

Stakeholder Engineering:

The '*Stakeholder Systems Engineering*' Discipline

By Tom Gilb

com/max/2906/1*5ntPVDwGUcZJKKeHye8a1Q.jpeg

A Deeper Understanding of Stakeholders, their Requirements, and their Dynamics, for Large and Complex Projects.

An advanced *systems engineering* exploration using 'Planguage'.

For fun, and coining a term 'Stakeholdeering'.

Version 1.012: 26July 2021
First complete draft.
LeanPub edition
leanpub.com/StakeholderEngineering

0. Contents

https://sef.psc.nsw.gov.au/sites/default/files/2020-12/stakeholder-engagement-framework-diagram.jpg

1.0 Overview of Stakeholder Engineering

Take notes and be prepared to send me feedback, if you feel so inclined. I'll put a reminder after every chapter.

No obligation at all, just an opportunity for you!

We writers love to know you care!

Even short thanks, will do!

1.0 Overview of Stakeholder Engineering

Hand Carved Wood Vampire Stake

Stakeholder Definition.

I have a slightly deviant definition of 'stakeholder' from common use and standards. But I believe my definition is essential and necessary, for a deep understanding of stakeholder nature.

Stakeholder: (Short Version) : **Any potential requirement source.** (See details in Glossary). Longer version below.

<u>Stakeholder</u>: anything which can influence our system or be influenced by it, from which we can derive our *own* system requirements, in order to manage successful long-term operation of 'our' defined system.

This definition requires further definitions:

Anything: absolutely anything, people, organizations, laws, standards, plans, contracts, competitors, enemies, the weak , protectors of the weak, other systems, natural phenomena.

System: any *defined* systems, and their *related* systems, consisting of all system components. Technology, people, organizations, processes, laws, rules, natural phenomena.

Influence our System: to cause our defined system to change its salient performance characteristics, and costs, relative to those attributes planned and valued, as primary and critical, by a defined group of core stakeholders.

Influenced by (our system): helped or hurt to any degree.

Derive: analyze the usefulness, for our primary purposes, of respecting and delivering (prioritizing) such stakeholder requirements; as opposed to *not* prioritizing them.

Requirements: any Performance Requirements [4], including any qualities, any resource requirements (Budgets), and constraints or restrictions, in values, or states and conditions.

Manage: consciously evaluate, prioritize, dynamically re-analyze, plan for, design for, fund, be responsible for.

Successful: delivering planned objectives, within specified constraints, while dealing with risks and changes, long term.

Re-edited 8 July

https://img0.etsystatic.com/050/0/10086201/il_570xN.680034122_cott.jpg

1.1 *Consequences* of the definition

1. **INHUMAN STAKEHOLDERS:** Many stakeholders are non-human (laws, nature) so any ideas you have about interviewing them, or asking for their needs, and requirements will not work.

2. **ETERNAL WATCH**: You do have to analyze and observe stakeholders, continuously.

3. **SUBTLE POWER**: You do not know much about the *relative power* that a stakeholder, and a particular requirement from them, actually *has*, until you identify the stakeholder, the requirement, *then* analyze its power over your system.

4. **BUTTERFLY EFFECTS**: A distant, obscure stakeholder, with a requirement you do not know about, can be critical suddenly, unexpectedly, to your systems success.

5. **SELFISH STAKEHOLDERS**: Stakeholders cannot be expected to aware of *other* stakeholders, and *their* conflicting requirements. *We* have to manage the prioritization.

6. **NEW REQUIREMENTS**: New stakeholders, and old stakeholders with new requirements, can be discovered at any time in the system lifetime, with potentially major adaptation to them necessary.

7. **COMPLEXITY TOOLS**: You are going to need some advanced planning tools, Technoscopes [2], to capture, digest, and draw conclusions about - known stakeholders, and their requirements, for the lifetime of the system.

8. **LIMITS TO CONCERN**: The set of requirements (performance objectives, budgets, constraints) we decide to deal with, is a small subset of all potential stakeholder requirements.

9. **ARCHITECTURE DECIDES**: Our specified requirements will determine our architecture (strategies, technology) for delivering them; and then changes in cost-effectiveness of available architecture will limit the ability to satisfy *some* stakeholder requirements, as well as enable us to satisfy *additional* requirements that we could not *initially* be responsible for.

10. **STAKEHOLDER PREEMPTS**: Deeply understanding the 'real' stakeholder requirements is, by definition, more critical than the consequent requirements, design and operation of the system: stakeholders have first priority - the rest is just interpretation of them.

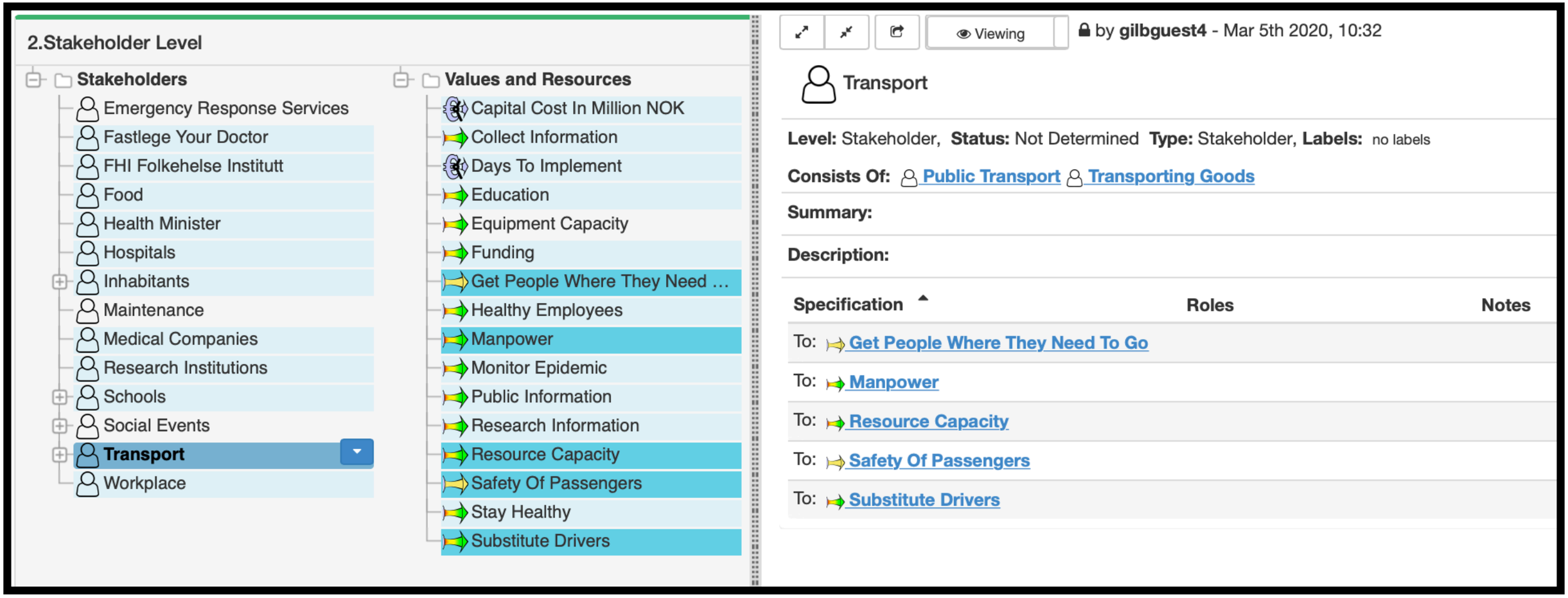


Figure 1.1 . Source Oslo Sw. Arch. OSWA, Workshop March 2020 Virus Control in Norway: Exercise. ValPlan app.
Left: a systems viewpoint. Right: The 'Transport' stakeholder viewpoint.

A simplified example of a set of stakeholders, and the relation for a single stakeholder (example 'Transport') which owns 5 of the Stakeholder Values. These 5 Values are 'needed' by both 'Transport', *and* probably also some of the other stakeholders.

One single Value Requirement for 'Transport', for example 'Safety of Passengers' can *simultaneously* be needed by any other stakeholders.

Not only that: the *level* of the Value might be different; and it might be needed *earlier* or *later*.

And all of these facts might change a week later. This was the beginning of Covid, March 2020!

This is a very simple example, reality is worse.

It gets tempting to ignore reality, except in this case the daily death rate reminded us we could not.

1.2 Some Basic Stakeholder Principles

STAKEHOLDER PRINCIPLES. (© gilb.com 2020, Governeering book)

1. Some stakeholders are *more critical* to your system than others.

2. Some stakeholder needs are *more critical* to your system than others.

3. Stakeholders are *undisciplined*: they may not know all their needs, or know them precisely, or know their value. But they can be analyzed, coached, and helped to get the best possible deal.

4. Stakeholders may be *inaccessible*, unwilling, inanimate, oppositional, and worse: but we need to deal with them intelligently.

5. Stakeholders might well ask for the *wrong thing, a 'means'* rather than their real 'ends'. But they can be guided to understand that. Or their requests can be interpreted in their own, real, best interests.

6. Stakeholders do *not want to wait years, get delays, invest shitloads of money, and then get little or no value.* They want as much 'value improvement' of their current situation, as they can get, as fast as they can get it. For as little cost as possible,

7. Stakeholders *cannot have any realistic idea of* what their needs and demands will *cost to satisfy*. So their adopted (by you) requirements need to be based on *value for costs*, not on value alone. Delivering small increments, based on high value-to-cost, is one smart way to deal with this.

8. If you think you have found 'all critical stakeholders', I think you should assume there is *at least one more*, and when you find that one, ... New Stakeholders will emerge, and they are not all identified at the beginning.

9. If you think you have found all critical needs of a stakeholder, there will *always be at least one more need*, hiding.

10. If you do not understand, and act on the principles above; you will blame your failure on 'system complexity', and the unexpected and wicked problems. But in reality it is *your own fault* and responsibility; deal with it - up front and constantly thereafter.

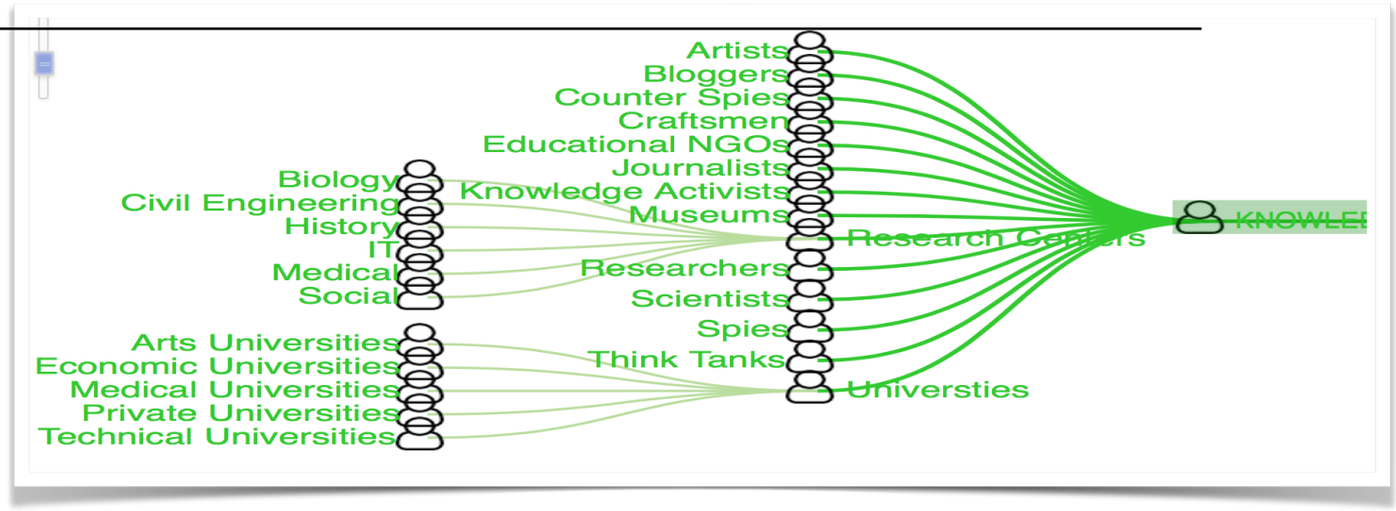


Figure 1.2 A. a hierarchical stakeholder map. (Education and Knowledge plan, Poland)

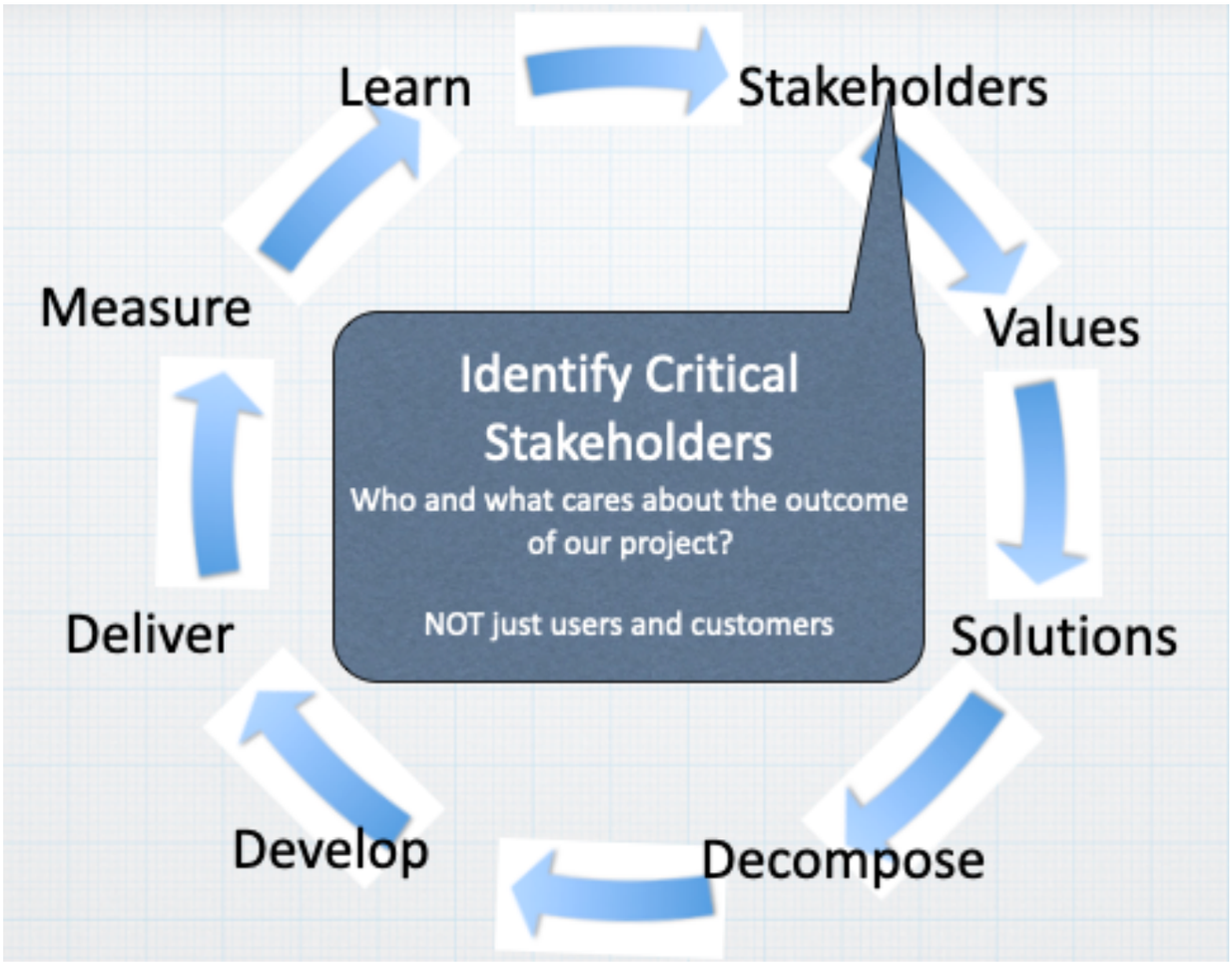


Figure 1.2 B. The Gilb Evo Cycle, where stakeholders are the logical prelude to value requirements, and Solution analysis. But there is no simple sequence. It repeats and we learn better.

1.3 Stakeholder Attributes

- **Some 'value attributes' of stakeholders**
    - **which can be defined in more detail,**
- **and can be quantified [3]**
    - **See Figure 1.3 B & C**
- **their current status level estimated**
- **and that level will *change through time***

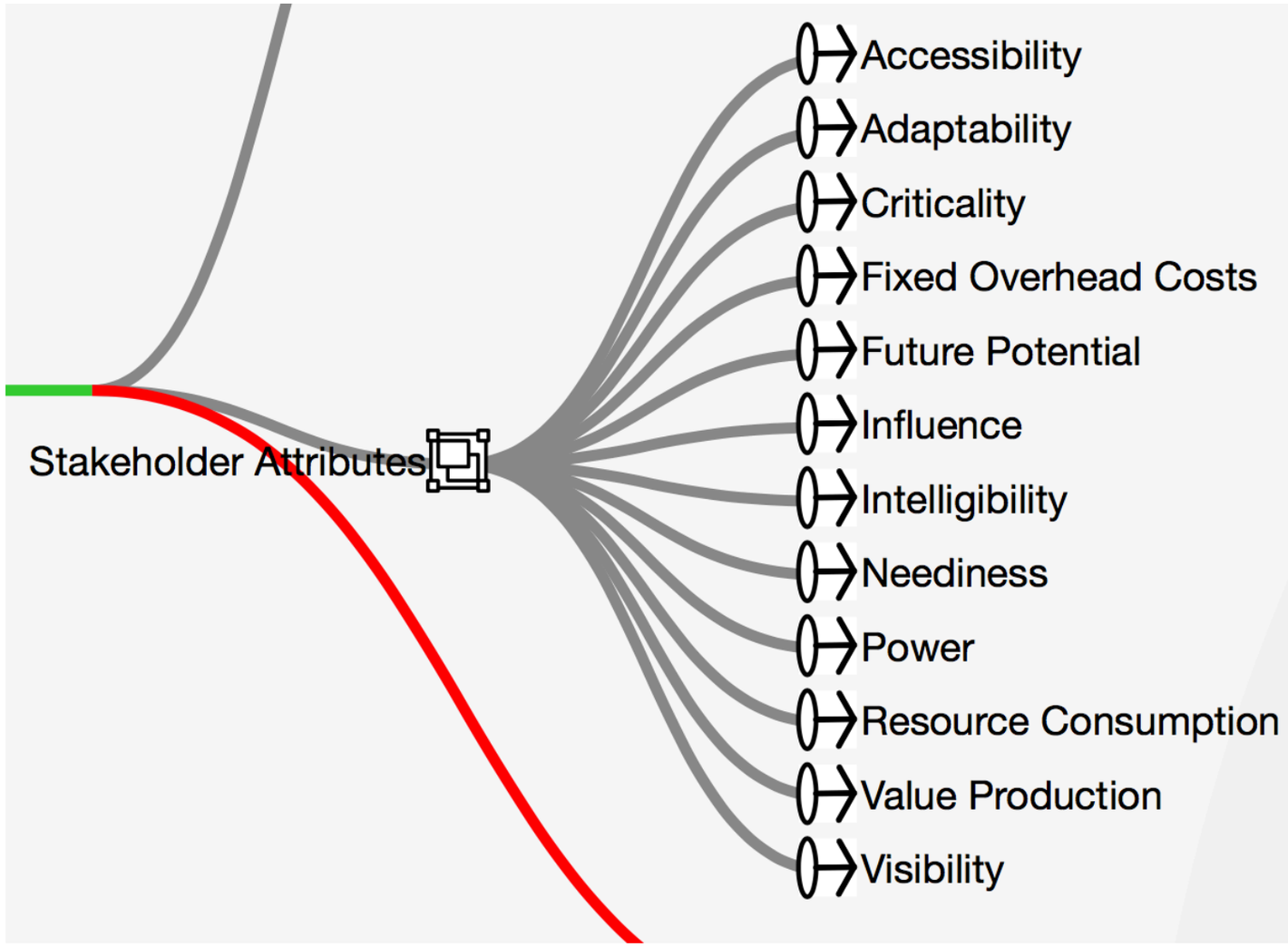


Figure 1.3 A. Some Value Attributes of Stakeholders.

1.3 Quantifying Stakeholder Attributes

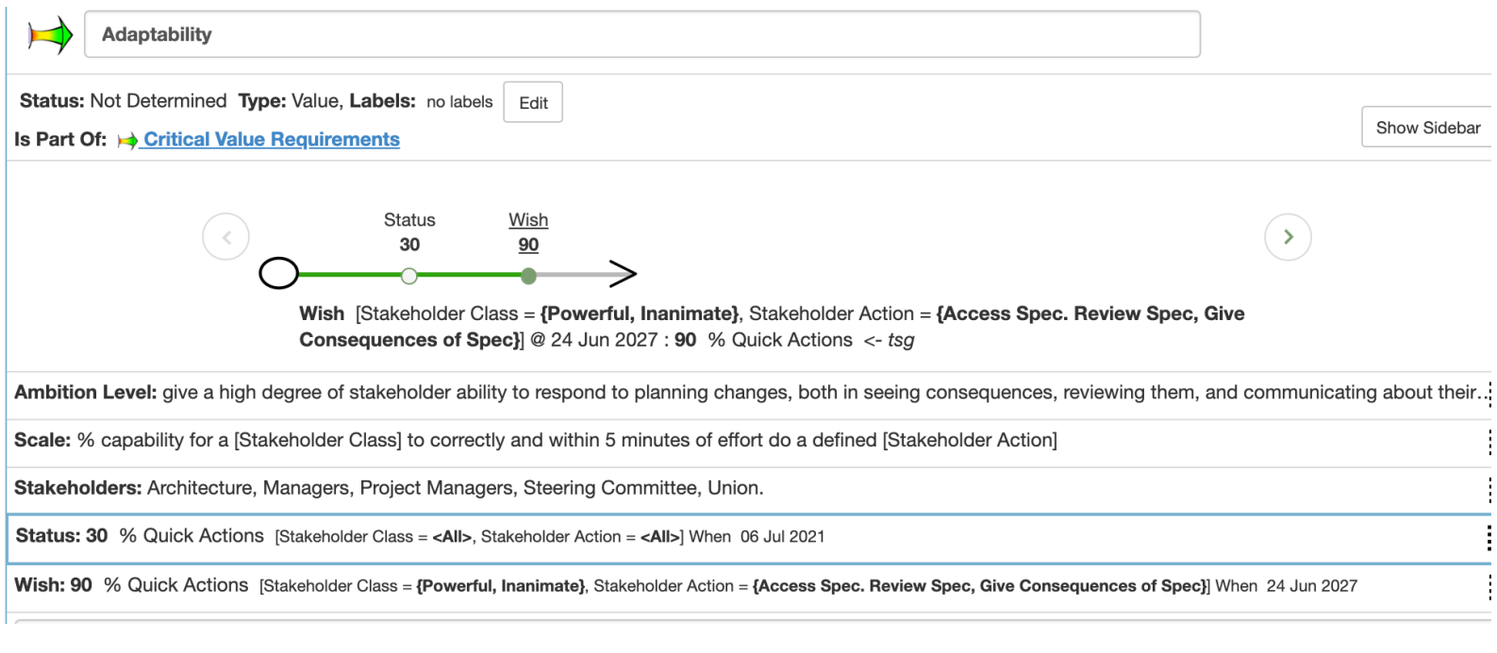


Figure 1.3 B. Here is a structured and quantified definition of Adaptability, for a requirement purpose.

These examples of defining and quantifying Stakeholder attributes should indicate that we are serious about our Stakeholder Engineering approach.

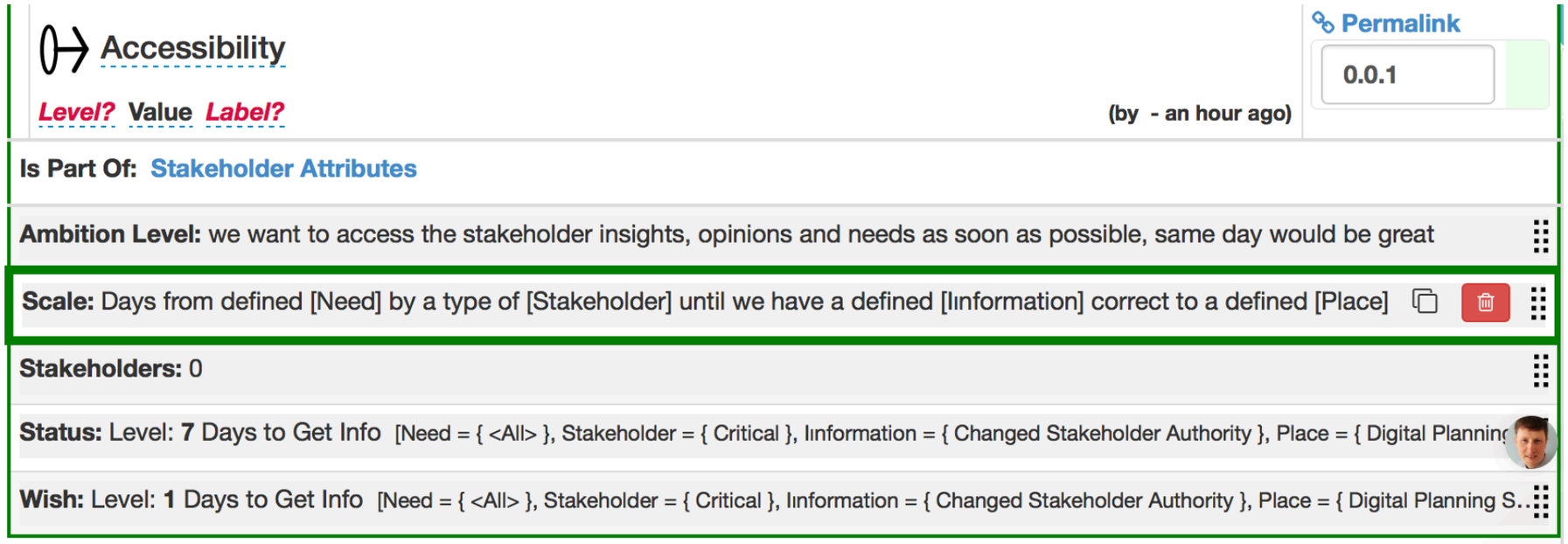
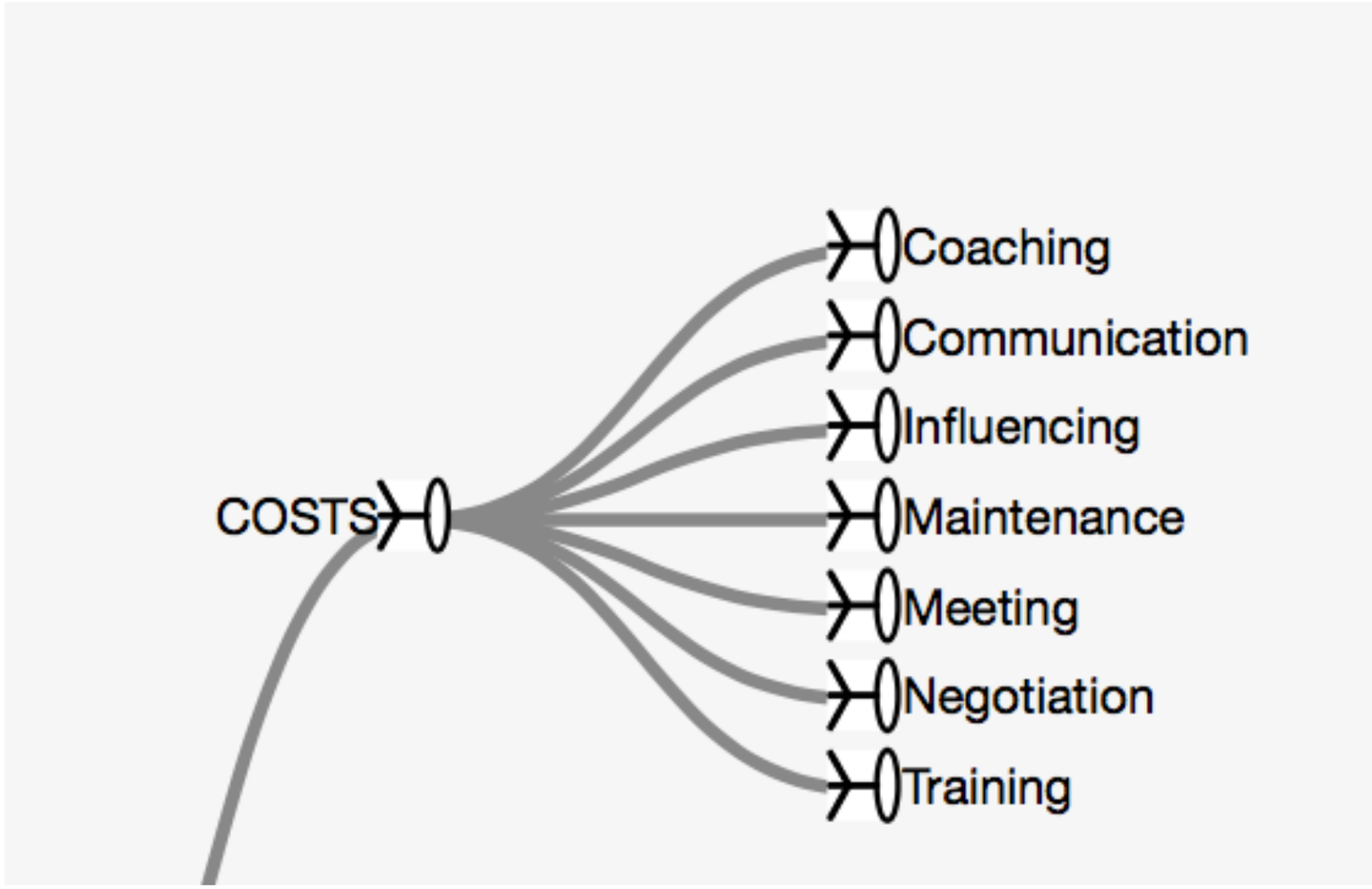


Figure 1.3 C. Here is another Stakeholder attribute defined, using the same Pattern. The key idea is a defined Scale of measure. See [3] Quanteer for detailed explanation.

1.4 Stakeholder Costs

Dealing with any stakeholder has various costs, money and time.



These costs need to be considered when we decide to analyze and engage with stakeholders

1.5 Stakeholder Engineering Principles

Dealing with Stakeholders, at a simple level is a ‘craft’. But for large and complex systems, like National systems, we believe that an engineering approach has advantages. Here are some principles to explain the ‘Stakeholder Engineering’ idea.

1. **DIGITIZATION**: All aspects of stakeholder knowledge will be digitized in a defined way (Planguage) to permit automation, and integration with other system aspects (Requirements, Design, QC, Project Management).

2. **CLEAR DEFINITION**: Stakeholders will be unambiguously defined, and be referred to with a unique Tag.

3. **TRACK STAKES HELD**: Stakeholders will be explicitly correlated with all their requirements, at any level of requirement maturity, specification, approval, and commitment.

4. **TRACK RELATED STAKEHOLDERS**: All requirements specifications, at any level of maturity, will be specified as ‘related to’ all non-trivial stakeholders for them.

5. **MULTI-QUANTIFICATION:** Stakeholder analysis will be based on all important dimensions of values and costs, and these will be quantified, and enriched with Background specifications..

6. **PRIORITIZATION:** priority of value-delivery actions will be based on current real-time calculation of 'overall values deliverable’ in relation to 'overall and long term costs'.

7. **FAIR BALANCE:** in the competition for resources we shall aim for a reasonable balance in stakeholder satisfaction, based on values for costs, good investments, and potential alternative uses of resources outside our system of concern.

8. **VALUE-DRIVEN LOGIC**: Decisions of all types will be made on the basis of quantified value Requirements, on approved processes and Rules, and on transparent decision processes.

9. **INCREMENTAL FEEDBACK:** system delivery will be based on an early continuous stream of attempted value deliveries, 2% of horizon budgets, with measurement and feedback the basis for validating the increments, or rejecting them, or modifying them.

10. **STAKEHOLDER ANALYSIS:** will be continuous, and based on value-for-resources.

