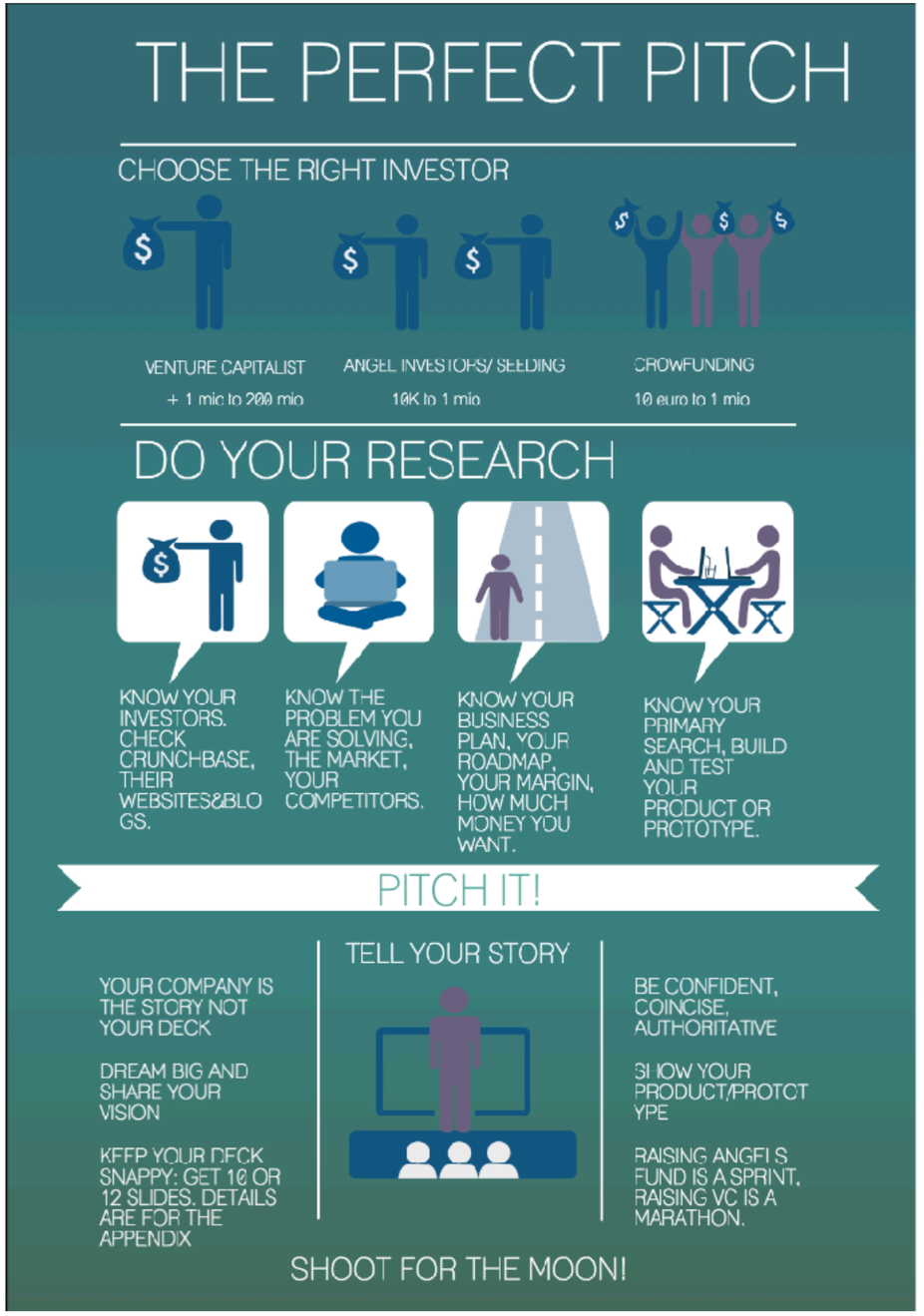
1. **Know your audience**: find out who your audience is, find out what their interests are, trial run with such an audience.
2. **Tell them what they want and need to know:** your accomplishments, status to date, short term plans, need for support (money, advice, contacts).
3. **Summarize on a single slide, and summarize that slide in a single sentence**
4. **Give them high payoff, low risk propositions.**
5. **Explain how you have reduced possibility of disappointment**
6. **Explain how you are focussed on Value Generation, incrementally, and show progress.**
7. **Ask questions, at beginning and midway, and at end to get some quick feedback**
8. **Give a slide with 10 reasons you are going to succeed: *including Planguage methods***
9. **Explain why you have made more efficient use of startup capital than others**
10. **Offer to share your methods with other startups they are investing in.**

19.2 Here are ten principles for effective investor presentations for social startups:

From Perplexity

1. **Time Management: Keep your presentation concise and respect the time constraints of investors. Aim for a 10-minute pitch, allowing additional time for questions.**
2. **Engagement: Make your presentation interesting and engaging. Use storytelling to capture attention and convey your message effectively.**
3. **Clarity and Focus: Be clear and to the point. Avoid unnecessary details and focus on conveying your core message quickly.**
4. **Problem-Solution Framework: Clearly define the problem your startup addresses and present a compelling solution. This helps investors understand the relevance and potential impact of your business.**
5. **Market Opportunity: Highlight the market size and opportunity. Demonstrate the demand for your solution and the potential for growth and profitability.**
6. **Business Model: Explain how your startup plans to make money. Describe your revenue model and growth strategies.**
7. **Competitive Advantage: Identify your competitors and articulate what sets your startup apart. Highlight your unique value proposition.**
8. **Team Introduction: Present your team and their competencies. Investors often look for passionate and experienced teams.**
9. **Financial Expectations: Clearly state how much investment you need and how it will be used. Outline your financial projections and milestones.**
10. **Confidence and Practice: Exude confidence through your body language and delivery. Practice your pitch multiple times to ensure smooth execution.**

These principles aim to create a memorable and compelling presentation that captures investor interest and demonstrates the potential of your social startup.



https://i.pinimg.com/originals/24/70/46/2470464c18bf449b3db5692200f8eec4.png



24. Case Studies

The Confirmit Case Study 2003-2009



Confirmit Case URL
https://www.slideshare.net/slideshow/tom-gilb-project-management-driven-by-the-top-ten-critical-improvements-quantified-bcs-apm-15-april-2013/19004735#64
Contains a set of Confirmit slides

SEE *this* page Presentation notes for more

[Confirmit]

Their product = confirmit®

Chief Storyteller =



Trond Johansen

Customer Successes in Corporate Sector



We gave them a 1 day briefing on our Evo method and Planguage

That’s all they needed to succeed!

They were Real engineers



Real Example of 1 of the 25 Quality Requirements

Usability.Productivity (taken from Confirmit 8.5, performed a set of predefined steps, to produce a standard MR Report. development)

Scale for quantification: Time in minutes to set up a typical specified Market Research-report

Past Level [Release 8.0]: 65 mins.,

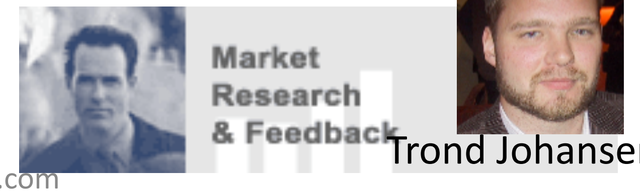
Tolerable Limit [Release 8.5]: 35 mins.,

Goal [Release 8.5]: 25 mins.

Note: end result was actually 20 minutes

☺

Meter [Weekly Step]: Candidates with Reportal experience, and with knowledge of MR-specific reporting features



Trond Johansen

Shift: from Function to Quality

- **Our new focus is on the day-to-day operations of our Market Research users,**
  - **not a list of features that they might or might not like. 50% never used!**
  - **We KNOW that increased efficiency, which leads to more profit, will please them.**
  - **The '45 minutes actually saved x thousands of customer reports'**
    - **= big $$$ saved**
- **After one week we had defined more or less all the requirements for the next version (8.5) of Confirmit.**

Design Process

Design Suggestions



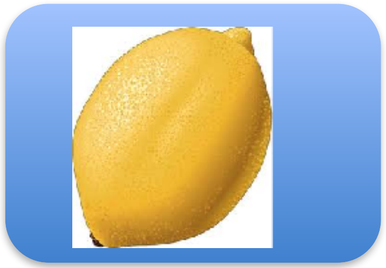
- Design A



- Design B



- Design C



- Design D

Impacts to Cost Evaluation

	A	B	C	D
Goal 1	30%	10%’	-10%	80%
Cost	10	50	1	20
G/C	3:1	1:5	?	4:1

Design Process: The winner

Design Suggestions



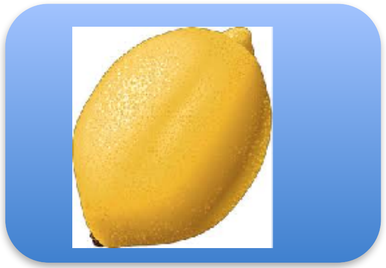
- Design A



- Design B



- Design C



- Design D

Impacts to Cost Evaluation

	A	B	C	D
Goal 1	30%	10%’	-10%	80%
Cost	10	50	1	20
G/C	3:1	1:5	?	4:1


Trond Johansen

- IET for MR Project – Confirmit (<-FIRM Product Brand) 8.5
- **Solution:** Recoding
  - Make it possible to recode variable on the fly from Reportal.
  - Estimated effort: 4 days
  - **Estimated** Productivity Improvement: 20 minutes (50% way to Goal)
  - actual result 38 minutes (95% progress towards Goal)

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

Evo – IET

- Product quality:
  - **Usability.Intuitiveness:**
  - **Scale: Probability that <secret name of stakeholders> can intuitively, and without any help, figure out how to do a set of defined, common, simple tasks correctly (without any**

Current Status		Improvement	Goals			Step 1 (7.-18.Aug)		Step 2 (21.-1.sep)		Step 3 (4.-15.sep)	
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact	Actual Impact	Estimated Impact	Actual Impact
Usability.Intuitiveness											
	9,0	9,0	18	12	8					8	9
	1,5	5,0	6,5	3,0	1,0					4,5	5,0

  - **Meter1: The time it takes for "secret name of stakeholders" (First time users) to create a SimpleSet1 of pre-defined authoring tasks**
  - **Meter2: The number of times "secret name of stakeholders" (First time users) are uncertain of how to perform a step in SimpleSet1**

New slide by Trond October 2 2006

EVO Plan Confirmit 8.5 in **Evo Step Impact *Measurement***

4 product areas were attacked in all: **25 Qualities** concurrently, one quarter of a year. Total development staff = 13

9

8

3

3

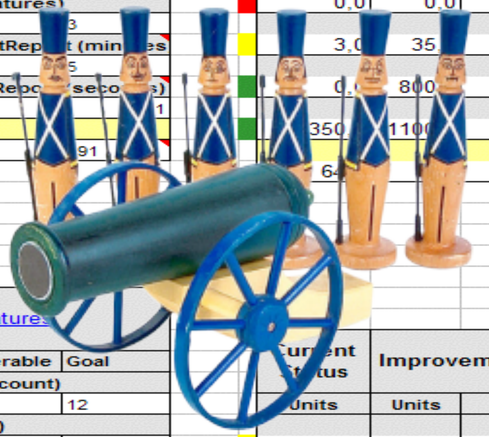
Impact Estimation Table: Reportal codename "Hyggen"

Current Status	Improvements		Reportal - E-SAT features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Intuitivness (%)		
75,0	25,0	62,5	50	75	90
			Usability.Consistency.Visual (Elements)		
14,0	14,0	100,0	0	11	14
			Usability.Consistency.Interaction (Components)		
15,0	15,0	107,1	0	11	14
			Usability.Productivity (minutes)		
5,0	75,0	96,2	80	5	2
5,0	45,0	95,7	50	5	1
			Usability.Flexibility.OfflineReport.ExportFormats		
3,0	2,0	66,7	1	3	4
			Usability.Robustness (errors)		
1,0	22,0	95,7	7	1	0
			Usability.Replacability (nr of features)		
4,0	5,0	100,0	8	5	3
			Usability.ResponseTime.ExportRep t (min es		
1,0	12,0	150,0	13	13	5
			Usability.ResponseTime.ViewRep se co s)		
1,0	14,0	100,0	15		1
			Development resources		
203,0			0		91

Current Status	Improvements		Survey Engine .NET		
Units	Units	%	Past	Tolerable	Goal
			Backwards.Compatibility (%)		
83,0	48,0	80,0	40	85	95
0,0	67,0	100,0	67	0	0
			Generate.WI.Time (small/medium/large seconds)		
4,0	59,0	100,0	63	8	4
10,0	397,0	100,0	407	100	10
94,0	2290,0	103,9	2384	500	180
			Testability (%)		
10,0	10,0	13,3	0	100	100
			Usability.Speed (seconds/user rating 1-10)		
774,0	507,0	51,7	1281	600	300
5,0	3,0	60,0	2	5	7
			Runtime.ResourceUsage.Memory		
0,0	0,0	0,0		?	?
			Runtime.ResourceUsage.CPU		
3,0	35	97,2	38	3	2
			Runtime.ResourceUsage.MemoryLeak		
0	800	100,0	800	0	0
			Runtime.Concurrency (number of users)		
350	1100	146,7	150	500	1000
64			Development resources		
			0		84

Current Status	Improvements		Reportal - MR Features		
Units	Units	%	Past	Tolerable	Goal
			Usability.Replacability (feature count)		
1,0	1,0	50,0	14	13	12
			Usability.Productivity (minutes)		
20,0	45,0	112,5	65	35	25
			Usability.ClientAcceptance (features count)		
4,4	4,4	36,7	0	4	12
			Development resources		
101,0			0		86

Current Status	Improvements		XML Web Services		
Units	Units	%	Past	Tolerable	Goal
			TransferDefinition.Usability.Efficiency		
7,0	9,0	81,8	16	10	5
17,0	8,0	53,3	25	15	10
			TransferDefinition.Usability.Response		
943,0	-186,0	######	170	60	30
			TransferDefinition.Usability.Intuitiveness		
5,0	10,0	95,2	15	7,5	4,5
			Development resources		
2,0			0		48





Confirmit **Evo Weekly Value Delivery Cycle**

	Development Team	Users (PMT, Pros, Doc writer, other)	CTO (Sys Arch, Process Mgr)	QA (Configuration Manager & Test Manager)
Friday	✓ PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting ✓ PM: Attend Project Mgmt meeting: 12.00-15.00 ✓ Developers: Focus on genereal maintenance work, documentation.		✓ Approve/reject design & Step N ✓ Attend Project Mgmt meeting: 12-15	✓ Run final build and create setup for Version N-1. ✓ Install setup on test servers (external and internal) ✓ Perform initial crash test and then release Version N-1
Monday	✓ Develop test code & code for Version N	✓ Use Version N-1		✓ Follow up CI ✓ Review test plans, tests
Tuesday	✓ Develop Test Code & Code for Version N ✓ Meet with users to Discuss Action Taken Regarding Feedback From Version N-1	✓ Meet with developers to give Feedback and Discuss Action Taken from previous actions	✓ System Architect to review code and test code	✓ Follow up CI ✓ Review test plans, tests
Wednesday	✓ Develop test code & code for Version N			✓ Review test plans, tests ✓ Follow up CI
Thursday	✓ Complete Test Code & Code for Version N ✓ Complete GUI tests for Version N-2			✓ Review test plans, tests ✓ Follow up CI

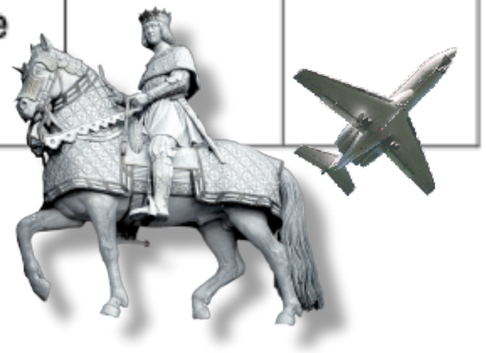




Evo's impact on Confirmit product qualities 1st Qtr

- Only 5 highlights of the 25 impacts are listed here

Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid	15 min	5 min
Performance.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and an response time<500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]	250 users	6000



© Tom@Gilb.com www.gilb.com

Initial Experiences and conclusions

- **EVO has resulted in**
  - **increased motivation and**
  - **enthusiasm amongst developers,**
  - **it opens up for empowered creativity**



- **Developers**
  - **embraced the method and**
  - **saw the value of using it,**
  - **even though they found parts of Evo difficult to understand and execute**

confirmit



Trond Johansen

Evo's impact on confirmit product qualities - 1

- The impact described is based on:
  - Internal usability test, productivity tests ++
  - Performance tests carried out at Microsoft Windows ISV laboratory in Redmond USA
  - Direct customer feedback
    - *"I just wanted to let you know how appreciative we are of the new "entire report" export functionality you recently incorporated into the Reportal.*
    - *It produces a fantastic looking report, and the table of contents is a wonderful feature.*
    - *It is also **a HUGE time saver." <- Customer***
  - "These leaps in product qualities would not have been achieved without Evo". <- TJ





Trond Johansen

Conclusions -

- **The method's positive impact on Confirmit product qualities has convinced us that**
  - **Evo is a better suited development process than our former waterfall process, and**
  - **we will continue to use Evo in the future.**
- **What surprised us the most was**
  - **the method's power of focusing on delivering value for clients versus cost of implementation.**
  - **Evo enables you to re-prioritize the next development-steps based on the weekly feedback.**
  - **What seemed important**
    - **at the start of the project**
    - **may be replaced by other solutions**
    - **based on knowledge gained from previous steps.**
- **The method has**
  - **high focus on measurable product qualities, and**
    - **defining these clearly and testably, requires training and maturity.**
  - **It is important to believe that everything can be measured,**
    - **and to seek guidance if it seems impossible.**