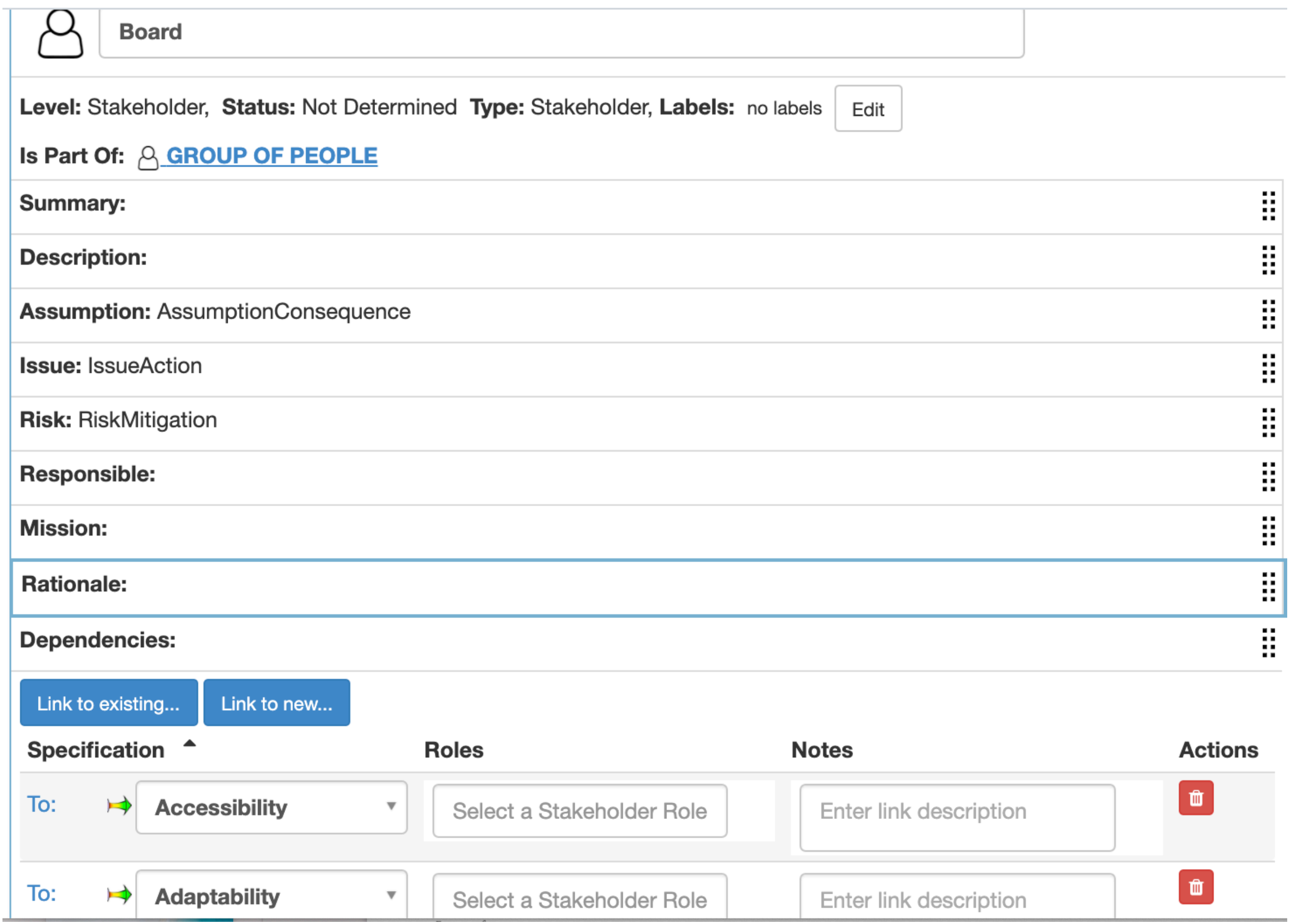


Figure 1.6 A. A Template Pattern for defining a stakeholder.

Stakeholder Specification Patterns from Planguage [4]

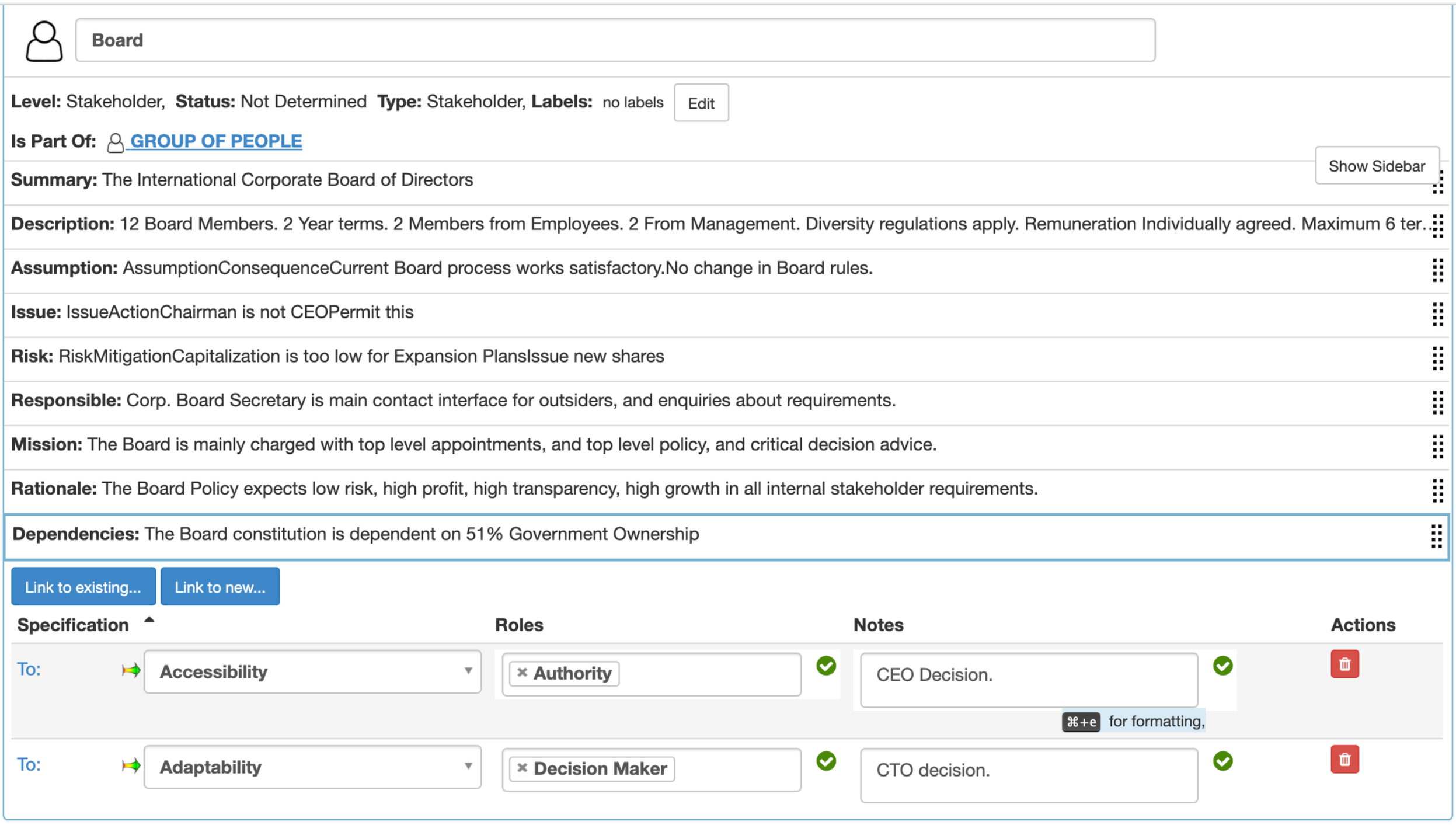


Figure 1.6 B. Example of filling out the parameters about the stakeholder (made up)

By using this structure, with tagged specification parameters (like 'Assumption') we are achieving several purposes. We can add apps (like ValPlan, or our own logic) which can analyze stakeholders, using well-defined parameters. This example is a one-liner format, and all of the parameters have more detail included and available, such as *who* and *when* the specification was made, and the *source*. Systematic unique Tags (Board, Accessibility) help us relate (digitally and manually) to the larger system of planning (Systems Engineering).

1.7 Stakeholder Attributes. Cost aspects, Effectiveness Aspects, and our strategies for handling stakeholders. These are 'arbitrary but useful' examples, and patterns. Tailor your own version.

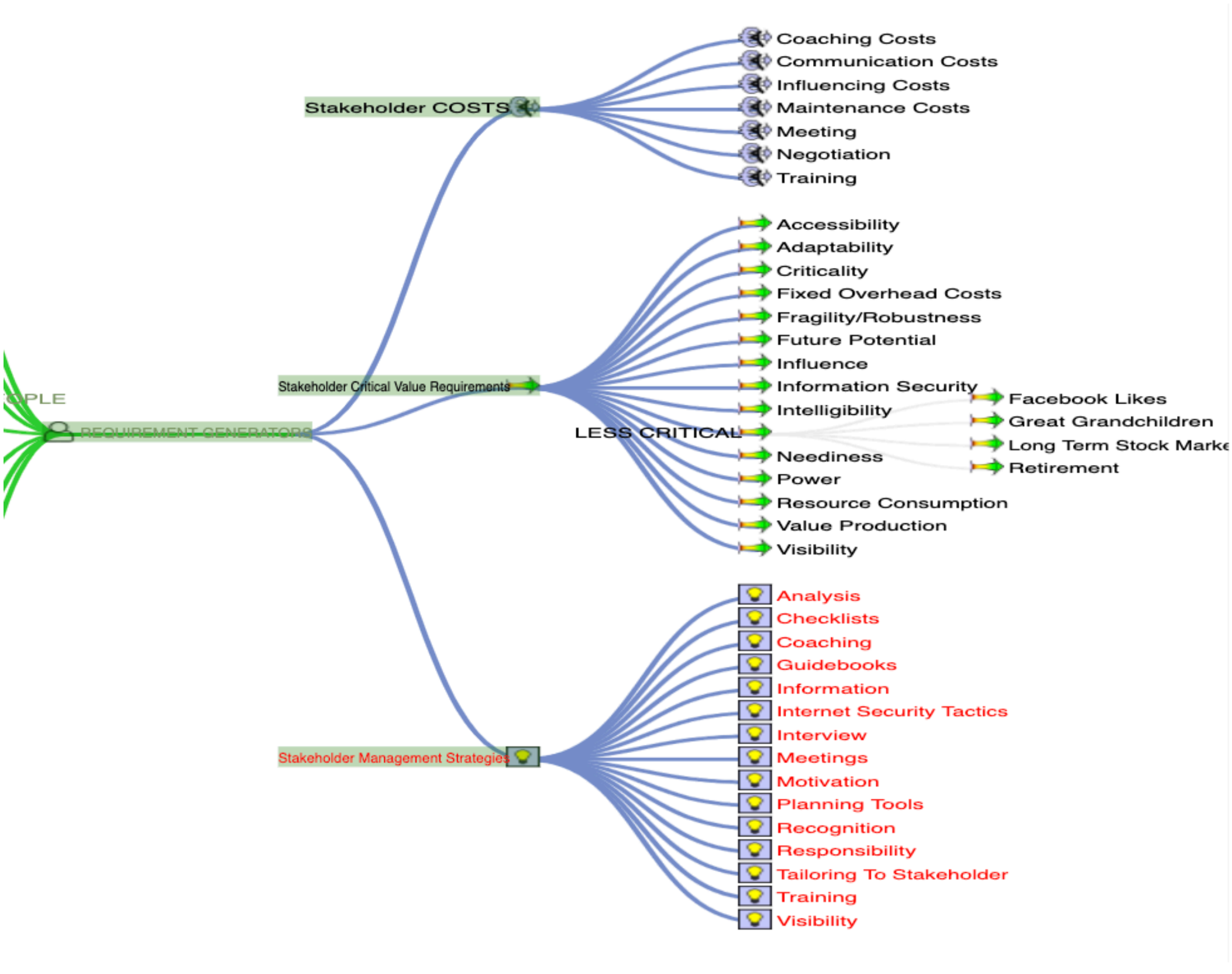
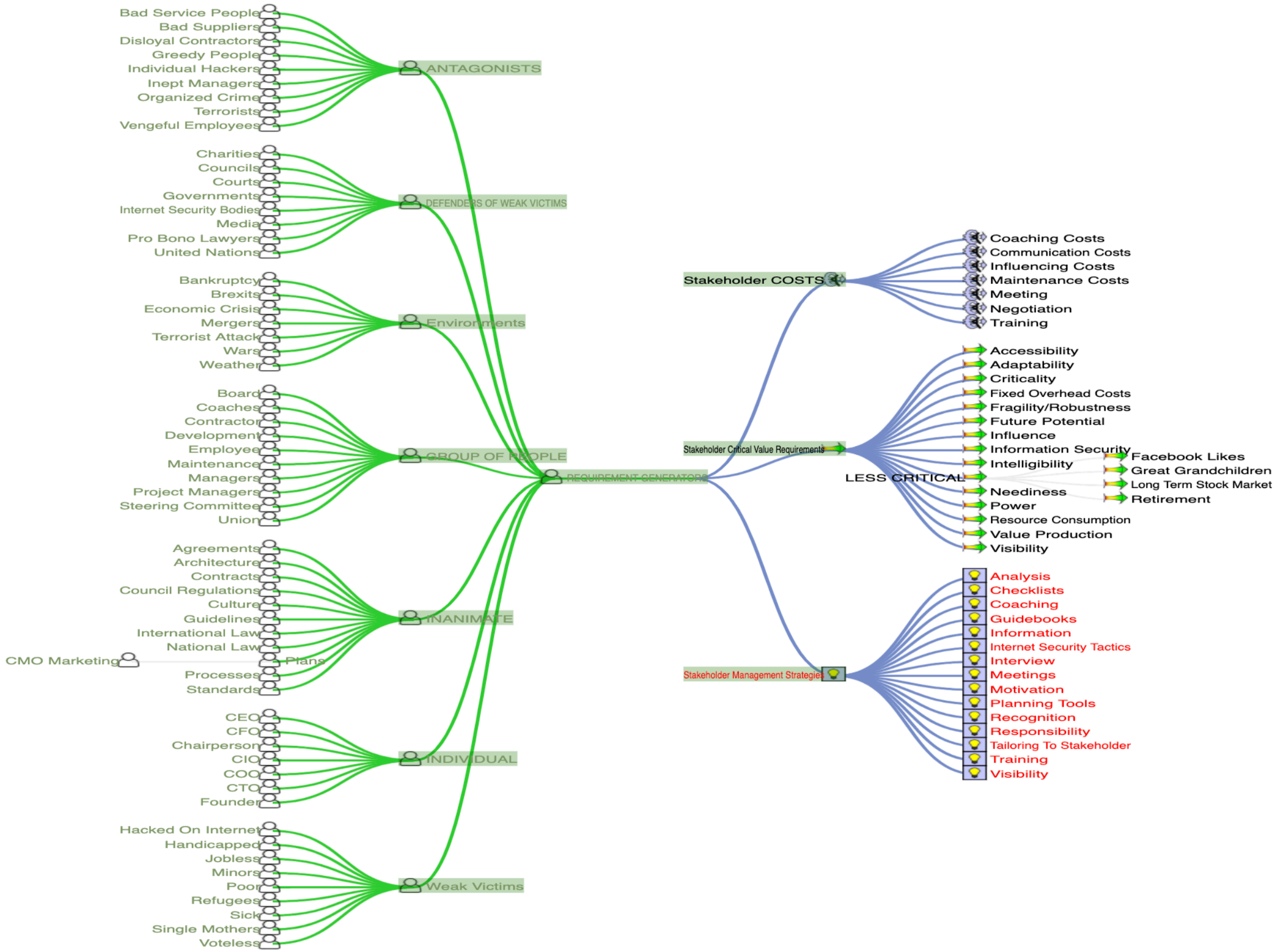


Figure 1.7 A. The strategies, are examples of investments and processes we can potentially make, so as to improve the Attributes of the stakeholders for us, for example to make them 'More Visible', by 'Meetings'

1.7 A generic hierarchical stakeholder pattern, with detailed examples of categories.



Figured 1.7 B This is a top-level example of an overview of a useful set of interesting stakeholders, together with our objectives in managing them (arrows), our potential strategies, for better stakeholder management (lightbulbs), and the associated cost aspects of managing stakeholders.

1.7 A 'Risk Management Perspective' of some different 'classes' of stakeholders

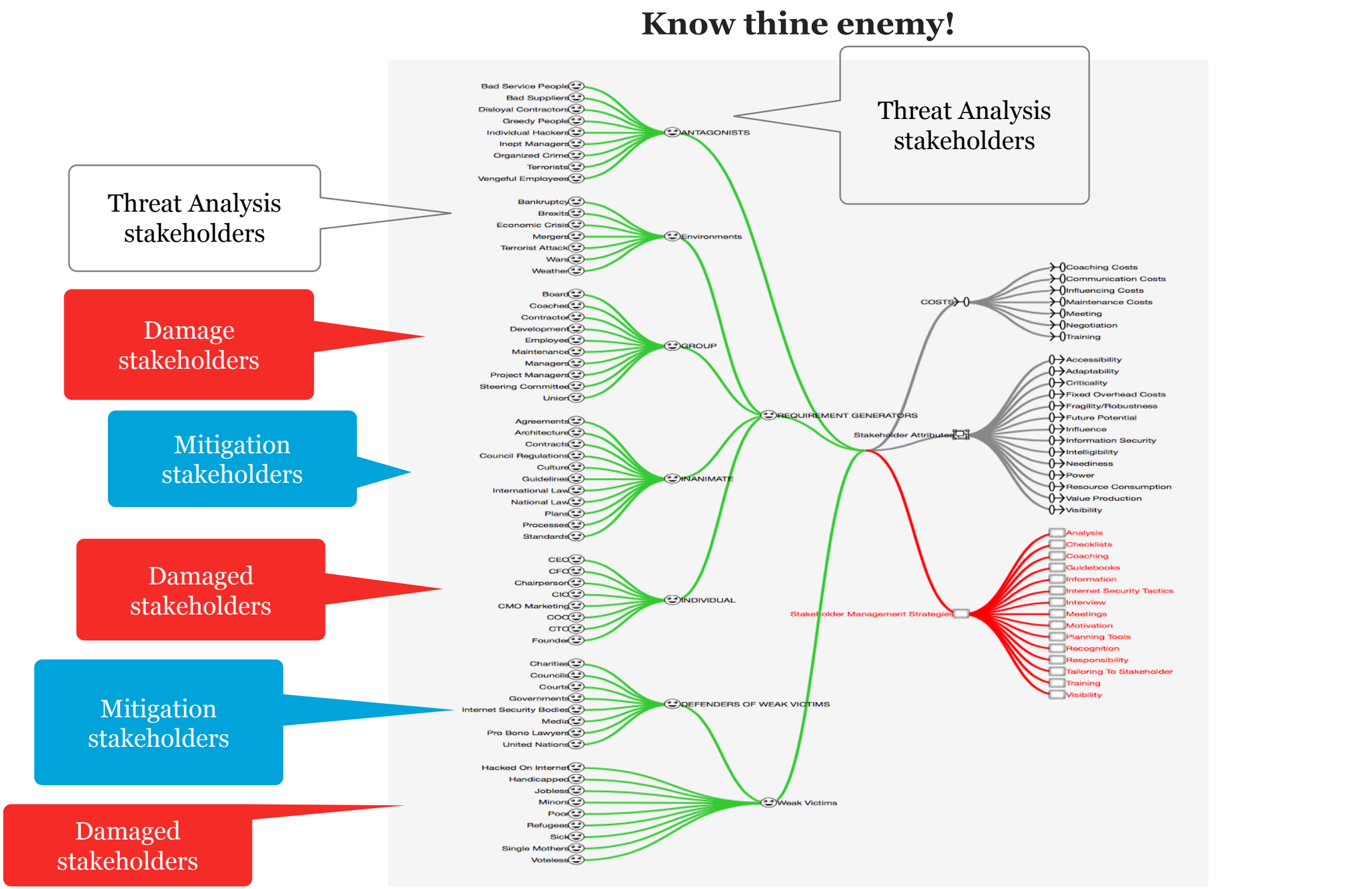
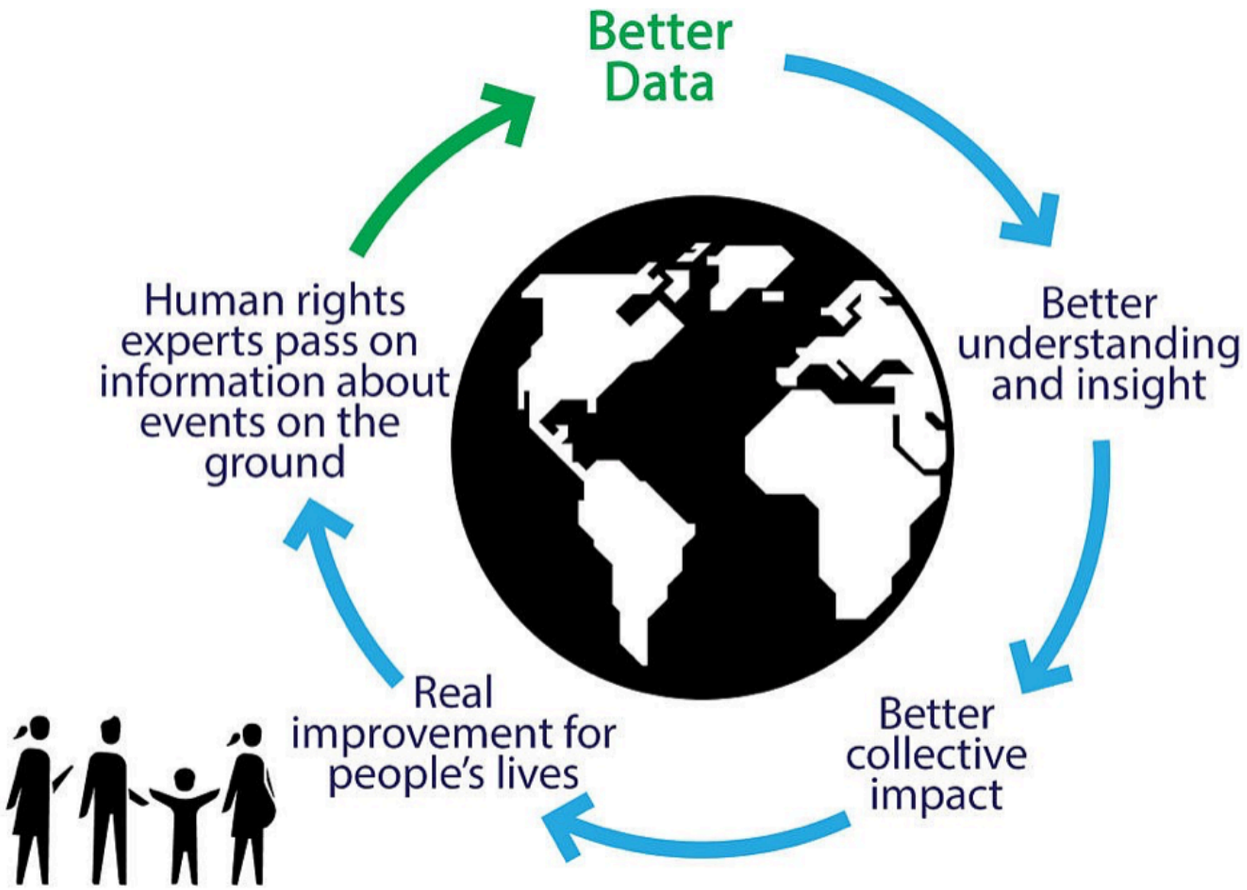


Figure 1.7 C. Not all stakeholders are consumers, customers or nice people.

1.8 Stakeholder Rights

Stakeholders should have the

- Right to have a voice
- Right to be consulted
- Right to be warned
- Right to suggest
- Right to review
- Right to measure
- Right to complain
- Right to be informed
- Right to change their mind
- Right to understand costs
- Right to understand value/resources
- Right to understand risks
- Right to set their priorities



https://humanrightsmeasurement.org/methodology/measuring-civil-political-rights/

1.9 Stakeholder Ethics

- **Stakeholders will have highly varied ethics, and motivations**
- **We can influence stakeholder ethics by a variety of actions**

https://www.chuckgallagher.com/2013/04/16/business-ethics-theories-which-theory-of-ethics-do-you-follow-stockholder-stakeholder-and-social-contract-theories-part-one/

The Logic of Design:
Design Process Principles.
Tom Gilb, 2016, Paper.
http://www.gilb.com/dl857

1.10
The Basic Design Steps Logic:
A summary.
Notice the emergence of the Stakeholder concerns.

1. **Constraints determine environments.**
2. **Environments determine stakeholders**
3. **Stakeholders have values and priorities**
4. **Values have many dimensions**
5. **Stakeholders determine value levels**
6. **Design hypotheses should be powerful and efficient ideas, for satisfying stakeholder needs**
7. **Design hypotheses can be evaluated quantitatively, with respect to all quantified objectives and resources**
8. **Designs can be decomposed, to find more efficient design subsets, that can be implemented early**
9. **Designs can be implemented sequentially, and their value-delivery, and resource costs, measured**
10. **Designs that unexpectedly threaten achievement of objectives, or excessive use of resources, can be removed or modified.**
11. **Designs that have the best set of effects on objectives, for the least consumption of limited resources, should generally be selected for early implementation.**
12. **A design increment can have unacceptable results, in combination with previous increments, and they, or it, might need removal or modification**
13. **When all stakeholder objectives are reached, the process of design is complete: except for possible optimization of operational resources, by even-better design.**
14. **When deadlined and budgeted implementation-resources are used up, it might be reasonable to negotiate additional resources with stakeholders; especially if the incremental values are worth the additional effort.**

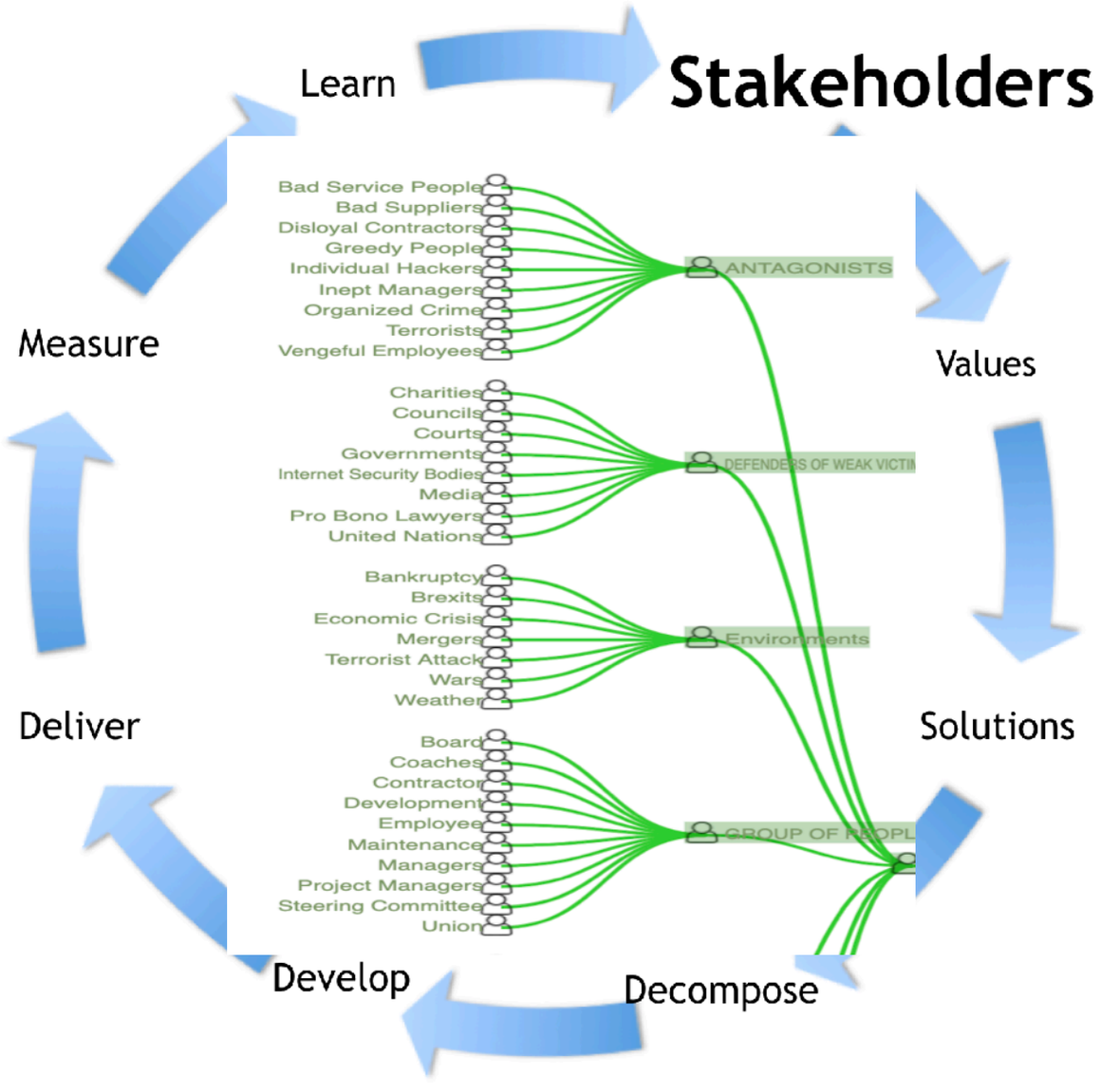


Figure 1.10 The Gilb Evo Cycle. An 'agile' development or improvement process.

2.0 Identification of Stakeholders

I would appreciate hearing your reactions, suggestions, difficulties understanding me, objections, references and links. After each chapter!
tom@Gilb.com

PS
Feel free to send to friends, and post link on social media, or to use in whole or part in training, lectures, blogs, your own books, papers, slides, *translations* with main link
https://tinyurl.com/StakeholderBook, and if you like:

2.0 Identification of Stakeholders

An eternal, continuous discovery process

We always have a current system, we always have a certain body of knowledge about our stakeholders. I believe we need to build and improve systems from this base, in increments.

The stakeholder value-delivery increment is there, partly to deliver some real value, in the short term; but it is also there to give us the *opportunity to learn from reality*. Some of those realities is that *new stakeholders emerge*, and need to be systematically analyzed, and integrated into your overall planning. Another reality is that new or changed *stakeholder needs emerge,* and need to be analyzed, and if appropriate, prioritized, and brought into the overall planning.

The main point here, is that 'stakeholder identification' is not an 'up front' process, not a one-off process. It is a continuous, complete 'life-cycle of system' process.

Failure to keep your organization discovering new stakeholders and new stakeholder needs, is a path to failure of your organization or system.

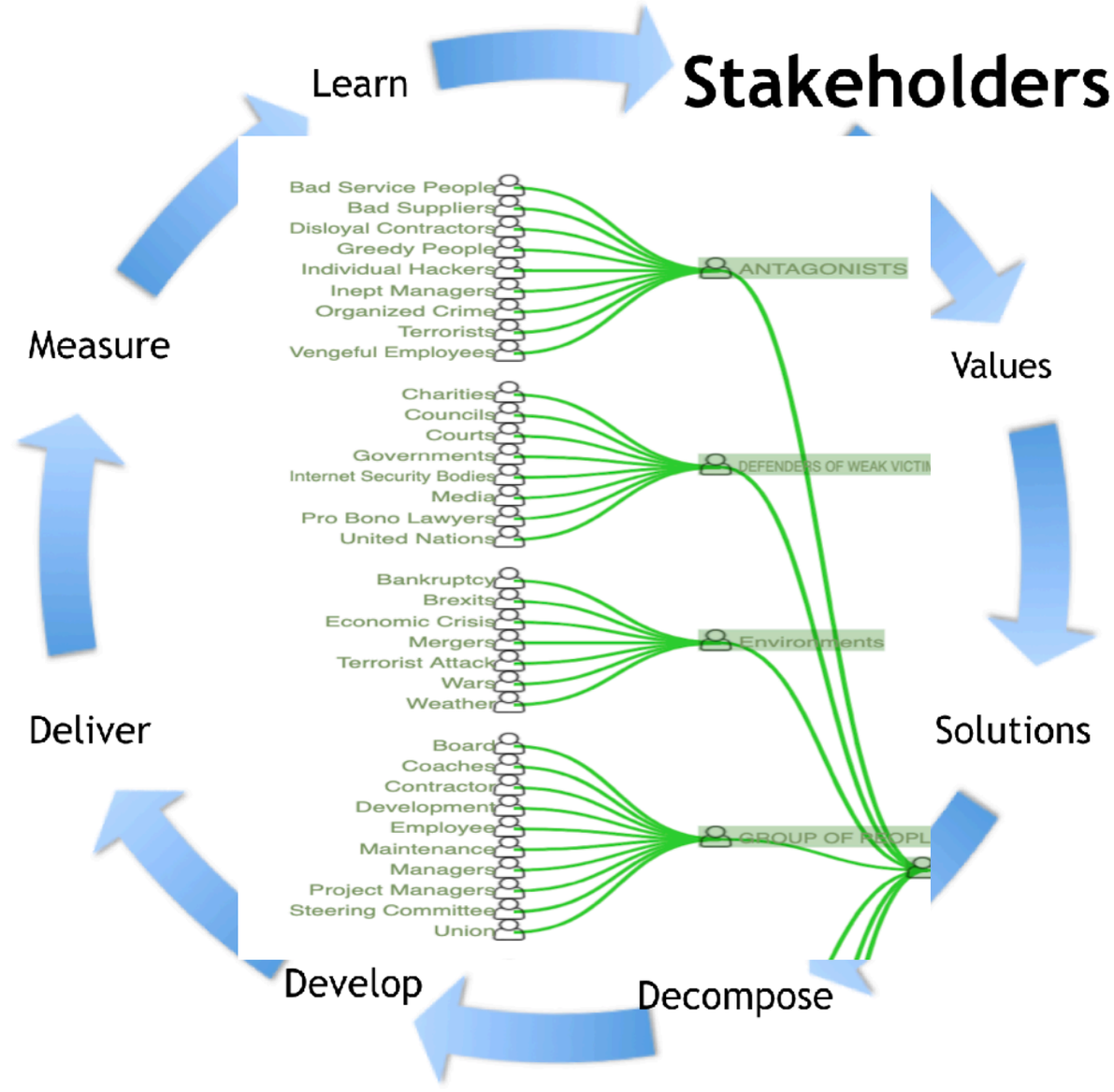


Figure 2.0 The Gilb Evo Cycle. An 'agile' development or improvement process.

Building knowledge about your stakeholder's identity, and their values is a long-term iterative maturity process, with no end.

- **Follow the money!** Whoever is paying is definitely a stakeholder. Also, if the program produces savings or additional costs for an organization, then the organization is also a stakeholder.

- **Follow the resources.** Every entity that provides resources, whether internal or external, labor or facilities, and equipment, is a stakeholder. Line managers and functional managers providing resources are stakeholders.

- **Follow the deliverables.** Whoever is the recipient of the product or service the program is providing is a stakeholder.

- **Follow the signatures.** The individual who signs off on completion of the final product or service (or completed phases of the product or service) is a stakeholder. Note: This may or may not be the recipient referred to in the previous bullet. Often there may be more recipients than signatories.

- **Examine other programs' stakeholder lists.** Include active programs and completed projects.

- **Review the organizational chart** to assess which parts of the organization may be stakeholders.

- **Ask** team members, customers, and any other confirmed stakeholder to help you identify additional stakeholders.

- **Look for the "Unofficial People of Influence**. These may be people who are trusted by high-level leaders or who wield a lot of power through influence and not position.

The goal of following these guidelines is to make sure every possible stakeholder is identified. Some of your stakeholders may play major roles, while others may have minor roles and little or no interest or interaction. Regardless of size or role, every stakeholder's needs must be assessed, and you cannot meet the needs of a stakeholder you have not identified.

https://www.sebasolutions.com/dev/newsletter/?id=104

http://web.archive.org/web/20160704230729/http://www.sebasolutions.com/dev/newsletter/?id=104

He does not suggest *directly*, weak stakeholders, antagonists, and non human stakeholders. So, by my stakeholder definition, this method will not identify 'every possible stakeholder'. But I think he gives pretty good advice on some practical ways to identify stakeholders

Here are some ideas of how to discover stakeholders.

2.1 Principles of Stakeholder Discovery

1. **PROBLEM TRACING:** Follow a ***problem***, of any kind, *back* to a stakeholder.

2. **MEDIA TRACE:** Follow *press and social media* back to stakeholder types

3. **ROOTS**: Analyze *root-cause* of problems, and discover 'root stakeholders'.

4. **HIERARCHIES:** Analyze the **stakeholders of 'your stakeholders'** - they influence each other

5. **PATTERNS:** Make use of ***stakeholder patterns lists***, in this book, in your other projects, on the internet. Search for 'Stakeholder lists', and find hits like https://www.stakeholdermap.com/stakeholder-list.html

7. **REQUIREMENT SOURCES:** Look at any requirement including any constraint, and trace it to its ***source***, and to its ***'sources source'.***

8. **DOCUMENT SOURCES:** When specifying requirements, make it a rigorous practice to ***capture the source*** *in writing*. They identify specific, and general, types of stakeholders.

9. **POTENTIAL STAKEHOLDERS:** Think **outside your box:** imagine future and potential customers, markets, applications, competitors, partners. Think of them as *Potential* Stakeholders: especially if you deliver certain requirements 'successfully enough'.

10. **COALFACE**: Ask your customer-facing, user service, and marketing people about missing **product and service quality** requirements, and *their* stakeholders.

11. **DOMAIN**: Study problems, and *their* stakeholders; from *other* projects in your organization: this is inside *your* domain, learn from it.

2.2 Specific Methods for Stakeholder Discovery

1. **Templates**: https://www.stakeholdermap.com/

2. **TIE TO REQUIREMENTS:** Make sure *every* specified requirement, has one or more stakeholders attached to it.

3. **USER STORY**: Use the User Story ‘as a <stakeholder>’ component, to identify stakeholders.

4. **USE CASES:** Use Cases identify stakeholders.

5. **COUNTER CASE:** Use the ***opposites*** of already-identified stakeholders: non-customers, lost-customers, previous users, negative reviews, silent majority, competitor customers,

6. **INANIMATE:** Think of *inanimate* stakeholders, requiring you to do things or avoid things, like *laws, rules, policies, plans, architecture, agreements, contracts*.

7. **ENVIRONMENTS:** Think of ‘abnormal environments’ as stakeholders: like *natural disasters, wars, out of office, on travel, on personal mobile, system hacked, system unavailable, Brexit, No majority government, damaging false PR, death of key people*.

8. **DECOMPOSE:** Decompose *existing* stakeholders into sub-categories. See next page 2.2 Simmons’ advice.

9. **CAPTURE THE STREAM:** Make sure you have a systematic *process*, and responsible *trained* people, to capture new stakeholder ideas, *as they occur* in daily work, like meetings, and requirements specifications or design, or planning work. Cumulate in digital databases, integrated with organizational planning.

10. **MEDIA ANALYSIS:** Use media reports in your Domain and specifically analyze for Stakeholder content.

Decompose stakeholder categories for more insight.

2.2 Go deeper into broad categories: Erik's advice.

> "Among my emphases with this group has been to characterize their stakeholders as part of the definition process.

Drawing on requirements engineering, the words we use to define stakeholders tend to be collective nouns.

This amounts to classification on common attributes (users, suppliers, sales & marketing...).

While there is some value in that classification, I've been emphasizing the **need to characterize and go beneath the broad collective noun to look at crucial differences among those broad stakeholder groups.**

In my experience, the **most valuable** elements of system delivery come from the **less-obvious aspects** of stakeholder sub-groups.

So decomposing stakeholder names and characterizing those groups as to their **differences** can be far more effective than just focusing on the abstract group.George Lackoff, in

Erik Simmons, at Swiss Conference, 2011

Metaphors We Live By, provides an anecdote using birds: To call a penguin, a robin, and an ostrich birds is not incorrect, but this classification is a reduction to lowest common denominator. If we instead characterize these individuals, we find one swims, one runs, and one flies. So while there is value in naming them all as birds, that *classification overlooks some of the most important aspects of the individual members of the class.*

BTW, I have asked this group to explicitly define each stakeholder's stake as part of this characterization. I'm seeing a lot of clarity and value come from the discussions of exactly what stake these entities have in the system, and from that stake derive many tacit stakeholder values."

Erik Simmons, Construx.com, Email 7 July 2021

2.2 *Planguage* Methods for Stakeholder Discovery.

1. **[Scale Qualifiers]:** Decompose existing stakeholders into sub-categories [Scale qualifier] combinations:

Scale: % [Customers] needing [Training] to [Skill Levels] by [Training Providers].

2. **Parameter Variety:** Clearly separate different stakeholder levels, conditions, and deadlines for the same generic Value (like 'Training')

***Goal** by 2027, 42%, Customers=New, Training=Advanced, Skill Levels=Expert, Training Providers=Third Party. Source=Stakeholder 1*

***Tolerable** by 2033, 64%, Customers=Old, Training=Simple remote, Skill Levels=Hire-able, Training Providers=Foreign Remote. Source=Stakeholder 2*

3. **Source Fanaticism**: for almost every parameter of specification in Planguage, there are dozens, insist on naming a Source for it. If you cannot, then why is it in the specification at all? If you can then you are at or near a stakeholder.

R13: **Source:** Specification statements shall contain information about their source of origin. Use the '<-' icon and state the source person and the date, or the source document with detailed statement reference.

Figure 2.2A, Source: Competitive Engineering [4], page 17

1.4

Rules: Generic Rules for Technical and Management Specification

Learning Time:
Scale: Mean Time to Learn defined [Task] by defined [Operator].

Fail [Outgoing Call, Beginner]: 3 minutes <- Marketing Requirement 3.4.5.
Risk: If the Mean Time is not lower, then Competitor Products will be perceived as better and we will lose <market share> <- Marketing Planner [Andersen].

Fail [Address List Update, Professional User]: 30 seconds <- Marketing Requirement 3.4.6.
Authority: External Consultants. "Outside consultants tell us we will be rated badly if we fail to beat this level."

Goal [Average Task, Average User]: 25 seconds <- Marketing Requirement 3.4.7.
Rationale: Marketing believes this will make us best in the Market.

Figure 2.2B. The Planguage keyed icon '<-' means 'Source:'. The 3 sources here identify stakeholders. In addition the Background specification parameters 'Rationale', 'Authority' and 'Risk' all give Stakeholder insights, not just 'identification', but 'Why?'.

2.2 Specific methods for stakeholder discovery. The Value Stream.

Based on Ralph Keeney [9].

At 'your personal level of planning', you have a set of downstream stakeholders, maybe several levels of them, which *you are going plan to deliver value to.* Let's call this '*your boss and up'.* They have a set of objectives, for you, which Keeney calls **'Fundamental Objectives'**. A very critical territory for you to search, for all kinds of related stakeholders.

The **'Strategic Objectives'** is Keeney's name for plans ***you*** are making, and are responsible for. This might be for 'your organization' and for many stakeholders in *your* organization, at many levels. A good place to search for stakeholders.

Finally, you may be delegating, outsourcing, all kinds of detailed planning, for products and services. These are, relative to you, sets of '**Means Objectives**' (the *means* to *your* 'ends'). If you analyze and explore these 'suppliers' you will find many stakeholders, too.

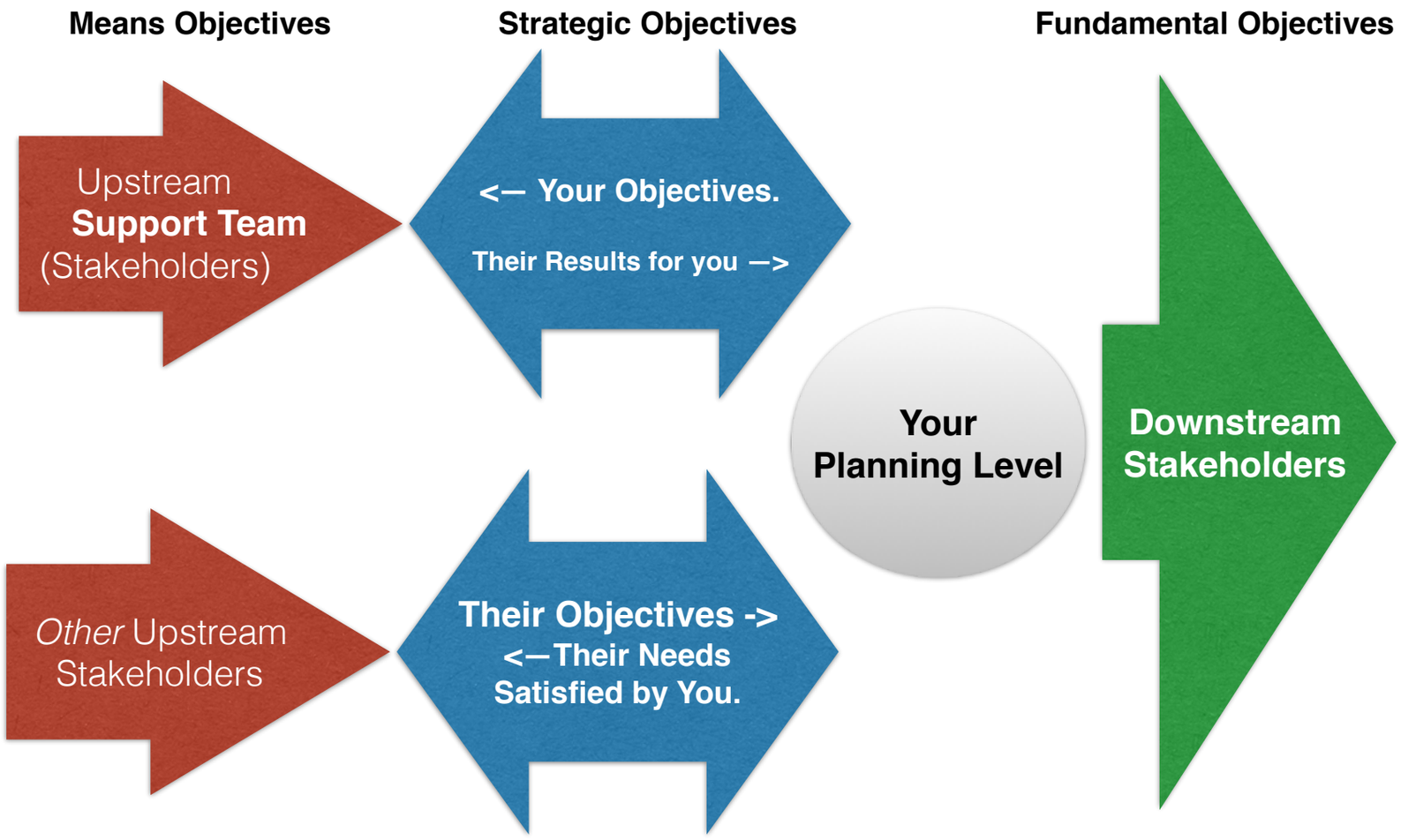


Figure 2.2 Sharing your objectives with your support team, so they can know what kind of support they need to align to. Upstream stakeholders are not your primary result delivery stakeholders. But they can require certain qualities and values from you anyway. For example they might require, payment, crediting, or quality in communication. <- Value Planning book 3.8, 2015 [7]

Several management planners I have exposed these ideas of Keeney to, have had an 'Ah-Ha' moment. They realized they had problems due to collapsing these levels of concern, to a single level, rather than clearly separating, the 3 levels of responsibility.

3.

Stakeholder Specification

I would appreciate hearing your reactions, suggestions, difficulties understanding me, objections, references and links. After each chapter!
tom@Gilb.com

PS
Feel free to send to friends, and post link on social media, or to use in whole or part in training, lectures, blogs, your own books, papers, slides with main link https://tinyurl.com/StakeholderBook

3.0 Basic Rules of Stakeholder Specification

1. **Tag:** Each stakeholder will have a unique Tag, possibly with synonym tags (UK: United Kingdom)

2. **Hierarchy:** Stakeholder Tags can be a systematic hierarchy UK={England, Wales, Scotland, Northern Ireland}. UK.Wales.

3. **Fit for Purpose:** The detailed description of a stakeholder should be suitable for purposes. More-critical stakeholders having more detail about them.

4. **Master Spec:** The Stakeholder description should ideally be one single description for many purposes. Not a new one, from scratch, for every project. But, if there is one stable global description, we can add special information related to our project, or a smaller domain. We are looking for re-use, and common updating. This might be achieved by pointing to their URLs for a Website, and Wikipedia, as a basis.

5. **Template:** There is a fairly stable template or pattern in Planguage for Stakeholders (Fig. 3.1), but you can tailor, from that Pattern for your local purposes.

6. **Relationships:** The information about stakeholder relations to Objectives, to other requirements, and to architecture strategies, can be achieved by pointers, by using their (Requirement, Design) Tags. If you have tool support, like ValPlan, the links will be digital and fairly automatic. But some links or relations might need to be specified explicitly and manually.

7. **Experience Data:** Any information, especially history and cases which can help us discover stakeholder requirements, needs to be included. Even speculation, and rumor: rather too much than too little.

8. **Risk:** A thorough, and continuously updated record of all risk elements (Issues, Risks, Assumptions, Dependencies) should be included, even when no resolution is yet available.

9. **Responsibles:** A thorough specification of who or what is responsible (position, current person) for various things should be included.

3.1 Stakeholder Specification Principles.

1. **Efficiency**: The investment in specification and updating a stakeholder should hold out a reasonable promise of rewards for selecting correct and complete requirements, and delivering successful systems value.

2. **Corporate Memory**: Many stakeholder specifications should be built, based on our previous experience with them, and kept up-to-date, for the long term. This is a form of Corporate Memory.

3. **Size Doesn't Matter:** The total volume of a digital stakeholder record, does not matter, as much as getting access to facts and insights. The Stakeholder Specification should not be optimized and summarized. We can easily hide detail, when we do not need it, and retrieve it when we do.

4. **Requirements is the Purpose:** Keep in mind that the primary purpose of *stakeholder* specification is so that we can glean accurate, updated, and complete insights *for our consideration* as requirements, for some current effort (project, improving a system, transformation). So everything that can give us even a hint to follow up currently, might easily be worthwhile.

5. **Sub-Stakeholders for Insight:** When a stakeholder specification is getting too complex and confusing, it might be time to decompose it into more-specialized sub-stakeholder types.

6. **Owner:** the stakeholder information needs to be responsibly cared for, and maintained. So at least one specification 'Owner', needs to be specified, like 'Marketing', or 'Customer Service'. Some group with lots of experience and interest in that class of stakeholder.

7. **Confidentiality:** some parts of the Stakeholder Spec might need to be confidential, and to be used only with permission, for example from an Owner.

8. **Reject Reason:** when a stakeholder is not seriously considered, or one of their needs is not taken into our committed project, the explicit reason why, and people involved, should be documented explicitly in the stakeholder specification.

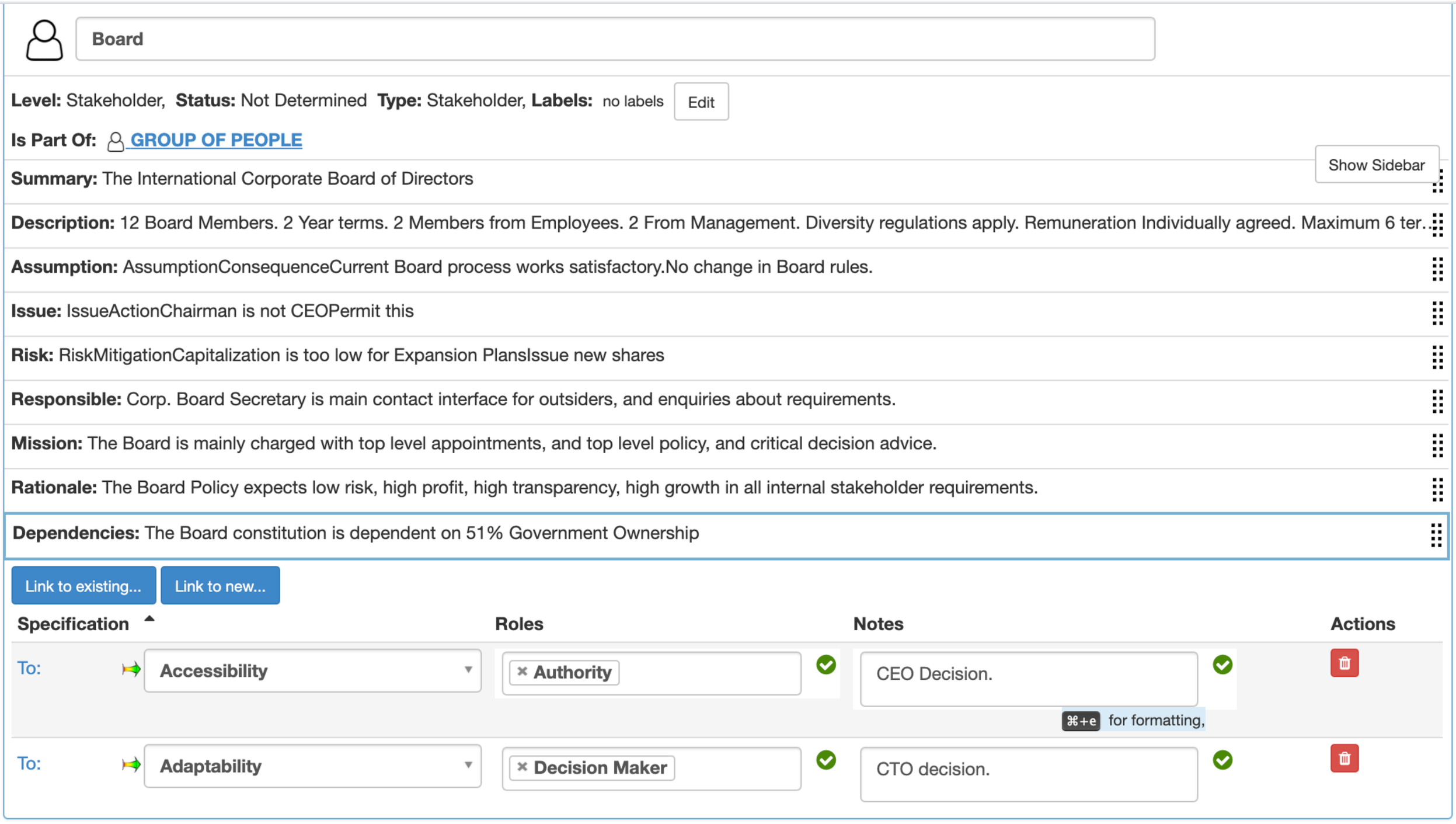


Figure 3.1 Example of filling out the parameters about the stakeholder (made up). A stakeholder template pattern.

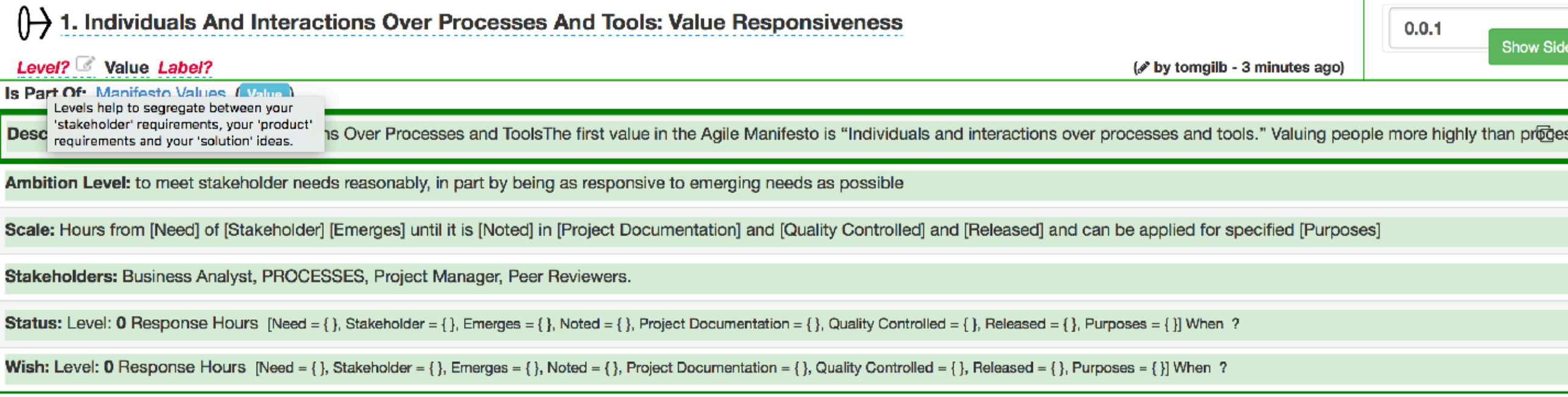
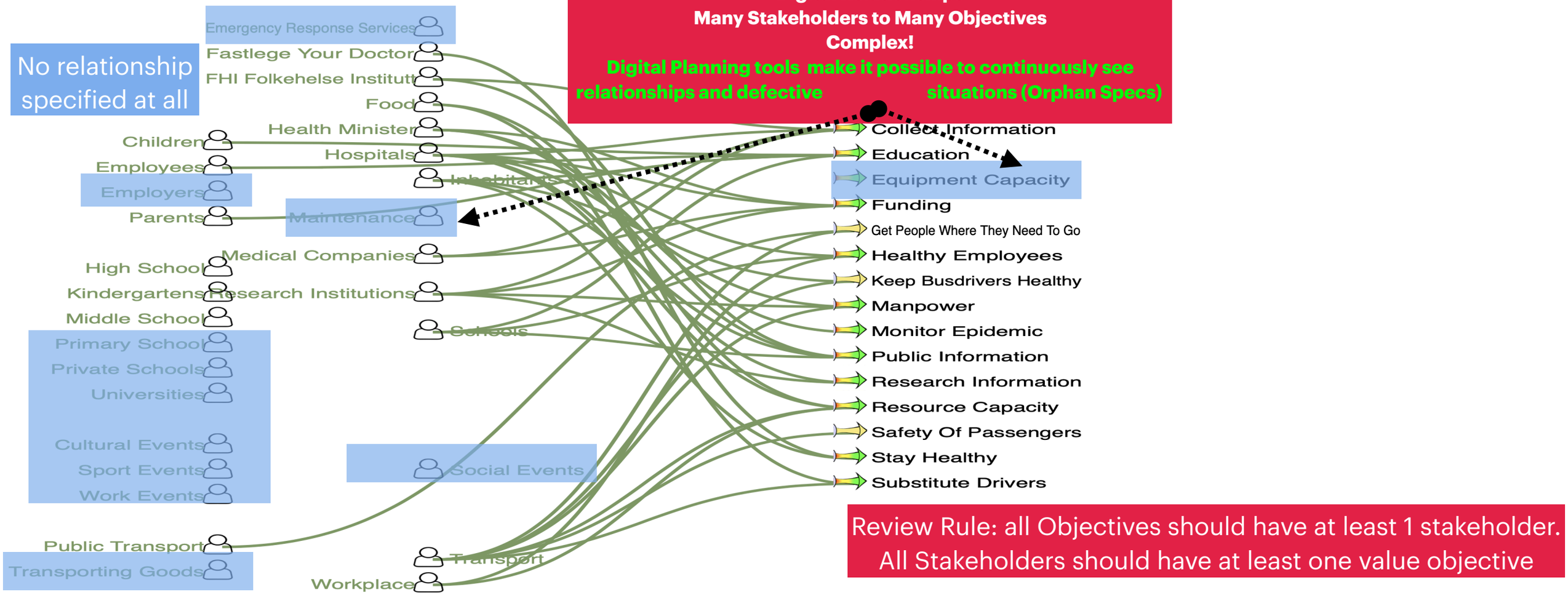


Figure 3.2 A simple example of a Value Requirement spec, which names related Stakeholders, with their Tags. Giving us digital access and updating, of stakeholder information.

In this example of specification of a Value objective, four Stakeholders are specified, as having an interesting degree of interest in this Value Objective. The ValPlan app will automatically, digitally, attach this fact, to each of the 4 the stakeholder specifications. As with the example in Figure 3.1 above where the stakeholder record has 'Accessibility' and 'Adaptability' attached at the bottom of a Stakeholder specification. This is dynamically and automatically updated as changes occur, changes in specifying the relationship (Stakeholder-Value), and in any changes to either the Stakeholder or the Value specification itself.

It is a fundamental Planguage rule that we specify *once*, and re-use *many* times. Duplicate parallel records are dangerously difficult to keep updated.

At this point the reader might have discovered something I take for granted, but which many people initially misunderstand. When I refer to stakeholder Values, I do not primarily mean financial ones. I means absolutely all values, things they have a 'stake' in. This then includes what many people call 'soft' values; and which they believe are 'not easily quantified' or 'engineered'. **I believe we can usefully quantify, and thus 'engineer', all these so-called 'soft' values.** [Quanteer, 3].



By definition, all stakeholders must have at least one 'stake' (requirement) and all requirements must have at least one stakeholder. So we can make 'orphan requirements' (shaded rectangles) or 'orphan stakeholders' (shaded rectangles) **visible**, for review purposes. It is a small step further in an app, to create a 'list of orphans', from this information. Automated Review.

When the lines from one stakeholder go to, for example, 3 different requirements (like 'Schools' in Fig. 3.3), these 4 (3+1) specification objects can be thought of as making up a ***Virtual*** Stakeholder specification. Not only that, but it is a 'living specification', sensitive to the most-detailed changes, as time goes on.

3.4.1 Primary stakeholder values – working notes

Value Analysis incubators and accelerators

Stakeholders	Addressable concerns	Met by	Delivered by features	Financial impact	Quantifiable gain	Intangible impact	Concerns	Benefit
Service owners	Profitability/ occupancy	Proof of results over time	Audit trail	Potential higher return from pipeline of applicants	Calculable from business model	Confidence in the future	Concern of exposure outweighs visible success	Forecastable revenue stream
	Visible success	Transparent high performance	Pitch results	Shorter time to investment or commercialisation	Depends whether faster turnover delivers profits	Reputation	"	Higher numbers and quality of applicants
	Fears exposure for poor perfomance	Knowing where to improve	Gap analysis showing where help is needed	"	Lower failure rate repaid if service has shares	Reduced fear of exposure	"	Confidence in methods
Funders/sponsors/ grant providers	The long period before the value of donations is known	Leading indicators of success	Individual and aggregate incubatee assessments	(for service) continued willingness to donate (invest)	(for service) running costs assured	Confidence of continued involvement	Uncertainty about outcome resolved faster	Concern allieviated management time saved
	Tangible evidence of value from donation (or investment)	"	Audit trail	(for service) greater/new willingness to give (invest)	(for service) opportunity to increase funds and improve	Confidence that greater involvement will pay off	Commitment justified or reinforced	Emerging positive story to tell
	Six/seven year wait for results that may expose failure	"	Individual and aggregate incubatee assessments	(for service) continued willingness to donate (invest)	(for service) running costs assured	Confidence of continued involvement	Uncertainty about outcome resolved faster	Concern allieviated management time saved
Service Management	Evidence for performance/ contribution	Visibile progress of individual and aggregate incubatees	Individual and aggregate incubatee assessments	Keeps job may earn bonus	Success bonus	Improved confidence in ability	Initial concern of exposure overcome by visible success	Respect of incubatees, confidence of owners
	Ways to improve poor incubatee performance	Knowing which mentor/coach can help	Gap analysis showing where help is needed	"	"	Ability to improve methods	Confident intervention	Respect of incubatees, confidence of owners
	Fears exposure for poor perfomance	Knowing where to improve	"	"	"	Ability to improve methods	"	Respect of incubatees, confidence of owners
Incubatees	Uncertain whether/how much progress they're making	Progressive proof of achievement	Assured optimal expert feedback and audit trail	Shorter time to investment or commercialisation	Depends on business model and time to market	Confidence that the team is on track	Sense of achievement fostering a will to succeed	Faster commercialisation
	Not knowing where their shortfall in knowledge lies	Clarity sbout which topics and abilities to work on	Gap analysis showing where help is needed	"	"	Confidence they are addressing the right issues	Sense of learning and growing	"
	Concern that mentors and coaches are too specialist	Knowing that mentors are focusing on their weak skills	Gap analysis showing where help is needed	"	"	Confidence they are receiving the appropriate	"	"
Mentors/coaches	Knowing of the limited scope of indivual experience	Synthesising coherent advice from multiple perspectives	Audit trail	More focused use of time	Freed up hours for more productive woek	Increased assurance of adding value		Freed up hours for more productive woek
	Objectively prioritising the most significant shortfalls	Recent reference data indicating problem areas	"	"	"	"		"
	Knowing when the incubatee has absorbed the lesson	Recent data that shows high performance	"	"	"	"		"

Figure 3.4.1 An example of David Stoughton's Stakeholder Analysis.

A nice practical example from one of my professional friends, of analysis of stakeholders. It is useful as a draft, and as an overview. But it is not, digitally and formally, *'capturing all the data about a stakeholder'* that we are discussing here. It only looks at *five* stakeholders. It does give innovative ideas about the kinds of data you *might* want to consider collecting; and about *language* in which to connect it.

Stakeholder Analysis of Value Entities (SAVE. !)

Draft by David Stoughton, for GilbFest 2020, david@value-kinetics.com

Stakeholder Analysis of Value Entities (SAVE. !)

Value Analysis - directly involved stakeholders

Stakeholders	**Concerns that could or should be addressed**	**Evidence that would show concern is satisfied**	**Function that delivers the evidence**	**Direct or indirect financial outcome** (thumbnail scale)	**Other tangible outcome** (thumbnail scale)	**Intangible or emotional outcome** (thumbnail scale)	**Secondary benefits**	**Concerns the proposed solution might generate**

Draft value proposition - long form

Figure 3.4.2 David Stoughton's Template for Stakeholder Analysis.

"I use this format for developing value propositions - I've left those headings below the table.

I've tried to come up with a generic form of the issues I'm addressing in each column. When I talk of primary stakeholders this is often the full purchasing group in an organisation including those they might turn to for advice but excluding the broader social stakeholders. That's so I can hone the value proposition(s) - often adapted as we go to each critical stakeholder as necessary."

Courtesy of David Stoughton, david@value-kinetics.com

4.
Requirements Elicitation

I would appreciate hearing your reactions, suggestions, difficulties understanding me, objections, references and links. After each chapter!
tom@Gilb.com

PS
Feel free to send to friends, and post link on social media, or to use in whole or part in training, lectures, blogs, your own books, papers, slides with main link https://tinyurl.com/StakeholderBook

4.0 Requirements Elicitation

Garbage In, Goodies Out.