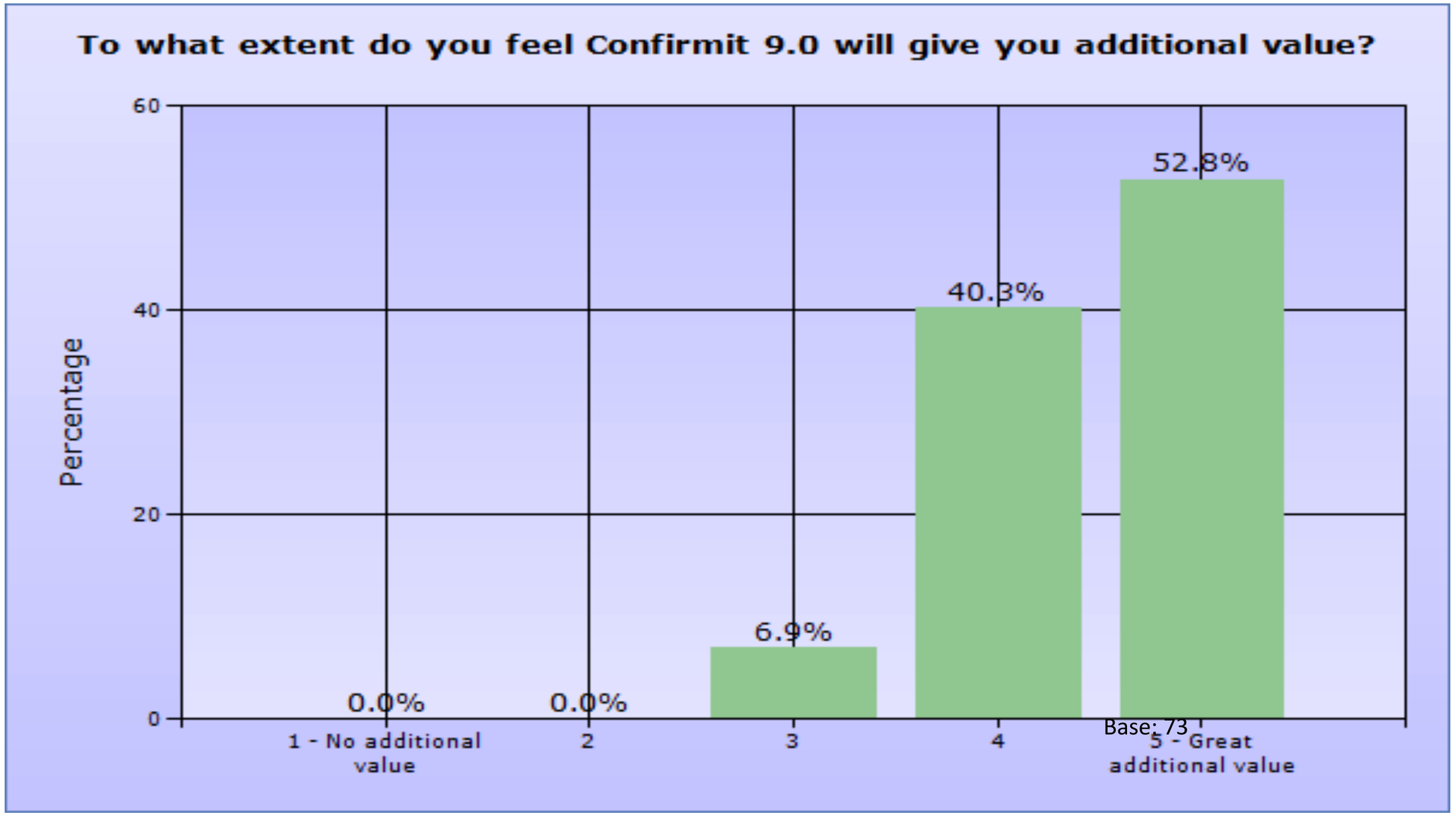


Trond Johansen

Initial Customer Feedback on the new Confirmit 9.0

November 24th, 2004

Initial perceived value of the new release (Base 73 people)



Evo's impact on Confirmit 9.0 product qualities
Results from the second quarter of using Evo. 1/2

Product quality	Description	Customer value
Intuitiveness	Probability that an inexperienced user can intuitively figure out how to set up a defined Simple Survey correctly.	Probability increased by **175%**
Productivity	Time in minutes for a defined advanced user, with full knowledge of 9.0 functionality, to set up a defined advanced survey correctly.	Time reduced by **38%**

Product quality	Description	Customer value
Productivity	Time (in minutes) to test a defined survey and identify 4 inserted script errors, starting from when the questionnaire is finished to the time testing is complete and is ready for production. (Defined Survey: Complex survey, 60 questions, comprehensive JScripting.)	Time reduced by **83%** and error tracking increased by 25%

© Tom@Gilb.com www.gilb.com

Evo's impact on Confirmit 9.0 product qualities
Results from the second quarter of using Evo. 2/2

Product quality	Description	Customer value
Performance	Max number of panelists that the system can support without exceeding a defined time for the defined task, with all components of the panel system performing acceptable.	Number of panelists increased by **1500%**
Scalability	Ability to accomplish a bulk-update of X panelists within a timeframe of Z second	Number of panelists increased by **700%**
Performance	Number of responses a database can contain if the generation of a defined table should be run in 5 seconds.	Number of responses increased by **1400%**

Code quality – ”green” week

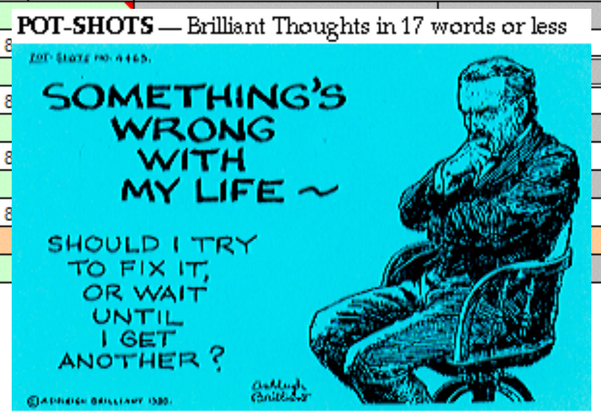
Confirmit (2005) Norway

decided to design ‘ease of change’ in, to a legacy system, in one-week delivery-cycles, per month, using ‘Evo’ Agile

‘Refactoring to reduce technical debt’ -> Re-Engineering

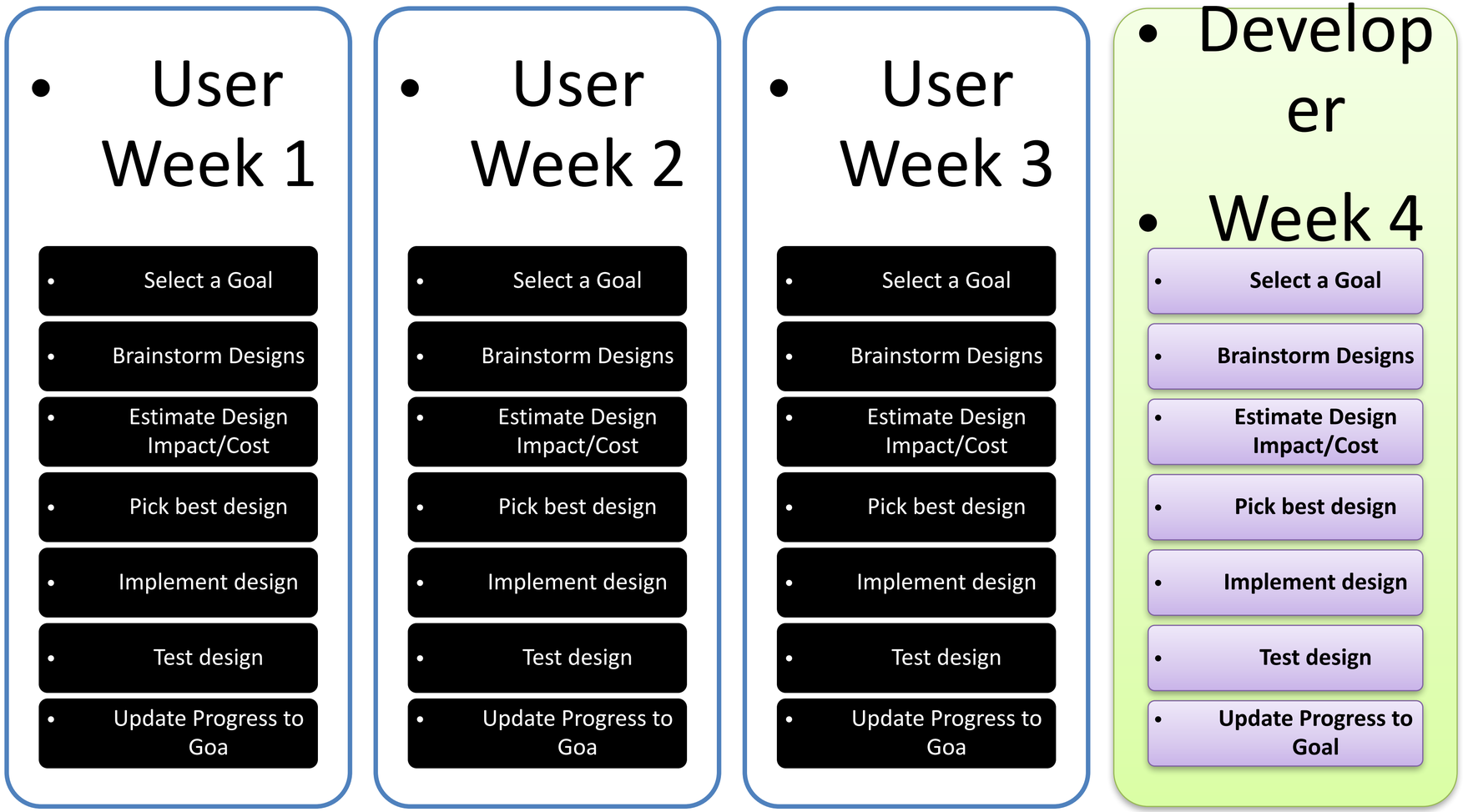
- In these ”green” weeks, some of the deliverables will be less visible for the end users, but more visible for our QA department.
- We manage code quality through an Impact Estimation table.

Current Status		Improvement	Goals			Step 6 (week 14)		Step 7 (wee	
	Units		Past	Tolerable	Goal	Estimated Impact	Actual Impact	Estimated Impact	A
	100,0	100,0	0	80	100			100	
Speed									
	100,0	100,0	0	80	100	100	100		
Maintainability.Doc.Code									
	100,0	100,0	0	80	100	100	100		
InterviewerConsole									
NUnitTests									
	0,0	0,0	0	90	100				
PeerTests									
	100,0	100,0	0	90	100			100	
FxCop									
	0,0	10,0	10	0	0				
TestDirectorTests									
	100,0	100,0	0	90	100			100	
Robustness.Correctness									
	2,0	2,0	0	1	2	2	2		
Robustness.BoundaryConditions									
	0,0	0,0	0	8					
Speed									
	0,0	0,0	0	8					
ResourceUsage.CPU									
	100,0	0,0	100	8					
Maintainability.Doc.Code									
	100,0	100,0	0	8					
SynchronizationStatus									
NUnitTests									



- Speed
- Maintainability
- Nunit Tests
- PeerTests
- TestDirectorTests
- Robustness.Correctness
- Robustness.Boundary Conditions
- ResourceUsage.CPU
- Maintainability.DocCode
- SynchronizationStatus

The Monthly ‘Green Week’

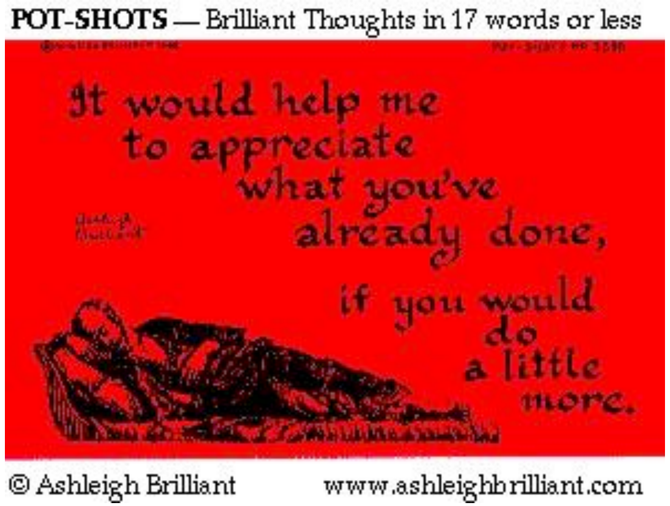


Initial qualitative feedback on the new release

" ... keep up the good work."

"It looks like you have listened to the people that actually use the software daily and aimed to make it easier for them ... "

“I was very impressed with the version 9.0”

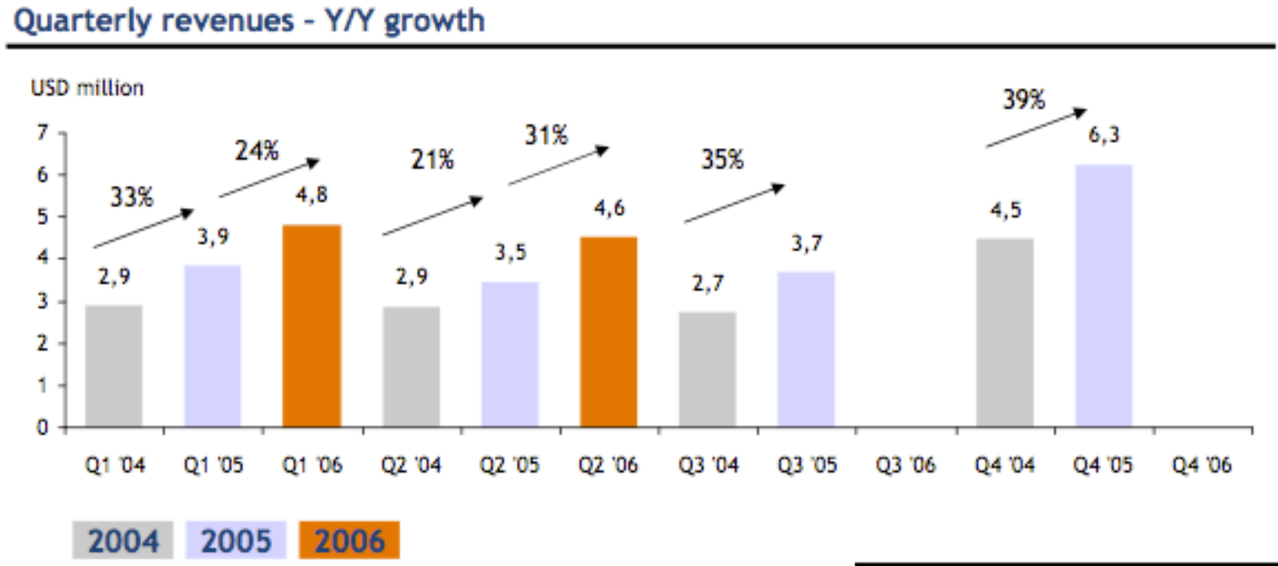


- Seminar observations
  - On several occasions, customers gave spontaneous "WOWs" and applauses!
  - The training room in London was literally packed with people eager to test the new version.
  - Several clients asked if they could access the test server from home as well.
  - Great participation rate; 95% of all registered people showed up.

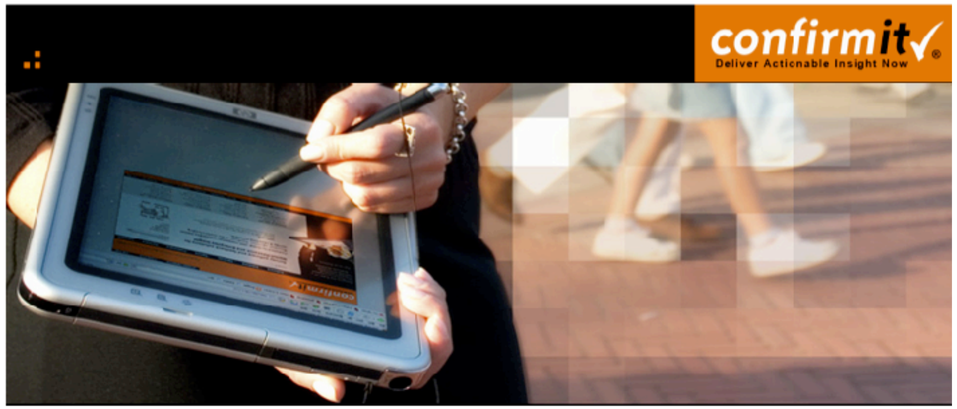
Confirmit Results Since Evo Method

.: Revenue growth

confirmit
Deliver Actionable Insight Now



- ✓ Full year 2005 revenue growth: 33%
- ✓ YTD Q2 2005 revenue growth: 27%
- ✓ YTD Q2 2006 revenue growth: 27%



http://www.newsweb.no/index.asp?symbol=FIRM&melding_ID=132091



25. Templates

These templates can be copied and pasted, then filled out

25.1 Requirements Template

With hints, for learning when doing

Critical Result Requirements Template

Elementary scalar requirement template <with hints>
Tag: <Tag name of the elementary scalar requirement>.

Type:
<{Performance Requirement: {Quality Requirement,
Resource Saving Requirement,
Workload Capacity Requirement},
Resource Requirement: {Financial Requirement,
Time Requirement,
Headcount Requirement,
others}}>.

========== Basic Information ==========
Version: <Date or other version number>.
Status: <{Draft, SQC Exited, Approved, Rejected}>.
Quality Level: <Maximum remaining major defects/page, sample size, date>.
Owner: <Role/e-mail/name of the person responsible for this specification>.

Stakeholders: <Name any stakeholders with an interest in this specification>.

Gist: <Brief description, capturing the essential meaning of the requirement>.
Description: <Optional, full description of the requirement>.
Ambition: <Summarize the ambition level of *only the targets* below. Give the overall real ambition level in 5-20 words>.

========== Scale of Measure ==========
Scale: <Scale of measure for the requirement (States the units of measure for all the targets, constraints and benchmarks) and the scale qualifiers>.

========== Measurement ==========
Meter: <The method to be used to obtain measurements on the defined Scale>.

========== Benchmarks ========== "Past Numeric Values" ====
Past [<when, where, if>]: <Past or current level. State if it is an estimate> <- <Source>.
Record [<when, where, if>]: <State-of-the-art level> <- <Source>.
Trend [<when, where, if>]: <Prediction of rate of change or future state-of-the-art level> <- <Source>.

========== Targets ========== "Future Numeric Values" ===
Goal/Budget [<when, where, if>]: <Planned target level> <- <Source>.
Stretch [<when, where, if>]: <Motivating ambition level> <- <Source>.
Wish [<when, where, if>]: <Dream level (unbudgeted)> <- <Source>.

========== Constraints ========== "Specific Restrictions"====
Fail [<when, where, if>]: <Failure level> <- <Source>.
Survival [<when, where, if>]: <Survival level> <- <Source>.

========== Relationships ==========
Is Part Of: <Refer to the tags of any supra-requirements (complex requirements) that this requirement is part of. A hierarchy of tags (For example, A.B.C) is preferable>.
Is Impacted By: <Refer to the tags of any design ideas that impact this requirement> <- <Source>.
Impacts: <Name any requirements or designs or plans that are impacted significantly by this>.

========== Priority and Risk Management ==========
Rationale: <Justify why this requirement exists**>.**
Value: <Name [stakeholder, time, place, event]: Quantify, or express in words, the value claimed as a result of delivering the requirement>.
Assumptions: <State any assumptions made in connection with this requirement> <- <Source>.
Dependencies: <State anything that achieving the planned requirement level is dependent on> <- <Source>.
Risks: <List or refer to tags of anything that could cause delay or negative impact> <- <Source>.
Priority: <List the tags of any system elements that must be implemented before or after this requirement>.
Issues: <State any known issues>.

Figure 4.12 *A scalar requirement template with hints <- Chapter 4 Competitive Engineering by Gilb [CE]*

25.2 Summary of All Requirements Template

With hints, for learning when doing

Requirements Overview:

Requirement Specification Template (A Summary Template)
Tag: <Tag name for the system>.
Type: System.
========== Basic Information ==========
Version: <Date or other version number>.
Status: <{Draft, SQC Exited, Approved, Rejected}>.
Quality Level: <Maximum remaining major defects/page, sample size, date>.
Owner: <Role/e-mail/name of the person responsible for changes and updates>.
Stakeholders: <Name any stakeholders (other than the Owner) with an interest in the system>.
Gist: <A brief description of the system>.
Description: <A full description of the system>.
Vision: <The overall aims and direction for the system>.

========== Relationships ==========
Consists Of: Sub-System: <Tags for the immediate hierarchical sub-systems, if any, comprising this system>.
Linked To: <Other systems or programs that this system interfaces with>.