When we identify a stakeholder (Ch. 2), and when we collect some data about them (Ch. 3); how do we know what their requirements *are*?

Do ***they*** even really know what their requirements are? Are they able to articulate them *clearly* to us? Even if ‘they’ are animate?

In the *next* Chapter, 5, we will ask, ‘do we even care to commit to deliver their ‘clarified’ values, or respect their constraints?’

The purpose of this Chapter 4, is to *figure out what the stakeholder requirements actually are*, and to *specify them in a clear enough manner for our evaluation,* and (Ch.5) *for our prioritization purposes*.

My observation is that most people are not trained well enough to do this. The job is rarely well done. But allow me to explain this assertion. In advanced ‘stakeholder engineering’ detail.

https://tinyurl.com/PLanalysisFree

A booklet with advice on how to analyze plans, and make them better

PLanalysis

© Tom Gilb, 2020
TOM@GILB.COM

(Feedback welcome)

https://tinyurl.com/PLanalysisFree

Figure 4.0. A booklet on analysis of poorly formulated, or implied requirements. (The photo is my cabin view of the Oslofjord, as I write.)

I find that the material in this booklet is so close to what I want to say that I have made a bold decision to copy the contents (48 pages), as the main base for Chapter 4.

Summary: Most Planning sucks. I'll help you do a much better job, for free. You will help make the world a better place.

4.0.0 Introduction

Surely you have seen bad plans? Have you ever seen a really great and admirable plan? What are the criteria for judging plans, and for declaring they are great?

Most all plans I see are terrible. They are totally lacking in clear ideas about the most critical objectives. They are incomplete, missing critical elements everyone where. And they are 95% ambiguous words, with no attempt at clear definitions.

The problem is so wide-spread, that I have guessed, that is why almost nobody reacts to it. Nobody cries foul. Nobody does anything about it. We just *live* with bad plans.

Well, it worries me a lot. It destroys productivity of the whole world's organizations, private and public. Things cost much more. Results are years late.

But nobody seems to care. Not the leaders, the top politicians, the C-level executives, not the business schools. Hardly a voice is raised. These plans are 'they way it is'. This book is for 'managers', who want to manage.

I guess you are reading this book because you are more interested than most. So, I want to help you out.

This is a very practical book. I am going to show you how to analyze plans, and identify the bad stuff. Then how to do something about it in practice.

If you are leading an organization, or even just a project, then you can expect things to get measurably better, faster, less risky, more productive.

I have a theory that the bad planning methods in widespread use are planted there by our enemies, a sort of planning disinformation.

The comfortable thing about this book, is that you can try out the ideas, immediately, in small steps, and augment the methods as you see earlier efforts succeed.

You do not have to buy a new planning religion, or change your whole organization. Just try things quietly, diplomatically, and you and your colleagues judge for yourselves. The ideas are absolutely free, no permissions, no licenses, no certifications, no expensive training. See the references, with about 90% free downloads from me.

I do not want your money, but I would like to help you make the world a much better place

Clarity is Contagious. Unless you want to sabotage and hide reality. We do need leaders, and I hope that is YOU.

4.1.1: One-Page PLanalysis: Here are some basic questions for looking at a plan text.

1. **COUNT AMBIGUITIES**: on a page or less mark and count all ambiguous words. Anything about 5 ambiguities per 300 words is very bad, and a sign of unintelligible plans. They probably all need definition.

2. **LINK WORDS**: on a page, underline or mark all 'link words', like in order to, so as , by, thus giving, by means of. They indicate 2 levels of concern (ends and means). These need separation (clear agreed ends, before means is specified). And the claim of a relationship needs documentation, not a claim without evidence and source.

3. **AND**: The use of '&'et 'and' in sentences indicates several different considerations, which need separation, identification, quality control, and justification.

4. **BULLET POINTS**: bullet points ('*', and similar (dash (-)) and even simple numbering (1. 2. 3.) are indicators of no stable identity of the idea. No one single approved instance of that idea which can be referred to as the master definition. No followup, no responsibility. You need stable Name Tags, or at least a unique number.

5. **DEGREES**: look for and mark words like, increased, enhanced, reduced, better, excellent. They indicate degrees of improvement: but you need numbers not words.

6. **GENERALITIES**: non specific words like, people, cases, productivity, organization, team, security, are a type of ambiguity that needs definition into a set of interesting instances (not merely a good definition)

7. **NO EVIDENCE**: when claims are made for good strategies, look for any evidence for the claim, and a named source of the evidence. Maybe even who is responsible for the good results, or blame if not.

8. **NO SOURCE**: for every claim, look for a specific source (like a URL to a study). If not, assume this is high risk.

9. **CAUSALITY**: anything If we do X then Y will happen, is a claim and evidence and responsibility need to be there.

10. **SIDE EFFECTS**: if claims If X there Y, do not even mention all associated costs, and side effects for all objectives and stakeholders, then the point is poorly researched, and you are at great risk something will go wrong.

4.1.2 Here are some nice actions you can take when you discover plan defects: PlanFix

1. Do not personally attack the planner. Attack the plan. Who ever wrote it did their best, as they understood the job, and got trained to do it. If there is any fault, it is 'management'.

2. Before you ever announce a plan defect to anyone else, you should yourself draft a pretty good correction to the plan. Like define terms, quantify objectives, give sources.

3. Before announcing defects to a group (by email, or in a meeting) discuss confidentially with a sympathetic person. Tell them what you are thinking of doing. And ask if they agree and are your ally. Ask what they think you should do. "**Never walk into a room without an ally**". (Trygve Lie, UN Sec. Gen. principle)

4. Consider taking up the defects with the plan author directly and confidentially, and non-threateningly. Offer your help to make their plan look better.

5. Point out that there are no official rules or standards yet, for some of the defect types, and offer to develop them,

6. At some point, make the point that the organization needs to improve their training and standards for planning, so as to reduce the plan defects. (Experiences says 'by 100x!). Offer to make it happen (train, standards). Have your new starts ready and drafted. Rewrite part of the plan to show how they work.

4.1.3.1 Core beliefs about plans. The Logic of Planning © Gilb 2020

1. **FUTURE**: Plans are most critically about moving towards balanced-sets of stakeholder-value targets.

2. **CLARITY**: If stakeholder-value targets are unclear, we cannot reach them through planning.

3. **MULTIPLE**: All plans must deal with *multiple* targets, and multiple constraints, simultaneously.

4. **COMPLETENESS**: If we fail to deal with *any single* critical target or constraint, the entire plan can fail to deliver, any or all, of the desired future states, and/or the expected level of resource constraints needed.

5. **CONSTANT CHANGE**: all plan elements (like objectives, strategies, constraints, stakeholders) are continuously subject to change, to mirror real-world changes.

6. **CONSTANT UPDATES**: if the plan does not get updated frequently enough, with those critical changes, then it is risking some degree of failure of results.

7. **STAKEHOLDERS**: plan stakeholders are *many*, with *many* needs, not just for a *class* of stakeholders (like ‘nurses’), but with *variations for individuals*; and the stakeholder needs picture is always changing.

8. **NEED CHANGE**: if we fail in our plans to plan for real critical-stakeholder needs, including their *need to change*, and to *be an individual*, then we risk undesirable results in the system being planned.

9. **CHANGING UNKNOWNS**: it is impossible to know all stakeholder needs, and all system requirements, in advance. They will be discovered gradually, and they will change.

10. **PLANNING ADAPTABILITY**: it is possible to plan any system, so that it can more-easily adapt to new changes during the system lifetime. Open-ended systems.

4.1.3.2 Planning *consequences* of the Core Planning Beliefs (in 4.1.3.1)

1. **STAKEHOLDER PLANS:** Thorough analysis, and specification of all critical stakeholders, and their needs, is required.

2. **OBJECTIVES QUANTIFIED:** Absolutely all critical objectives must be specified *quantitatively*, for intelligibility.

3. **MULTIPLE RESOURCES**: All critical resources, *people, time, money, space; both short term and longer term,* must be budgeted, and managed, during *design* and *implementation*.

4. **STRATEGY VALUES ESTIMATION:** Potential strategies *(means, designs, architecture)* must be quantitatively evaluated, against the *quantified objectives* and constraints, together with strategy *risks and uncertainties*.

5. **STRATEGY DECOMPOSITION**: Large strategies, need to be decomposed into smaller strategies, and deployed incrementally: scaling up when proven, and modified when disappointing.

6. **DYNAMIC PRIORITIZATION:** Prioritization needs to be dynamic, in small increments, to cope with changes and new insights. It needs to be based on a selected policy; like 'best value for resources, and risks'.

7. **INCREMENTAL DETAIL**: It is sufficient to do *detailed planning* for the *near term* increments. It is premature to plan in detail, *too* far ahead (as in chess)

8. **CORE PLAN**: The main 'plan implementation controls' can be a one-page table, showing the most-critical objectives and resource budgets, together with current planned value progress, and resource consumption. Nothing else is essential. [UN Case, 15, 17 Goals]

9. **VALUE VALUES**: The essential planning question, at all times is: *'how much progress have we made towards our planned value targets, in relation to remaining budgeted resources?'*

10. **CONTRACT**: as far as legally and practically possible, all payments and rewards for plan implementation, should be based on the degree-of-measurable (and stable, locked in) delivery-of-values, within budgeted constraints.

4.1.3.3 Good Questions about plans

One way to identify good plans.

You can put these on the back of your business card. I did

Twelve Tough Questions

1. NUMBERS
Why isn't the improvement *quantified*?

2. RISK
What is degree of *risk* or uncertainty; and *why*?

3. DOUBT
Are you *sure*? If not, w*hy* not?

4. SOURCE
Where did you get *that* from? How can *I* check it out?

5. IMPACT
How does *your* idea affect *my goals*, measurably?

6. ALL CRITICAL FACTORS
Did we *forget anything critical* to survival?

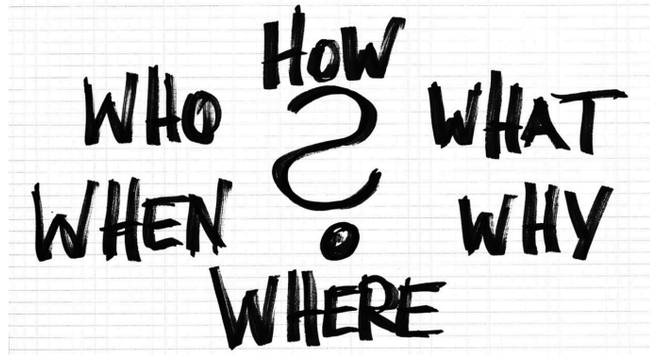


Figure 4.1.3.3

7. EVIDENCE
How do you *know* it works that way? Did it '*ever*'?

8. ENOUGH
Have we got a *complete* solution? Are *all* objectives satisfied?

9. PROFITABILITY FIRST
Are we planning to do the '*profitable* things' first?

10. COMMITMENT
Who is responsible for failure, or success?

11. PROOF
How can we be *sure* the plan is working, *during* the project; *early*?

12. NO CURE
Is it *no cure*, *no pay*, in a contract? Why not?

PLanalysis Checklists

4.1.4. Plan Knowledge Verification

Making sure you can trust and use the knowledge

My basic ways of verifying knowledge:

1. **Search for case study facts, or research, on the internet.**

2. **Challenge the source, to supply *evidence*, facts, numbers, measures, references, studies, case studies.**

3. **Try it out, in your own work. If you are in my profession, you get your clients to try it for you, but make sure you get their results later.**

4. **Challenge people, maybe those with competing ideas, to 'show fault', in your evidence base, or to show *better* evidence, for *their* competing ideas. Notice I did not say 'argue with words'. I said 'show data'.**

Figure 4.1.4 Plan Knowledge Management & 'Accounting'

https://www.semanticscholar.org/paper/Knowledge-management-and-measurement%3A-a-critical-Ragab-Arisha/f994db4aeffd79f92ef2ddc2c9b1eb20bea0ddeb

4.1.5 Technical Plans, Requirements

Paper: 20 Tough Questions 2016
http://concepts.gilb.com/dl876

10 Tough Questions You can ask about *Plan Objectives*

1. Have you **agreed** a set of your top-10 critical-value objectives for the product?

2. Are those objectives **unambiguously clear,** to *all* who might *have to* understand them; the intended readership?

3. Is it clear *which* requirements the **stakeholders support,** and are interested in?

4. Are the requirements really values, qualities and **results**: *not* the *technology,* we *think* will get us results.

5. Is it clear - *what* the **worst acceptable value** delivery level is? (Tolerable level)

6. Is it clear - what the **Wish level** is, and that this is not a commitment *yet* (Goal level): until we find technology and resources, to reach a promised 'Goal' level?

7. Is it clear what the requirement's knock-on **value** is, for example 'economic', or in terms of higher-level objectives, if we reach the Wish or Goal level. What is it worth?

8. Do we know the **defect density** of our specifications? If you can see more than 10 unclear or ambiguous words on a requirements page, is this a threat to understanding your project? (See Terzakis, Intel, [D1])

9. Do we have other **major stakeholder levels that need a *separate* specification of requirement**s? Like; Business Level, Stakeholder Level, Product Level or Sub-Product Level.

10. Is there any *requirement, which is* arguably **more-critical** than the top-ten, that we failed to include or specify? Now that we *think* we have a complete set: what is missing?

Edited 20 Q slightly 190720, edit 060820

Intel has used my knowledge methods for over 20 years for over 20,000 trained engineers. This was part of an invited speech I held for them in 2016

The Keynote Slides: 'Power Planning Principles'
http://concepts.gilb.com/dl874 (pptx version)
20 April 2016
'Accelerate Results' Intel Conference,
Hillsborough Oregon

The Video
https://www.gilb.com/blog/power-planning-principles?cid=87f388e7-e0bc-4796-ab1a-c7faad2674d3

3rd Gen Intel Xeon Scalable processor, dubbed 'Cooper Lake'

Image: Intel

Figure 4.1.5 Intel Product.

4.1.6 Technical Plans: Designs & Architecture

These 20 Tough Questions (10+10) are not for beginners. They assume training and experience in my methods of knowledge (Planguage, Spec QC, Evo) which 20,000 Intel Engineers had been trained in, and used. Video is at https://www.gilb.com/blog/power-planning-principles?cid=87f388e7-e0bc-4796-ab1a-c7faad2674d3

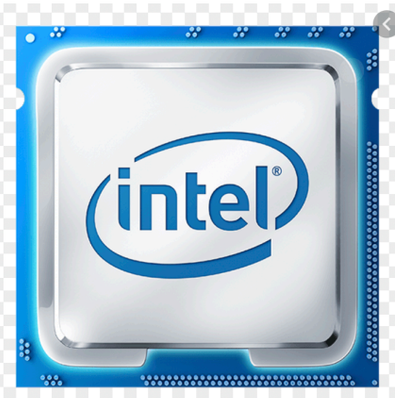


Figure 4.1.6 Intel Logo

10 Tough Questions You can ask about Solutions, ***Design & Architecture***

11. Are the **designs/solutions specified so unambiguously and clearly,** so that nobody can **inadvertently misunderstand them, including *what* to estimate and *what* to implement?**

12. Have you estimated the short-term and life-cycle **costs**, in both time and money, for **each** major strategy, design, or solution?

13. Have you looked at the ratio of solution-impacts over their costs (solution impacts/solution costs): so you can select the most **efficient solutions**?

14. Have you looked at the **worst-worst case** (for 'credibility' '±uncertainty') for all *value* impacts, and all *resource* impacts?

15. Can you consider implementing the most efficient (effects/costs) solutions **early, to get feedback,** learning, and possibly deliver *real value* to the field?

16. Can you *decompose* *any* design solution, into **smaller,** independently-implementable, sub-solutions? High-value sub-solutions can *then* be done *earlier*.

17. Have you invited **competitive imaginative engineers, to come up with far more cost-effective solutions** than *you* can show them, on your Impact Estimation Tables? Using the Impact Estimation Table as a *provocative baseline* for discussion.

18. Is it possible to improve **the Impact Estimates [2,3,4]**, and improve *certainty*, by better research, on existing experience of the solutions, or by experiments, or pilots? Can you get better solution *credibility,* for 'deciding-what-to-do early'?

19. Can we conduct simple, short-term, this week, *A/B experiments*, to get better data and experience, on *some* of the solutions?

20. What can we do to **motivate the best design engineers (and architects)** to analyze our ideas, and **come up with *better* ones?** Both up front, and after delivery-cycle feedback?

Edited 20 Q slightly 190720, edit 060820

20 Tough Questions 2016
http://concepts.gilb.com/dl876

4.2.0 Term Analysis

How to analyze ‘terms’ into useful categories.

We are looking at terms in plans, 1 or more tightly related sequential words, for the following purposes:

1. CLARITY: To see if they are unambiguously clear, or need better definition

2. CLASS: To determine classification, as to planning object.

3. RELATION:To see if they are useful in defining other terms.

4. RULES: to see if they violate rules or standards

5. LINK WORDS: indicating a bad mix of ends and means.

Here are some simple examples.

Part of a typical plan, with many objectives.

What the NHS Long Term Plan will deliver for patients

These are just some of the ways that we want to improve care for patients over the next ten years:

Making sure everyone gets the best start in life	• reducing stillbirths and mother and child deaths during birth by 50% • ensuring most women can benefit from continuity of carer through and beyond their pregnancy, targeted towards those who will benefit most • providing extra support for expectant mothers at risk of premature birth • expanding support for perinatal mental health conditions • taking further action on childhood obesity • increasing funding for children and young people’s mental health • bringing down waiting times for autism assessments • providing the right care for children with a learning disability • delivering the best treatments available for children with cancer, including CAR-T and proton beam therapy.
Delivering world-class care for major health problems	• preventing 150,000 heart attacks, strokes and dementia cases • providing education and exercise programmes to tens of thousands more patients with heart problems, preventing up to 14,000 premature deaths • saving 55,000 more lives a year by diagnosing more cancers early • investing in spotting and treating lung conditions early to prevent 80,000 stays in hospital • spending at least £2.3bn more a year on mental health care • helping 380,000 more people get therapy for depression and anxiety by 2023/24 • delivering community-based physical and mental care for 370,000 people with severe mental illness a year by 2023/24.
Supporting people to age well	• increasing funding for primary and community care by at least £4.5bn • bringing together different professionals to coordinate care better • helping more people to live independently at home for longer • developing more rapid community response teams to prevent unnecessary hospital spells, and speed up discharges home. • upgrading NHS staff support to people living in care homes. • improving the recognition of carers and support they receive • making further progress on care for people with dementia • giving more people more say about the care they receive and where they receive it, particularly towards the end of their lives.

Figure 4.2.0 *Source:https://www.longtermplan.nhs.uk/wp-content/uploads/2019/01/the-nhs-long-term-plan-summary.pdf*
The full plan: https://www.longtermplan.nhs.uk/publication/nhs-long-term-plan/

4.2.1 Term Analysis, *based on a set of 'Rules'*

Observations: on Figure 2.0 NHS Objectives

1. The '•' (bullet point) is a sort-of 'term' indicating, a new statement. But it also indicates that the statement has **no identity** (nameless), and cannot be cross referenced later ('bullet point 23' ?) and itself is not referencing any particular specification in the rest of the plan. This is probably also a rule violation ('All statements will have a unique Name Tag'), like T1 (Figure 2.1)

2. Notice the terms starting the statements, like *'reducing', 'ensuring', and 'preventing'*. Clearly these indicate a 'degree of improvement' for a stakeholder value. In many cases a *number* is specified. But in many cases *no* number is specified. Most of these statements, I would classify as an 'Ambition Level' (Mgt BS); and expect much-more-detailed specification *somewhere*, to explain this 'headline'.

3. There is a very large number of ambiguous terms *('mother and child', 'benefit', further action', 'support')*, in addition all the scalar terms *(increasing, expanding). T3, T4.*

4. There are no references to the **basis** for the decision (T6), or the **responsible** instance for the result *(T8)*

5. *There are no useful classifications of the nature of the statements. The heading says 'will deliver for patients' and some of them are indeed objectives. But some of them are clearly NOT, such as 'spending money', and 'diagnosing more cancers'*

6. *There are many violations of T9, no 'Link Words ('preventing up to,', 'by diagnosing' , 'to prevent') these terms imply guaranteed causality. They choose and determine strategies, **before** we even have a 'clear objective', and without showing us, 'how a selection was made' of all possible strategies.*

7. *There is more, but the density of violations of clear planning rules is so pervasive here, both* badly-specified *things, and* omissions *of information, that the **plan defec**t (rule violation) **density** exceeds any reasonable level*. So the conclusion is not, to fix it up in bad spots. A total proper **rewrite** is required.

8. The obvious excuse that, 'this is just a summary', is invalid since there is no direct reference to clearer better plans. We cannot read **these** plans and understand them.We cannot review or QC them. Management and politicians **cannot make decisions** to do these things on this basis.

9. I looked at the *detailed* plan, and the level of 'objectives specification' is almost identical to these 'summaries', as bad as the summaries.

SOME BASIC RULES
OF PLAN SPECIFICATION;
Which impact 'TERM ANALYSIS'

T1. TAG: All statements will have a Name Tag, for unique identity, or will refer to a Name Tag as it's primary specification.

T2.STATEMENT TYPE: All statements will be proceeded by a declaration of the statement nature, using a defined Term (like Note, Goal, Scale)

T3. UNAMBIGUOUS: All terms will be unambiguous, or defined somehow in the plan glossary.

T4. CLEAR: All terms will be clear enough to be correctly interpreted, and tested, by the Intended Readership, and QC.

T5. QUANTIFY: All value improvement objectives will be specified quantitatively (Scale, Benchmark, Constraint, Target, Deadline).

T6. SOURCE: A reference to the source of the decision to specify this shall be included.

T7. SET TAGS: An unique Identity Name will be implied, or explicitly for every statement, and any referenced set of related statements.Often as a hierarchical set like 'Database.Quality.Scale'

T8. RESPONSIBILITY: An explicit or group of statements reference will be made to the entity responsible for delivering the results indicated. 'RESULT RESPONSIBLE: CTO'

T9: NO TYPE MIX: Link Words (Achieve X thru Y) are prohibited. Means and ends will be separated, and justified.

Figure 4.2.1 Some typical Rules that impact our planning specification, and term analysis
It is amazing how many organizations do not actually have standards like these for planning. Do you?

4.2.2.1 Quantitative Plan Analysis by 'Defect Density' & 'Rules Violations'

Numeric Analysis of a plan. Plan QC.

If you would like a systematic, repeatable, *cheap* method of finding out if the terms violate too many rules, then you can use my Spec QC: Specification Quality Control [4], CE. Here is a report, with with long-term, large-scale successful use, at Intel.

The method is based on 'Rules for planning', very similar to those above [Fig. 4.2.2.1]. A small team (2 to 4 people) takes a representative sample (1 to 3 pages) and they count Rule violations. Every violation is a threat to the success of the plan.

If the density of 'Rule Violations' (defects per page) is too high *(would not pay off, would cost more if we used it, than if we fix it)*, the plan is **refuse 'exit' -** to the next use of it (for example to *architecture* or *strategy planning*). The plan authors have to do, 'whatever it takes' to reduce defects. Sloppy planning is not tolerated.

In this case (Fig.4.2.2.1) it took 6 attempts. The defects were reduced by 98%. Planners learned to follow best practice rules, in *practice*. And their productivity went up 233%. In other words, it did **not** increase costs to do this, it increased value of professional work. 'This stuff works!' (Erik Simmons, Intel, [CE Book [4].

Intel Measures of Gilb Methods 2013

The Impact of Requirements on Software Quality across Three Product Generations

John Terzakis

Intel Corporation, USA
john.terzakis@intel.com

Abstract—**In a previous case study, we presented data demonstrating the impact that a well-written and well-reviewed set of requirements had on software defects and other quality indicators between two generations of an Intel product. The first generation was coded from an unorganized collection of requirements that were reviewed infrequently and informally. In contrast, the second was developed based on a set of requirements stored in a Requirements Management database and formally reviewed at each revision. Quality indicators for the second software product all improved dramatically even with the increased complexity of the newer product. This paper will recap that study and then present data from a subsequent Intel case study revealing that quality enhancements continued on the third generation of the product. The third generation software was designed and coded using the final set of requirements from the second version as a starting point. Key product differentiators included changes to operate with a new Intel processor, the introduction of new hardware platforms and the addition of approximately fifty new features. Software development methodologies were nearly identical, with only the change to a continuous build process for source code check-in added. Despite the enhanced functionality and complexity in the third generation software, requirements defects, software defects, software sightings, feature commit vs. delivery (feature variance),**

ays from project
he second to the

, requirements
quality, multi-

II. PRODUCT BACKGROUNDS

The requirements for Gen 1 that existed were scattered across a variety of documents, spreadsheets, emails and web sites and lacked a consistent syntax. They were under lax revision and change control, which made determining the most current set of requirements challenging. There was no overall requirements specification; hence reviews were sporadic and unstructured. Many of the legacy features were not documented. As a result, testing had many gaps due to missing and incorrect information.

The Gen 1 product was targeted to run on both desktop and laptop platforms running on an Intel processor (CPU). Code was developed across multiple sites in the United States and other countries. Integration of the code bases and testing occurred in the U.S. The Software Development Lifecycle (SDLC) was approximately two years.

After analyzing the software defect data from the Gen 1 release, the Gen 2 team identified requirements as a key improvement area. A requirements Subject Matter Expert (SME) was assigned to assist the team in the elicitation, analysis, writing, review and management of the requirements for the second generation product. The SME developed a plan to address three critical requirements areas: a central repository, training, and reviews. A commercial Requirements Management Tool (RMT) was used to store all product requirements in a database. The data model for the requirements was based on the Planguage keywords created by Tom Gilb [2]. The RMT was configured to generate a formatted Product Requirements Document (PRD) under revision control. Architecture specifications, design documents and test cases were developed from this PRD. The SME

ort paper [1] that

provided training on best practices for writing requirements, including a standardized syntax, attributes of well written requirements and Planguage to the primary authors (who were

TABLE I: GEN 2 REQUIREMENTS DEFECT DENSITY

PRD Revision	# of Defects	# of Pages	Defects/ Page (DPP)	% Change in DPP
0.3	312	31	10.06	-
0.5	209	44	4.75	-53%
0.6	247	60	4.12	-13%
0.7	114	33	3.45	-16%
0.8	45	38	1.18	-66%
1.0	10	45	0.22	-81%
Overall % change in DPP revision 0.3 to 1.0:				***-98%***

Figure 4.2.2.1 [12] TERZAKIS INTEL 2011 AND 2013. Practical industrial cases. SQC and Planguage
https://selab.fbk.eu/re11_download/industry/Terzakis.pdf

4.3.0 Phrase Analysis.

Phrase Analysis: sentences and statements

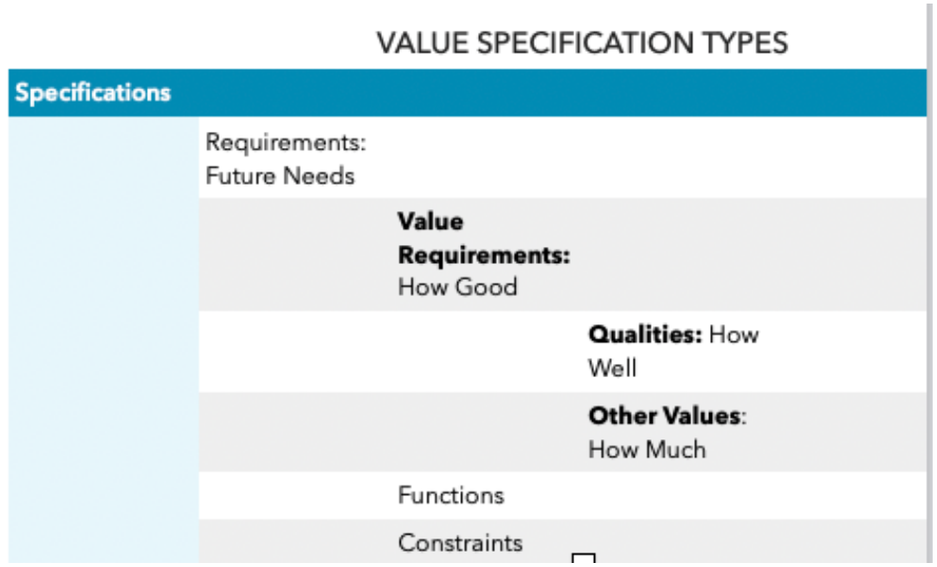


Figure 4.3.0 Stakeholder value types.

The purpose of phrase analysis

Is to identify and separate significantly different types of specification (an Objective like those in Fig. 4.2.0 (NHS) is a phrase).

So that they can get necessary respectful treatment, specification and evaluation

Unfortunately, it is common practice to mix together, even in one sentence, several very different types of planning object. For example 'Objectives + Strategies + Background Info'.

These 'phrases' need to be *separated*, so that they can be properly specified, and then *later* 'linked together', to show their relationships.

If we allow this customary 'sloppy mixture' of very different planning elements to persist, it will destroy the effectiveness of our plans. We will not get well-defined and clear objectives. We will be burdened with the wrong strategies, because they were 'born prematurely', and selected without respect to many other concurrent requirements. One-dimensional thinking is dangerous as a planning method.

Another persistent analysis problem, related to the 'ends **link word** means' problem, but not identical to it, is that the plans relate to very **different levels of concern** (as in organization hierarchy) and they are often not clearly separated. This leads directly to problems with *responsibility* and *traceability*.

4.3.1 Phrase Analysis. Basic Types

Basic Plan Elements

The basic types of plan elements we need to identify, to separate out, and to refine the specification of ,are

1. Requirements: **Future Desired States**

2. Solutions: **strategies, means, architecture to get to future states**

3. Background Information: **all kinds of useful specs related to the plans: responsibility, risks, priorities, issues, etc.**

4. Actions: **plans to do stuff, like invest, get sanction, implement.**

Why separate? So the plan will be clearer, and the planning process will produce more successful plans.

Planning elements and their relationships.

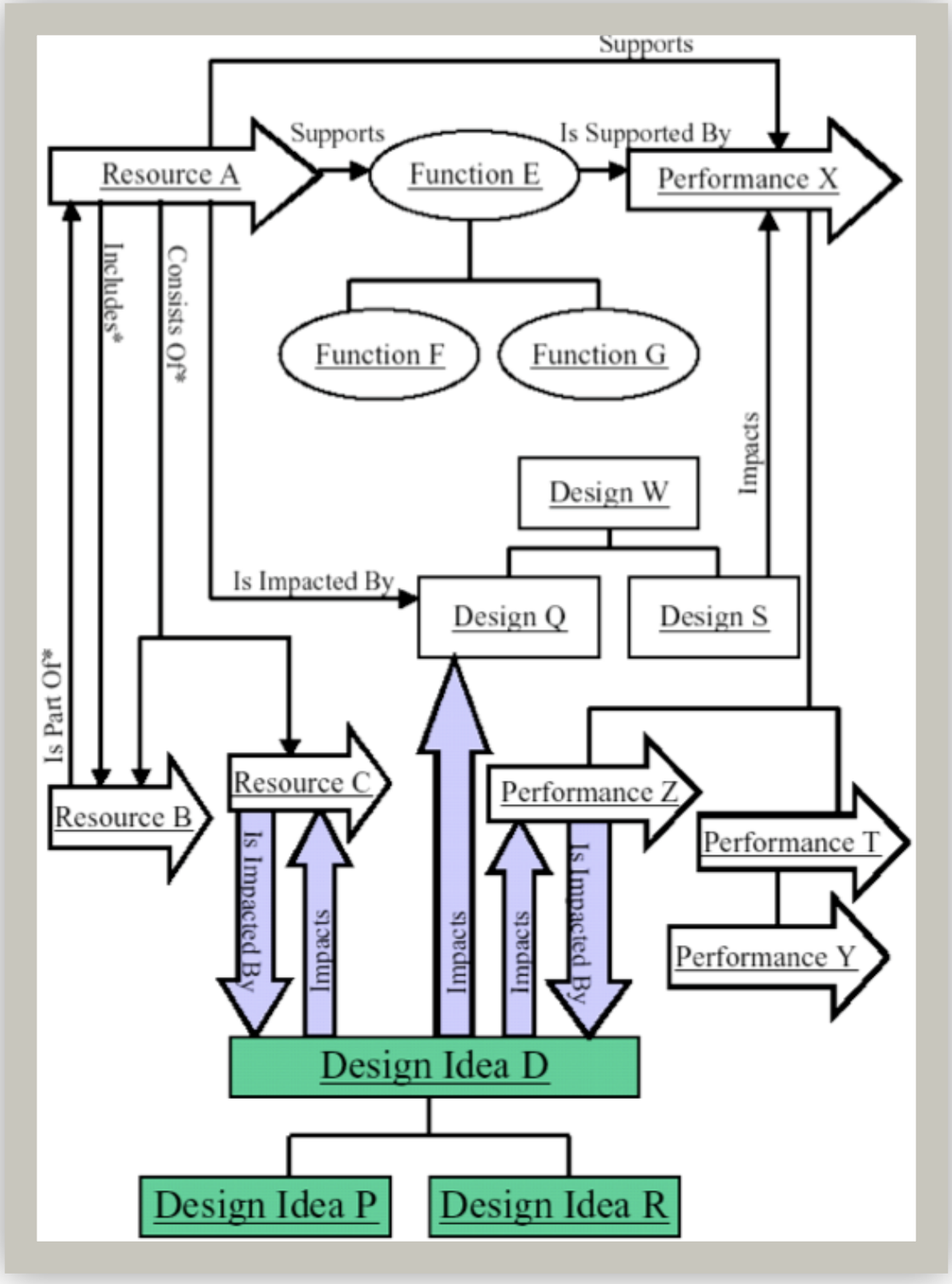


Figure 4.3.1. Source 'Competitive Engineering' [4]

4.3.2.1 Phrase Analysis. More-detailed sub-classes of planning phrase types

Requirements

A requirement is something ‘desired in the future’. But it is important to distinguish between different types of requirements. Figure 3.2.1 A.

A **Function**, is a binary, thing. Function is what a system **does**. It does not require quantification to specify it. But it does require enough precision and detail to *test* that it is there, and to *order it* from a supplier. You may not have to provide it, or build it, because it could *already be* in the older system, you are building on.

1. A ***Performance* requirement**; stakeholder **values** and **qualities;** are *always* a scalar variable. You must define them *quantitatively*, and specify, in your plan *which level you want*; *when*, for *whom* and under which *conditions*. Most of your <u>design</u> (architecture, strategizing) is explicitly directed towards how to deliver these performance levels.

2. **Constraints**: are many types, some are binary (‘use designs we have patents on’) and some are variables (‘no less than 18 degrees C’), some about resources (Budget = X) But they must be *respected*, when choosing strategies, and validating strategies.

Various types of requirements, Objectives, & Visions

Planguage Concept Glossary **401**

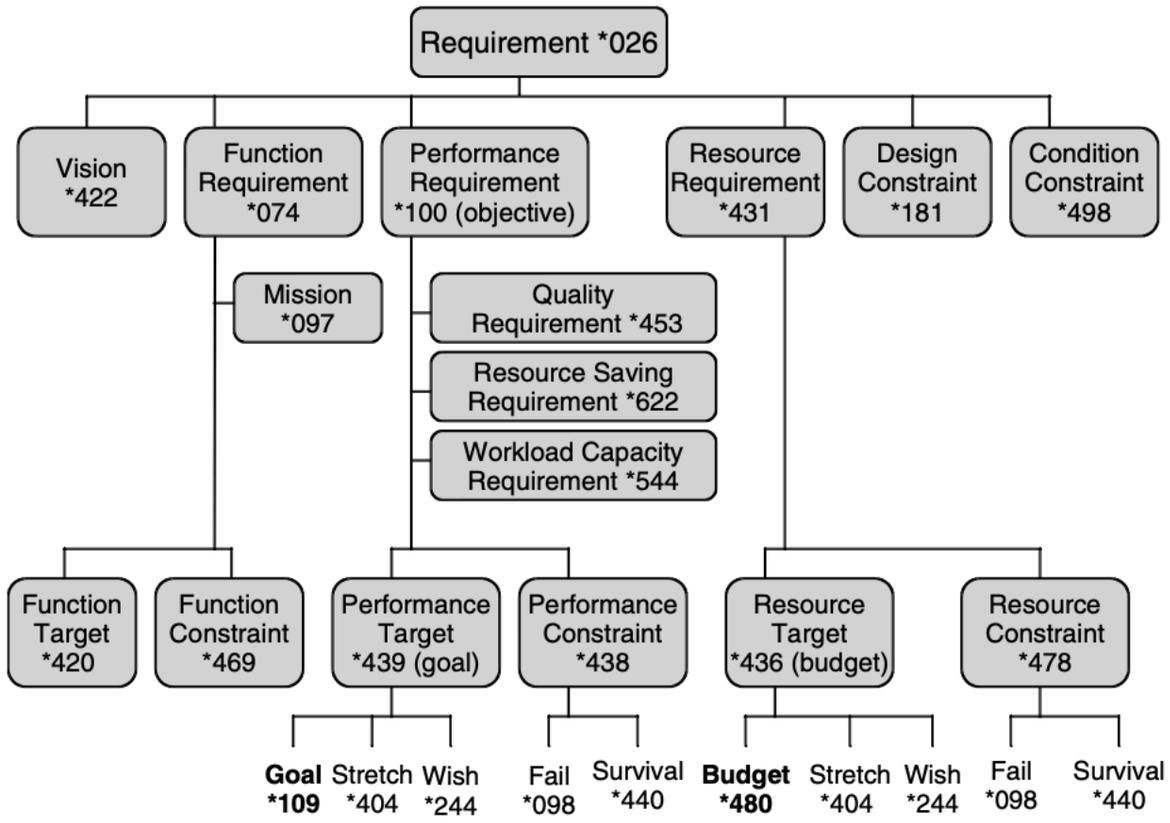


Figure G20
Requirement Concepts.

Figure 4.3.2.1 A. Source CE book [4]

*The *-number, like ‘***026**’ for Requirement, indicates that these concepts are <u>formally defined</u> in Planguage.*

*You can look them up in one of many **free glossaries**, like [B1,] and [P4], and <u>ValPlan.net</u> <u>https://www.gilb.com/valplan</u> (free trial)*

*A ‘Design Constraint’ is interesting because it is **both** a requirement and a design.*

4.3.2.1. Ok Let's take UN Sustainability Goals Poverty Target 1.5 as an example

"By 2030, **build** the **resilience** of the **poor** and those in **vulnerable situations** and reduce their **exposure and vulnerability** to **climate-related** *extreme events and other economic, social and environmental* **shocks and disasters**"

Quote source: [https://sdgs.un.org/goals/goal1

Initial comment: this is a lot of words.
It is not clear at all.
NO Goal-level number for 'Resilience' building.
If we look at it in context (below) it is a sub-goal of UN Goal 1, End Poverty
What is it? A Goal, means objective, a strategy?
And how many ambiguous words do you find here, a simple count, a % of all words? I made them **bold &** underlined.

4.3.2.1 Phrase Analysis. More detailed sub-classes of planning types

Here is an exercise you can do, before you look at my analysis next page:

1. How many ambiguous terms can you find in 1.5, and then in 1.A. ? (in Figure 3,2,1 B here).

2. Do you think the statements are clean enough (defect density low) to publish internationally, and save the poor?

3. Goals 1.5 and 1.A are sub-Goals of UN Goal 1: End Poverty.

    1. Are they the main values, real goals or objectives?
    2. **Or** are they some selected 'means objectives' (a type of strategy) to support the main goal (end poverty)

3. If so, how many other Means Objectives are there, and why were these chosen by the UN?

4. Who selected these 'means objectives', and why?

5. What are these, what UN calls 'targets' (1.5.1,- 1.5.3, and 1.A1, 1.A.2) are they our real goals for ending poverty, are they KPIs (Key Performance Indicators)? Are they 'Sub-Means Objectives? Are they just there to make it look measurable? Or because there might be some statistics at UN for these factors? It is OK to feel confused.

Figure 4.3.2.1 B. Source SustainableDevelopment.UN.org, https://sdgs.un.org/goals/goal1

4.3.2.1 My Analysis Is below

UN-Clear Sustainability Goals

Let me spell it out, to leave no doubt in your mind.

1. Notice 1.5 and 1.A 20 and 28 pitfalls. By my rough count these statements contain 20 (1.5) and 28 (1.A) ambiguous and undefined words.
   1. Like **'resilience', 'exposure', 'ensure', 'significant', 'dimensions'.**
   2. There is **no hope of any 2 people on the planet understanding** all such terms as intended by the author (UN).
   3. T**wo 'Fuzzys'** (1.5 and 1.A) **do not make a Clear Idea** (SDG1), (End Poverty).
   4. If all (48+) ambiguous terms were *somewhere* defined, it might *help* reduce ambiguity.
   5. But there is **no** hint or pointer to such a **glossary** in the UN material. But there are some glossaries! See later.
   6. So everyone is on their own.
   7. Dictionary definitions will not be helpful.

2. In a **desperate attempt to clarify or define**, they **specify a few 'measures'** (Indicators 1.5.1 etc, and 1.A.1 etc.).

   But guess what? **Same ambiguity problem!** What is a '***disaster***'? What are '***resources***'?

If there were some UN statistics for *these* categories, they *should be referenced, right here.*

1. This is a **messy mixture of <u>ends</u> and <u>means</u>,** many levels of them.

2. Phrases like 'in order to' [1A] and 'to (end poverty)'[1A] are what I call **'link words'**. They link a suggested ***means*** *(strategy, solution)* to a specified ***end***.

3. The situation is that **we have not defined 'end poverty' at all.**

We have suggested some **specific strategies** ('mobilization of resources' (1.A), 'predictable means') (1.A) to reach a **badly-defined goal** ('end poverty').

Premature specification of <u>strategies</u> to solve badly-defined problems, is a bad planning idea.

4. We *cannot know* if these various nice-sounding ambiguous strategies are **cost-effective**, because we do **not have a clear definition** yet of 'end poverty', **to judge them by**.

X

A selection of The UN 'Targets' and Indicators for SDG1 (End Poverty)

sustainabledevelopment.un.org

SUSTAINABLE DEVELOPMENT GOALS KNOWLEDGE PLATFORM

HOME SDGS HLPF STATES SIDS UN SYSTEM STAKEHOLDERS

ABOUT

1.5 By 2030, build the resilience of the poor and those in vulnerable situations and reduce their exposure and vulnerability to climate-related extreme events and other economic, social and environmental shocks and disasters

<- 20 Pitfalls

1.5.1 Number of deaths, missing persons and persons affected by disaster per 100,000 people

1.5.2 Direct disaster economic loss in relation to global gross domestic product (GDP)a

1.5.3 Number of countries with national and local disaster risk reduction strategies

1.A Ensure significant mobilization of resources from a variety of sources, including through enhanced development cooperation, in order to provide adequate and predictable means for developing countries, in particular least developed countries, to implement programmes and policies to end poverty in all its dimensions

<- 28 Pitfalls

1.A.1 Proportion of resources allocated by the government directly to poverty reduction programmes

1.A.2 Proportion of total government spending on essential services (education, health and social protection)

Figure 4.3.2.1 C.

A detailed treatment of 'Goal' 1.5 is in found in the Sustainability Planning book https://tinyurl.com/UNGoalsGilb, page 14, 18, 19, 22, 24, 28, 46-48, 51, 56-59 Including how to redefine it for clarity.
Apologies for the detailed analysis. Feel free to skip it if you are already convinced the goal specification needs a lot of help.

A discussion of Target 1.A will be found in the Sustainability Planning book https://tinyurl.com/UNGoalsGilb, page 33,
"Target 1.A Resource Mobilization[17]. Analysis "
The detail, at left might be worth studying, if you want to learn some tricks of plan analysis.

4.3.2.1 Here is some more UN Sustainability planning: "Goal 8, Decent Work, and Economic Growth"

I have analyzed it, in detail, on the next page, but if you would like to think about it, before you see my comments, **here are some questions?**

My apology for the dense detail, but I want my argument to be convincing, for skeptics. Skip the detail if you are convinced that the UN Goals are 'wonderful, but unclear'. The blah blah words overwhelm.

1. Can you highlight, or mark up things, that are just *background commentary* (ie neither objective, strategy, or constraint)?

2. Where is the central **Goal 8** mentioned?

3. This 'framework', is **this** Goal 8, now re-formulated as a set of 5 Objectives (called 'streams'). Why are these 5 streams not measurable? They are clear 'variables'

4. Is the *'build back better' 'imperative' which is 'connecting' the 5 streams,* a strategy to facilitate them, or a 6th stream ? (3rd line from bottom, sorry no tags here)

5. I have **bold underlined** some terms, what *are* they?

6. I **have marked up some blue terms what are they?**

7. **What is your count of ambiguous undefined terms?**

"In April 2020, the United Nations released a framework for the immediate socio-economic response to COVID-19, as a roadmap to support countries' path to social and economic recovery.
It calls for an extraordinary scale-up of international support and political commitment t**o ensure that** people everywhere have access to essential services and social protection.

The socio-economic response framework consists of five streams of work:

1. Ensuring that essential health services are **still available and protecting** health systems;
2. Helping people cope with adversity, **through** social protection and basic services;
3. Protecting jobs, supporting small and medium-sized enterprises, and informal sector workers **through** economic response and recovery programmes;
4. Guiding the necessary surge in fiscal and financial stimulus **to make macroeconomic policies work for** the most vulnerable and strengthening multilateral and regional responses; and
5. Promoting social cohesion **and investing in** community-led resilience and response systems.

These five streams **are connected by** a strong environmental sustainability and gender equality imperative to build back better.
The UN Secretary-General has stressed that the recovery from the COVID-19 crisis must lead to a different

Figure 4.3.2.1 D Goal 8 Decent Work, in Covid-19 Times

4.3.2.1 Phrase Analysis. More detailed sub-classes of planning types

Looking for requirements and objectives first

8 DECENT WORK AND ECONOMIC GROWTH

BE CAREFUL TO ASK FOR WHAT YOU REALLY WANT:
You need to be very conscious of the difference between
'Ends' (Value Goals) and 'Means' (Strategies for delivering the Ends),
so that you really get your intended sustainability value improvements.
Even when your 'best strategies' turn out surprisingly bad,
and even deliver results *later*, than your initial goal planning specified.

'means' in the 'ends'

"In April 2020, the United Nations released a framework for the immediate socio-economic response to COVID-19, as a roadmap to support countries' path to social and economic recovery.
It calls for an extraordinary scale-up of international support and political commitment to **ensure that** people everywhere have access to essential services and social protection.

The socio-economic response framework consists of five streams of work:
1. Ensuring that essential health services are **still available and protecting** health systems;
2. Helping people cope with adversity, **through** social protection and basic services;
3. Protecting jobs, supporting small and medium-sized enterpri workers **through** economic response and recovery program
4. Guiding the necessary surge in fiscal and financial stimulus **policies work for** the most vulnerable and strengthening multilateral and regional responses; and
5. Promoting social cohesion **and investing in** community-led resilience and response systems.

These five streams **are connected by** a strong environmental sustainability and gender equality imperative to build back better.
The UN Secretary-General has stressed that the recovery from the COVID-19 crisis must lead to a different economy."

https://www.un.org/sustainabledevelopment/economic-growth/

- **This example is from recent COVID-19 updates to UN Goal 8 'Decent Work and Economic Growth'**
- **The underlined and bold words are 'link words'**
- **They link 'ends' and 'means'**
- **This helps us see the difference between UN Goals (ends) and suggested UN Strategies**
- **Notice that both of these are badly defined, ambiguous,**
  - **Goals are not quantified**

helping people cope with adversity,

  - **Strategies have no estimate impact on the bad goals**

social protection and basic services;

- **This is one of the 17 goals**
- **And there are 7 link-word cases, in this Goal alone.**
- **And dozens of unclear words, political slogans. So this is *not* a basis for serious planning and economic decisions, and prioritization.**
  - **Simple question: which one of the 7 or so strategies, at left, would you do in the short term, and why? (difficult to answer because of fuzziness)**

4.3.2.2 Phrase Analysis. Looking at Solutions, Strategies, Architecture, Enablers

Solutions

How we will deliver the ambitions of the NHS Long Term Plan

To ensure that the NHS can achieve the ambitious improvements we want to see for patients over the next ten years, the NHS Long Term Plan also sets out how we think we can overcome the challenges that the NHS faces, such as staff shortages and growing demand for services, by:

1. **Doing things differently: we will give people more control over their own health and the care they receive, encourage more collaboration between GPs, their teams and community services, as [?]primary care networks[?], to increase the services they can provide jointly, and increase the focus on NHS organisations working with their local partners, as [?]Integrated Care Systems[?], to plan and deliver services which meet the needs of their communities.**

2. **Preventing illness and tackling health inequalities: the NHS will increase its contribution to tackling some of the most significant causes of ill health, including new action to help people stop smoking, overcome drinking problems and avoid Type 2 diabetes, with a particular focus on the communities and groups of people most affected by these problems.**

3. **Backing our workforce: we will continue to increase the NHS workforce, training and recruiting more professionals [?] including thousands more clinical placements for undergraduate nurses, hundreds more medical school places, and more routes into the NHS such as apprenticeships. We will also make the NHS a better place to work, so more staff stay in the NHS and feel able to make better use of their skills and experience for patients.**

4. **Making better use of data and digital technology: we will provide more convenient access to services and health information for patients, with the new NHS App as a digital [?]front door[?], better access to digital tools and patient records for staff, and improvements to the planning and delivery of services based on the analysis of patient and population data.**

5. **Getting the most out of taxpayers' investment in the NHS: we will continue working with doctors and other health professionals to identify ways to reduce duplication in how clinical services are delivered, make better use of the NHS[?] combined buying power to get commonly- used products for cheaper, and reduce spend on administration.**

P1.3. **NHS PLANS,** TWO PAGE SUMMARY: https://www.longtermplan.nhs.uk/wp-content/uploads/2019/01/the-nhs-long-term-plan-summary.pdf

4.3.2.2 Phrase Analysis. Looking at Solutions, Strategies, Architecture, Enablers

Solutions

HOW MUCH: If '*objectives*' are what we **want** to achieve, or to say it better - "**how *much*** we want to achieve', then...

HOW TO: then '**Solutions'** are **how** we 'propose **to** achieve those value levels'. What we propose to ***do.*** *To* ***implement.***

The set of 5 things, in the NHS 10 year plan, are clearly labeled as **HOW**. So these are some class of Solutions. But what are they? Are they the main strategies for reaching the primary Patient Health Goals? (Figure 2.0).

If we read carefully, they seem to be strategies for *enabling* the organization (NHS) able to carry out other medical and organizational strategies, so as *then* to reach the *medical* goals (through entirely different medical *strategies*). So *these* '*organizational*' strategies are at a different level. They serve a *different set of objectives* than the *medical* goals.

But if we take a closer look, the blue statements, are in fact the NHS **Organizational Objectives** (it just does not say that directly, and they are not clear, and quantified. And the stuff in BLACK type is a 'stream of consciousness-brainstormed, badly defined set of strategies. NO: there is a *mixture* of some badly-defined objectives in there too ('to increase the services they can jointly provide..''). I'd call it an unintelligible mess, with no hope of success.

Source: [P1.3.] **NHS PLANS,** TWO PAGE SUMMARY: https://www.longtermplan.nhs.uk/wp-content/uploads/2019/01/the-nhs-long-term-plan-summary.pdf ———>

How we will deliver the ambitions of the NHS Long Term Plan

To ensure that the NHS can achieve the ambitious improvements we want to see for patients over the next ten years, the NHS Long Term Plan also sets out how we think we can overcome the challenges that the NHS faces, such as staff shortages and growing demand for services, by:

1. **Doing things differently: we will give people more control over their own health and the care they receive, encourage more collaboration between GPs, their teams and community services, as [?]primary care networks[?], to increase the services they can provide jointly, and increase the focus on NHS organisations working with their local partners, as [?]Integrated Care Systems[?], to plan and deliver services which meet the needs of their communities.**

2. **Preventing illness and tackling health inequalities: the NHS will increase its contribution to tackling some of the most significant causes of ill health, including new action to help people stop smoking, overcome drinking problems and avoid Type 2 diabetes, with a particular focus on the communities and groups of people most affected by these problems.**

3. **Backing our workforce: we will continue to increase the NHS workforce, training and recruiting more professionals [?] including thousands more clinical placements for undergraduate nurses, hundreds more medical school places, and more routes into the NHS such as apprenticeships. We will also make the NHS a better place to work, so more staff stay in the NHS and feel able to make better use of their skills and experience for patients.**

4. **Making better use of data and digital technology: we will provide more convenient access to services and health information for patients, with the new NHS App as a digital [?]front door[?], better access to digital tools and patient records for staff, and improvements to the planning and delivery of services based on the analysis of patient and population data.**

5. **Getting the most out of taxpayers' investment in the NHS: we will continue working with doctors and other health professionals to identify ways to reduce duplication in how clinical services are delivered, make better use of the NHS[?] combined buying power to get commonly- used products for cheaper, and reduce spend on**

4.3.2.2 Phrase Analysis. Looking at Solutions, Strategies, Architecture, Enablers

Solutions: Some 'Rules for Solutions', please?

SR1. Solutions will be marked with a **Type** (Solutions, Strategies, etc.) declaring what we think it is.

SR2. The **Level** of the Solutions will be specified, in relation to the Objectives it pretends to serve (Organization, Health, Medical Staff).

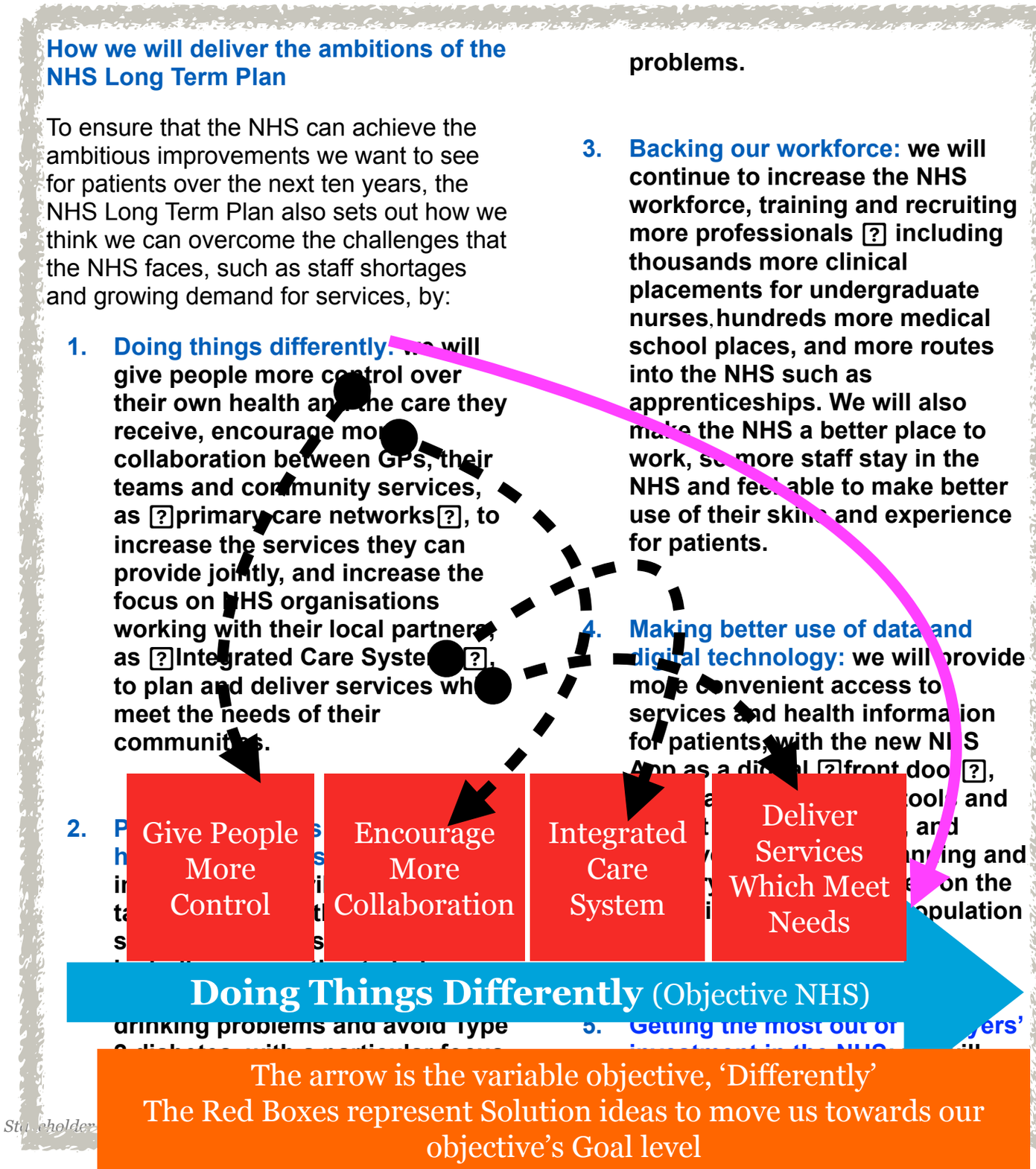
SR3. All Solution ideas will have a stable **Unique Name Tag**

SR4. Solutions may be specified in a **Summary** (entitled 'Summary') of any convenient length, or even several summaries for different summary purposes.

SR5. The only *official and binding specification* of the solution, will be entitled '**Description**', which is short for 'Most Detailed Official Unique Solution Description'. It must be precise, explicit, complete, and unambiguous for purposes.

SR6. All other **specifications related to the Solution** will be collected in a Solution Specification, or at least *referenced* there.

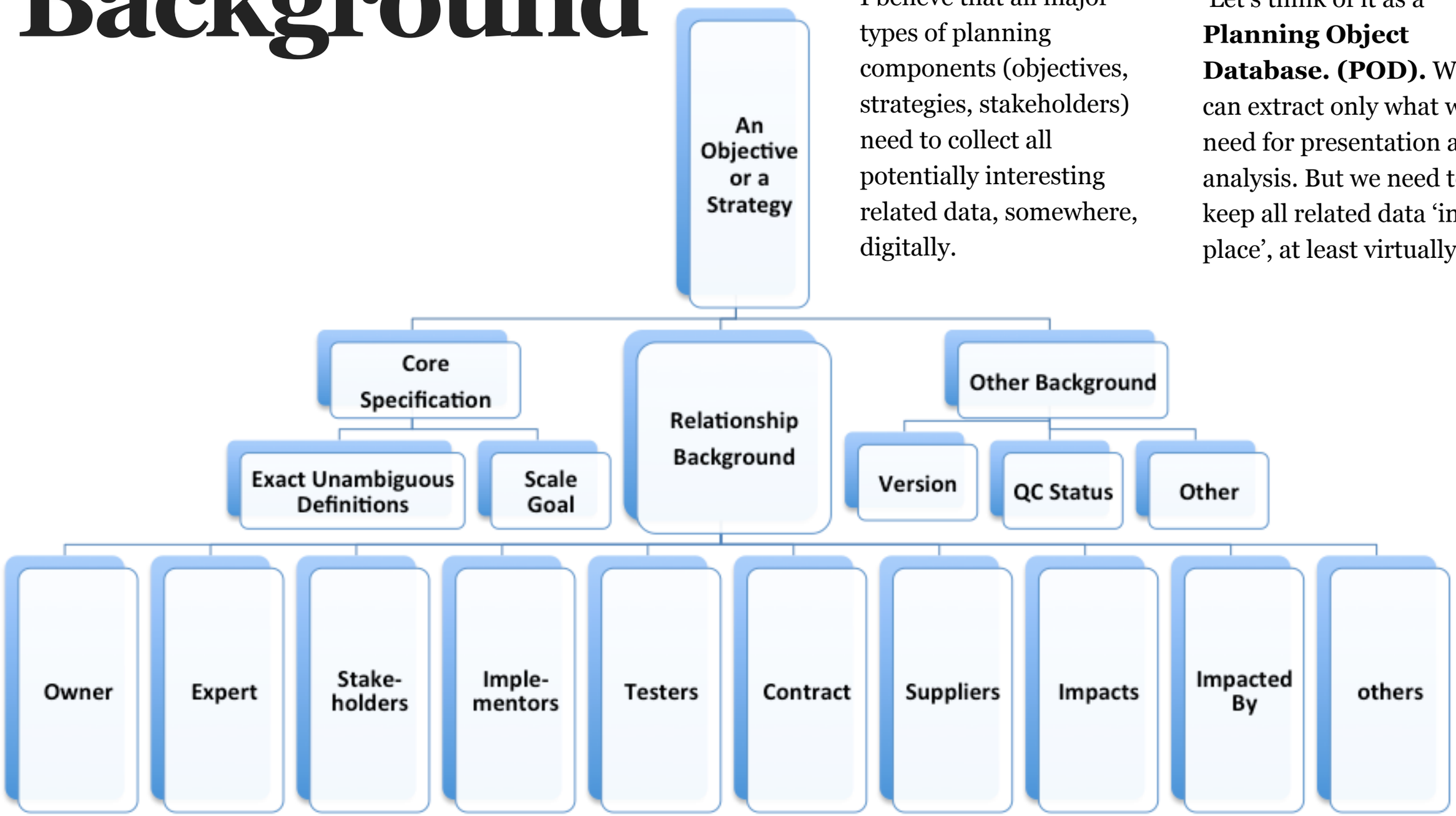
P1.3. **NHS PLANS,** TWO PAGE SUMMARY: https://www.longtermplan.nhs.uk/wp-content/uploads/2019/01/the-nhs-long-term-plan-summary.pdf



The arrow is the variable objective, 'Differently'
The Red Boxes represent Solution ideas to move us towards our objective's Goal level

4.3.2.3 Phrase Analysis. More detailed sub-classes of planning

Background



I believe that all major types of planning components (objectives, strategies, stakeholders) need to collect all potentially interesting related data, somewhere, digitally.

Let's think of it as a **Planning Object Database. (POD).** We can extract only what we need for presentation and analysis. But we need to keep all related data 'in one place', at least virtually.

Figure 3.2.3 A Source: Value Planning [7]. There is much more to a planned objective or strategy, than the 'core' statement. And we need to keep track of it.

Some examples of types of 'Background' data elements, potential parameters as part of a specification, like a strategy.

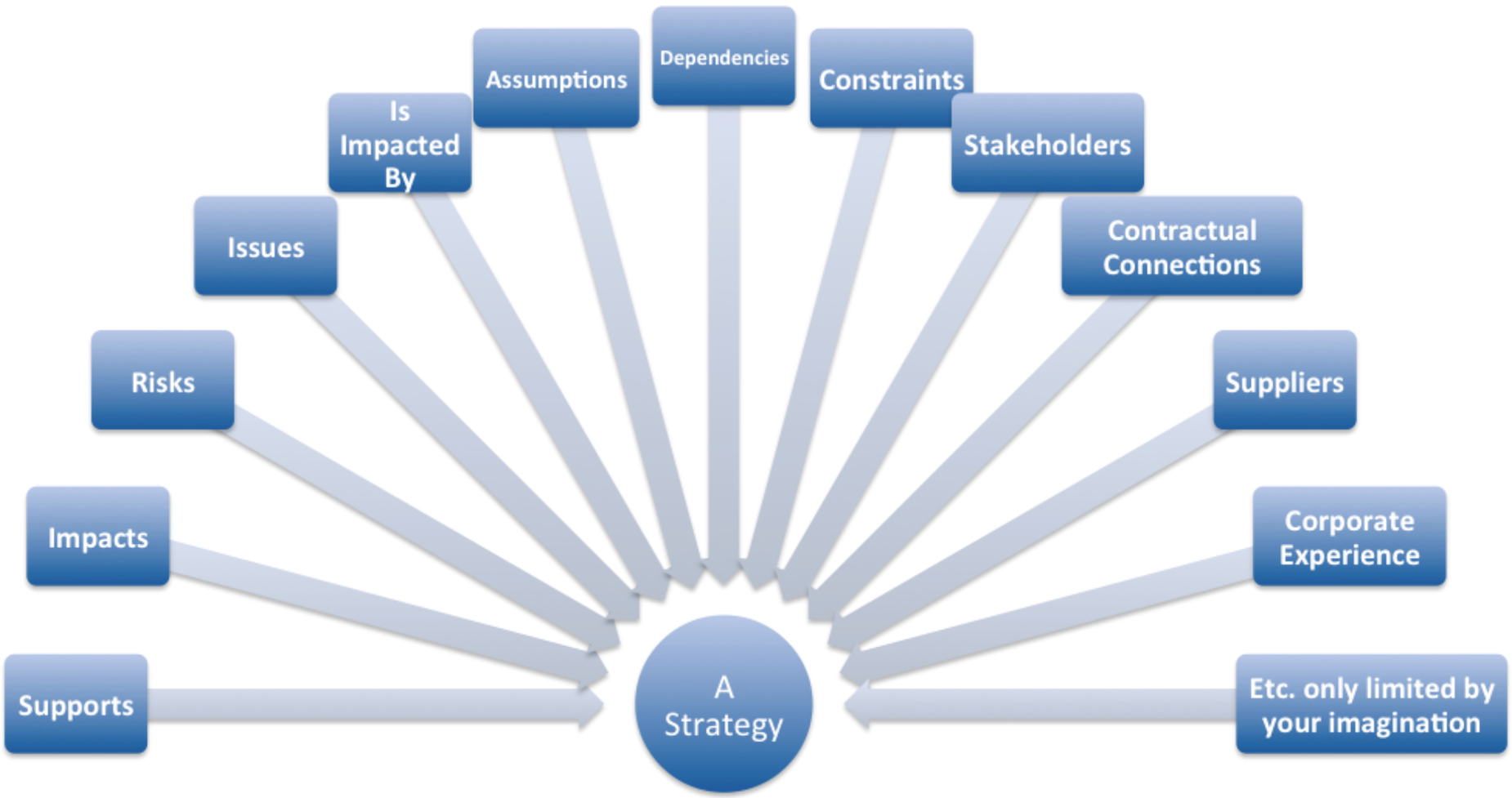


Figure 4.3.2.3 B. Source [7] Value Planning, Fig. 2.2 and 0.2A. An illustrative example of Background Specs.