========== Function Requirements ==========
Mission: <Mission statement or tag of the mission statement>.
Function Requirement:
<{Function Target, Function Constraint}>: <State tags of the function requirements>.
Note: 1. See Function Specification Template. 2. By default, 'Function Requirement' means 'Function Target'.

========== Performance Requirements ==========
Performance Requirement:
<{Quality, Resource Saving, Workload Capacity}>: <State tags of the performance requirements>.
Note: See Scalar Requirement Template.

========== Resource Requirements ==========
Resource Requirement:
<{Financial Resource, Time Resource, Headcount Resource, others}>: <State tags of the resource requirements>.
Note: See Scalar Requirement Template.

========== Design Constraints ==========
Design Constraint: <State tags of any relevant design constraints>.
Note: See Design Specification Template.

========== Condition Constraints ==========
Condition Constraint: <State tags of any relevant condition constraints or specify a list of condition constraints>.

========== Priority and Risk Management ==========
Rationale: <What are the reasons supporting these requirements? >.
Value: <State the overall stakeholder value associated with these requirements>.
Assumptions: <Any assumptions that have been made>.
Dependencies: <Using text or tags, name any major system dependencies>.
Risks: <List or refer to tags of any major risks that could cause delay or negative impacts to the achieving the requirements>.
Priority: <Are there any known overall priority requirements? >.
Issues: <Unresolved concerns or problems in the specification or the system>.

========== Evolutionary Project Management Plan ==========
Evo Plan: <State the tag of the Evo Plan>.

========== Potential Design Ideas ==========
Design Ideas: <State tags of any suggested design ideas for this system, which are not in the Evo Plan>.

Figure 2.10: Requirement Specification Template. This is a summary template giving an overview of the requirements. [CE]

25.3 Design Template

With hints, for learning when doing

Design – Strategy

Design Specification Template <with Hints>
Tag: <Tag name for the design idea>.
Type: {Design Idea, Design Constraint}.
========== Basic Information ==========
Version: <Date or version number>.
Status: <{Draft, SQC Exited, Approved, Rejected}>.
Quality Level: <Maximum remaining major defects/page, sample size, date>.
Owner: < Role/e-mail/name of person responsible for changes and updates>.
Expert: < Name and contact information for a technical expert, in our organization or otherwise available to us, on this design idea>.
Authority: <Name and contact information for the leading authorities, in our organization or elsewhere, on this technology or strategy. This can include references to papers, books and websites>.
Source: <Source references for the information in this specification. Could include people>.
Gist: <Brief description>.
Description: <Describe the design idea in sufficient detail to support the estimated impacts and costs given below>.

<Term Tag here>: Definition: <Use this to define specific terms used anywhere in the specification>.
"Repeat this for as many definitions as you need"

Stakeholders: <Prime stakeholders concerned with this design>.
========== Design Relationships ==========
Reuse of Other Design: <If a currently available component or design is specified, then give its tag or reference code here to indicate that a known component is being reused>.
Reuse of This Design: <If this design is used elsewhere in another system or used several times in this system, then capture the information here>.
Design Constraints: <If this design is a reflection of attempting to adhere to any known design constraints, then that should be noted here with reference one or more of the constraint tags or identities>.
Sub-Designs: <Name tags of any designs, which are subsets of this one, if any>.
========== Impacts Relationships ==========
Impacts [Functions]: <List of functions and subsystems which this design impacts attributes of>.
Impacts [Intended]: <Give a list of the performance requirements that this design idea will positively impact in a major way. The positive impacts are the main justification for the existence of the design idea!>.
Impacts [Side Effects]: <Give a list of the performance requirements that this design idea will impact in a more minor way, good or bad>.
Impacts [Cost]: <Give a list of the budgets that this design idea will impact in a major way>.
Impacts [Other Designs]: <Does this design have any consequences with respect to other designs? Name them at least>.

========== Impact Estimation/Feedback ==========
For each Scalar Requirement in Impacts [Intended] (see above):
Tag: <Tag name of a scalar requirement listed in Impacts [Intended]>.
Scale: <Scale of measure for the scalar requirement>.
Scale Impact: <Give estimated or real impact, when implemented, using the defined Scale. That is, given current baseline numeric value, what numeric value will implementing this design idea achieve or what numeric value has been achieved?>.
Scale Uncertainty: <Give estimated optimistic/pessimistic or real ± error margins>.
Percentage Impact: <Convert Scale Impact to Percentage Impact. That is, what percentage of the way to the planned target, relative to the baseline and the planned target will implementing this design idea achieve or, has been achieved? 100% means meeting the defined Goal/Budget level on time>.
Percentage Uncertainty: <Convert Scale Uncertainty to Percentage Uncertainty ± deviations>.
Evidence: <Give the observed numeric values, dates, places and other relevant information where you have data about previous experience of using this design idea>.
Source: <Give the person or written source of your evidence>.
Credibility: <Credibility 0.0 low to 1.0 high. Rate the credibility of your estimates, based on the evidence and its source>.
========== Priority and Risk Management ==========
Rationale: <Justify why this design idea exists>.
Value: <Name [stakeholder, scalar impacts and other related conditions]: Describe or quantify the knock-on value for stakeholders of the design impacts>.
Assumptions: <Any assumptions that have been made>.
Dependencies: <State any dependencies for this design idea>.
Risks: <Name or refer to tags of any factors, which could threaten your estimated impacts>.
Priority: <List the tag names of any design ideas that must be implemented before or after this design idea>.
Issues: <Unresolved concerns or problems in the specification or the system>.
========== Implementation Control ==========
Supplier: < Name actual supplier or list supplier requirements>
Responsible: <Who in your organization is responsible for managing the supplier relation?>
Contract: <Refer to the contract if any, or the contract template>
Test Plan: <Refer to specific test plan for this design>
Implementation Process: <Name any special needs during implementation>
========== Location of Specification ==========
Location of Master Specification: <Give the intranet web location of this master specification>.

Figure 7.7. Design Specification Template [CE] . This is a form to fill out, with <hints in fuzzy brackets>.

25.4 Evo Step Template

With hints, for learning when doing

Evo Step

A TEMPLATE FOR EVO STEP SPECIFICATION
Tag: <Tag name for the step>.
Type: Evo Step.
========== Basic Information
==========
Version: <Date or version of last update to step specification>.
Status: <{Specification Stage [{Draft, SQC Exited, Approved}], In Evo Plan, Scheduled Next, Under Implementation, Delivered awaiting Feedback, Feedback Obtained}, date> <- <Source (who says 'Status' is true?)>.
Quality Level: <Maximum remaining major defects/page, sample size, SQC date>.
Owner: <Who is taking responsibility for the step in terms of specification>.
Stakeholders: <Who are you going to deliver requirements to? >.
Implementers: <Who is in charge of implementing this step>.
Gist: <Brief description of the main idea of this step>.
Description: <Give a detailed, unambiguous description of the step, or a tag reference to a place where it
is described. Remember to include definitions of any local terms>.

Implementation Details: "Includes relevant details, such as <which product>, <which area of application system>."
Evo Plan: <Tag of the Evo Plan that this step is associated with>.
Step Content: <Step Elements: {Design Ideas, Functions, Tasks, re-used step definitions}>.

========== Measurement
==========
Test: <Refer to tags of any test plan and/or test cases, which apply to this step>.
Step Validation/Feedback:
Specification Quality Control (SQC): <outcome, date>,
Pre-Delivery Test: <outcome, date>,
Post Delivery Results: <{problems, stakeholder feedback}, date>,
Certification Specification: <refer to the certification plans>.

========== Priority and Risk Management
==========
Constraints:
< Any legal, political, economic, security or other constraints imposed on implementation>
<- <Source (who says this is true?)>.
Assumptions: <Any assumptions that have been made>.
Dependencies:
<Anything which must be in place, finished, working properly, for us to be able to start this Evo
step or to complete it> <- <Source (who says this is true?)>.
Risks: <Any risks that need to be taken into account>.
Priority:
<Name, using tags, any system elements, which must clearly be done *after* or must clearly be done *before*. Give any relevant reasons>.
Issues: <Unresolved concerns or problems in the step specification or the system>.

========== Benefits and Costs
==========
Rationale: <Justify the existence of this step>.
Step Value:
<Real measurements or estimates of numeric value to stakeholders>. "Value in terms of meeting the requirements. At least, the value on scale 0 (none) to 9 (highest)."
<- <Source (who says this is true?)>.
Step Cost:
<Budgets or real costs>. "For example, financial costs and engineering hours. These must be
constrained by the Evo 2% policy. At least, the value on scale 0 (very cheap) to 9 (high and
unpredictable)." <- <Source (who says this is true?)>.

Figure 10.10: [CE] A possible specification template for a one page Evo step. Notice that the parameters are designed to give you enough information to decide on the order for step sequencing in an Evo Plan.

25.5 Evo Step Tracking Template

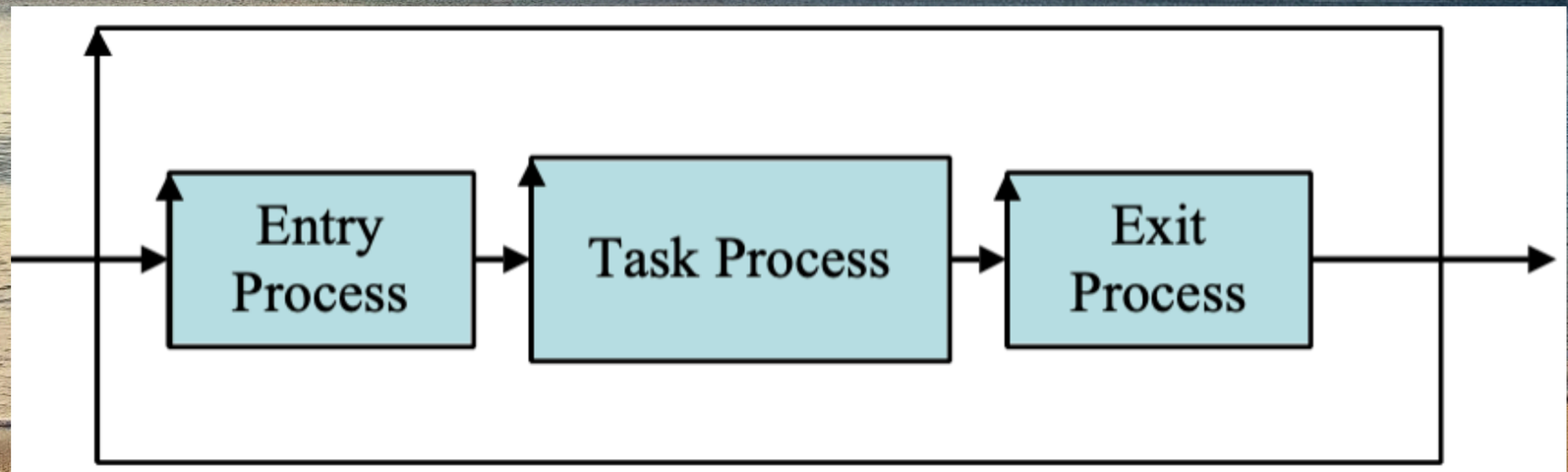
Note there is a tool for this [ValPlan]

Step Target Requirement	Step 1 Plan % (of Target)	Actual %	Deviation %	Step 2 to Step 20 Plan %	Plan % cumulated to here	Step 21 [CA, NV, WA] Plan %	Plan % cumulated to here	Step 22 [all others] Plan %	Plan % cumulated to here
Performance 1	5	3	-2	40	43	40	83	-20	63
Performance 2	10	12	+2	50	62	30	92	60	152
Performance 3	20	13	-7	20	33	20	53	30	83
Cost A	1	3	+2	25	28	10	38	20	58
Cost B	4	6	+2	38	44	0	44	5	49

Evo Plan and Project Tracking in form of Spread sheet

- ***Figure 10.5: This is a conceptual example. Three goals (performance targets) and two resource targets are having the real impacts on them tracked, as steps are delivered. The same IE table is also being used to specify the impact estimates for the future planned steps. So at each step, the project can learn from the reality of the step's deviation from its estimates. Plans and estimates can then be adjusted and improved from an early stage of the project. <- Competitive Engineering, Gilb.***

27. Processes



A process with its three main component processes.

See many examples of Process in [CE] and later in this book.

27.1 Process

A work activity, defined sequence

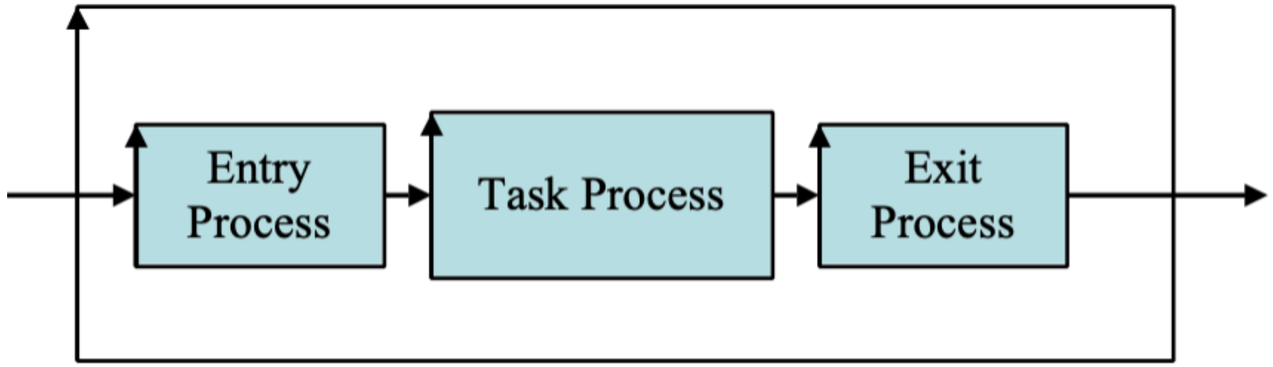
Process Concept *113 February 23, 2003tg

A process is a work activity consisting of:

- an entry process, which examines entry conditions
- a task process, which follows a procedure defining the task. There might also be an associated verification process, such as test or quality control
- an exit process, which examines exit conditions

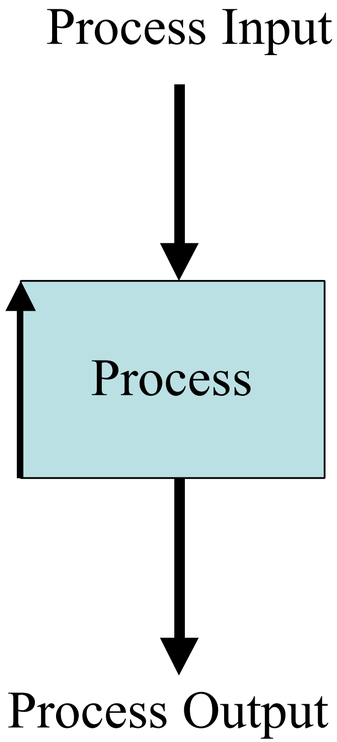
Processes transform inputs to outputs, using resources, and display their own performance and resource (cost) characteristics.

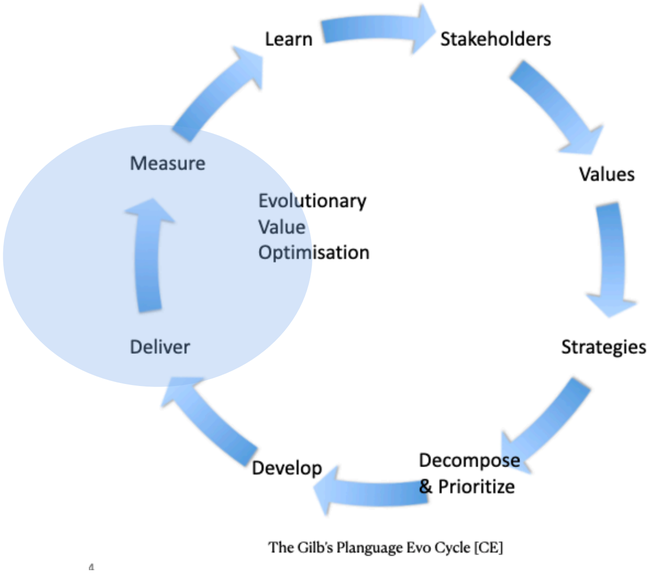
Processes, like 'Rules', capture wisdom, logic, culture, and well-intended advice, from the experienced, to the less-experienced.



A process with its three main component processes.

See many examples of Process in [CE] and later in this book.





The Evo Cycle Process

27.2 The EVO Process, the One-Page Summary

A Systems Engineering Process For incremental value delivery management

A *PLANNING* CYCLE

Clarify your environment: critical-stakeholders' territory

1. **STAKEHOLDERS:** Identify *critical stakeholder's* and their critical **values**
2. **VALUES: Quantify** and **clarify** y*our project's* critical values: what *degree* of values do you expect the design to deliver, to which stakeholders
3. Identify design **constraints**: legality, political, cultural, policy, other plans
4. Identify design **resource-limitation**s: time, money, operational costs for example.

Design (Solutions): Architecture Level

6. **SOLUTIONS:** use the many Value and Resource requirements, AS EXPLICIT TOOLS, to find solutions, to deliver Values, within constraints. Aka Designs, Strategies, Architecture
7. **DECOMPOSE:** Decomposition of top-level architecture into design components
8. **PRIORITY:** Identification, *detailed* specification and *prioritization* of top-level architecture, and of any *decomposed* design solutions..

A *VALUE DELIVERY* CYCLE: An 'Evo-step'.

9. The **Evo Value delivery steps** (for example, a week, or 2% of total project budget)
   1. Select your highest **priority value**, and the most-critical scale-parameter attributes (*the who, what, where, when - slices of reality*).
   2. **DEVELOP**: Find a design component which will deliver good **value-for-resources** to your critical priority requirements.
   3. Ready the design-component for delivery: for ***integration*** **to the existing system.**
   4. **DELIVER: Deliver** the design component to the **real system**
   5. **MEASURE: Measure** the results (values and costs) of the design increment
   6. **Dynamic-Design-To-Requirements:** If results are **not OK**, attempt design **improvement, and redeliver.**
   7. **KEEP CYCLING:** If results *are* **OK,** then **repeat** this 'value delivery cycle', scale up, until 'done'.

Project Completion: All Value is delivered

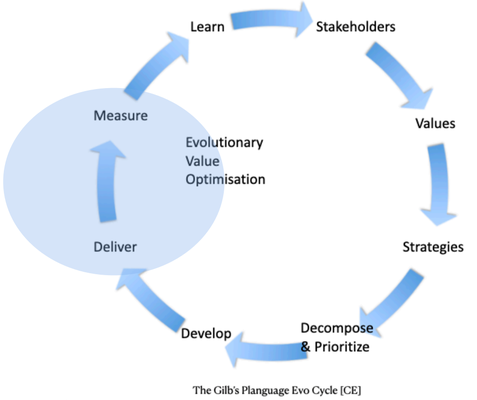
10. **DONE?**: When **all value requirements are reached**, or when all **critical resources are used up**. Stop.



Gilb Value Cycle ('EVO')

Copyright 2020 kai@Gilb.com

27.3 Evo Process (simplified)



The Gilb's Planguage Evo Cycle [CE]

The evolutionary project management process

A Simple Evolutionary Project Management Method

Tag: Quantified Simple Evo Project. Version: July 8, 2003 (3). Owner: Tom@Gilb.com. Status: Draft. Published in Cutter IT Journal July 2004.* The Evo Cycle Process

Project Process Description

1. Gather from all the key stakeholders the top few (5 to 20) most critical performance (including qualities and savings) goals that the project needs to deliver. Give each goal a reference name (a tag).
2. For each goal, define a scale of measure and a 'final' goal level. For example: *Reliability: Scale: Mean Time Between Failure, Goal: >1 month.*
3. Define approximately 4 budgets for your most limited resources (for example, time, people, money, and equipment).
4. Write up these plans for the goals and budgets (*Try to ensure this is kept to only one page*).
5. Negotiate with the key stakeholders to formally agree the goals and budgets.
6. Plan to deliver some benefit (that is, progress towards the goals) in *weekly* (or shorter) increments (Evo steps).
7. Implement the project in Evo steps. Report to project sponsors after each Evo step (weekly, or shorter) with your best available estimates or measures, for each performance goal and each resource budget.

- O*n a single page,* summarize the *progress to date* towards achieving the goals and the costs incurred.
- Based on numeric feedback, and stakeholder feedback; *change whatever needs to be changed to reach goals.*

8 When all goals are reached: "Claim success and move on" [Gerstner, 2002]. Free the remaining resources for more profitable ventures

Project Policy

1. The project manager, and the project, will be judged exclusively on the relationship of progress towards achieving the goals versus the amounts of the budgets used. The project team will do anything legal and ethical to deliver the goal levels within the budgets.
2. The team will be paid and rewarded for 'benefits delivered' in relation to cost.
3. The team will find their own work process, and their own design.

*** 'Adding Stakeholder Metrics to Agile Projects'**

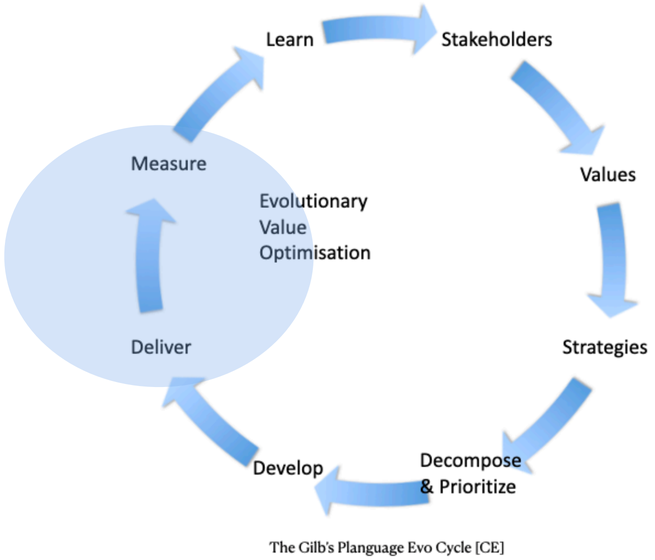
A version was published in Cutter It Journal: The Journal of Information Technology Management, July 2004, Vol. 17, No.7, as 'Adding Stakeholder Metrics to Agile Projects', pp31-35. www.cutter.com.

27.4 Evo Cycle Process

Process: Agile Evo:

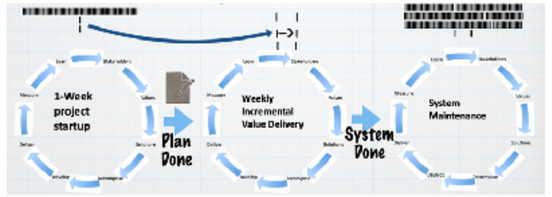
- Status: Draft
- Version: Sept 1 2004
- Author: Tom Gilb
- Owner: Richard Jeans
- Process Summary:

- Process Detail:
  - **Entry** Conditions:
    - Evo.E1: Project Top Level Critical Result Requirements are exited from Spec QC (<10 Majors/Page remain), and then Approved by Programme Manager.
    - Evo.E2: Constraints:
      - all known constraints are specified and exited from Spec QC. Resource Constraints and Condition Constraints.
  - Evo.P1: **Strategies**:
    - Develop Top Few (up to 10) Strategies to satisfy the Result Requirements within the defined Constraints.
  - Evo.P2: **Estimation**:
    - Estimate quantitatively, using Impact Estimation, the impacts on all Requirements (Result and Constraint) of each of the Strategies.
  - Evo.P3: **Selection**:
    - Select the single strategy that appears to have the most deliverable stakeholder value towards specified Goal levels in Result Requirements (if a tie, then the one with lowest costs goes first).
  - Evo.P4.**Decompose**:
    - Decompose this strategy so that it does not exceed the Evo Cycle Policy limits (2% to 10% resources, 1 week to 1 month).
    - Do this by focussing on
      - Limited result delivery (not *necessarily* all the way to the Goal level, at one step)
      - Limited Stakeholder focus (not *necessarily* to all stakeholders at once)
      - Proving Concept of delivery (not *necessarily* the full scale industrial strength version – just enough to actually prove the result can be delivered in practice and stakeholders agree).
  - Evo.P5.**Build**:
    - Translate the planned step into a deliverable reality.
    - Ideally following all necessary development process steps of testing and planning required for a final deliverable in the stakeholder environment.
    - Ready for final use.
    - Certain exceptions, to prototype results under controlled conditions may be approved, but should be exceptional.
    - The idea is finish the job, deliver results, claim victory, and move on.
  - Evo.P6.**Deliver**:
    - The resulting system increment will be handed over to the stakeholder environment, in such a manner as to guarantee the planned estimated impacts on the critical business, or stakeholder, results.
    - All documentation, training, briefing, hand-holding, data provision, transition tactics necessary must be planned for and provided for. Everything necessary for success. No excuses and no exceptions.
  - Evo.P7.**Measure**:
    - Immediately (same day) and continuously (days and weeks afterwards) we will gather data on the results and costs of the delivery step.
  - Evo.P8.**Study**:
    - Results will be compared to estimated expectations and conclusions as to cause of deviation drawn.
  - Evo.P9.**Act**:
    - Based on the results, and on any updates to requirements and strategies, the next step will be planned.
  - **Exit** Conditions:
    - Evo.X1: This cycle will be repeated until all valid Result Requirements are achieved; or until finite budgets are used up.



The Evo Cycle Process

27.3 One-Week Project 'Startup' Process



Minimize Overhead, Maximize Value Flow, even for very large projects

Time Plan:

Day 1: **Project Objectives**: The top few critical objectives quantified.
- Objective: Determine, clarify, agree critical few project objectives – results – end states
- **Process**:
  1. Analyze current documentation and slides, for expressed or implied objectives (often implied by designs or lower level objectives)
  2. Develop list of **Stakeholders** and their needs and values
  3. Brainstorm 'top ten' critical objectives names list. Agree they are top critical few.
  4. Detail definition in Planguage – meaning quantify and define clearly, unambiguously and in detail (a page)
  5. Quality Control Objectives for Clarity: Major defect measurement. Exit if less than 1.0 majors per page
  6. Quality Control Objectives for Relevance: Review against higher level objectives than project for alignment.
  7. Define Constraints: resources, traditions, policies, corporate IT architecture, hidden assumptions.
  8. Define Issues – yet unresolved
     - Note we might well choose to several things in *parallel*.
- **Output**: A solid set of the top few critical *objectives* in quantified and measurable language. *Stakeholder* data specified.
- **Participants**: anybody who is concerned with the business results, the higher the management level the better.
- **End of Day Process**: meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.
- **Note**: this process is so critical and can be time consuming, so if necessary it can spill over to next day. Perhaps in parallel with startup of the strategy identification. *Nothing is more critical or fundamental than doing this well.*

Day 2: **Project Strategies and Architecture**: the top few critical strategies for reaching the critical objectives
- **Objective**: to identify the top 'ten' most critical strategic decisions or architectures; the ones that will contribute or enable us most, to reach our primary objective goal levels on time.
- **Process**:
  1. Analysis of current documentation and slides to identify candidate strategies, implied or expressed.
  2. Brainstorming of the 'names' of the specific strategy list, the top ten and a set of less powerful ideas (say 11-30)
  3. Detail each top ten strategy sufficiently to understand impacts (on objectives, time and costs)
  4. Specify, for each strategy all critical related information (like stakeholders, risks, assumptions, constraints, etc.)
  5. Quality Control for clarity – correct unclear items. Exit based on defect level, or not.
  6. Likely that work will need to be done in parallel in order to do ten strategies to a rich level of specification.
- **Output**: A formal strategy specification, ready for evaluation, and decomposition and delivery of partial value results.
- **Participants**: system architects, project architects, strategy planners. And members of the project team who will be in on the entire weeks process. The major input here is technical and organizational strategy (the means to reach the objectives)
- **End of Day Process: :** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 3: Evaluation of Strategies using Impact Estimation: our best estimates with experience and risk. How sure are of the major strategy decisions.
- **Objective**: to estimate to primary effects and all side effects of all top critical strategies on all top critical objectives, and on some resources (time, cost, effort). The estimates will be backed up by evidence, or their credibility will be rated low.
- **Process**:
  1. Using the objectives and strategies developed on first 2 days as inputs
  2. Populate an Impact Estimation table (aka Value Decision Table) with estimates of the expected result of deploying defined strategies. Estimate main intended impacts
  3. And all side effects (on other core objectives)
  4. And on all resources (time, money. Effort)
  5. Estimate ± ranges
  6. Specify evidence and sources for estimates
  7. Determine Credibility level
  8. Quality Control the IE table against standards (Rules for IE in CE book), for possible 'exit' (meets standards)
  9. Lots of parallel work needed and expected to do a good job.
- **Output**:
  - A fairly decent Impact Estimation table, possibly a several level set of them.
    - This will tell us if it is safe to proceed (we have good enough strategies)
    - And it will help us prioritize high value deliveries soon.
- **Participants**: architects, planners, anybody with strong views on any of the strategies. The team for the week.
- **Note**: *it might be necessary and desirable, now or later, to do this impact estimation process at 2 or 3 related levels (Business, Stakeholder, IT System) in order to see the Business-IT relationship clearly. This might exceed time limits and be done parallel or later.*
- **End of Day Process**: meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 4: **Evolutionary Step Decomposition**: what are the high value short term value delivery steps we can execute.
- **Objective**: to identify near team candidates for real value delivery to real stakeholders. What can we do for real next week!
- **Process**:
  - Identify highest value (to costs) strategies and sub-sets of strategies
  - Decompose into doable subsets in weekly to monthly cycles of result delivery
  - Plan the near steps (1 or more) in detail so that we are ready to execute the step in practice.
    - Who does it, main responsible, team.
    - Expected measurable results and costs
    - Stakeholder involved in receiving
    - Test process (for value)
- **Output**: 1 or more potential steps for value delivery to some stakeholders, a plan good enough to approve and execute in practive.
- **Participants**: Project Management, architects prepared to decompose architecture in practice. The weeks team for this start up study.
- **End of Day Process:** meet 30 minutes with any responsible interested managers to present the outputs, and to get preliminary corrections and go-ahead.

Day 5
- This is normally used to present the plan to management and get approval to go forward the next week.
- In our case we have chosen a 4 days model due to Easter Holidays. So we have to find another way to present and approve.
- **Objective**: To present the entire set of plans to responsible executive(s) and discuss them, with approval if possible, or approve with changes.
- **Process**:
  - Present all planned outputs
  - Discuss them and answer questions
  - Take corrections
  - Get approval for the next implementation step.
- **Output**: Approval for next implementation step, corrections
- **Participants**: project tem + key manager above the project manager.
- **End of Day Process: none, unless corrections needed before execute OK.**
  - Possible Corrections and ready to execute a delivery step next week

27.5.2 Principles of Processes [KEN]

https://tinyurl.com/KENGilb

Chapter 6. Educational Content

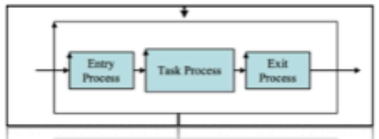


Figure 6.4.8 Process with Entry Exit sub processes.

6.4. Principles of Processes (© Gilb.com 2307-2020)

1. PROCEDURE: A process is *specified by* a 'set of **instructions**', on 'how to carry out a particular task'. 'Process Description/Specification', or 'Procedure'.

2. MEDIUM: The process '**instructions**' can be written, oral culture, computer programmed, videos, or embedded in machines or biological systems.

3. EXPERTISE: A process represents *known best-practices*, best advice from experienced experts. Cultural expertise.

4. ATTRIBUTES: The multi-dimensional **attributes**, qualities and costs, of a process defines how cost-effective it is (process attribute results), and are all *measurable*. See below, 6.4). Example using Deming's 'Statistical Process Control'.

5. MUTATION: When there is a need for better results, *a process can be re-designed;* a process mutation is created, and measured, to see if it gives the desired better results.

6. STRATEGY: A particular process, is a *design or strategy*, with known historical attributes, which can be deployed, in order to get those known cost-effective attributes.

7. ENTRY: The 'safe conditions' for *starting* a process and *succeeding*, are **'process entry conditions'**, which can be defined as a formal *standard*, and tested in a *process Entry Process*, to make sure the process does not fail, for want of sufficient safe conditions. See Fig. 6.4.7 example, E1.

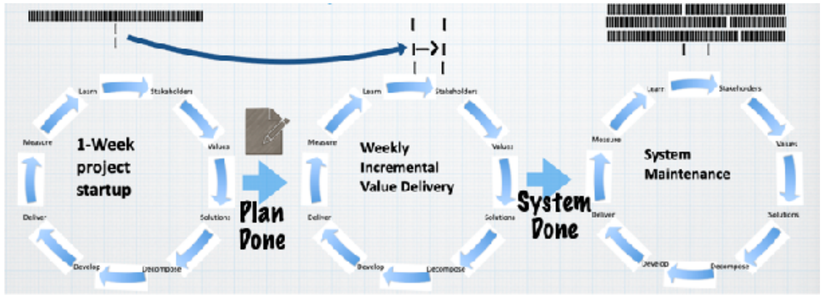
8. EXIT: It also improves the success of a process, to have a set of defined standard **Exit Conditions**, which are tested by an **Exit Process**, before delivery of process results. These conditions are 'organizational' or 'cultural' *learning*, and is a tactic for systematic *organizational improvement*. *See Fig. 6.4.7 example, X1.*

9. FAILURE*:* A process can fail for several reasons: inappropriate process for task at hand, misinterpreted process instructions, lack of good Entry and Exit controls, and conscious sabotage, etc. .

10. BAD BETTERMENT: It is generally better to have a bad-but-formal process, which is *systematically improved by experience* and feedback, than to let individuals use their own personal opinion of how best to do things. If they have better opinions they should feed them for tryouts to the *process improvement process*, so everybody can benefit from their creative ideas. This is key to Statistical Process Control.

27.4 Evo Policy

An Evo ‘Framework’, and Motivation



It would be useful if the management responsible would clearly adopt and spread a policy regarding the ‘Evo’ work, here is a *template*.

IT Project Policy.
Version: 23 March 2010 tg

1. VALUE FIRST: The main purpose of our projects is to deliver agreed and quantified value to our stakeholders.

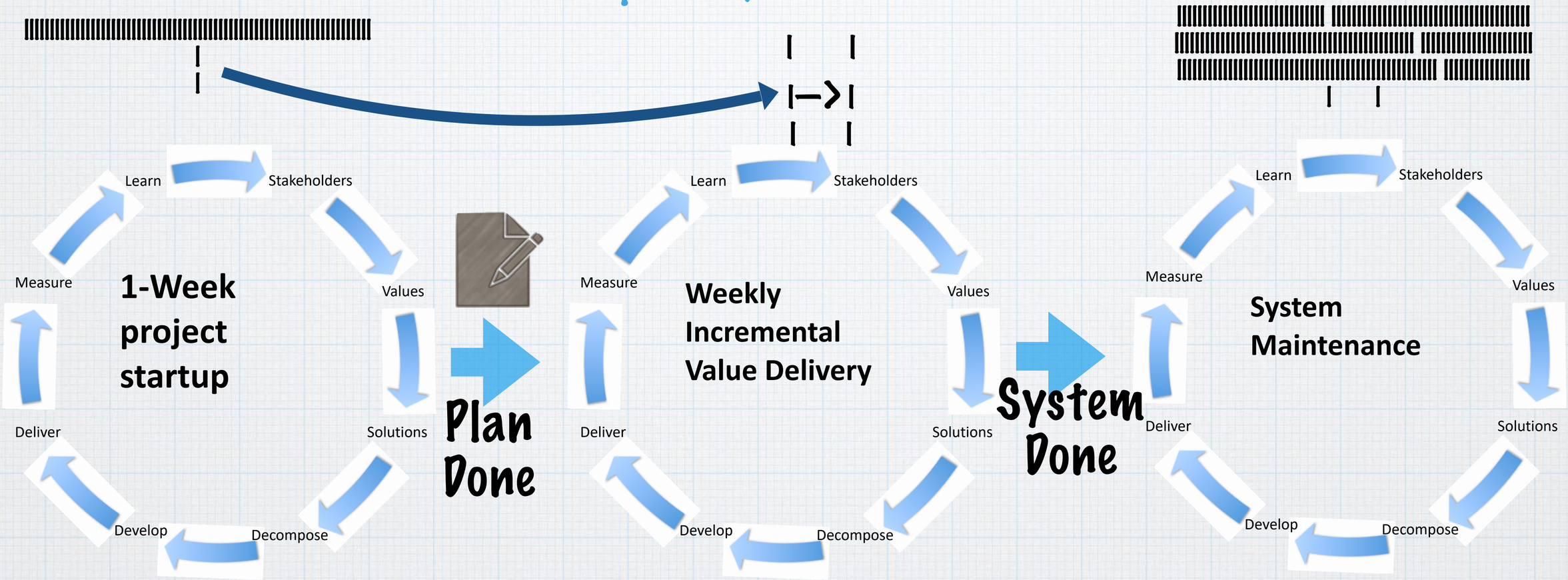
2. VALUE EARLY: We want real and proven value delivery as early as possible, and frequently and continuously as possible.

3. VALUE PRIORITY: The value delivery is our highest priority, and we can modify or scrap any plan, process, culture, habit or other artifact that stands in the way of delivering that value on time, except legally binding external constraints.

4. VALUE REALITY: We have a complex environment, and it is impossible to foresee the costs and effects of technological change on our business; but we can probe, experiment, test early, measure early and deal explicitly with risks early – and learn early what works and what it really costs.

5. VALUE TUNING: We can then modify plans to fit with the emerging reality of our experience, and simultaneously with the changing reality of our project-external environment.

27.5 'Evo' Development Process
Deep Start, Incremental Build



Wide View
(system-wide, top-10 stuff)
With selection of a deep slice
Of action.
Monday-Wednesday (Wide)
Thursday (Deep slice)

Many '2%' cycles
of a slice of 'value delivery'
until
'Requirements Met'

Keeping the system in
Good shape

The 'Evo Cycle' applied high and wide:
A 'development process' simplification
SLIDES:https://www.researchgate.net/publication/381918278_The_'Evo_Cycle'_applied_high_and_wide_A_'development_process'_simplification



28. Rules

28.1 Rule

'How' to specify well

Rule **Concept *333 May 25, 2003**

A rule is any statement of a standard on how to write or carry out some part of a systems engineering or business process.