The idea is that the planning organization commits to incrementally collecting, in the Planning Object Database (POD), certain types of background information that we know from experience is needed, and is a best practice to collect and keep available. A second idea is that there is **only one single Master Object** (like a given requirement), which is the official, updated, quality controlled, securely stored, legally valid, and richest source of data about the Planning Object. This is used for the widest variety of longer-term purposes. Since it is digital, we need not worry about collecting a very large variety and detail (like meeting notes and photos) of related data.

4.3.2.3 Phrase Analysis. More detailed sub-classes of planning types: Background

You don't just write a plan. A plan emerges, until is it no longer needed. Then the Plan Archeologists can have fun figuring out why the system was the way it was. New planners might pick up some ideas for their new plans.

Figure 4.3.2.3 C. Source: Value Planning [7]. 9.2B.
The gradual and eternal process of maturing a specification object, and the Planning Object Database.

What are the reasons for collecting a lot of background data about a Planning Object?

1. Auditing
2. Quality Control
3. Integration
4. Reviewing
5. Prioritization
6. Managing Change
7. Managing Implementation
8. Risk Management
9. Legal Governance
10. Responsibility
11. Motivation
12. Organizational Learning
13. Agile Value Delivery
14. Corporate Memory
15. Many other purposes!

4.3.2.3 Phrase Analysis. More detailed sub-classes of planning types: Background

Extending the Background specs to the Administrative background specs.

Here are some very specific examples of core, background, and '**administrative background**' parameters to describe a Plan Object, like an objective. In this case these are concepts defined in Planguage [B1] and in the 700+ item Planguage Glossary [P4].

The Core is the set of ideas for the main planning object itself, the requirement, the design, the stakeholder. The *Background* is everything else, non-core.

Things about the past, history, older versions of the plan, suggestions that are not yet the official plan, minority opinions, defects found lists, review reports, relationships to everything especially stakeholders, and to other Plan Objects like Objectives and Strategies, and Contracts.

Finally we can distinguish a set of Background specs which are not related to the plan object idea *itself* (the requirement, the strategy) but are all about the **administration** of the Plan Object Core&Background.

Things that help us change, review, take responsibility for planning, for implementation, system testing and measurement.

A simple initial way to exploit these concepts, is as a **checklist when analyzing any plan**.

Ask if what you are reading is one of these.

Ask if this is *missing* from what your are reading. But maybe it is available from people or other sources.

Maybe it is well worth recording in the POD. (Plan Object Database).

Most of these things are already available somewhere. Digital storage makes it interesting to capture, collect, organize and exploit.

Core specs, Background and Administrative parameters of a Planning Object

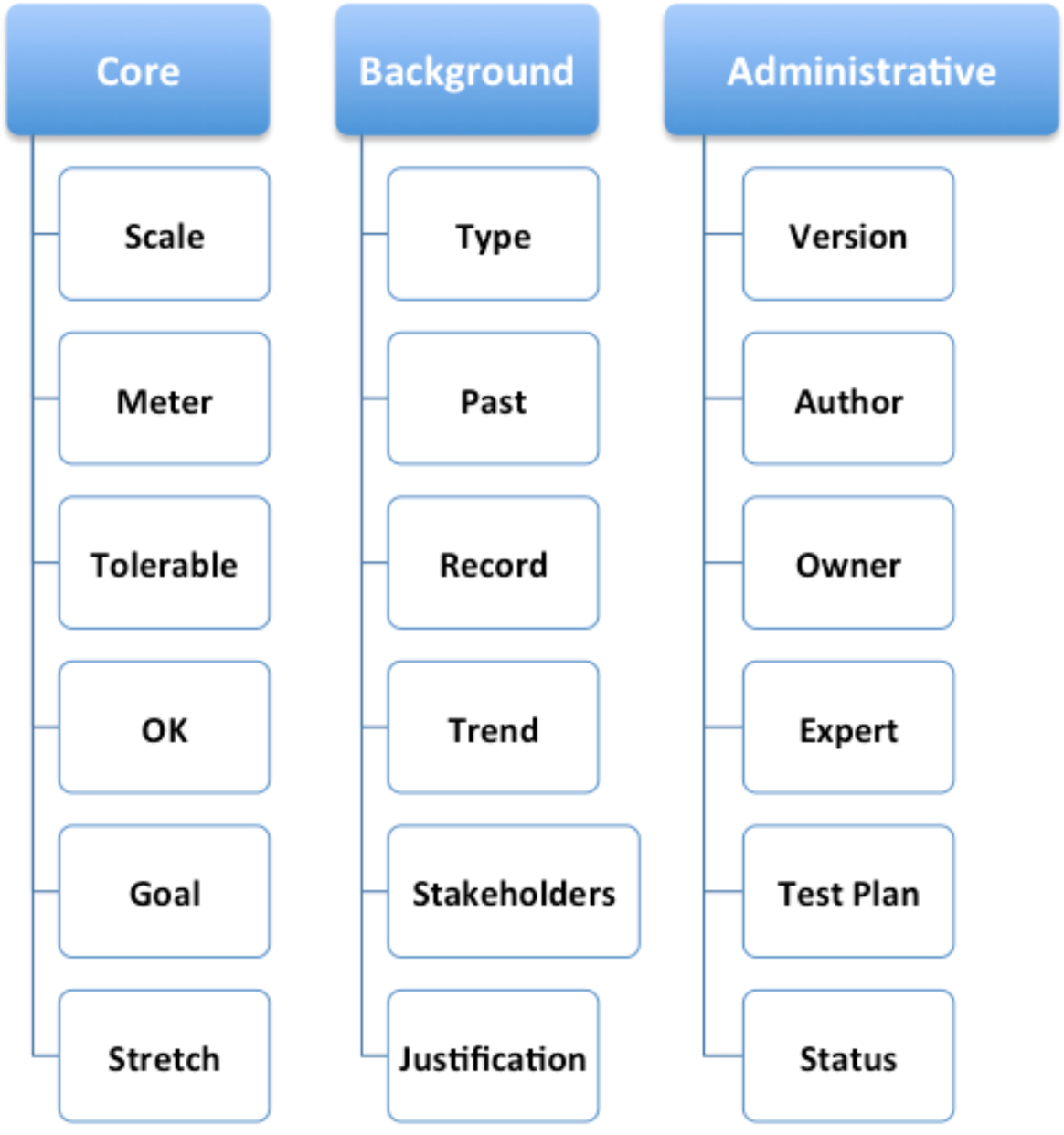


Figure 4.3.2.3 D. Source: Value Planning 4.3 [7].

Background statements example

Here is a constructed example of an objective with lots of background statements, in blue.

You can use these parameter statements, like

Supports: Corporate Profitability.

As a checklist, of required items to document about a plan object,

Or you can **add it into** the Plan Object, as needed. From a larger list of defined concepts.

In automated systems, the **Parameter Heading** (like Type, Stakeholders), is helpful as a definition. We specify what it contains, and can then ask questions, like 'What responsibilities does John Doe have in our projects?' We currently use them to automate plan overview diagrams with automated linkages, for example.

AI Plan Analysis Becoming Reality

In advanced AI systems, it is already possible to let the system decide from a less structured text, what various items are to be classified as. This is currently (2020-1) being done at GraphMetrix for construction plans. And we intend to rapidly extend the capability to all types of plans.

Market Flexibility:
Type: Marketing Director Objective.
Supports: Corporate Profitability.
Stakeholders: Marketing, Production, Distribution.
Potential Strategies: Greater Supplier Flexibility, Greater Distributor Flexibility.
Spec Owner: VP Marketing.
Expert: Marketing Consultant Jane X.
Constraints: EU legal considerations.
Ambition: 10 x faster market adjustment **<- CEO.**

Scale: The average calendar time needed to make a defined [**Adjustment**] in a defined [**Market**], for a defined [**Product**].

Past [General average] 4 weeks **<- Expert.**

Goal [By = End Next Year, Adjustment = **Additional Distributors**, Market = Asia, Product = Toys] less than 1 week. **<- Mkt Dir.**

Risk: EU and specific Asian import countries regulations for safety, might be vastly different. **<- Legal Department.**

Figure 4.3.2.3 E Source [CE, 4] An example (Constructed) of a Plan Object, with a series of Background statements related to it in the Master Plan Object Record: the requirements.

4.3.2.3 Valplan [13] **Automated Planalysis**. Of Background Specifications, like ‘Stakeholders’.

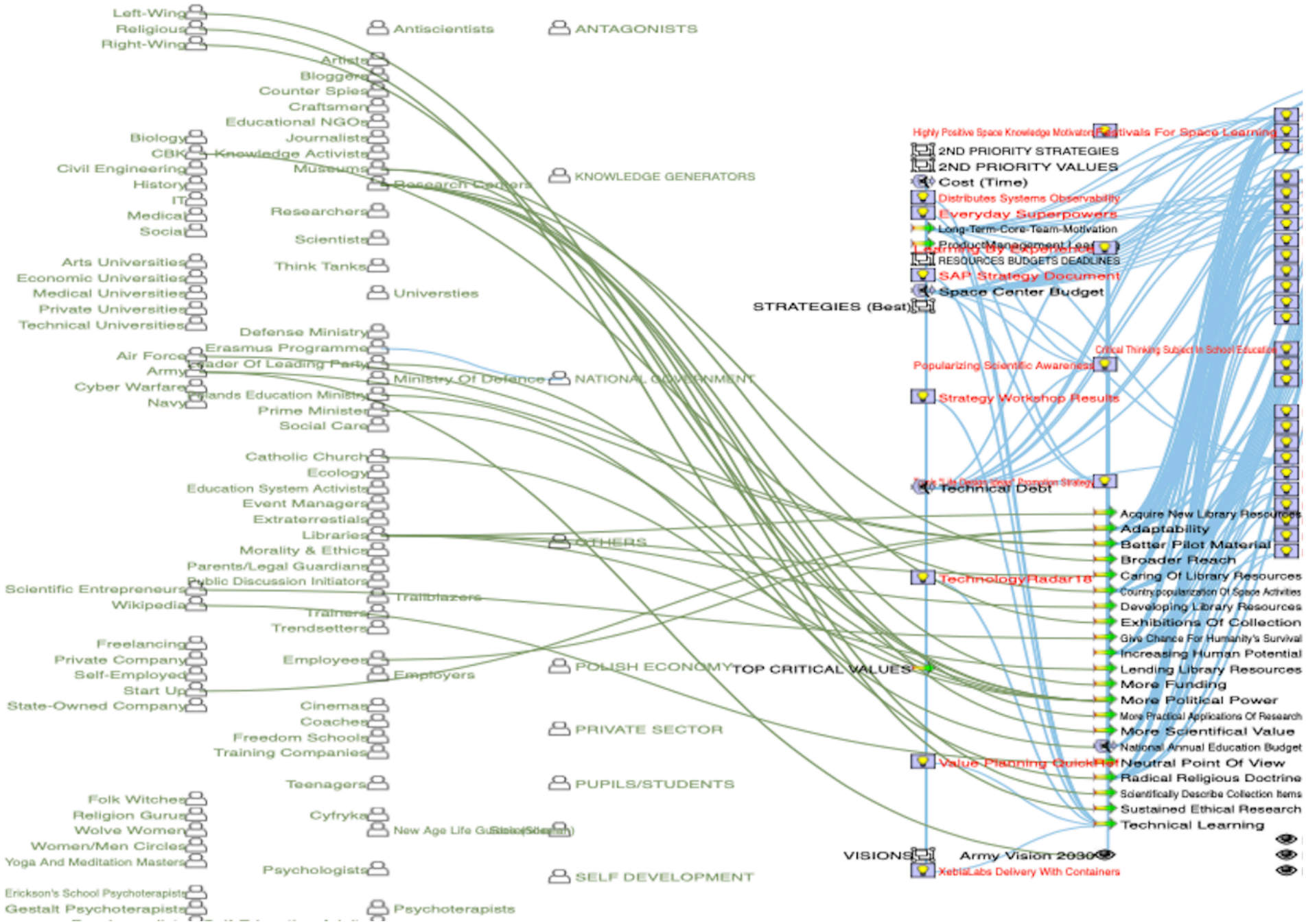


Figure 4.3.2.3 F *Source [13, ValPlan] doing* ***automated Plan Analysis***
of a 5-day Master-course, Plan for ‘Polish Knowledge and Education’, Katowice, 2018
The left side green lines couple Stakeholders to their Objectives.
The right-hand blue lines show the digitally known relationship between all Objectives (arrows) and their supporting Strategies (lightbulbs).
Derived automatically from Impact Estimation Tables.
Now a ***deeper type of Plan Analysis is possible****. For example: Why do many stakeholders (head icon) have no registered objectives (arrow icon)? and Why do several objectives (arrows) have no corresponding strategies (lightbulbs). That is a ‘manual’ or human plan analysis.*
But we can ***automate the analysis*** *by asking*
“Show a list of all stakeholders with no objectives”. *We can of course select any sub-set we are interested in.*
And another question example:
“Show. A list of all Strategies (lightbulbs) which are orphaned, by not supporting any particular Objective.

4.3.2.4 Principles of Plan Analysis:

From Blah-Blah to AI. © Gilb, 2020-Aug-8

1. **GARBAGE IN PLANS**: We can analyze sloppy unstructured plans, and identify esstentially-different plan objects, which need more clarification, completeness, and detail.

2. **RULES REIGN**: if we establish formal rules for plan specification, we can raise the quality level and usefulness of a plan, as well as *quantify* the plan *quality,* in terms of defect density.

3. **NO GARBAGE OUT**: If we can measure the technical goodness; the conformance to our best practice standards, of a plan, then we have an effective tool for teaching better planning, and making sure bad plans do not survive. No 'Exit' if defects abound.

4. **VALUES FIRST**: The most critical purpose of plan analysis is to make sure that critical **objectives** are clarified, complete, drivers of all systems development effort.

5. **NO LINK-WORD PLANS**: The second most critical purpose of plan analysis is to make sure that all **strategies** are *identified, detailed clearly, analyzed for impacts on objectives and constraints*; and NOT assumed to be 'objectives in themselves', ever, until they have proven their place in real implementation.

6. **BACKGROUND**: in addition to the core plan objects (ends, means, constraints) we need to be able to extract, or recover from elsewhere, a large number of related information (like sources, stakeholders, responsibility, risks, assumptions) about the core plan objects, and join them manually or automatically in a Plan Object Database (POD).

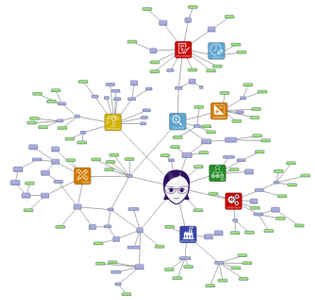


Figure 4.3.2.4. Trinity's Plan Network Symbol [14, Graphmetrix].

7. **POD LEARNING**: A Plan Object Database (POD) should contain an extensive, but useful, or potentially useful set of all related data pieces of information, about the plans, and about the implementation actions and results of the plan: together with the ability to allow research and analysis for organizational learning about planning, and plan execution. See Figure 4.3.2.4, IBM practice.

8. **INTELLIGENT DATA**: Plan data can be systematically annotated, structured, and quantified to make it more intelligible for digital exploitation. (Planguage has done this for decades [4], mainly at the level of spreadsheets).

9. **DIGITALIZATION OF PLANS**: apps can be written, like ValPlan to exploit the structured planning data even better than spreadsheets.
Valplan by Richard Smith to exploit planning structures, contents, and metrics, far beyond what spreadsheets can do. This was based on Planguage [4], and considerable inputs to design from Tom and Kai Gilb for 5 years.

10. **AI PLAN ANALYSIS**: The application of Web 3.0, Trinity Graph Data Relationships-RDF, Ontologies, Solid, and AI [GraphMetrix.com] is the next generation in planning technology. It allows automatic analysis of old and unstructured plans (example 50-year old blueprints and pdf files) to understand data elements, and classify them digitally. Then it has the ability to talk to almost any files and file formats, anywhere, in *real time, securely*. Then build AI apps to exploit the highly intelligible data about the plans. *This is currently working for construction industry planning (August 2020) and will quickly spread to many other classes of planning. See Fig 4.3.2.4*

Plan for a week's value-increment delivery to stakeholders

7.7 Evo Value Delivery Step Template

Defining a stakeholder value delivery *step*, for 'Evo' [4, Ch 10] process of incremental value to stakeholders.

This template is usable in a word processor, and some clients pull it up, using a macro. Copy and paste is fine too.

The <Fuzzy Brackets> are hints and instructions, built-in to the template. They *can* be deleted, when a specification is made.

This template is far more detailed than other agile 'sprints'. But it can be modified in any useful direction. Some readers might want to think of this as agile step planning, serious engineering style.

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 10.8
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.

Figure 7.7 Evo step Template, Source CE [4]

7.8.0 Impact Estimation Table Template

This IET Template explains more of 'why' we use the various components of the table.

Form Follows function (Louis Sullivan).

The IET structure is designed to deliver many things. A summary of these capabilities is in Figure 7.8.

The sequence and positioning on a template is not random, they serve well-thought out purpose.

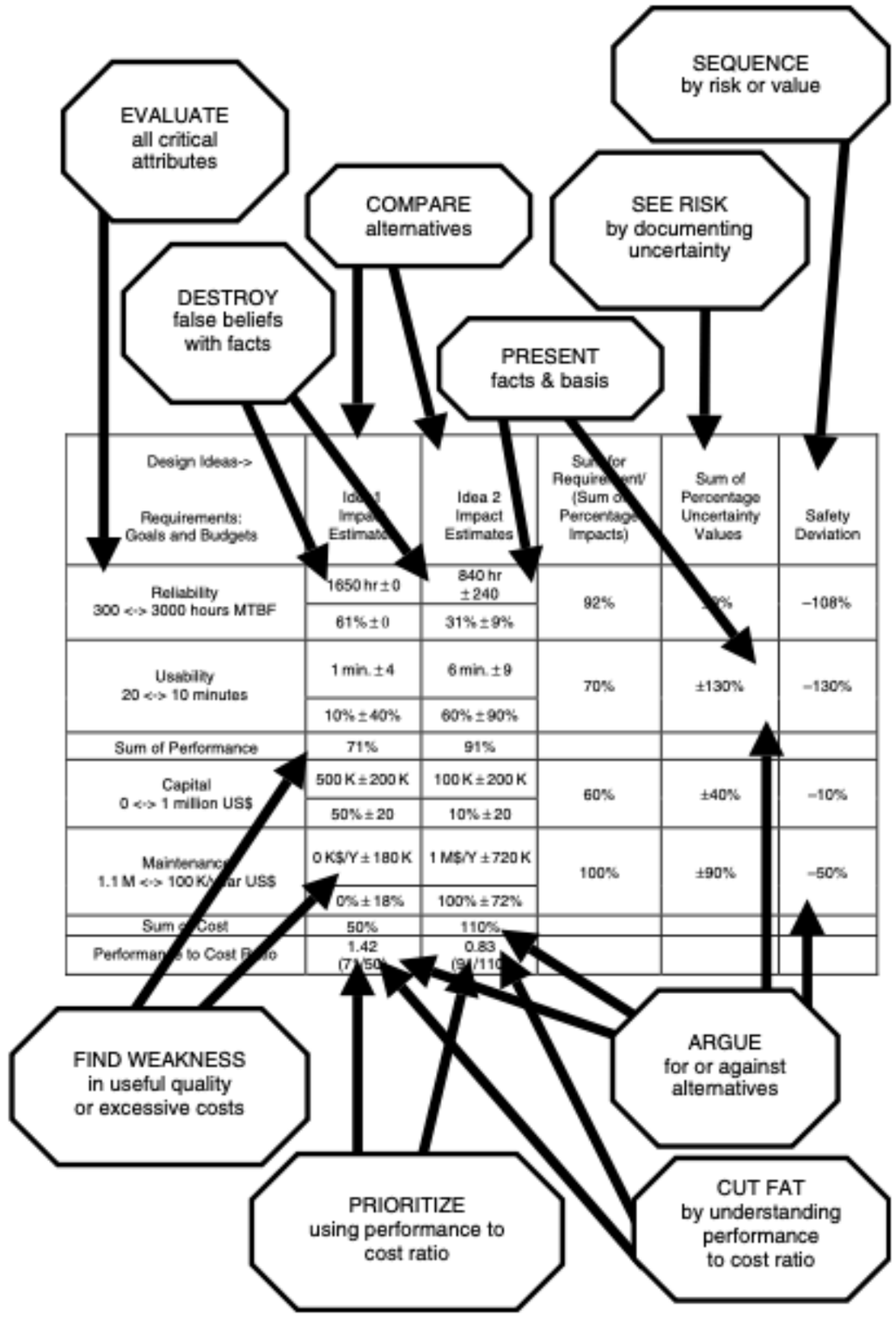


Figure 9.12
Multiple Purposes for IE. Impact Estimation serves many purposes. Here are some headlines and some symbolic pointers to the parts of the IE table which influence these purposes. A list of the main purposes can be found in Figure 9.2.

Figure 7.8 IET Template. Source CE Book [4]

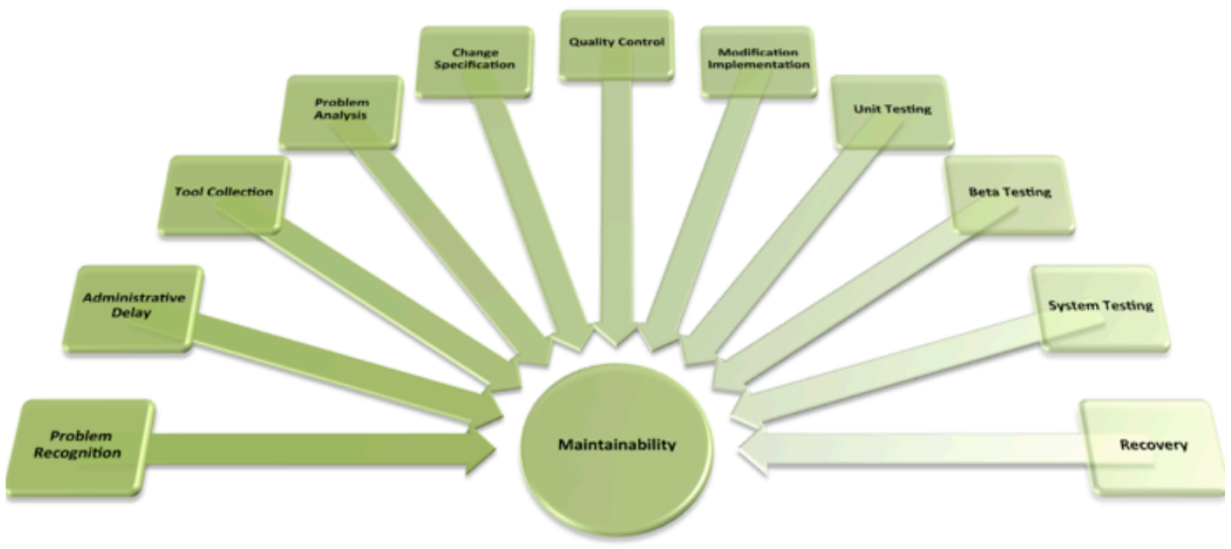
7.8.1. A simple Impact Estimation Table filled out with stakeholders (like 'Board'), and stakeholder attributes (like Criticality).

Requirements	Board Function	Contractor Function	Employee Function	Managers Function
Accessibility Δ: Status: **7** → Wish: **1** Days to Ge...	-3.5 58 %	-1 17 %	-6 100 %	-0.5 8 %
Adaptability Δ: Status: **30** → Wish: **90** % Quick Ac...	33 55 %	2 3 %	60 100 %	3 5 %
Criticality Δ: Status: **40** → Wish: **1** % Failure	-20 51 %	-30 77 %	-2 5 %	-39 100 %
Influence Δ: Status: **2** → Wish: **6** -10+10	2 50 %	0.5 13 %	0.5 13 %	3.8 95 %
Sum Of Values: Σ%:	214 %	110 %	218 %	208 %
Stakeholder COSTS Δ: Status: **5** → Budget: **14** Rel.0to10	2 22 %	4 44 %	8 89 %	1 11 %
Sum Of Development Resources: Σ%:	22 %	44 %	89 %	11 %
Value To Cost:	9.70	2.50	2.40	18.90

Figure 7.8.1 The Impact Estimation Table filled out partly

Maintainabilty Sub-Scales Patterns

7.9.1 Scale and Quality Templates



Maintainability:
Type: Complex Quality Requirement.
Includes: {Problem Recognition, Administrative Delay, Tool Collection, Problem Analysis, Change Specification, Quality Control, Modification Implementation, Modification Testing {Unit Testing, Integration Testing, Beta Testing, System Testing}, Recovery}.

Problem Recognition:
Scale: Clock hours from defined [Fault Occurrence: Default: Bug occurs in any use or test of system] until fault officially recognized by defined [Recognition Act: Default: Fault is logged electronically].
Administrative Delay:
Scale: Clock hours from defined [Recognition Act] until defined [Correction Action] initiated and assigned to a defined [Maintenance Instance].
Tool Collection:
Scale: Clock hours for defined [Maintenance Instance: Default: Whoever is assigned] to acquire all defined [Tools: Default: all systems and information necessary to analyze, correct and quality control the correction].
Problem Analysis:
Scale: Clock time for the assigned defined [Maintenance Instance] to analyze the fault symptoms and be able to begin to formulate a correction hypothesis.
Change Specification:
Scale: Clock hours needed by defined [Maintenance Instance] to fully and correctly describe the necessary correction actions, according to current applicable standards for this.
Note: This includes any additional time for corrections after quality control and tests.
Quality Control:
Scale: Clock hours for quality control of the correction hypothesis (against relevant standards).
Modification Implementation:
Scale: Clock hours to carry out the correction activity as planned. "Includes any necessary corrections as a result of quality control or testing."
Modification Testing:
Unit Testing:
Scale: Clock hours to carry out defined [Unit Test] for the fault correction.
Integration Testing:
Scale: Clock hours to carry out defined [Integration Test] for the fault correction.
Beta Testing:
Scale: Clock hours to carry out defined [Beta Test] for the fault correction before official release of the correction is permitted.
System Testing:
Scale: Clock hours to carry out defined [System Test] for the fault correction.
Recovery:
Scale: Clock hours for defined [User Type] to return system to the state it was in prior to the fault and, to a state ready to continue with work.

Source: The above is an extension of some basic ideas from Ireson, Editor, Reliability Handbook, *McGraw Hill, 1966* (Ireson 1966).

Figure 5.4
A more detailed view of Maintainability.

Figure 7.9.1 Source CE [4] Chapter 5. Scales of Measure

7.9.2 Usability Scale Pattern

This pattern was incremented over decades of experience.

It gives a pretty good set of Usability sub-values. It contains suggested Scales of measure, with [Scale Qualifiers]

Like any pattern we can modify it in any useful way.

But like good patterns it is the result of decades of practical experience.

It also is frequently applied amongst my clients.

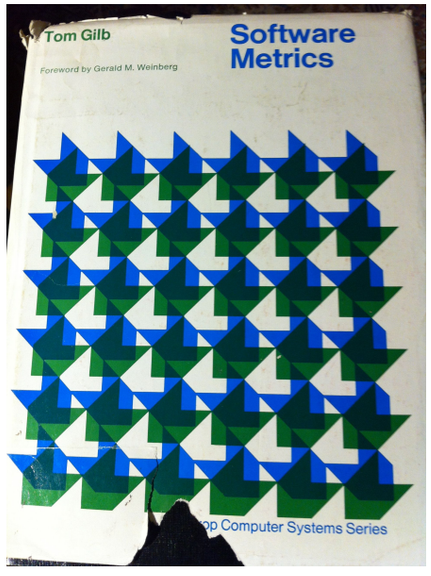
EXAMPLE

Usability:
Type: Complex Quality Requirement.
Includes: Type: Elementary Quality Requirement {Entry Conditions, Training Requirement, Computer Familiarity, Web Experience Level, Productivity, Error Rate, Likeability, Intuitiveness, Intelligibility}.
Entry Conditions:
Scale: <Grade Level of User>.
Training Requirement:
Scale: Time needed to read <any instructions> or get <any help> in order to perform defined [Tasks] successfully.
Computer Familiarity:
Scale: Years of <experience with computers>.
Web Experience Level:
Scale: Years of <experience with using the web>.
Productivity:
Scale: Ability to correctly produce defined [Work Units: Default: Completed Transactions].
Error Rate:
Scale: Number of Erroneous Transactions requiring correction each <session>.
Likeability:
Scale: Option of <pleasure> on using the system on scale of −10 to +10.
Intuitiveness:
Scale: Probability that a defined [User] can intuitively figure out how to do a defined [Task] correctly (without any errors needing correction).
Intelligibility:
Scale: Probability in % that a defined [User] will correctly interpret defined [Messages or Displays].

Figure 7.9.2 Usability. Source: CE [4].

7.9.3 Scale and Quality Templates

2005



1976-7

This diagram is an overview of the more-detailed Scale Patterns in the Competitive Engineering book (2005), ideas **pioneered** in 1976 Software Metrics book.

Performance Attribute Hierarchy Pattern.

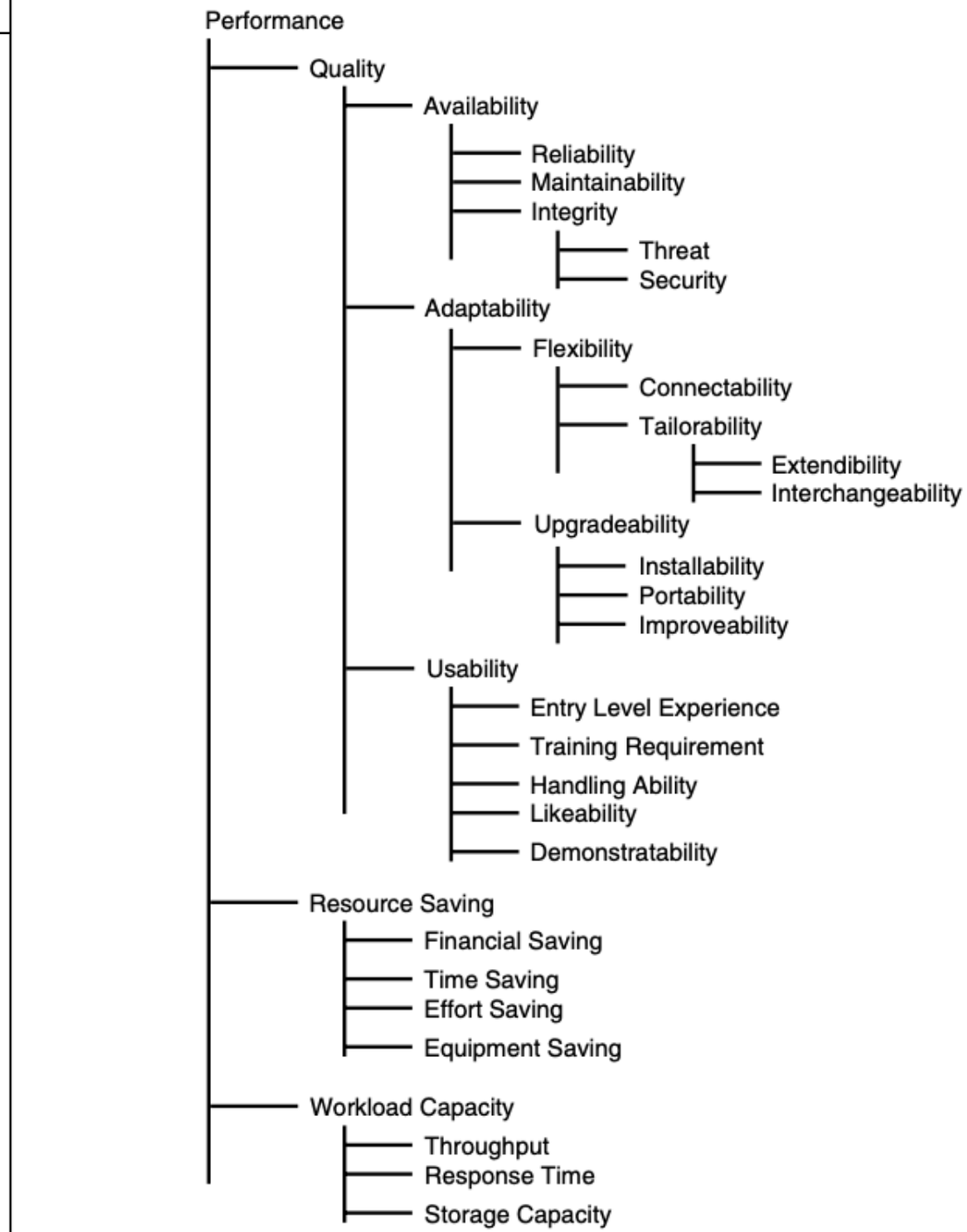


Figure 5.3
One decomposition possibility for performance attributes with emphasis on the detail of the quality attributes.