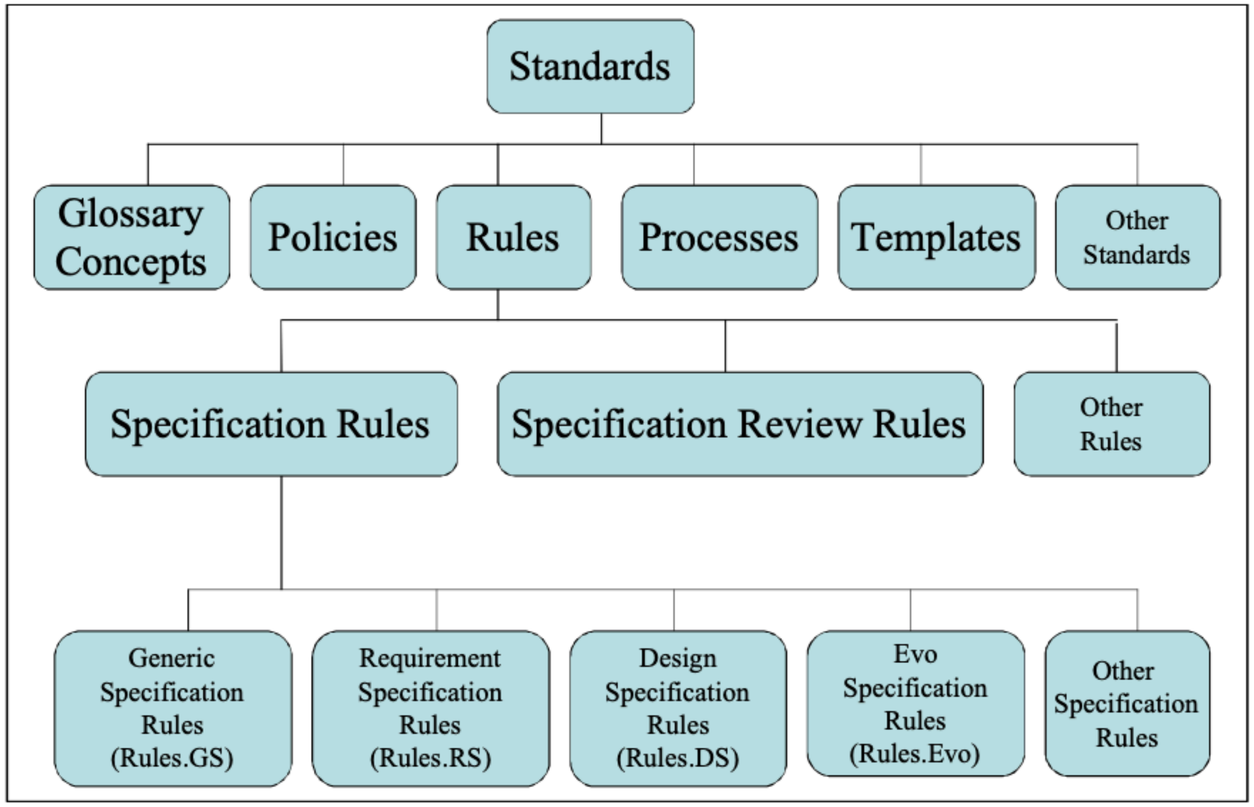
Rules are intended to capture experience, advice, wisdom: good guides to getting things right.

They make up part of our our 'culture'.

Improved definition (2024):

Rule
A standard for **specification**.

Note: By contrast, a *Process* is a standard for **activity**.



Rules, in the larger context of other 'Standards'.

See many examples of Rules in [CE] and later in this book.

28.2 *Requirements* Rules

Advice on well-written requirements, objectives, goals, visions

REQUIREMENTS RULES
Version: March 3, 2010
Writers: Tom@Gilb.com & Kai@Gilb.com

Purpose (of this document): a draft of Rules for the Requirements specification and consequently the Quality Control process from Gilb to Cititech.

Rules: Rules.RQ (main tag)

RQ.Readers: The complete and correct list of all INTENDED READERS will be attached to all specifications.

RQ.Clear: Specifications must be CLEAR as to meaning, intent, use, to any intended reader.
Annotation: where any doubt might arise in the mind of any intended reader, however inexperienced, the writer shall include annotation to the specifications to improve clarity.
For example {Justification, Rationale, Impacts, Impacted By, Depends on, Supports, [qualifiers], defined Scales of measure, Note, and any other specification parameter that can improve immediate and correct interpretation of the specification}.

RQ.Unambiguous: Specifications must be UNAMBIGUOUS to any intended reader.
Definitions: all potentially ambiguous terms must be clearly defined somewhere in the documentation or the corporate glossary.
Defined: All defined terms must signal their formal definitions by being written with 'Capital Letters' on every word.

RQ.Consistent: All specifications must be CONSISTENT with any (exited) source specifications upon which they are based.
Source: all specifications must be annotated with explicit and detailed reference to their source information. Use source arrow '<-', or 'Source: parameter.
Source-Relationship: to other important 'relationship' information such as: {Impacts, Test case, Impacted By, Supports, Depends On, and other relationship parameters}

RQ.Tag: All elementary specifications will be REFERENCE-ABLE by means of a 'tag'
General: use a name tag on all elementary specifications with Capital Letters to emphasize that it is defined formally.
Parameters: In addition, certain parts of a specification will be reference-able by using parameter names (eg Goal, Scale) and qualifiers (eg [Europe, 2005]).
Comment: the use of bullet points is effectively 'banned'. Use of numbering for reference is considered dangerous (because of automatic renumbering in Word). Paragraphs with text containing several ideas and sub ideas without unique names are banned.

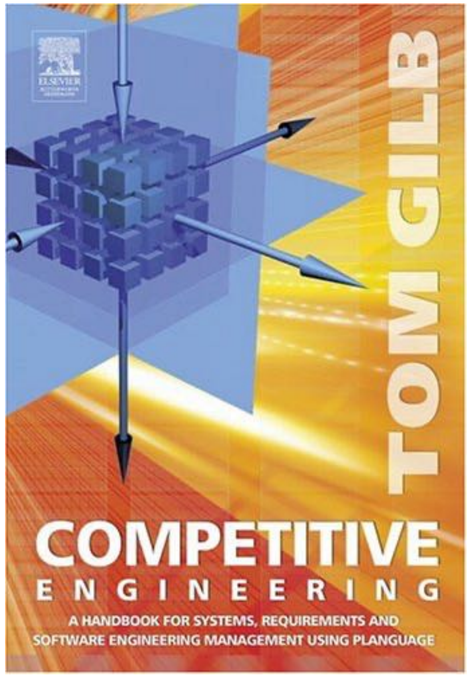
RQ.Once: A specification of any kind, including definitions of any kind, will be stated only once, in the entire project-wide documentation. This will be the MASTER version. All other use of that idea will be done by cross-referencing that tag (and parameter/qualifier as needed).
Rationale: we want above all to avoid inconsistency because of differing versions of the same requirement, design or other definitions or specification. Cutting and pasting is effectively banned. Electronic reference using for example URLs or live links is encouraged. 'Versioning' of the master specifications, at least at the requirement or design level, is strongly encouraged (it is in the templates).

RQ.Core: A clear distinction will be made between Core Specification and Background Specification (aka Commentary).
Background: Background will appear in italics.
Comment: Either manually or by automated conversion, using parameters and other clues. Eg Past is always background, Goal is always Core).
Core: Core specification will not appear in *italics*.
Comments: Core specification is the essential real definition of the future state of the system. Defects in Core Specs are inevitably Major defects (they have the potential to escalate in cost, to delay and to deliver wrong systems). Defects in Background specs are probably minor defects (can't hurt real system much).

RQ.Requirements: Specifications that are Required by a Stakeholder that we decide to serve will be called Requirements.
Performance: mark as: Performance Requirement
Function: mark as: Functional Requirement
Design: mark as: Design Constraint
Stakeholders: Include the Stakeholder requiring the Requirement.

RQ.Performance-Function-Design: Specification must be separated into:
Performance: Performance Attributes
Functions: Functional Specifications
Design: Design Specifications
Comment: Performance Attributes explain HOW WELL the system performs. The added values and the system abilities, like Usability, Reliability etc. They are measurable.
Functional Specifications define purely WHAT the system does. Functions are binary, they are either there or not.
Design specifications are the HOW the system is built.

RQ.Templates: Corporate Templates for specification will be used as a strong guide to detailed content and terminology rules in addition to these generic rules.



Many more Rule sets here [CE]

28.3 Evo Project Management *Plan* Rules

A pattern for writing an Evo Plan

Rules: Evolutionary Project Management <- Gilb, Competitive Engineering, Evo Chapter.

Gist: Rules for Evo Plan Specification.
Tag: Rules.EVO. Version: Fall, 2004. Owner: TG. Status: Draft.

Base: The rules for generic specification, <Citi **Rules.GS**> apply as well as all other Planguage rules needed to express requirements and design.

R1: **Tags**: All steps of an Evo plan will have a unique tag to enable cross-referencing from other specifications (such as test planning or costing).

R2: **Detail**: All detailed design idea specifications shall be kept separate from the Evo plan. For brevity, use Planguage step descriptions only. Any Evo plan elements yet to be defined in detail must be specified by a unique tag in fuzzy brackets (<Tag Name 1>). This will indicate that the detail is not specified yet. *Rationale: we need to avoid the clutter of design idea definitions in the Evo plan itself. Tags are sufficient.*

R3: **Cost**: Any planned step, that has an estimated incremental impact, for any resource attribute, which exceeds 10% of the total budget planned level, will be re-specified into smaller steps, to reduce risk. An average of <10%-of-budget steps is desirable (*as risk of economic loss is then at 10% maximum*), but individual projects *may* specify their own budget constraints. All planned steps still exceeding these single step budget constraints must be agreed by authorized signature.

R4: **Time**: Any step, which would take more than 10% of the total project calendar time (from project start up to the main long-term deadline), must be divided into smaller steps. An average of <10%-of-time steps is desirable, but individual projects may specify their own time constraints. All steps exceeding the 10% time constraint must be agreed by authorized signature. *Rationale: control time to deadline.*

R5: **Priority**: The 'next step', at any point in the project, should ideally be selected using an Impact Estimation table to evaluate step options. Steps that you estimate to deliver the greatest benefits (Sum of Percentage Impacts) to stakeholders, or that have the best benefit to cost ratio, shall generally be done earliest, wherever logically possible, and when 'other considerations' (such as a customer contract or request) do not have higher priority. Any specific priority factors, which override going for the greatest benefits first, shall be clearly documented. There must be some *specified* clear rationale, policy or rule behind prioritizing steps differently from this rule. This could be some estimate of value of a step, which is outside the scope of Impact Estimation, which might have priority.
Example:
Step 44:
Type: Step.
Consists Of: ABC [UK]: <- Contract Requirement 6.4.
Rationale: The contract demands we deliver this step at this point.
Optionally, there can be a project-defined constraint of a step having to achieve a minimum estimated Percentage Impact, or benefit to cost ratio, or value (financial growth or saving), before being considered for implementation at all.

R6: **Next**: Only the current step, or the approved next step, has 'commitment to implementation' (and even then, it could be terminated mid-implementation, if seen not to be delivering to plan). The sequencing specification of subsequent steps is not necessary, and is certainly not fixed. *In practice, there is likely to be a tentative step sequencing mapped out, which captures any dependencies.*

R7: **Impact**: The next step must be numerically estimated *in detail* for its impacts on all the critical performance and resource requirements. Other later steps may be more roughly estimated, either individually or in relevant groups. *They will be estimated in greater detail as their 'turn' approaches. Rationale: to force us to estimate, measure and consider deviation in small immediate steps.*

R8: **Learn**: The actual results of the steps already implemented (that is, the cumulative impacts on all requirement levels to-date) and the estimated results for the next step must be specified in an IE table (see Figure 10.5 in Competitive Engineering book example). Specific comment about negative deviations already experienced, and what you have specifically done in your plan to learn from them, should be included in some form of footnote or comment. *(Note: we assume the use of an IE table, but other formats are possible.)*

R9: **Completeness**: *All* the specified design ideas for a system, implemented or not, must be represented *somewhere* on an Evo plan. (Remember you can use tags, and you can declare a large set of designs with a single tag. For example, A: Defined As: {B, C. D, E, F}).
Rationale: This is because failure to include all the specified design ideas somewhere on the Evo plan, causes confusion. It leaves us to wonder:

- *Was it forgotten inadvertently?*
- *Why is it specified, if it is planned never to be implemented? (If you are just keeping the idea in reserve, be specific).*



29. Nick's Concerns

Nicholas Coutts Concerns

On 1 Aug 2024, at 10:38, Nicholas Coutts <nick.coutts@sense-ims.com> wrote:

Nick Coutts, is a veteran of coaching a very large number of social startups in Europe. His experience is valuable and respected. [Coutts].

I asked him for advice on making this book useful, and this, at right, is the set of ideas.

I am going to use it as a checklist when writing the book. And I hope both Nick and others will contribute more ideas.

I thought that I could usefully draft a set of direct answers, here at the end of the book. Because I can refer, now, and after the book is more developed, to the full text of the book.

I also hope this early condensed answer, will provoke readers to come up with even more concerns based on startup experience.

Tom

Would be great to demonstrate how Planguage can be used to address these issues facing start ups:

How to design their project. Most start developing a product or service without having designed their project. The project design has to include a design for the business, not just an MVP. The design of the business can affect the design of the MVP and vice versa. The challenge is to use a design for a project that allows information to emerge and be incorporated.

Lack of understanding of what competences are needed and when they will be needed. Lack of competence = risk. Risk to time, cost, quality.

Decisions based on assertion and assumption when decisions need to be based on evidence.

Spending too much time on doing what they CAN do rather than what they SHOULD do.

Testing only a single design rather than evaluating options, ie design alternatives.

Budgeting based on cost rather than value.

Rewarding team members in advance with equity rather than paying in arrears for value created.

In summary, the big challenge is to design the project when you don't know what competences you will need to reach your goal - the competences in what tools to use and when to use them to run tests to get the evidence needed for the next iteration. Hence the need to focus on how to get evidence but having to default to making something that you know you can do when you don't know what tests to run and how to run them. And how do I size the iterations, ie the work in progress, so that I minimise the risk of going in the wrong direction? I may need to alter the size of each iteration.

We can of course use a credibility rating for our confidence in the action, eg using the tool or test and ouf confidence in the evidence produced by the test.

All this points to an evolutionary approach to project design and management.

Nick

Startup Project Design

"How to design their project.
Most start developing a product or service without having designed their project.
The project design has to include a design for the business, not just an MVP.
The design of the business can affect the design of the MVP and vice versa.
The challenge is to use a design for a project that allows information to emerge and be incorporated." <- N.C.

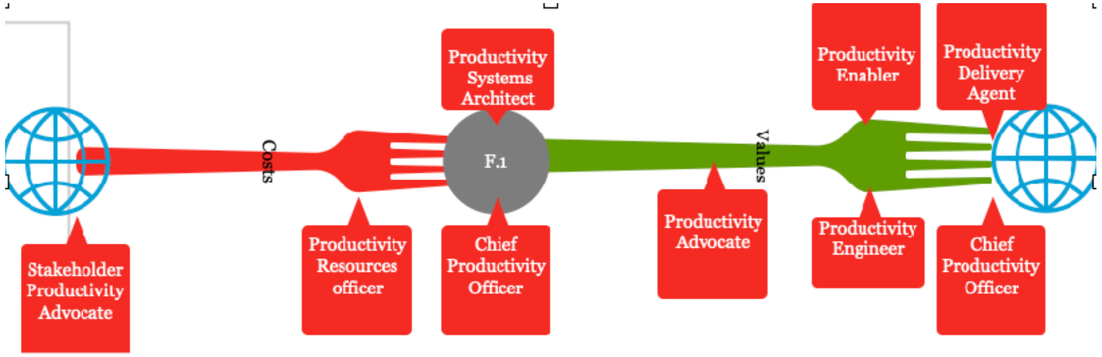


- **Project Design**: defined as, *deciding on the culture, and tools needed to bring an uncertain process, to a successful completion.*
- It is the project culture and tools, which will help you decide competitively, on the qualities, costs, efficiency, competitiveness, and success of the products and services, which your startup produces.
- We suggest that the main, initial step, is to choose the Evo-cycle. The Evo-cycle contains most of the necessary components to succeed. It does not contain the human components of competence and motivation. But it does contain tools for motivated and competent people to succeed.
- We have detailed the 'how' this works in detail in this book, and in the supplementary references.
- The Evo cycle makes sure that, you consider your critical *stakeholders*, their critical *values*, and that you *design* your products and services, to meet a realistic set of value-objectives, within necessary constraints. It makes sure that you take small (2%) steps, and *then* get feedback from reality. So that your decision-making is based on reality, not weak hopes and loose assumptions
- The 'Evo cycle decision' applies to your two main design areas: your *startup organization*, and then your startup *products and services*.
- Once you have make the strategic decision to use the Evo-cycle, for your business, you can use it to plan other parts of your organization, in addition to the Evo process. You can start with your organization *stakeholders*, then their critical *values*, then design some organizational *strategies* to deliver those values (like motivation, communication, competencies).
- You can continue this Evolutionary organizational process as you grow, go through stages of maturity, and get feedback or weaknesses, about your products and services, with the current organization.
- If you start with the Evo Cycle, you will have started with a *powerful thinking and decision-making pattern,* which will help you *see* and *solve* problems, earlier.

Competencies

"Lack of understanding of what **competences are needed** and **when they will be needed.**
Lack of competence = risk. Risk to time, cost, quality." <- N.C.

- **Competencies**: *defined as*: *human skills for developing successful efficient long-term products and services.*
- Startups start small; grow rapidly, and cannot ever recruit the best most-experienced experts in every needed domain.
- Competencies are many, and they are multi-dimensional (*knowledge, experience, up datedness, motivation*)
- In general, 'smart motivated people' are a very good start. An attitude, ability, and willingness to learn anything, and try to be good at anything, is a good start.
- An ability to seek feedback, correction, and ideas - from AI, from our Evo-cycle, and from outsiders, is a useful corrective, to youthful inexperience (or to old timer's cherished habits)
- The primary competency for founders, leaders, and primary planners, initially is probably making really smart decisions, about strategies that they *can* competently execute, with any help, inside or outside, they can actually *get*.
- One of the Value-requirements for their decisions, can be *'the degree to which they have access to sufficient competency to execute the strategy or architecture well' ('*Organizational Competency Requirements'). 'Design to Our Competencies'
- In small growing organizations, general competencies, and flexibility in learning, and in accessing outside specialities, is a necessary virtue.
- Lack of ideal competency is going to increase the probability of less than perfect qualities, and time and financial costs. But our smart strategic decisions, and excellent feedback mechanisms will still give us early and incremental value stream production. [Productivity Stream].



Source: [Optima] some imaginative areas of competence, among many others.

The good news is that these skills are not well taught at schools of higher education. You are going to have to learn and teach them on-the-job

Some *'Value Planning'* Roles to Consider

From [Value Design]

Organizing the Value Design Planning process. Who are they players to make value delivery really happen? What responsibilities and skills do they need?
Here are some suggestions for specialist roles. These are mainly responsibilities, rather than full time jobs. But they do require training and knowledge.

Value Analyst: analyzes stakeholder needs, and priorities, and selects critical, or possibly critical, needs and specified them as requirements, at least at the 'Wish' level (potential Goal requirement).

Specification Owner: a person (or group) which has undertaken responsibility, by name, for the update and maintenance of a specification object, such as a requirement, a design, or a table.