Figure 7.9.3 Performance Attributes, a general pattern and hierarchy. Source CE [4] 5.3

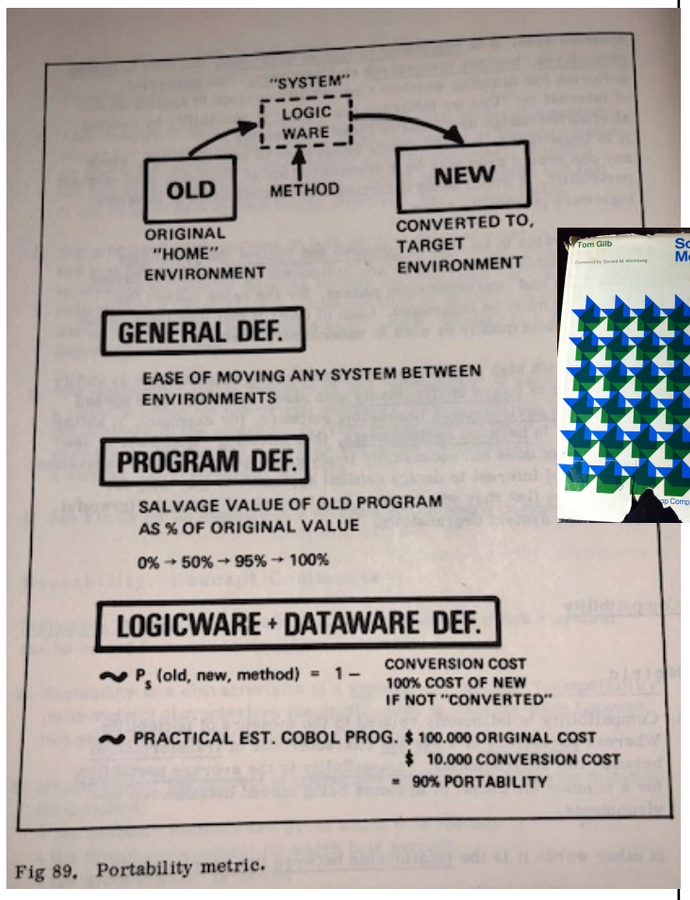
How flexible *is* a system, and can you quantify and define *that* value?

7.9.4 Scale and Quality Templates

Various 'Flexibility' attributes are defined, and their pattern 'Scales of Measure' are here and in [4] in more detail.. Of course anybody can have a different opinion about the meaning of these attribute terms. But we have formally defined them for our method and textbook purposes. You can modify to taste. They are not fixed at all.

They need to be tailored to the stakeholder environment, not least using the [Scale Qualifiers] and their ***conditions.*** Conditions for each [Scale Qualifier] are not suggested in detail here, because they are so varied in different environments. This is where you need to analyze your stakeholders deeply.

I started publishing early versions of these **flexibility** ideas, as such 'patterns', in my 1976-7 Software Metrics book (p. 161-178). So they have matured over time. I practiced the ideas with real clients in the 1960's; like contracting for app 'value' Portability at my stakeholder University of Oslo Press.



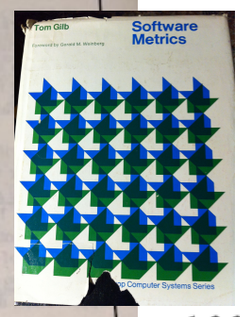


Figure 7.9.4 B Portability diagram. Source Software Metrics 1976

The Flexibility Attribute Defined

1.2.1 **Flexibility**:
Gist: This concerns the 'in-built' ability of the system to adapt or to be adapted by its users to suit conditions (without any fundamental system modification by system development).
Type: Complex Quality Requirement.
Includes: {Connectability, Tailorability}.

1.2.1.1 **Connectability**: '*The cost to interconnect the system to its environment.*'
Gist: The support in-built within the system to connect to different interfaces.

1.2.1.2 **Tailorability**: '*The cost to modify the system to suit its conditions.*'
Type: Complex Quality Requirement.
Includes: {Extendibility, Interchangeability}.

1.2.1.2.1 **Extendibility**:
Scale: The cost to add to a defined [System] a defined [Extension Class] and defined [Extension Quantity] using a defined [Extension Means].
"In other words, add such things as a new user or a new node."
Type: Complex Quality Requirement.
Includes: {Node Addability,
Connection Addability,
Application Addability,
Subscriber Addability}.

1.2.1.2.2 **Interchangeability**: '*The cost to modify use of system components.*'
Gist: This is concerned with the ability to modify the system to switch from using a certain set of system components to using another set.
For example, this could be a daily occurrence switching system mode from day to night use.

1.2.2 **Upgradeability**: '*The cost to modify the system fundamentally; either to install it or change out system components.*'
Gist: This concerns the ability of the system to be modified by the system developers or system support in planned stages (as opposed to unplanned maintenance or tailoring the system).
Type: Complex Quality Requirement.
Includes: {Installability, Portability, Improveability}.

1.2.2.1 **Installability**: '*The cost to install in defined conditions.*'
This concerns installing the system code and also, installing it in new locations to extend the system

Figure 7.9.4 Flexibility. Source CE [4]. Partial text.

Tailor by defining conditions for each [Scale Qualifier].

7.9.5 Scale and Quality Templates. Digital Library Templates and Patterns

Patterns and Templates can be inserted into a digital library, which is ‘Corporate Learning’ and can be re-used, as we work with new plans.

The Patterns are a ‘pretty good suggestion’, from experience. But the planner, guided by stakeholder information, can change anything about a pattern, like adding several new [Scale Qualifiers], and of course Tagging the specification with any Tag in any Language they want.

Selecting A Usability Scale, from the App Library

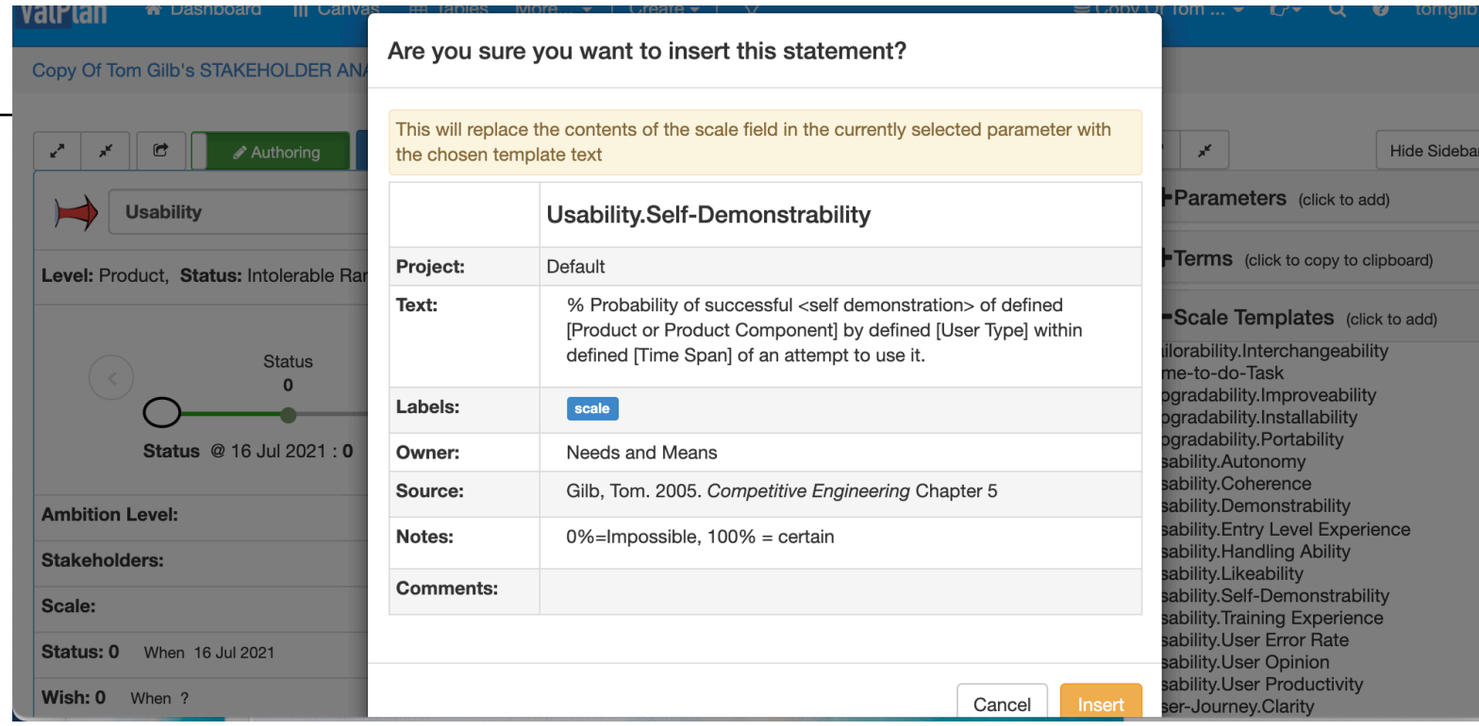


Figure 7.9.5 A. Source ValPlan file on Stakeholders

The ‘Usability Self-Demonstratability’ Scale is inserted into the Scale, and the [Scale Qualifiers] invite us to define them; usually with a set of Conditions.

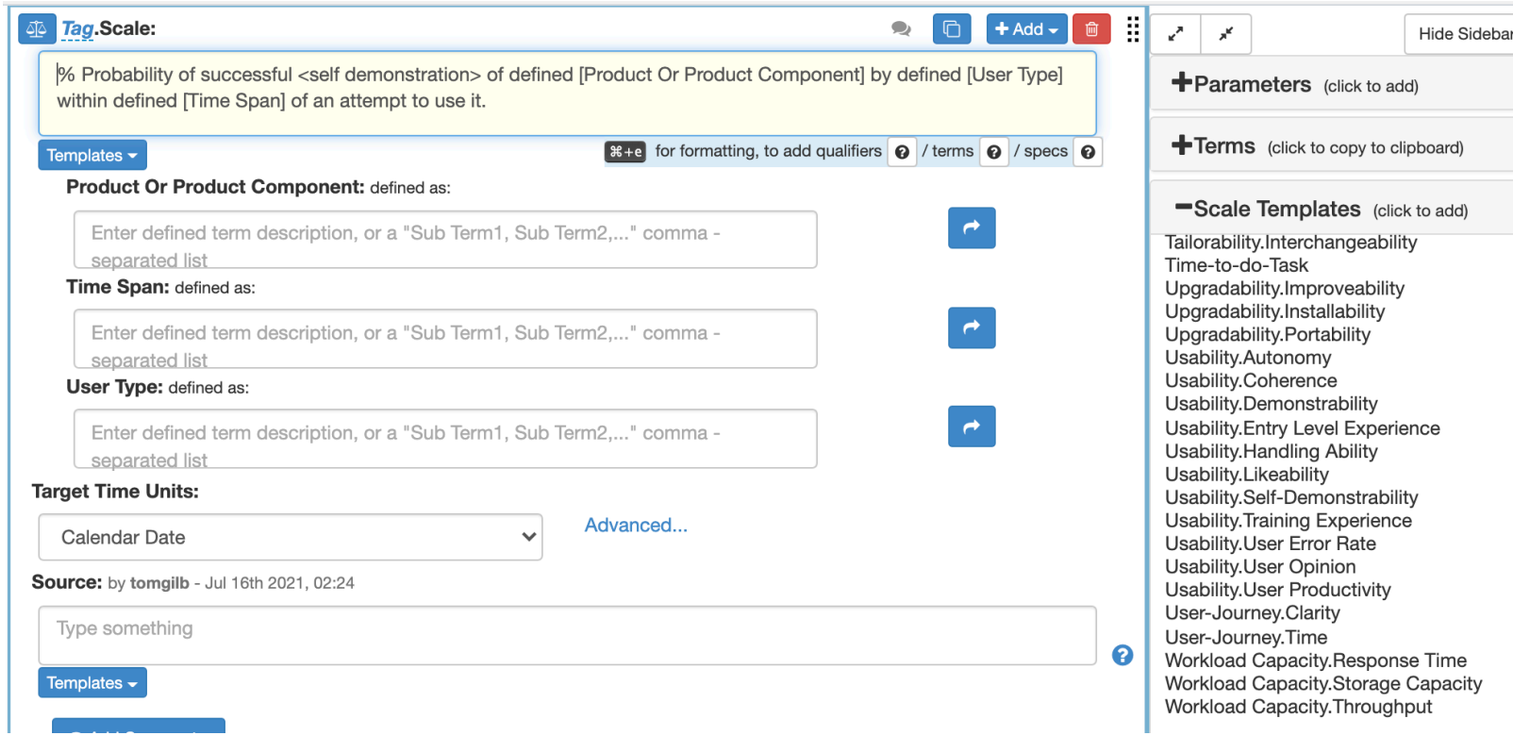


Figure 7.9.5 B. The selected Scale is automatically inserted into the Scale window, and we are ready to tailor with Condition sets for ‘User Type’ etc.

8. Other ‘Patterns’

I would appreciate hearing your reactions, suggestions, difficulties understanding me, objections, references and links. After each chapter!
tom@Gilb.com

PS
Feel free to send to friends, and post link on social media, or to use in whole or part in training, lectures, blogs, your own books, papers, slides with main link
https://tinyurl.com/StakeholderBook

8.0 Pattern =

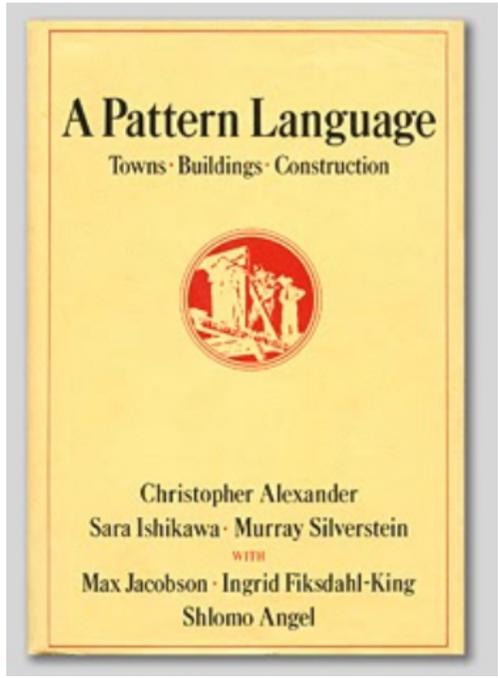
Pattern **Concept *655 (Planguage Glossary)**

- A ‘Pattern’ is a general tool
- that defines any idea structure,
- or sets of structures,
- in any intelligible format,
- which can be exploited by tailoring it to,
- or copying it into,
- a specific domain.

Patterns are a form of stored wisdom or knowledge. The main idea is to allow people to organize and reuse knowledge, rather than re-inventing it from scratch or memory.

Examples of patterns in Planguage and the CE book are: Principles, Scales of measure, Examples, Case Studies, Processes, Procedures, Concept Glossary Definitions, Entry and Exit Rules, Rules (for Specification), Standards.

***Planguage** as defined in Competitive Engineering is a **set of diverse patterns** intended to inform the user of useful structures which can either be copied or tailored to specific purposes and domains. (Version 2006, 2013, 2021)*



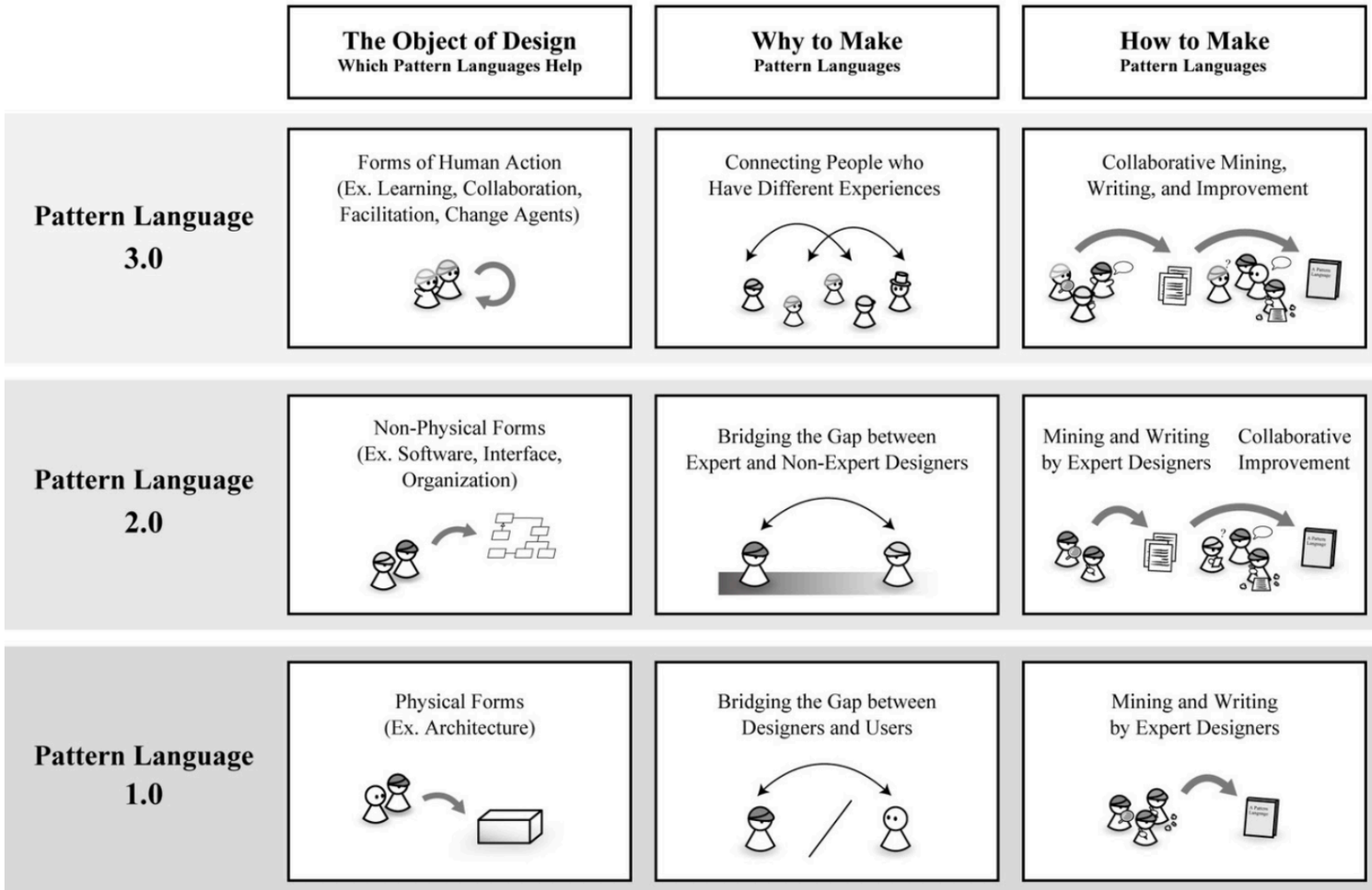
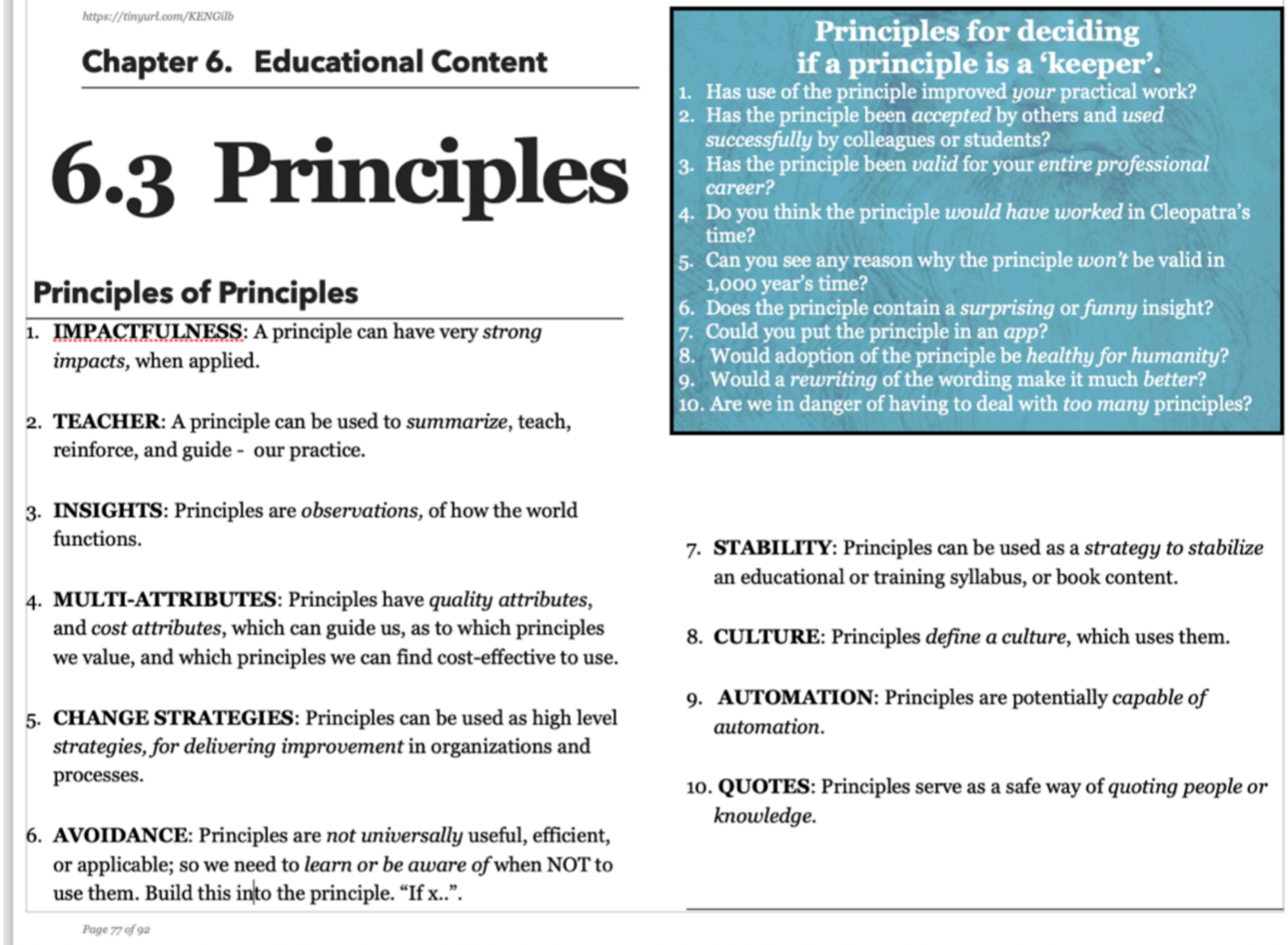


Figure 8.0
Source: http://2.bp.blogspot.com/-usweDeCuKVY/TijxxjAF7rI/AAAAAAAAARI/qtkko1qyHTU/s1600/PatternEvolutionMatrix_New.jpg

8.1


https://tinyurl.com/KENGilb

Chapter 6. Educational Content

6.3 Principles

Principles of Principles

1. **IMPACTFULNESS**: A principle can have very *strong impacts,* when applied.
2. **TEACHER**: A principle can be used to *summarize*, teach, reinforce, and guide - our practice.
3. **INSIGHTS**: Principles are *observations,* of how the world functions.
4. **MULTI-ATTRIBUTES**: Principles have *quality attributes*, and *cost attributes*, which can guide us, as to which principles we value, and which principles we can find cost-effective to use.
5. **CHANGE STRATEGIES**: Principles can be used as high level *strategies, for delivering improvement* in organizations and processes.
6. **AVOIDANCE**: Principles are *not universally* useful, efficient, or applicable; so we need to *learn or be aware of* when NOT to use them. Build this into the principle. "If x..".
7. **STABILITY**: Principles can be used as a *strategy to stabilize* an educational or training syllabus, or book content.
8. **CULTURE**: Principles *define a culture*, which uses them.
9. **AUTOMATION**: Principles are potentially *capable of automation*.
10. **QUOTES**: Principles serve as a safe way of *quoting people or knowledge.*

Page 77 of 92

Principles for deciding if a principle is a 'keeper'.

1. Has use of the principle improved *your* practical work?
2. Has the principle been *accepted* by others and *used successfully* by colleagues or students?
3. Has the principle been *valid* for your *entire professional career?*
4. Do you think the principle *would have worked* in Cleopatra's time?
5. Can you see any reason why the principle *won't* be valid in 1,000 year's time?
6. Does the principle contain a *surprising* or *funny* insight?
7. Could you put the principle in an *app*?
8. Would adoption of the principle be *healthy for humanity*?
9. Would a *rewriting* of the wording make it much *better*?
10. Are we in danger of having to deal with *too many* principles?

Figure 8.1. 'Principles of Principles'. Source KEN Eduneering, 6.3 [23]

We have introduced earlier, **Stakeholder Principles** (1.2), and **Stakeholder Engineering Principles**.(1.5)

Planguage is explained with over 100 Principles (10 in each of 10 Chapters) [4]. Each of my principles, I believe, is fairly **universal** (applies to many things), **eternal** (won't go out of style in your lifetime), and **powerful** (will really help you do your systems engineering work much better).

My principles for deciding if a principle is a useful pattern is in the blue box above ``**Principles for Deciding if a principle is a 'keeper'**.``

The *concept* is the thing, *not* the word

8.2 Concepts

"When I use a word", Humpty Dumpty said, in a rather scornful tone, "it means just what I choose it to mean—neither more nor less." "The question is", said Alice, "whether you can make words mean so many different things." "The question is", said Humpty Dumpty, "which is to be master—that's all." Lewis Carroll, Through the Looking Glass, *Chapter VI (Humpty Dumpty), 1871.*

Figure G2
Alice meets Humpty Dumpty[3].

Figure 8.2 A. Source [4, Glossary, Figure G2, by Permission]

A 'Concept' is an idea, defined by words, and possibly by diagrams. It is a very strong pattern, for synchronizing stakeholders, and for stored knowledge about stakeholders.
We 'take charge of a Concept' by creating and publishing, deep and powerful definitions, which bring us into the 'systems engineering culture', as opposed to many other informal uses of a term.
We assign 'Terms' to *point* to that concept definition, and Capitalize the Term to *demand* the use of our formal definition. We sometimes color and underline it, as a hot link to the concept definition. We (Planguage users) then become the master of the word.

Attribute **Concept *003**

An attribute is an observable characteristic of a system. Any specific system can be described by a set of past, present and desired attributes. There are four main categories of attribute:

- Performance: 'How Good the System Is'
- Function: 'What the System Does'
- Resource: 'What the System Costs'
- Design (or Architecture): 'The Means for delivering the System'

All attributes are qualified by Conditions, which describe the time, place and events under which the attributes exist.

> *Attribute:* "A characteristic of an item; for example, the item's color, size, or type."
>
> *Source:* Dictionary of Computing Terms, *IEEE 630-90.*

Notes:

1. Performance and resource attributes are scalar (described by a scale of measure). Function and design attributes are binary (either present or absent).
2. Attributes can be complex. They can be defined by a sub-set of elementary attributes.
3. An attribute may be described by any useful set of Planguage parameters.

Example:
Reliability: "The attribute tag name."
Ambition: High duration of operation. "*Summary of the target.*"
Scale: Hours of <uninterrupted service>. "*Defining the measure.*"
Goal [Next Release]: 6,000 hours. "The required target level for the attribute."
The tag (Reliability) and the parameters (Ambition, Scale and Goal) provide a systematic framework for defining and referring to a scalar attribute's components.

Synonyms: Characteristic *003; Property *003.
Related Concepts: Performance *434; Function *069; Resource *199; Design *047.

Figure 8.2 C. Source [4, Glossary]

Concepts, include various sub-patterns to help us understand and use them. Like 'Synonyms', and 'Example:'

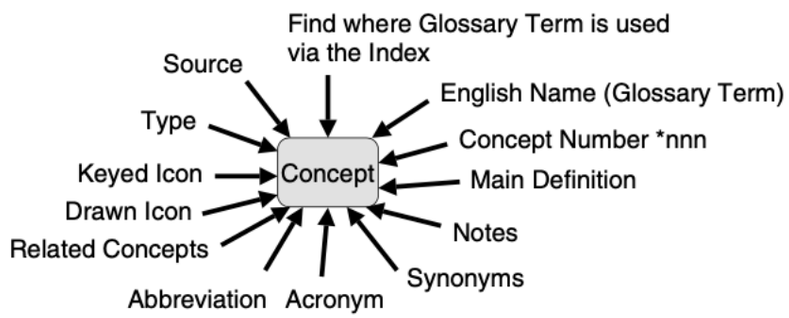


Figure 8.2 B Source [4, Glossary]

Any number and type of pointers can refer to the concept definition. This is a useful sub-pattern.

8.3 Rules

R9: **Clear**: Specifications should be 'clear enough to test' and 'unambiguous to their intended readers.'

R10: **Simple**: Complex specifications should be decomposed into a set of elementary, tagged specifications.

R11: **Fuzzy**: When any element of a specification is unclear then it shall be marked, for later clarification, by <fuzzy angle brackets>.

R12: **Comment**: Any text which is secondary to a specification, and where no defect in it could result in a costly problem later, must be clearly identified. It can be written in *italic text* statements, or headed by suitable warning (such as Note, Rationale or Comment), or written in "quotes," and/or moved to footnotes. Non-commentary specification shall be in plain text. *Italic* can be used for emphasis of single terms in non-commentary statements. Readers should be able visually, at a glance without decoding the contents, to distinguish between 'critical' and *'non-critical'* specification.

R13: **Source**: Specification statements shall contain information about their source of origin. Use the '<-' icon and state the source person and the date, or the source document with detailed statement reference.

R14: **Assumptions**: All known assumptions (and any relevant source(s) of any assumptions) should be explicitly stated.

The 'Assumption' Planguage parameter can be used for this purpose. But there are also a number of alternative ways, such as {Risk, Source, Impacts, Depends On, Comment, Authority, [Qualifiers], If}. In fact, any reasonable device, suitable for the purpose, will do.

Figure 8.3 A, A Sample of Rules [4], 1.4

Rules: Generic Rules for Technical and Management Specification. One Rule is that Rules for a topic should not exceed one page.

Rule **Concept *333**

A rule is any statement of a standard on how to write or carry out some part of a systems engineering or business process.

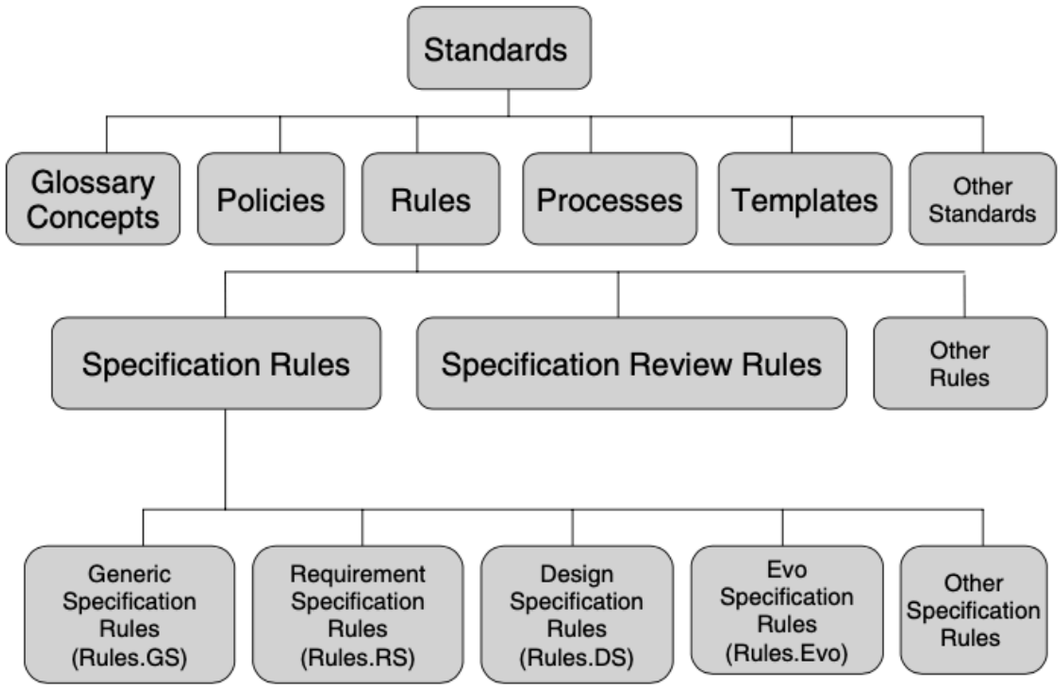


Figure G24
Rules as standards. Some of the different types of Specification Rules are shown.