Implementation Responsible: a person (or group) which has taken named responsibility, as specified in the specification object, for actual practical implementation of a design object. This can be for a requirement level (reach the requirement Goal), or for a design (deliver the design and try to get the maximum value from it).

Value Designer: a generic (all possible design areas) designer (or team) who undertakes to identify possible design components to reach a Value Requirement level, on time. To research them as to all side-effects and costs, documenting such facts in the design object and corresponding Value Tables. The Value Designer might hand over exploration of a design idea to a Specialist.

Value Engineering Specialist: a designer with a narrow speciality (usability, security, performance, organizational improvement, AI) who is updated on the state of the art, and has a good international network of people and sources to find good specialist designs.

Value Architect: A person or team, who sits at the Apex of the system, and synchronizes all ongoing efforts in order to get maximum necessary value for available resources. Manages the top critical values, and the top level design architecture.

Stakeholders: the dozens or more sources of requirements which the designers need to know about, and make sure their design fits them.

Value Process Manager: a person or team responsible for getting a best possible value stream flowing from the other people involved. Sort of like old project manager, except they are focussed on the Values/Costs numbers, not building stuff. They allocate resources (money, time), and assign people to specialist tasks.

Value Quality Control: these people carry out Specification Quality Control of specifications, to make sure the Defects Per Page is economically low enough before Exit to any other process.
They are also responsible for measuring value levels and costs after incremental implementation. They will check that designs are in fact implemented as specified by suppliers for Exiting to integration delivery.

Value Policy: this is the written policy that gives clear guidance to the Value process, from organizational management. Perhaps Chief Technical Officer level.

Value Contracting: this contracting can be done at both internal levels and external suppliers, and basically motivate suppliers to deliver value and get rewarded for it.

Value Suppliers/Sub-contractors: internal and external to the organization people or organizations who undertake a specific responsibility, for construction, implementation, organizing, designing, or any other activity needed to deliver value.

Value Design Builders: people and organizations who build, physically, logically or organizationally, any design component or related activity.

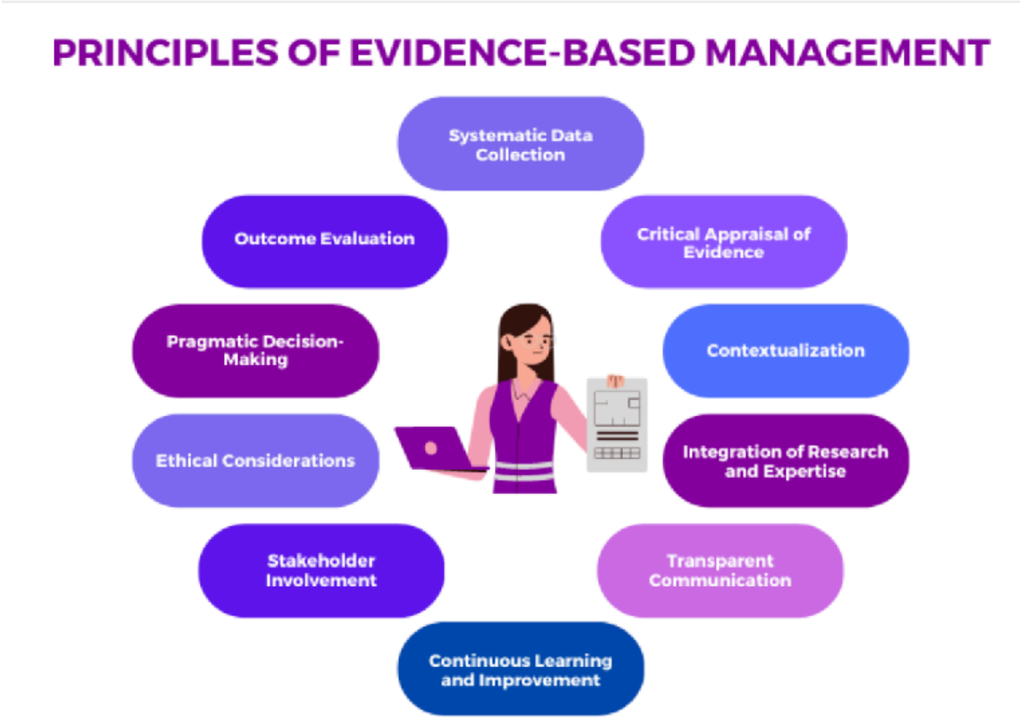
Value Director: the person, or group, responsible for focusing on the Value Delivery, and reporting to a steering committee or Board about the plans and accomplishments to date in Value

Source [Optima 12.1]

Sound Decision-making

Decisions based on **assertion** and **assumption** when decisions need to be based on **evidence**." <- N.C.

- It is tempting to make decisions based on appeals to 'common sense' (*'it is obvious'*) or authority (*'Corporation X does this'*), but if this is a wrong decision for your unique little startup, you can pay too heavy a price, for this simplicity in decision-making.
- Your culture *needs* to be skeptical.
  - As Musk's First Principle says, *Trust no-one, assume the requirement or design is wrong, and work towards improving it* [MM].
- We need a culture that will always politely question bland assumptions, with at least *'How can we be more sure it will really work best for us, now?'*
  - *If nobody can provide convincing evidence, then*
    - *Move to an alternative safer, more well-founded strategy*
    - *Or at least plan a short cheap evo cycle to prove or disprove your hypothesis*
- Caution, in the face of unwarranted optimism, is a very small price to pay, to avoid terrible costly strategies. A drop of prevention is worth a truckload of cure.

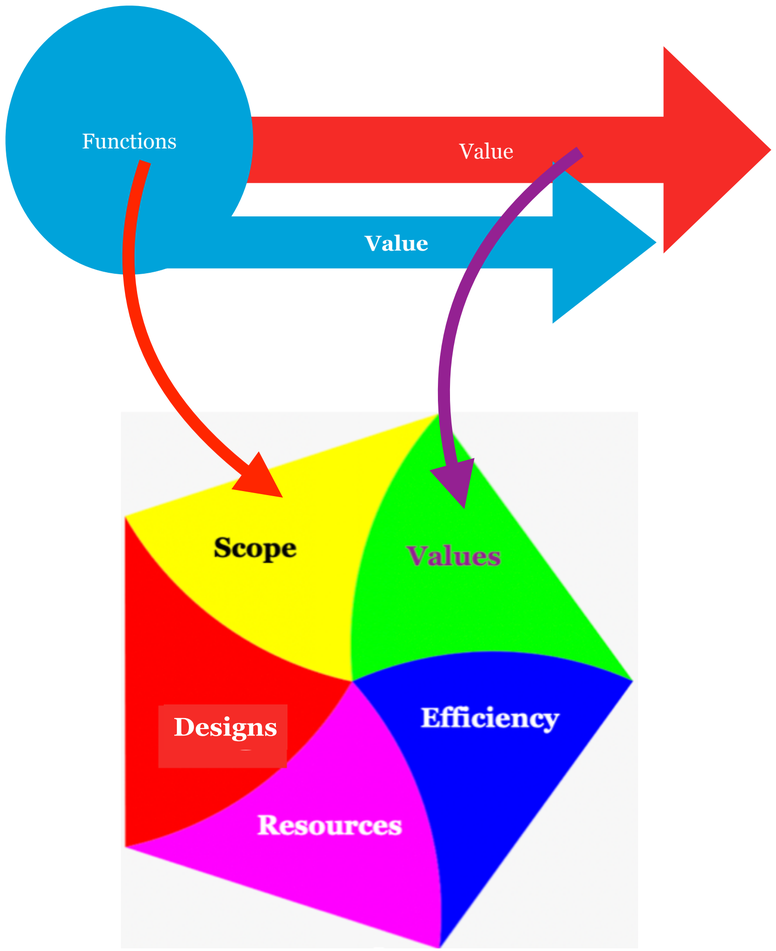


https://chisellabs.com/glossary/evidence-based-management-why-is-it-important/

Smart Priorities

"Spending too much time on doing what they CAN do rather than what they SHOULD do." <- N. C.

- People will, if you pay them, do absolutely useless work, in the hope that you know or believe it is a good idea.
- So somebody has to be in the business of determining what the smart things to do, actually are ('prioritization')
  - The 'prioritizer' can be a planner or a self-managing team, or both.
  - The *main* ingredient is the *basis* for prioritizing the important things.
- That basis has to be primarily **'Value Delivery',**
  - And secondarily *'cost reduction'.*
- Most organizations, conventional cultures, are really bad, at even stating their critical values clearly. They never learned or experienced it.
  - 'They use ambiguous words
  - They fail to quantify values
  - So they *rarely have* a sound basis for smart (value-based) prioritization.
  - This is incredible, and a total failure of many types of higher education. Many scientific, medical and engineering disciplines are much better than say, IT, and management.
- So get your act straight! Define your value priorities carefully and clearly, then use them to decide what you should be doing now, *this* Evo-cycle.
- Use an Impact Estimation Table [VIET, CE] to visualize the necessary logic of 'what to do next' for best 'value production'.
  - That is what your social startup is all about!



[Simple, 2022]

Design Optimization

PROBLEM: “Testing only a single design, rather than evaluating options, i.e. **design alternatives” <- N.C..**

- It is all too easy to focus on a sexy-sounding design idea. Get it done!
- But, an hour spent looking at *possibly sexier* alternative designs, is probably well spent time, before building things that can take weeks.
- There should be *explicit friendly competition* to find a better design, maybe just in one critical-dimension of value or cost, but making it significantly better
- Any design will have to compete on at least 10 values, and several cost aspects, so a simple ‘gut feel judgement’ is NOT smart, and NOT likely to be the ‘best *overall* design’ you can find.
- Let people with potential ideas, enter the friendly competition and let them argue their ideas with estimates and evidence on an Impact Estimation Table.
  - Let the best design win. It takes a day of ‘mind work’, before you expend weeks of body work.
  - This is what architects and engineers do that makes them more successful than ‘managers’ and IT people. [Fylvbjerg]

STRATEGIES –––> Values / Objectives	Digital Transformation	Carrier-Centric Payment	Customer Experience	Debt Restructuring
CarrierSatisfaction	70 %	90 %	80 %	10 %
CarrierPaymentSpeed	80 %	100 %	40 %	60 %
ShipperRetention	60 %	20 %	90 %	20 %
ShipperSatisfaction	70 %	30 %	90 %	10 %
EmployeeRetention	50 %	10 %	30 %	30 %
EmployeeSatisfaction	60 %	20 %	40 %	40 %
DebtReduction	30 %	-20 %	50 %	100 %
CreditRating	40 %	30 %	50 %	90 %
ComplianceRate	80 %	40 %	30 %	20 %
SystemModernization	100 %	60 %	70 %	30 %
Total Impact	640 %	380 %	570 %	410 %

SOURCE 2.4.2 [Plan-gineering]

Looking at 4 design options through a Value lens.

Projects that were rigorous in design evaluation
Were the ones that survived

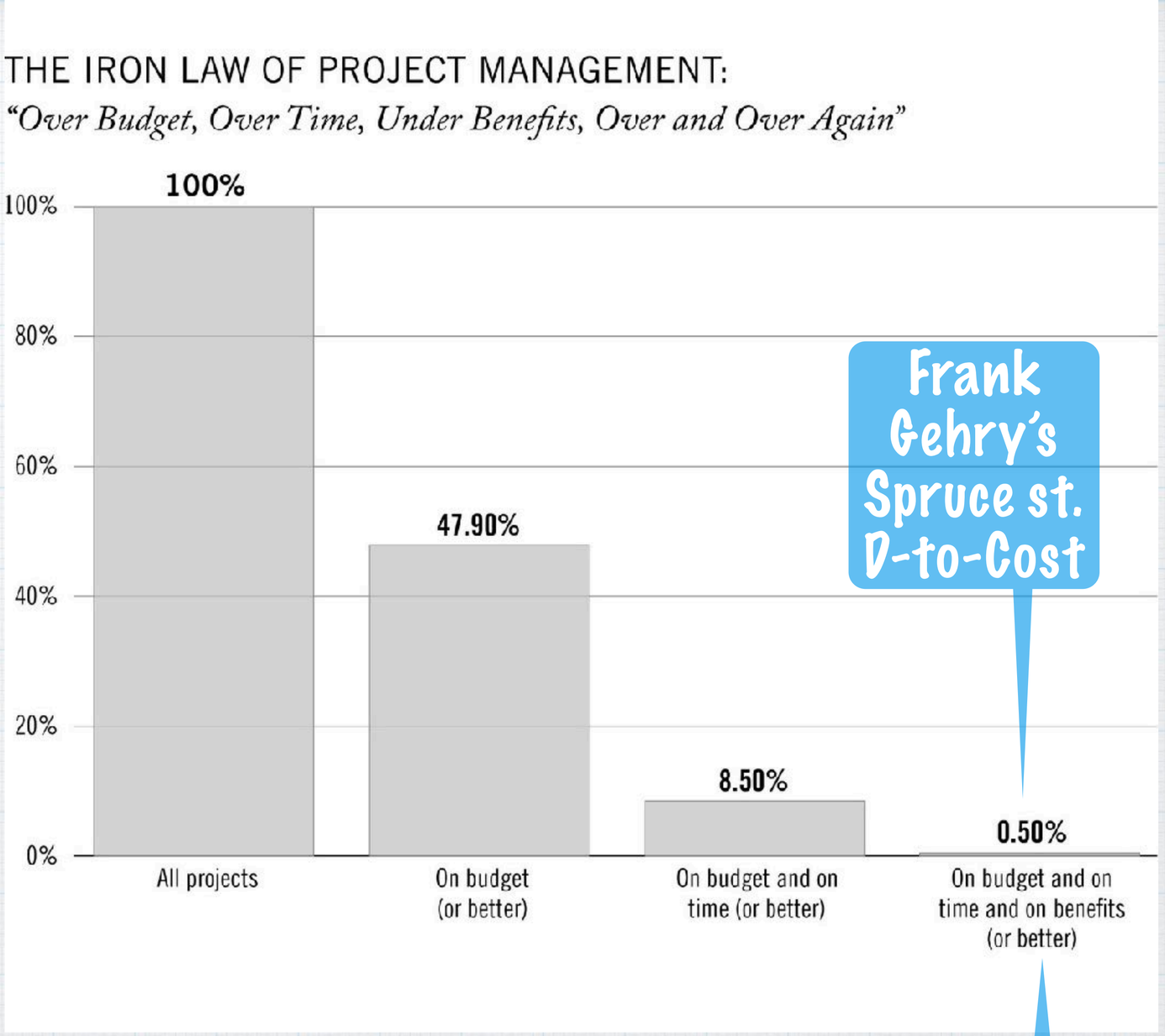


RECOMMENDED BOOK!



Bent Flyvbjerg

- Database , for 16,000 Big Projects
  - Including many IT
  - 99.5% failed to deliver 'benefits' on time and under budget
  - My definition of Project 'Success' ($, t, V)
- Those that did: spent a lot of time up front planning in depth, 0.5% !!
  - They did NOT just 'start agile' without requirements, design and quality control



16,000 project base, (Flyvbjerg)

FAILURE RATES

Value Budgeting FIRST: Cost Budgeting secondary.

<u>Problem</u>: "Budgeting based on *cost*, rather than <u>value</u>." <- N. C.

- Your startup problem is NOT primarily cost.
- If you produce enough value early and clearly, your funders will be motivated to increase your funding to get more of values.
- Focus on value first.
- Define a minimum value 'budget', called the Tolerable level, and get there initially.
- Defined a Wish or Goal 'Target' level for full satisfaction, and get there before you 'really' run out of money.

Security:

Scale: % probability of detecting a hacker within 5 seconds

Status: 10% last year.
(Benchmark level)

Tolerable: 80% by End this year.
(Constraint Level)

Wish: 98% by End Next Year.
(Target Level)

Source: [Value Management] Slides

Value Motivation

Problem: "Rewarding team members in advance with equity rather than paying in arrears for value created." <- N. C.

- "Rewarding team members in advance with equity" is a habit from the Money Greed Startups.
- We are social value creation startups: so our motivation has to be reward for value created and maintained.
- We can and do quantify *both,* our *values* and the *levels* of the values we want, and which we have *attained*.
- There is no technical problem in rewarding teams for achieving those values.
- That, of course will focus their attention!

Bill of Rights

- You have a right to *know* precisely what is expected of you.
- You have a right to *clarify* things with colleagues, anywhere in the organization.
- You have a right to *initiate* clearer definitions of objectives and strategies.
- You have a right to get objectives presented in *measurable, quantified* formats.
- You have a right to *change* your objectives and strategies, for better performance.
- You have a right to *try out* new ideas for improving communication.
- You have a right to *fail* when trying, but must kill your failures quickly.
- You have a right to *challenge* constructively higher-level objectives and strategies.
- You have a right to be *judged* objectively on your performance against measurable objectives.
- You have a right to *offer* constructive help to colleagues to improve communication.

Original version in (Gilb 1988 Page 23)

Value Motivation

Not Greed

Principles for ValueMotivation

1. If the value is not *clear*, you cannot motivate

2. Incompetent managers and professionals are quite happy to hide behind fuzzy value statements.

3. *Quantified* Values, motivate suppliers, often through contracts and payments.

Policies for ValueMotivation

1. All critical values will be clear, quantified, structured for detail, and measurable in practice.

2. People will be trained and equipped to quantify and measure critical values.

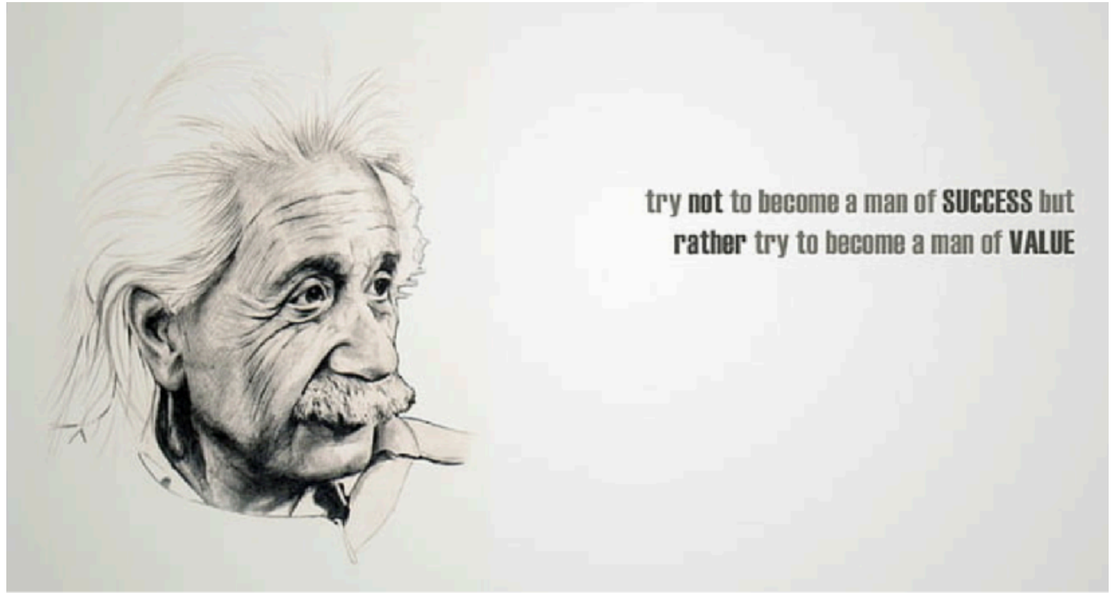
3. Real value delivery will be rewarded, acknowledged, credited and honored.

ValueMotivation Checklist

1. How clearly do you credit individuals and especially teams with value delivery? Publish, talk about it, document it, award it.

2. Have you made it clear that managers and teams do not *deserve*, and will not be *given*, budget and responsibility; if they are *not* working towards, and delivering, their critical Value set?

3. What would happen if you were to make use of the **Bill of Rights for Company Communication?** (previous page))

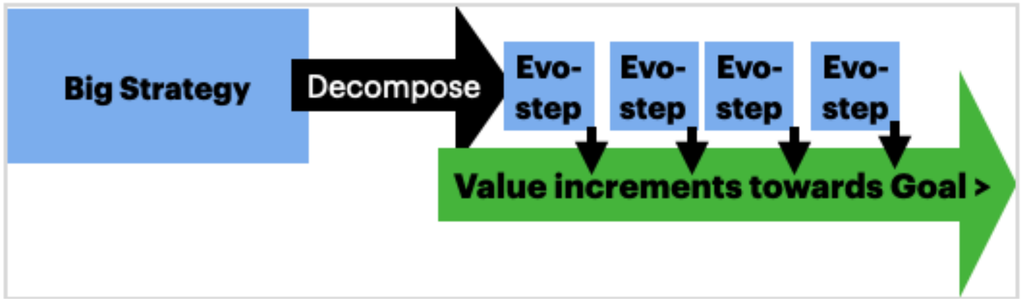


SOURCE: PAGE OF 76 101 © 2019-2024GILB.COM [VALUE MANAGEMENT]

Smart Evo-step Sizing: 'Decomposition'

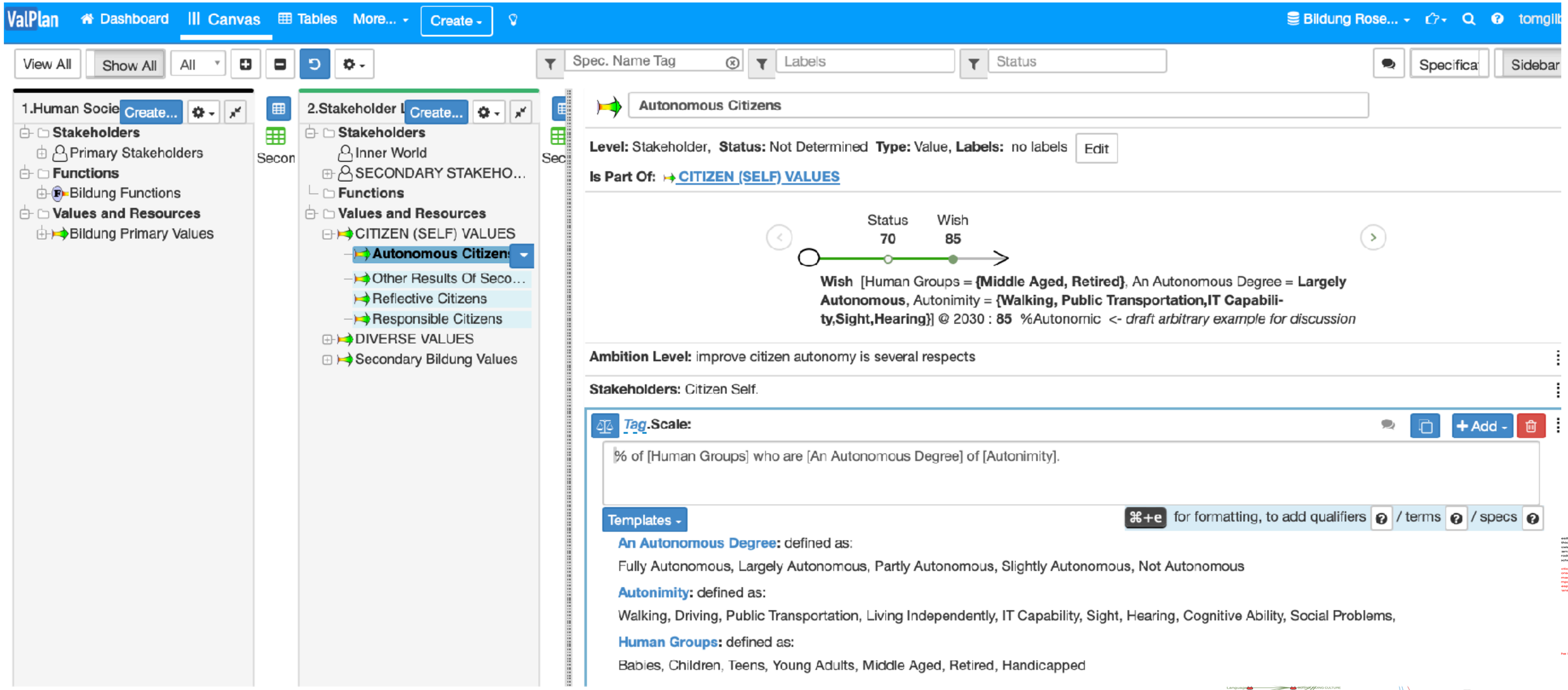
"And how do I size the iterations, ie the work in progress, *so that I minimise the risk of going in the wrong direction?* I may need to alter the size of each iteration." <- N. C.

- I use a mixture of sizing strategies.
- The first and simplest is 'always have weekly increments'. Then, by definition, you cannot lose more than a week.
- We have decomposition methods, in this and other books, whereby any strategy, of any initial size, can be decomposed into weekly increments.
  - My AI systems (see [Optima, 4]) have no problem giving me such decompositions.
  - You just have to remind them, as I have to remind my classroom students, that each sub-strategy has to be, NOT a task, but an Evo value implementation step, independent of all other sub-strategies.
- Musk [MM] has single day, 10 hour, Mobbing, increments which deliver real value at the end of day. Pretty extreme for a car production situation, but they do it. It helps that they can change the car software immediately for the production line! Other manufacturers cannot.
- DevOps, and Lean Startup with a stream of 'code changes' is not necessarily the same thing. They will fix bugs, and change functionality, but will not necessarily improve any top 10 critical Value Objective.
- Sometimes building a strategy demands a lead time, far-exceeding a week, in which case we put it in the 'Backroom' and let it develop, which we do other value delivery steps, until it is ready for delivery [CE].
  - Think 'kitchen' (backroom), and duration of food preparation, then fast delivery to restaurant guest (frontroom).



Source [Optima] Figure 4.2.3 B. **Decomposing into Evo-steps, not 'tasks', enables us to get measurable results, feedback, and planning improvement faster**

Decomposition by ‘Condition’ (where, what, who, why, etc.); uses ‘Planguage Scale Qualifiers



Source [Plan-gineering] 3.4.4 On the left, canvas mode, overview. A click on **’Autonomous Citizens’** gives us the detailed specification (above right), including the detailed [Scale Qualifiers], like for ‘Human Groups’ which are a Planguage tool, for decomposition in many simultaneous dimensions. This capability eases finding high-value, short, delivery steps.



References

[In Square Brackets]

[**AE**] **AGILE ENGINEERING, Why we need to succeed at scale.**, India Lite Keynote TALK 6th SEPT. 2022, SLIDES: https://tinyurl.com/IndiaGilbSlides2022, SLIDES (FULL SET) https://www.dropbox.com/scl/fo/jz4lh97cvgbmmcvb35uoy/h?rlkey=kb7lux8smmg20ti29y4iccb4u&dl=0, VIDEO: https://www.youtube.com/watch?v=qU_LLaPD9h4

[**Agile Manifesto**] Gilb, **How Well Does the Agile Manifesto Align with Principles that Lead to Success in Product Development?',** https://www.ppi-int.com/wp-content/uploads/2021/01/SyEN-062.pdf

[**Akson**] Norways One Patient One Journal IT scandal. Translate via Google

https://tidsskriftet.no/2023/01/leder/helseplattformen-en-it-skandale-i-midt-norge

https://tidsskriftet-no.translate.goog/2023/01/leder/helseplattformen-en-it-skandale-i-midt-norge?_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp

[**Anthes**] Anthes, Richard A., Mark W. Maier, Steve Ackerman, Robert Atlas, Lisa W. Callahan, Gerald Dittberner, Richard Edwing et al. "**Developing priority observational requirements from space using multi-attribute utility theory.**" Bulletin of the American Meteorological Society 100, no. 9 (2019): 1753-1774.

https://journals.ametsoc.org/view/journals/bams/100/9/bams-d-18-0180.1.xml?tab_body=fulltext-display

[**Bible**] **"Bible Analysis": a multidimensional analysis of some holy writings, using Planguage,** FOLDER W UPDATES: https://www.dropbox.com/scl/fo/2efrx4wsycluuuuwowxlb/ADmmke5SC962FbEFbA_mjPs?rlkey=elnq2fpcz8ytzc5qod8pw4zem&dl=0

[**BOOK**] **'BOOK',** https://www.dropbox.com/scl/fo/40ktw5dxitbpnsb8s1cyi/ANQhp2RyCcPyP_a3rYx2wIk?rlkey=qubhfbmadq1xzv5g25p999lo2&dl=0, June 2024, Analysis of a book manuscript by AI Gemini Pro.

[**BRODIE**] PHD, Lindsey Brodie, PhD, 2015, Title: "Impact Estimation: IT Priority Decisions", Middlesex University online publications., https://www.dropbox.com/scl/fi/1pxg9s6pgap49zc8a8kuh/Lindsey-Brodie-PhD-Thesis?rlkey=zxweopmd4vwiijzej757qkzia&e=1

https://repository.mdx.ac.uk/download/1aba a4e1235e6fc864c30161a581b6f27f993ab730653fda524d5209827829c3/19805254/LBrodieThesisEMBARGO.pdf

[**BSC**] https://hbr.org/1993/09/putting-the-balanced-scorecard-to-work#:~:text=The balanced scorecard, first proposed,coherent set of performance measures., see also book Value Requirements 2019 for comment on BSC

[VR] Value Requirements book, https://tinyurl.com/ValueRequirementsBook

[**CE**] **Competitive Engineering** (2005) book free, https://www.researchgate.net/publication/237129623_Competitive_Engineering_A_Handbook_for_Systems_Engineering

Chapter 10. Evolutionary Project Management

[**CG**] ConceptGlossary (see below CG set)

[**CGP**] **A Conceptual Glossary for Systems Engineering: Define the Concept, don't quibble about the terms.**,Gilb. 2006, Paper INCOSE 2007, https://www.researchgate.net/publication/286238165_104_A_Conceptual_Glossary_for_Systems_Engineering

[**CG Full**] **New Full Glossary**, 692 Concepts July 2024, https://www.researchgate.net/publication/381861406_Glossary_PLANGUAGE_MASTER_010724_Edit_for_Researchgate, July 2024, DOI: https://doi.org/10.13140/RG.2.2.18838.77121, LicenseCC BY-SA 4.0

[**CG-CE**] **Planguage Concept Glossary as edited in Competitive Engineering book** 2005, CE Glossary, https://www.researchgate.net/publication/381861274_321_Glossary_from_CE_copy_for_glossary_folder, no DOI possible on this chapter.

[**Confirmit**] see Case studies , and presenter notes on first page of Confirmit Case. Confirmit Case URL
https://www.slideshare.net/slideshow/tom-gilb-project-management-driven-by-the-top-ten-critical-improvements-quantified-bcs-apm-15-april-2013/19004735#64
Contains a set of Confirmit slides

[**Cost Engineering**] Tom Gilb: "**Cost Engineering**:"**How to get 10X better control over resources, and value for money"**, 2023, , https://tinyurl.com/CostEngFree, https://leanpub.com/costengineering

[**Coutts**] Nicholas Coutts,: GilbFest 2024 Slides: https://www.researchgate.net/publication/381880176_Some_examples_of_technology_for_good_Gilbfest_Nick_Coutts?channel=doi&linkId=66832f8af3b61c4e2ca6504c&showFulltext=true

https://www.sense-ims.com/about, Imperial College, University. Barcelona, Report on 220 startups, with 83% survival rate, over 3 years. Using [CE] and Planguage methods. Based on Talk held June 2024 at GilbFest (On line). Slides available (above), and a specific reference email about the role of Planguage (3.5). Tom Gilb has repeatedly taught at Imperial for NC classes, and directly helped with some startups (https://www.loowatt.com/our-team.html, and https://claned.com) [https://www.sense-ims.com/about] Nicholas Coutts, https://www.researchgate.net/publication/381880176_Some_examples_of_technology_for_good_Gilbfest_Nick_Coutts, (same as DOI above) *"All these challenges can be understood and managed using the principles in Competitive Engineering by Tom Gilb, especially the concepts and practical approaches in Scales of measure (chapter 5) and Impact estimation (chapter 9). Tom has with great generosity given me permission to make a pdf of his book available to all the students participating in the programmes in which I am involved and I know that his thinking has had a significant impact on improving the survival rates of startups." See 14.3.*

[**Cyber-Resilience**] Tom Gilb, Cyber Resilience Planning, with Jussi Mäkelä, 177 Pages, July 2023
https://www.dropbox.com/scl/fo/5aabtekld0i68fw915hwp/h?rlkey=alctzhrack8zc14igudf44s49&dl=0

[**Datocracy**] **Datocracy and a detailed architecture of a possible political future.**, By Tom Gilb, leanpub.com/Datocracy (Cheap), https://tinyurl.com/Datocracy (FREE)

[**DE**] Tom Gilb**, Decision-eering,** https://tinyurl.com/Decision-eering, booklet, pdf, 2022-3, https://www.researchgate.net/publication/383747726_DECISION-EERING_Decision_Engineering_rigor_and_logic_in_a_through_multi-_dimensional_decision-_making_process

(The RG link incase access to the talk slides below)

[DE-talk] **Simple Decisions: Decision-eering**, Slides GilbFest Talk 270623, https://www.dropbox.com/scl/fo/fbs61kftw6u1k1e25vowp/ALmrh3g_MnU8oQBztgoeQOI?rlkey=sxyzxx13bhdb6wbnx3eh8bztc&dl=0

[**Decomposition**]. https://tinyurl.com/VPDecomposition, Value Planning book, Chapter 5.

[**Dion**] https://insights.sei.cmu.edu/documents/1138/1995_005_001_16415.pdf
Raytheon Electronic Systems Experience in Software Process Improvement

[**DL**] Tom Gilb, **Design Logic Paper:Design Process Principles**. October 2015-2024, Researchgate PDF: https://www.researchgate.net/publication/381918566_The_Logic_of_Design_Design_Process_Principles?channel=doi&linkId=66848e06714e0b03153f4825&showFulltext=true

[**Dowie**] Risk Decision and Policy 4 (1), 57-73 (1999), Against risk, JACK DOWIE,, Applied Social Sciences, The Open University, Milton Keynes, MK7 6AA, UK, https://www.researchgate.net/publication/249015754_Against_risk

[**DoD**] Systems Engineering Guidebook, https://ac.cto.mil/wp-content/uploads/2022/02/Systems-Eng-Guidebook_Feb2022-Cleared-slp.pdf, (2022) Uses the Evolution concept often. This is not the same as my Evo method. But it is close and shows terms used.

[**Do No Evil**], Tom Gilb **"Don't Do Evil Systems",** 2024, Gilbfest

SLIDES: https://doi.org/10.13140/RG.2.2.25270.92487, https://www.dropbox.com/scl/fo/1slcn3gn9yfcgo83hjtew/ACJkxe1RoCh7Dw_VYvr5RCc?rlkey=g1o1axmuphjlygpg9wijfdz72&dl=0

[**DPP**] slides collection DBX Talks/DPP
Defect Prevention Process -IBM
https://www.dropbox.com/scl/fi/b3kytqnz95qix4jrrume2/DPP-SLIDES-PDF-2014-AT-DBX-2024.pdf?rlkey=d54zkfbh7e7j0aro3bgxftfiu&dl=0. See also Software Inspection book

Mays and Jones IBM SJ paper on **Experiences with DPP**
http://agileconsortium.pbworks.com/w/file/fetch/1527643/Mays1990ExperiencesDefectPreventionIBMSysJ.pdf see also [**Mays**]

[**DtC**] **Estimation: A Paradigm Shift towards Dynamic Design-to-Cost and Radical Management**
July 2012 INCOSE International Symposium 22(1):1929-1946, by Tom Gilb
https://www.researchgate.net/publication/285403494_Estimation_A_Paradigm_Shift_towards_Dynamic_Design-to-Cost_and_Radical_Management#fullTextFileContent
(linktested 030924)

[**Evo1985**] https://www.researchgate.net/publication/213878217_Evolutionary_Delivery_versus_the_waterfall_model

[**EvoKai**] https://www.gilb.com/offers/g56Dh5Gn/checkout, 'Evo eBook
– Evolutionary Project Management & Product Development,' Author: Kai Gilb, 14 EUR, 2007

[**EvoBook**] Evo: Evolutionary Value Optimization, Tom Gilb, 2024 (September 2 to …)

https://www.dropbox.com/scl/fo/kgy4wnqsz94nj7xahhp6p/ALsE-c603GAV0_942TdSMjg?rlkey=jw4jivjswcfsxujpep2dn148x&dl=0

[**Evo Cycle**] Tom Gilb, **The 'Evo Cycle' applied high and wide: A 'development process' simplification,** SLIDES:https://www.researchgate.net/publication/381918278_The_'Evo_Cycle'_applied_high_and_wide_A_'development_process'_simplification, FOR GILBFEST 27 JUNE 2023 5 MINUTE TALK

[**EvoWeek**] **An Agile Project Startup Week; Evo Start,** https://www.researchgate.net/publication/381155073_An_Agile_Project_Startup_Week_-_Evo_Start, 2 Files here. The second is the day by day syllable for the startup week.

[**Flyvbjerg**] Flyvbjerg et al: **How Big things Get Done**: https://sites.prh.com/how-big-things-get-done-book, 2023. See 15.13/14 here in Evo book.

[**Governeer**] Tom Gilb,
Governeering: Government Systems Engineering Planning.
https://leanpub.com/Governeering, 2021, $5-$10, 2020, https://tinyurl.com/Governeering (FREE), 2020

[**Guides**] Tom Gilb:
Guides: A broader and more-advanced 'constraints' theory. Theory of Guides (ToG),
https://leanpub.com/guides, 83 Pages, August 2023, update 020324,
https://www.dropbox.com/scl/fo/4amgzl6wuieo8vfy4hgk0/h?rlkey=rkkszv3yrtrv0twoprdnnm5pl&dl=0
https://www.researchgate.net/publication/381913042_A_broader_and_more-advanced_'constraints'_theory_Theory_of_Guides_ToG

[**HP**] —-HP Cases Evo— – – – – – – – – – – –
The Evolutionary Development Model for Software
by Elaine L. May and Barbara A. Zimmer
August 1996 Hewlett-Packard Journal
@inproceedings{May1996TheED,
title={**The Evolutionary Development Model for Software**},
author={Elaine L. May and Barbara Zimmer},
year={1996}, url={https://api.semanticscholar.org/CorpusID:18883699}}
and also (Full issue) https://vtda.org/pubs/HP_Journal/HP_Journal_1996-08.pdf

[HP2] Evolutionary Fusion: A Customer- Oriented Incremental Life Cycle for Fusion
by Todd Cotton
https://vtda.org/pubs/HP_Journal/HP_Journal_1996-08.pdf
ACCESSED 2024, August 1996 Hewlett-Packard Journal

[**HP3**] **RAPID AND FLEXIBLE PRODUCT DEVELOPMENT: AN ANALYSIS OF SOFTWARE PROJECTS AT HEWLETT PACKARD AND AGILENT** (2001)
by
Sharma Upadhyayula
https://dspace.mit.edu/handle/1721.1/29167
https://dspace.mit.edu/bitstream/handle/1721.1/29167/48272019-MIT.pdf?sequence=2&isAllowed=y, Accesssed 2024 full paper

[**HP4**] **Best Practices for Evolutionary Software Development**, Darren Bronson
https://dspace.mit.edu/bitstream/handle/1721.1/80490/42757317-MIT.pdf?sequence=2

[**Innovative Creativity**] Tom Gilb, 2018, 'INNOVATIVE CREATIVITY' BOOKLET FOLDER
https://www.dropbox.com/sh/pxbn5l4idssiu5t/AABPwpMvgo3wGg12ZbyhqZ_Ta?dl=0, Tom Gilb, 'INNOVATIVE CREATIVITY' 124 pages €14
https://www.gilb.com/store/QMMQhn2g

Innovative Creativity: How to deliver measurable values fast.
66 Slide Talk to Startups and Angel Investors on 5April 2019
at Bærum International Hub, Norway.
First time I ever held this talk.
Contains case studies, and analysis of Design Sprint methods (2 slides)
https://www.dropbox.com/scl/fo/ckvypm7fla7rxiayzcwdi/AObIJZS6GG-yAV58_S4IwOc?rlkey=y0pr36t27bihiwhqy85zdu7kt&dl=0

[**Keeney**] Keeney, Ralph L. 1992. Value-focused Thinking: A Path to Creative Decision-
making. Cambridge, MA and London: Harvard University Press. ISBN 0-674-93197-1.
While I credit Keeney with a 3 level planning hierarchy (Fundamental, Strategic, Means) I have noted that in later writings he decided to propose a two-level hierarchy. More like 'Ends and Means' alone. I do not think it matters much.

[**KEN**] Tom Gilb

KEN: The FREE Knowledge Edu-Neering booklet
https://tinyurl.com/KENGilb
Later see Researchgate and Leanpub

[**Life Design**] - eBook
https://www.dropbox.com/sh/persq32ar7f7lr3/AAAs7GGJ7yRQ0RNDaqpfb3Xna?dl=0
FREE PDF COPY. Share with your family!

Gilb: **LIFE DESIGN** Booklet €14
https://www.gilb.com/store/kCBGcG6L
PAID copy for my rich ethical friends. Income goes to charities.

[**Mays**] R Mays et al, IBM Systems Journal 1/1990, **"Experiences with Defect Prevention Process"**, DPP

http://agileconsortium.pbworks.com/f/Mays1990ExperiencesDefectPreventionIBMSysJ.pdf

SEI CMM Level 5
"**Key Practices of the Capability Maturity Model**"
SM, Version 1.1, Mark C. Paulk
Charles V. Weber, Suzanne M. Garcia, Mary Beth Chrissis, Marilyn Bush
SEI CMM Level 5 (DPP)
http://www.sei.cmu.edu/reports/93tr025.pdf

Also see Chapter 7 and 17 in Tom Gilb, **Software Inspection**, Book (1993) which are on the DPP method. 17 is by Mays, as above.
And see [CE] book 2005 Chapter on Specification QualityControl (and DPP)

[**Mills**] Mills, H. 1980. **The management of software engineering: part 1: principles of software engineering.** IBM Systems Journal 19, issue 4 (Dec.):414-420.
Direct Copy
http://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan

Includes Mills, O'Niell, Linger, Dyer,Quinnan p- 466 on

[**MM**] **'Musks Methods': A collection of insights into Musk's Methods, which parallel my Planguage/Evo. With my interpretation of Musk's Intent. Collected and Edited by Tom Gilb from books and videos.**
https://tinyurl.com/MusksMethods
A Folder, with continuous updates, a work in progress.
July 21 2023 113 pages, 200 MB.

[**OKR**] Tom Gilb
OKR Objectives and Key Results: what's wrong and how to fix it.
https://www.researchgate.net/publication/381316727_OKR_Objectives_and_Key_Results_what%27s_wrong_and_how_to_fix_it?channel=doi&linkId=6668a046b769e769192ce360&showFulltext=true

[**Optima**] Tom Gilb,,"OPTIMA : , Balancing Critical Values"
FOLDER: https://www.dropbox.com/scl/fo/11o605py2783rnhepylko/AOYiLmQ__TE9BBf_PWvaEpM?rlkey=npdqf3qgkjgn9hf99jeajgala&dl=0
WITH UPDATES: https://leanpub.com/optimabalancingcriticalvaluesthegoodthebadandtheuglythepowerfuluseofaiforbetterfasterplann. $8
https://www.researchgate.net/publication/382649938_Optima

[**PMI**] https://www.pmi.org/learning/library/risk-management-9096, " On the contrary, risk management must be seen as preparation for possible events in advance, rather than responding as they happen."

[**Penta**] https://twitter.com/imtomgilb/status/1790030887129678169,
The Penta Model Paper Alone August 2022
https://tinyurl.com/PentaPaper
also **Chapter 4 of this (SIMPLE) book**
https://tinyurl.com/SIMPLEGilb

The Penta Model:
A simplified 'Planguage'
Penta SLIDES PDF: 29JUNE2023 GILBFEST
https://www.dropbox.com/sh/e8i719o4kh82knl/AADZxxd6GE8KtbHwnJufvR2za?dl=0

[Plan-gineering] Plan-gineering Book, in Folder, by Tom Gilb, July 2024
http://tiny.cc/Plan-gineering **(free),** leanpub.com/Plan-gineering **(paid)**
https://www.researchgate.net/publication/382298075_Plan-gineering_PDF_1st_complete_draft_1607242137 **(free)**

[**PolComm**] Tom Gilb, PolComm Better Political Communication Using 'PoLanguage', https://doi.org/10.13140/RG.2.2.23228.04483
21 pages 'booklet', in pdf, An attempt to create a method for better Politician communication., Feel free to share with anyone who might be interested.

[**PoSEM**] Gilb, **Principles of Software Engineering Management**, 1988, on page 17 you will find my quote, wrongly attributed to Albert Einstein. "Things Should Be as simple as possible, but No Simpler" (my © anyway!. There is very much still useful Planguage etc in this book, if you are looking for historical claims to ideas. I was 48 years old then (1988), with 30 years since I joined IBM. So, fairly experienced and mature. This book is credited by many as inspiring agile. See Chapter 15 for the history there. Chapter 15, Deeper Perspectives on Evolutionary Delivery
https://www.researchgate.net/publication/380874956_Ch_15_Deeper_perspectives_on_Evolutionary_Delivery_later_2001_known_as_Agile_in_Gilb_Principles_of_Software_Engineering_Management

[Priority] "Managing Priorities, A Key to Systematic Decision-making". With Mark Maier, 2005 (paper), **https://www.researchgate.net/publication/228951438_Managing_Priorities_A_Key_to_Systematic_Decision_Making** see also [Anthes] which is also with Maier.

[**Productivity Stream**] **The Productivity Stream**
https://www.dropbox.com/scl/fo/gaeg661f2syp1eif2toif/h?rlkey=kksi8g6g5gi1ucuqbfbpv5xkr&dl=0
Started 17 Aug. 2023- 15 Sept, 141 pages

[**Project Failure**] https://www.uio.no/studier/emner/matnat/ifi/nedlagte-emner/INF5700/h08/undervisningsmateriale/Project-Management-Capers-Jones.pdf

[**Quanteer**] Tom Gilb, "**QUANTeer: The Art of quantifying your value ideas**", https://tinyurl.com/Quanteer, *SLIDES:* https://tinyurl.com/QUANTEERINGeducationROME, LEANPUB.COM/Quanteer, *VIDEO: Workshop "Quantifying IT System Qualities: 'Value Agile as it should be'" , September 16th 2020, 2 Hour Workshop and Lecture* , https://lnkd.in/dxbhkU2

[**Resilience**] Tom Gilb and Jussi Mäkelä 2023,, **Cyber Resilience Planning**: For BCS 'Cyber-Resilience Planning' Workshops., https://leanpub.com/cyber-resilienceplanning/, PDF Copy, Samples, and Paid, about 170 pages July 21 2023, Dropbox pdf copy, https://tinyurl.com/ResilienceText, Free.

[**R I S K**] *R I S K - Robust Innovative Safe-Keeping, Subtitle: V A L U E : Value Architecture Lifecycle Universal Efficiency, Gilb dropbox folder with PDF:* https://tinyurl.com/RISKpdf, https://www.researchgate.net/publication/383216194_RISK_Robust_Innovative_Safe-Keeping, https://leanpub.com/RISKrobustinnovativesafe-keeping/

[**Scale Free**] Gilb, T.: **Scalable Requirements: One Size Can Fit All**. *IEEE Software, July-Aug. 2021*

https://www.computer.org/csdl/magazine/so/2021/04/09461035/1uCdHyJJMAw

https://www.dropbox.com/sh/zpiy1lo4ayeoffo/AAAetdQFh5AAMHhYTS2W3WcEa?dl=0, *(Proof copy free), See Simmons quote here on Scale Free experience. He led the effort to use Planguage at Intel . He also wrote a Foreword in [CE]*

[**SE**] Tom Gilb, **Stakeholder Engineering.**, leanpub.com/StakeholderEngineering, Released 27 July 2021, Leanpub, 177 pages., Free copy, https://tinyurl.com/StakeholderBook

[**SEA**] https://tinyurl.com/SysEntArchBook, **Systems Enterprise Architecture** (SEA) BOOK, https://leanpub.com/SysEntArchBook

[**SimPlan**] **SimPlan: Simple Planning Language: Designed for 'Essence' sharing of 100 Planguage component**s.

https://www.dropbox.com/sh/3h6iwlz29vi3tvm/AACuH_ufpPgIZF9NrnmAos31a?dl=0

[**Simplicity**] Tom Gilb, Simplicity Talk., "Simplicity: Simple Simplicity Principles", https://slideplayer.com/slide/16355870/, ACCU Conference, and UK, GilbFest 2016, Includes Minsky, 10 Principles, and Einstein

[**SI**] Gilb &Graham, **Software Inspections**, 1993, https://www.amazon.com/Software-Inspection-Tom-Gilb/dp/0201631814, "Software Inspections +", (The Manuscript edition)

"A handbook for advanced quality improvement. Process Control over Peer Checking of all software engineering documents, from contracts and requirements to test plans and code.", 1993 AS SENT TO PUBLISHER, SUMMARY AND LINK TO MS., https://www.researchgate.net/publication/31655477_Software_Inspection_T_Gilb_D_Graham_ed_de_S_Finzi

[SM] Gilb, **Software Metrics**, 1976 Studentlitteratur, Sweden, 1977 Winthrop USA. Discusses Evo page 214.

[Startup Fail] https://explodingtopics.com/blog/startup-failure-stats#, "Most VCs do poorly because early stage VCs fail on 80% of their ventures and there are few home runs to offset the many failures. Marc Andreessen, a Top 20 VC, notes that the Top VC funds invest in about 200 ventures per year and do well in about 15." (7.5% do well rate)

"Sep 1, 2023 — On average, 63% of tech startups don't make it, 25% close down during the first year, and **only 10% survive in the long** run. Venture-backed" ...

[**Success**] Tom Gilb, **SUCCESS : Super Secrets & Strategies for Efficient Delivery in Projects, Programs, and Plan**s, Book Folder, tinyurl.com/SUCCESSGilb, October 2021. FREE, VIDEO 1 HOUR WESTFALL https://www.youtube.com/watch?v=8jnnBS-dNog, (7 SEPT 2022), Success Engineering Leadership University Lecture:, https://www.youtube.com/watch?v=YQ6rRMLoe_s, 3Hours video, see also slides , https://www.dropbox.com/sh/re0a3qlip9dncga/AACHHu5Qu6hZyQKKz1iyMG56a?dl=0

[**Technoscopes**]: Tools for understanding complex projects, By Tom Gilb, 2019, https://www.dropbox.com/s/11arbpu4mtwc7i9/TECHNOSCOPES%209Jan2019.epub?dl=0, Epub, Free Copy, Tested 9aug24, https://www.gilb.com/store/Pd4tqL8s, Price €14, https://leanpub.com/Technoscopes, $8.

[**TEDx**] Quantify the un-quantifiable,tinyurl.com/GilbTedxTom Gilb, Trondheim, 28 September 2013

[**Terzakis**] John Terzakis,"**The impact of requirements on software quality across three product generations," ,** 2013 21st IEEE International Requirements Engineering Conference (RE), Rio de Janeiro, 2013, pp. 284-289., https://www.thinkmind.org/download.php?articleid=iccgi_2013_3_10_10012

[**TIA**]: **Toolkit for Institutional Architecture** (Digital Book), by Tom Gilb, 2024, DROPBOX UPDATED PDF IN FOLDER: https://www.dropbox.com/scl/fo/j1zze3umyxy24vg7mc7so/AEhbtiiHM261HZjpRFx9Tm8?rlkey=7v7khid4dwvh7sannv5dvc2nh&dl=0, A proposal for TIAL.org, at TG Initiative., https://www.researchgate.net/publication/382049723_TIA_BOOK_MASTER_0707240005_1st_Edition, Leanpub https://leanpub.com/tiaatoolkitforinstitutionalarchitecture

[TIAL ANALYSIS] ANALYSIS OF the TOOLKIT AND SOME SUGGESTIONS FOR POLICY AND PRINCIPLES. ALONG WITH AN AI EXAMPLE OF BUILDING A MODEL IN PLANGUAGE FOR A UNIVERSITY INSTITUTION , https://www.dropbox.com/scl/fo/90finr2411ldhiaev5539/AHDlPxCxhKfnq0OW9-j6jYc?rlkey=dg9ml2t9v5hdjudu9a3x4ta15&dl=0, PDF slides: DOI: https://doi.org/10.13140/RG.2.2.18340.69762

[**Twelve Tough**] Tom Gilb, **12?: 'Twelve Tough Questions for Better Management',** 19 pages,

https://tinyurl.com/12TOUGH, (Folder in Dropbox with 19 pages pdf & epub & visiting card, free)

Gilb, Tom, 2020, 12?: **'Twelve Tough Questions for Better Management'**

https://leanpub.com/12ToughQuestions, $4.99 to $9.99 and updates free. , Free sample of part of it here.

[**Upstream**] Gilb: **The Upstream Journey: An overview of QC Test QA Practices**, 2024

https://www.dropbox.com/scl/fo/dvudhzsh5zd8c6hz1qnz4/h?rlkey=zq85uxj17tx5108p1lb9dhlfi&dl=0

Invited Chapter for book (The Knight's Journey?) by John Watkins 2024

[User Stories] '**User Stories: A Skeptical View'**, Tom and Kai Gilb, 2011, https://www.researchgate.net/publication/381155041_User_Stories_A_Skeptical_View, Gilb's Mythodology Column in Agile Record Magazine.

[**ValPlan**] https://www.gilb.com/valplan

[**Value Design**] Tom Gilb, **Value Design ,** BOOK leanpub.com/ValueDesign $5-10, VIDEO https://www.youtube.com/watch?v=y_FaiH5jt6E&list=PLKBhokJoqd3_wlvroj85YhmNfNj8ZJ8M-&index=41 , SLIDES RESEARCHGATE PPTX: https://www.researchgate.net/publication/382158693_Value_Design_SLIDES_MASTER_V_300520_plus_Haley_DS_slide_77?channel=doi&linkId=668fe71db15ba55907535l2c&showFulltext=true, Free:https://tinyurl.com/FreeValueDesign, 2019-2021

[**Value Management**] Value Management book and course slides and video 2020, Value Management, book FREE FOLDER 2021 https://tinyurl.com/ValueManagementFolder

Value Management **BOOK** (PAID ($5-$10), leanpub.com/ValueManagement

Video 'Value Management', 2.5 hours, 13 May 2020, BCS, https://www.youtube.com/watch?v=mr9gUFWj4Jg

Slides, 'Value Management'= TOMS DBX: https://tinyurl.com/ValueMgtBCS

RESEARCHGATE: https://www.researchgate.net/publication/382162365_Value_Management_Version_13052016S7_Course_Slides_PPTX

DOI: https://doi.org/10.13140/RG.2.2.23850.63685

[**Scale-Free**] Gilb, T.: Scalable Requirements: One Size Can Fit All. IEEE Software, July-Aug. 2021 , https://www.computer.org/csdl/magazine/so/2021/04/09461035/1uCdHyJJMAw, https://www.dropbox.com/sh/zpiy1lo4ayeoffo/AAAetdQFh5AAMHhYTS2W3WcEa?dl=0, (Proof copy free). Sarah Gregory Intel experience.

[**Simple**] **SIMPLE: Simplification Ideas & Methods & Principles, Logic & Engineering**,, https://tinyurl.com/SIMPLEGilb, 2022

[**Value First**] Manifesto, Kai Gilb & Tom Gilb, 2017, , https://www.researchgate.net/publication/383022244_The_Value_Manifesto_Manifestos_simplified

[**Value Planning, or VP**] Tom Gilb, **"Value Planning. Practical Tools for Clearer Management Communication"**, Digital Only Book. 2016-2019, 893 pages, €10, https://www.gilb.com/store/2W2zCX6z, https://leanpub.com/ValuePlanning, This book is aimed at management planning. It is based on the Planguage standards in 'Competitive Engineering' (2005). It contains detailed practical case studies and examples, as well as over 100 basic planning principles. **ONE PAGE VERSION FREE:** https://www.dropbox.com/scl/fi/0v52jb971ypxl1lzo2gr5/Value-Planning-One-Page-book.pdf?rlkey=u18zusxiqfo6jicejvvi2n2a3&dl=0, AND LINK TO TEDX 17 MINUTE TALK ABOUT IT.

[**VIET**] **Value Impact Estimation Table Book,** https://tinyurl.com/VIEbooklet, VIE BOOK PDF FOLDER 10 Feb. 2022

VIDEO: Westfall VIET Talk, July 12 2023, 46 minutes, https://www.youtube.com/watch?v=TsT8xKh9zYU&t=105s

[**VR**] Tom Gilb, **Value Requirements Book,** 2019, https://leanpub.com/ValueRequirements, 291 pages, $4.99 to $9.99

(**FREE**), https://tinyurl.com/ValueRequirementsBook, Critique of other methods Chapter 15.

VIDEO 1 Hour: https://www.youtube.com/watch?v=TZ6IrEyQ4Kk, Dec 14 2022, Westfall

Value Requirements Slides , 3 Hour BCS course, 22 April 2020,

Slides = https://doi.org/10.13140/RG.2.2.21157.10720

BCS Video course 3 hrs. https://www.youtube.com/watch?v=ZHrwQtG6IMw&list=PLKBhokJoqd3_wlvroj85YhmNfNj8ZJ8M-

[**Vision Engineering**] **"Value Planning: Top Level Vision Engineering", How to communicate critical visions and values quantitatively. Using The Planning Language.,** PDF: https://leanpub.com/visionengineeringtoplevelvalueplanning/ , A 64 Page pdf book. Aimed at demonstrating with examples how top management can communicate their 'visions' far more clearly.

[**100PPP**] **100 Practical Planning Principles**. Booklet PPP, https://www.researchgate.net/publication/383639680_100_Practical_Planning_Principles_260818, 100 Practical Planning Principles. Booklet PAID, https://www.gilb.com/store/4vRbzX6X