Figure 8.3. Source CE Book, Glossary, [4], Fig. G24.
System Engineering Standards are patterns of desired professional practices. Like other patterns, they are based on hard won experience, after long practice, perhaps informal practice; before being written up and made obligatory.
Violation of Rules are called 'Specification Defects', and are detected by Quality Control and Reviews.

Rules as we define the concept (to avoid the many other interpretations of that word) are *strongly recommended* stakeholder systems engineering ***practices***. They have been developed because they are significant practices, and failure to follow the Rules, can lead to various types of faults, and failures. Rules are a **major pattern;** regulating the practice of a profession, and allowing it develop, as it gains experience.

8.5 Cases

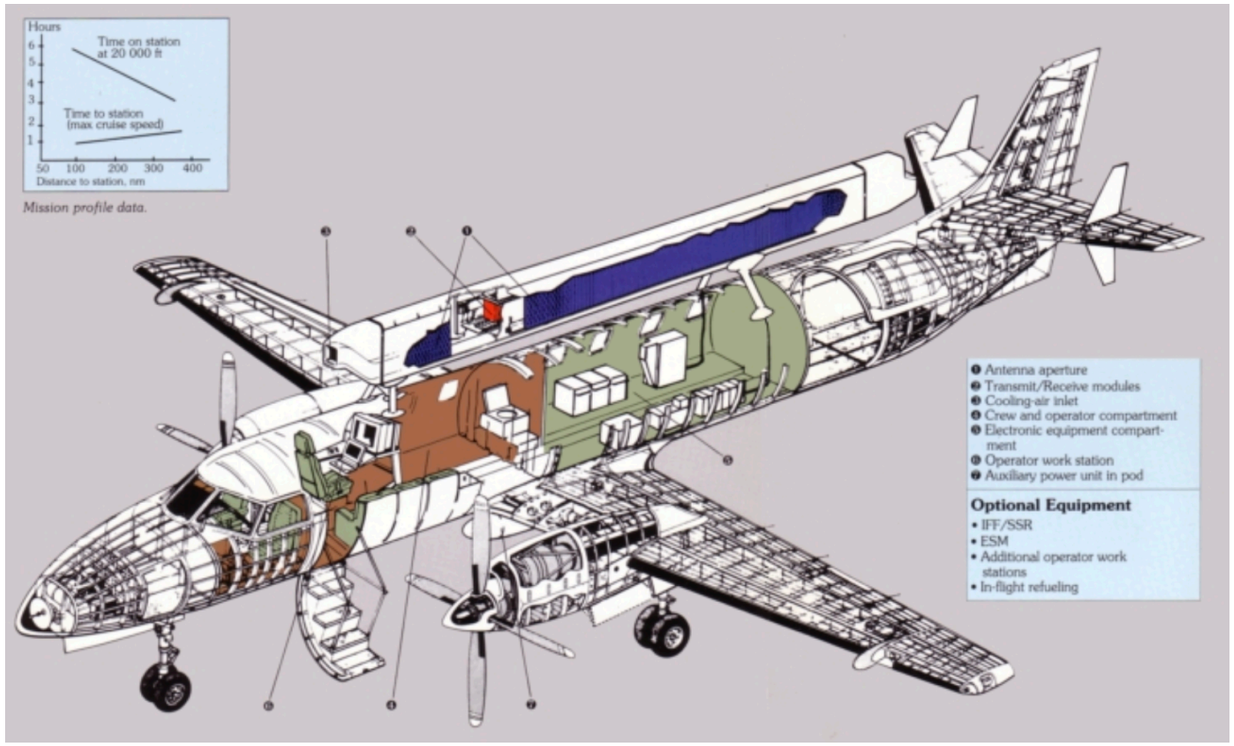


Figure 8.5 A. EriEye. Source: https://media.defenceindustrydaily.com/images/AIR_S100B_Argus_AEWC_Cutaway_lg.jpg

Case studies, and *real practical examples* (as opposed to made-up examples, and symbolic drawings) are a very important type of Pattern. Case Patterns are not oversimplified. They may be necessarily abbreviated (and even anonymized), but they tell a real verifiable story, of something that **really happened** and was really done. We get to see the 'ugly' detail, and can see *real stakeholders interacting with other real stakeholders*.

When I read papers, slides, books, see presentations and videos which do *not* give me realistic cases, I get *very suspicious* that the author has *no real experience*. I cannot trust them. I should not waste time listening or reading them. See other cases here [10, 11, 12, 35] and in the References; books like CE [4], VP [7A]. I have over 100 published personal cases.

EXAMPLE

Usability:
Ambition: Operator ease of learning & doing tasks under <all conditions> should be maximum possible ease & speed of performance with minimum training & minimum possibility of <unchecked error(s)>.

Usability.**Intuitiveness**:
Ambition: High probability that an operator will within a specified time from deciding the need to perform a specific task (without reference to handbooks or help facility) find a way to accomplish their desired task.
Scale: Percentage Probability that a defined [Individual Person: Default: Trained Operator] will find a way to perform a defined [Task Type] without reference to any written instructions, other than the help or guidance instructions offered by the immediate system screen (that is, no additional paper or on-line system reference information), within a defined [Time Period: Default: Within one second from deciding that it is necessary to perform the task].
Comment [Intuitiveness:Scale]: "I'm not sure if one second is acceptable or realistic, it's just a guess" <- MAB.
Meter: To be defined. Not crucial this 1st draft <- TG.
Past [System R]: 80%? <- LN.
Record [Mac User Interface]: 95%? <- TG.
Fail [**Trained Operator, Rare Tasks** [{<1/week, <1/year}]]: From 50% to 90%? <- MAB.
Goal [**Tasks Done** [<1/week (but more than 1/Month)]]: 99%? <- LN,
[**Tasks Done** [<1/year]]: 20%? <- JB,
[Turbulence, **Tasks Done** [<1/year]]: 10% ? <- TG.
========== User Defined Terms ==========
Trained Operator: Defined As: Command and Control Onboard Operator, who has been through approved training course of at least 200 hours duration.
Rare Tasks: Defined As: Types of tasks performed by an Onboard Operator less than once a week on average.
Tasks Done: Defined As: Distinct tasks carried out by Onboard Operator.
========== ========== ==========
Usability.**Intelligibility**:
Ambition: High ability for an operator to <correctly> interpret the meaning of given information.
Scale: Percentage Probability of <objectively correct> interpretation(s) of a defined [Set of <Inputs>] by a defined [Individual Person: Default: Trained Operator] within a defined [Time Period].
Meter [Acceptance]: Use about 10 Trained Operators, and use about 100 <representative sets of information per operator within 15 minutes?> - MAB.
Comment [Meter]: "Not sure if the 15 minutes are realistic" <- MAB.
Comment [Meter]: "This is a client & contract determined detail" <- MAB.
M1: Past: [XXX, 20 Trained Operators, 300 <data sets>, 30 minutes]: 99.0% <- Acceptance Test Report from XXX, MAB.
Record [XXX]: 99.0%. "None other than XXX known by me" <- MAB.
Fail [First Delivery Step]: 99.0%? <- MAB.
Fail [Acceptance]: 99.5%? <- MAB.
Goal [XXX, 20 Trained Operators, 300 <data sets>, 30 minutes]: 99.9% <- LN.

Figure 8.5 B ***My Ericsson Case Study of the Aircraft Project, 'EriEye' defining a tricky very-critical quantified quality, on the first morning of a successful real project. Source CE [4] Chapter 5 Scales of Measure.***

8.6 Processes

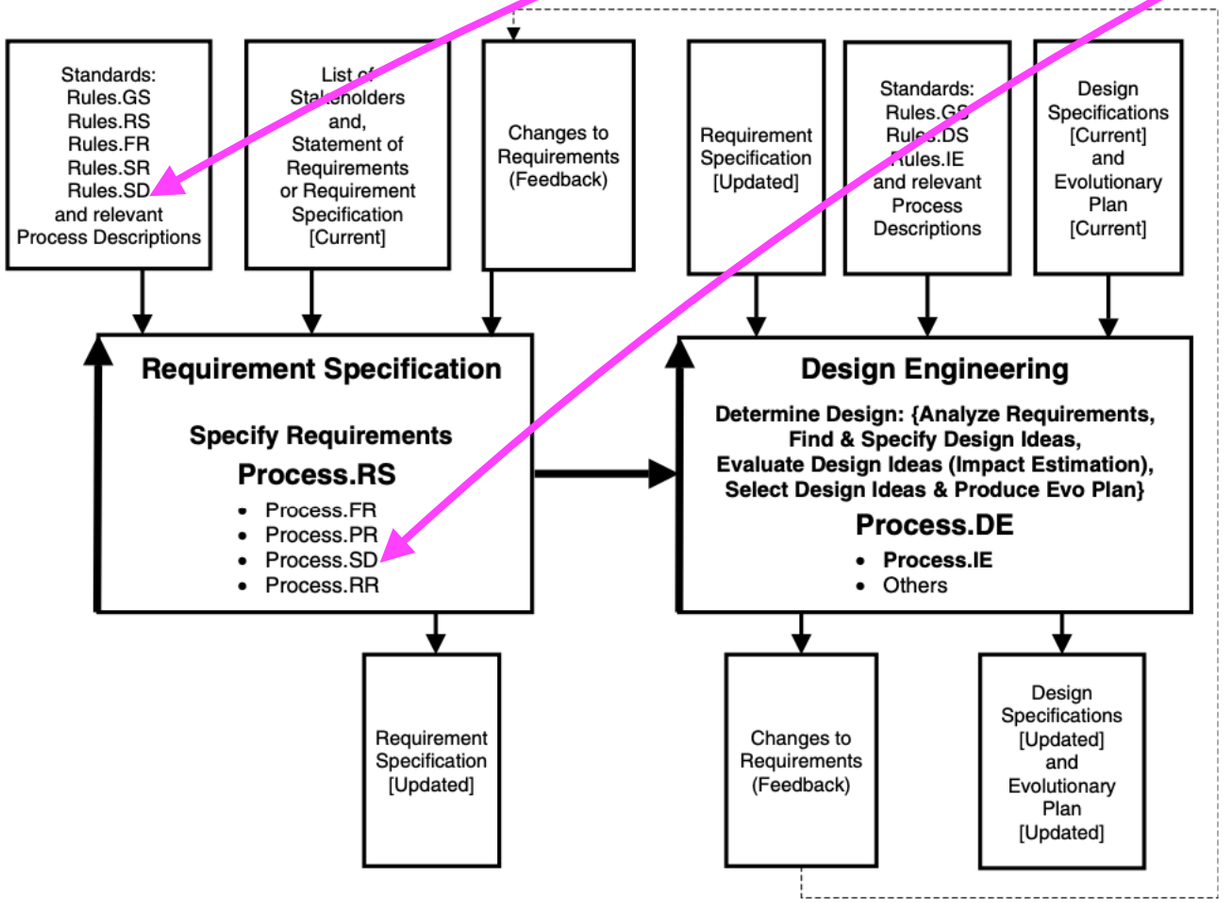


Figure 8.6 A. Some Systems Engineering Processes of Planguage [4]. Notice that a set of Rules for Scale Development is also part of the Patterns for Scale development.

Procedure for Scale Definition ('SD')

P1: Ensure that you have derived an elementary attribute (from a complex requirement), and that you are not trying to use a complex requirement, which needs decomposition into its elementary attributes. (Trying to find a single Scale for a complex (multi-Scale) requirement doesn't work well. It is usually the cause of trouble when people fail to find a suitable Scale.)

If you find you do indeed have a complex requirement, then decom- pose it and try to find Scales for its components. You might well find that further (second-level and more) decomposition is required!

P2: Ensure that the elementary attribute that you are developing a Scale for has a suitable tag and a Gist or Ambition parameter that adequately describes the concept in outline terms.

P3: Using the Gist or Ambition, analyze how a 'change' of degree in the scalar attribute level would be expressed. What would a user experience or perceive? For some examples, see Table 5.1, 'Examples of Scales of Measure'.

Sometimes you can keep things simple, and 'make do', by controlling the details at a higher level of abstraction:

- by deciding to use one dominant Scale only, and consciously ignoring the potential other scales.
- by aggregating several scales of measure to express one summary scale of measure.
- by defining a complex attribute as the 'set' of other Scales and definitions. P4: Specify the critical [time, place, event] qualifiers to express differ- ent benchmarks, constraints and target levels.
  P5: If there is no appropriate standard Meter (or test), start working on a Meter. Try to imagine a practical way to measure things along the Scale, or at least sketch one. Try thinking about any measures that are currently being carried out (this could even help you start developing ideas for scales of measure). Also, think about whether any current system could be modified, or have its settings changed, to perform additional measurement.

P6: Try out the Scale. Define some reference points from the past (benchmarks) and then, on the basis of benchmarks, specify future requirements (targets and constraints).

P7: Repeat this process until you are satisfied with the result. Try to get approval for your Scale from some of the stakeholders. Does it quantify what they really care about?

P8: Consider putting embedded parameters into the Scale definition. Rationale: To enable a Scale to be reused both within a project and in other projects.

Figure 8.6 B A Procedure for Scale Development (SD) [4] page 150, is part of the SD Process, which includes SD Rule, and SD Entry/Exit Conditions.

Intellectual work processes, patterns, are another way of storing experience and wisdom, including for the purpose of systematic improvement, and tailoring to other similar domains. As this diagram (A) overviews, the processes tie together various stakeholders. The external *stakeholders*, represented through the requirements, talk to the design engineers (internal *stakeholders*) for example.

8.7 Entry Exit

As a stakeholder engineering process, sends specifications from one stakeholder *(maybe a Customer, to a Business Analyst, then to a Systems Architect, then to a Project Manager)* to another; and they are **all under pressure** to deliver quickly, to meet deadlines; we need some mechanisms to make sure the sender has done a good enough job. This is a **pattern** known as Entry/Exit **Conditions**, which are examined (part of Quality Control) by a corresponding Entry/Exit **Process**.

Like other good patterns these E/X Conditions are developed based on experience, senior wisdom, and root cause analysis.

These E/X patterns are one specialist part, of the organizational **standards** for processing work.

A powerful example of the effect of using these patterns, is in the Terzakis Intel case: (4.2.2.1). The incoming requirement defect rates, finally exited, were **50x better** (10/ 0.22) than when they started trying to get exit approval.

Do you believe in carefully researched published client facts? I am not just telling you this is a 'nice idea': I am reporting what the stakeholder using my methods measured and published.

Exit Conditions

Exit Conditions are used to evaluate if the task is reliably and economically completed. They specify the safe and economic conditions for exit from a process to a 'next' process. Exit conditions are also built on experience from previous releases to the next work process.

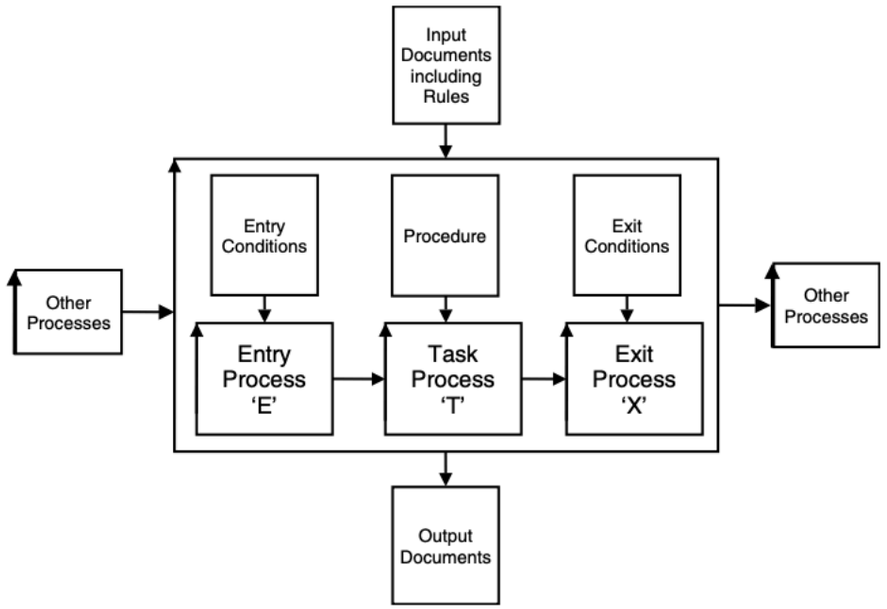


Figure 1.4
Diagram of a simple process showing its sub-processes and its relationship to other processes and documents. The input documents for each process include the rules, the entry conditions, the procedure and the exit conditions. The diagram also shows how the 'ETX' concept for a process is derived. A rectangle is the symbol for a 'written document.' A rectangle with arrow is a 'process' symbol. An example of such a process could be 'Requirement Specification.'

*Figure 8.7 A **Entry Exit Conditions, Patterns.** Source [4] Figure 1.4.*

Entry Conditions for Scale Definition Procedure.

E1: The Generic Entry Conditions apply. Input documentation includes contracts, marketing plans, product plans and the requirement specification. The relevant rules should also be available: the generic specification rules (Rules.GS), the requirement specification rules (Rules.RS), the rules for scalar requirement specification (Rules.SR) and, the rules for scale definition (Rules.SD).

E2: Do not enter this procedure if company files or standards already have adequate quantification devices. Preferably use the existing Scales and Meters found in the standards' libraries.

Figure 8.8 B, Sample real Entry Conditions, for Scale Definition (SD) procedure. Source [4] 5.5, page 149-150

Are you up for Extreme Depth?

8.8.0 The STARS ODM [1]

Are you up for 509 pages of Stakeholder advice?

The STARS ODM document [1], INFORMAL TECHNICAL REPORT

For
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

TASK: PV03 CDRL: A025 14 June 1996

Organization Domain Modeling (ODM) Guidebook Version 2.0 is the most in-depth set of **stakeholder process advice** I am aware of.

It is overwhelming. But it is 'US Air Force', and they have a big stakeholder playground. Some of you might be in that league, and find my advice too simplistic for your level of complexity.

The report is free and speaks for itself, in **509 dense pages**.

I made contact with 2 of the authors and they report they are thinking about doing something more with the ideas.

I decided I would put some samples of their thinking, mainly in the form of diagrams here, and the reader can decide for themselves if they want to access more detail.

I feel good that I see many similarities between my own ideas of stakeholder engineering, and theirs.

The reader will notice that I prefer to quantify objectives more, and to estimate effects more numerically (IET). Using methods I published a decade before they wrote this (1988, Principles of Software Engineering Management). But sometimes simple '+ 0 -' are suitable enough for purpose. These guys are very systematic.

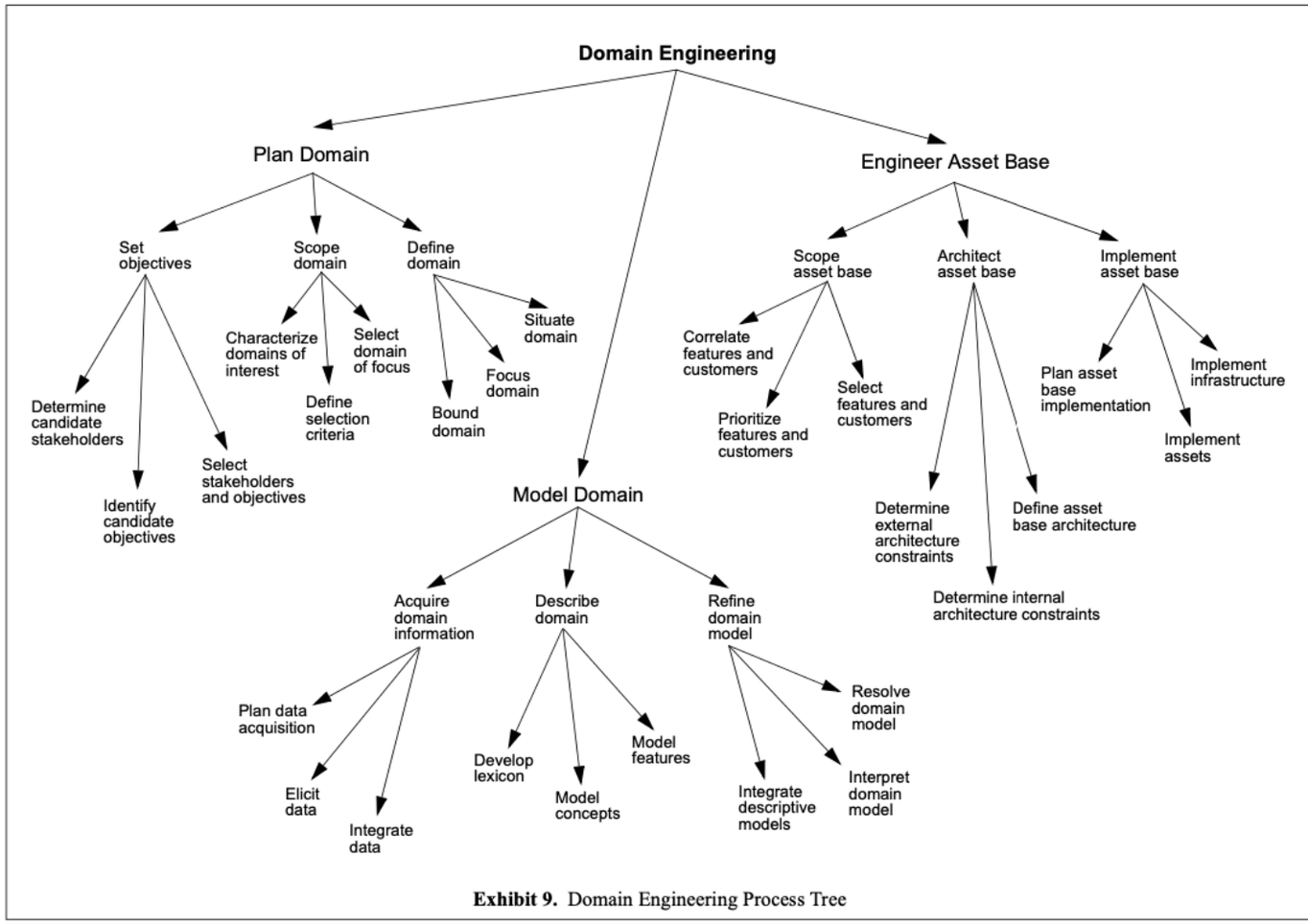


Exhibit 9. Domain Engineering Process Tree

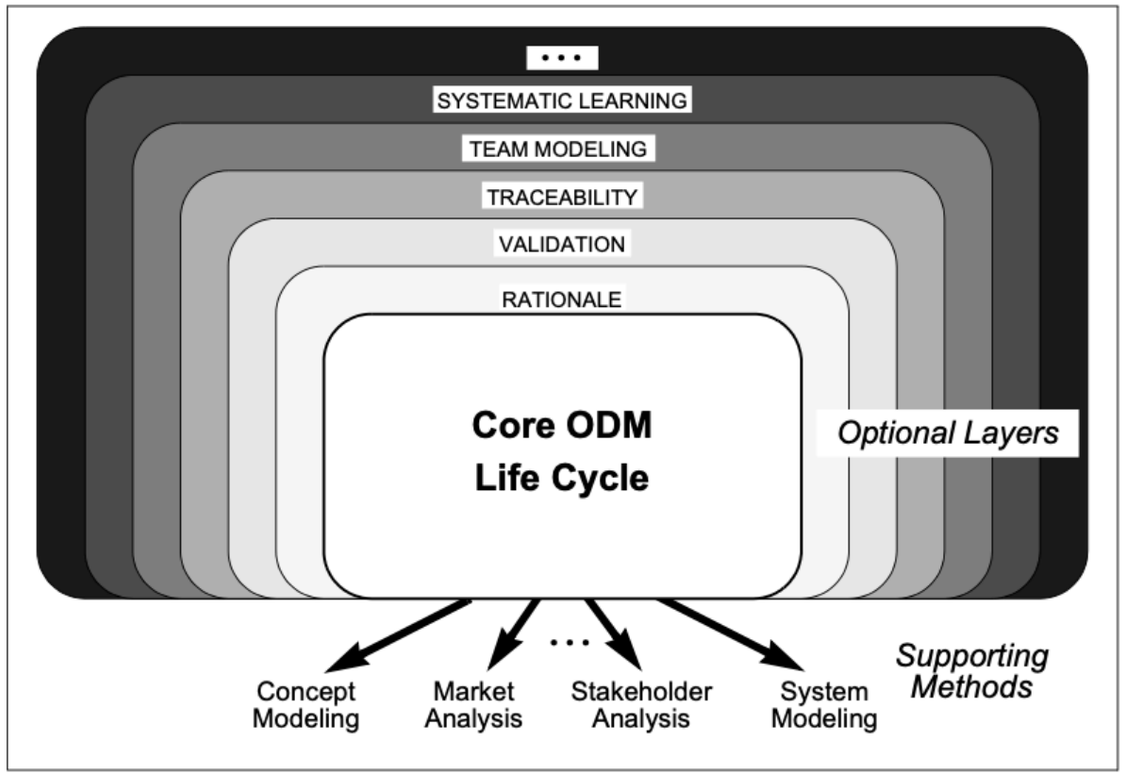


Exhibit 10. ODM Layers and Supporting Methods

8.8.1 The STARS ODM [1]

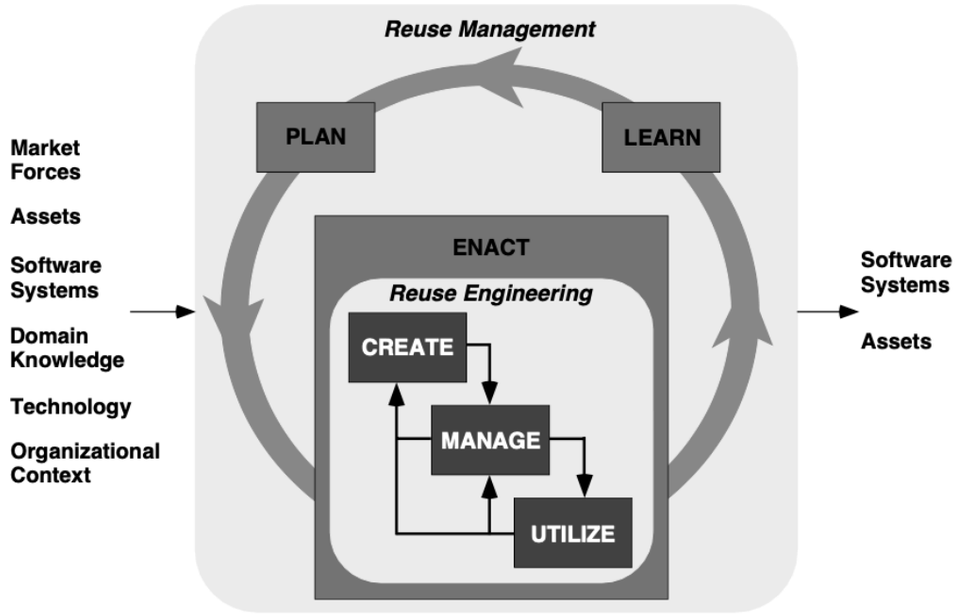


Exhibit 12. STARS Conceptual Framework for Reuse Processes (CFRP)

‰ Identify stakeholder roles and candidates for each role

Ask questions based on critical roles to identify stakeholders who fill these roles. Exhibit 21 shows a starter list of questions, based on roles that are often critical to a domain engineering project.

Who are core or key customers of the domain engineering project?
Who is funding the project?
Who defines success for the project? (e.g., funder, the funder's customers?)
Who will be performing the project?
Who are experts on application areas within the organization's focus?
Who are potential customers for assets to be developed by the project?
Who is providing technology for the project?
Who might be interested in seeing the project *not* succeed?

Exhibit 21. Starter List of Stakeholder Roles

Other questions to ask include the following:

- Based on organization relations (e.g., external market relations, internal customer-supplier relationships), which organizations are potential asset developers, asset base managers, utilizers of assets? Who are stakeholders within these organizations?
- Look at previous reuse efforts within the project's organizations. Results of previous domain engineering projects, both formal and informal, need to be considered early in the *Determine Candidate Stakeholders* task. These results can lead to the identification of stakeholders who were involved in the previous projects.

Knowledge evolves in all domains, and domain engineering methods must provide support for this evolution over time. Nevertheless, there is a distinction between incremental learning and evolution and that suggested by domain engineering. We view domain engineering as a window of parallelism within a larger evolutionary pattern that might begin with a slower building up of knowledge through cases and examples. A domain model represents a "snapshot" derived by parallel analysis of multiple instances. Once a domain model has been established the incremental process begins again, enriching or possibly "decaying" the domain model structure. Eventually, the model may need to be substantively revised; or the domain may be absorbed into boundaries of neighboring domains or due to the obsolescence of the defining technology for the domain.

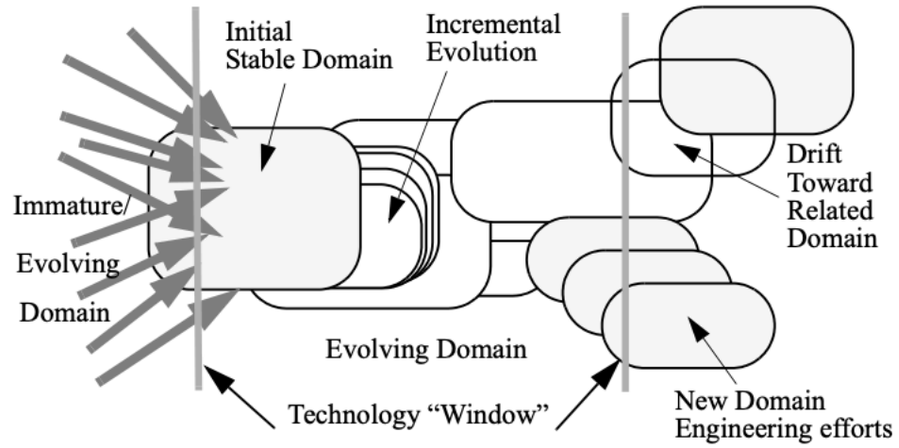


Exhibit 15. Domain Evolution

This means it is important to pick reasonable domains for application of domain engineering as an "accelerating function," and critical to establish realistic expectations for the results of the domain engineering effort. Domain engineering could not have stabilized window management systems much before the basic metaphors settled into common practice, nor could domain engineering help to salvage, say, a model of punch-card printing routines in the modern era. Biggerstaff's Rule of 3 says that a good domain for reuse should have at least three legacy systems (sufficient maturity) and at least three new systems should be anticipated (sufficient economic justification). Biggerstaff also introduced the notion of the "technology window" within which reuse and (more specifically for our purposes) domain engineering efforts would be cost-effective.

Stakeholders		Roles		
		DE	Asset Utilizers-Internal	Asset Utilizers-External
Key Stakeholders	Webonautics		X-3	
	Maskara		X-3	
	Persona DE Team	X-2		
Other Stakeholders	JSoft			X-4
	MMI			X-4

Exhibit 25. *Example Excerpt:* Revised Project Stakeholders/Roles Matrix

8.8.2 The STARS ODM [1]

Stakeholders Interests			Enablers	Barriers
Key Stakeholders	C. Connell	Greater collaboration between Webonautics and MasKara	Production of single library of interest to both	Separate projects
	F. Morrow	Give Persona reputation that attracts good technical people	Enter future-oriented technically viable area	Follow a fleeting fad
	Webonautics	'Real work' as opposed to management's 'flavor of the week'	Use of established methods (OO)	Overemphasis on non-technical reuse issues
Key Stakeholders	MasKara	Reduce production costs to be more competitive on government contracts	Systematic reuse	
		'Real work' as opposed to management's 'flavor of the week'	Use of established methods (Ada generics)	Overemphasis on non-technical reuse issues
	J. Perricone	Partner with MMI to get big CDC contract	PC look and feel compatibility	Solution bound to a single platform
			Vertical niche - medical	
Other Stakeholders	MMI	Get contract with CDC	PC-compatible (most popular system at CDC)	Insufficient attention to security

Exhibit 26. *Example Excerpt:* Project Stakeholders Interests Table

Project Objectives	Domain Selection Criteria										
	Use in Web and CORBA applications	**Cuts across platforms**	**Visible in popular systems**	**Not in COTS System**	**When in COTS, not accessible via API**	**Impacts turnkey systems**	**Of use in planned systems**	**No reliance on technology soon obsolete**	**Knowledge available**	**Commonality across systems**	**Little performance impact from extra functionality**
Provide capabilities not currently available from COTS APIs	X APIs as candidates for ORB or Web	0	X	+	0	0	+	+	0	0	0

Exhibit 39. *Example Excerpt:* Project Objectives/Criteria Matrix

Project Stakeholders (Selected & Prioritized)		Project Objectives (Selected & Prioritized)			
		Produce platform independent library for both Web and CORBA	Assist in production of systems with functionality familiar from popular systems	Provide capability not currently available from COTS systems through API	Close partnership contract with MMI and CDC
Key Stakeholders	C. Connell	+	0	0	+
	F. Morrow	+	+	+	+
	Webonautics	+	0	+	0
	Maskara	+	0	0	0
	Persona	0	0	+	+
	J. Perricone	-	0	0	+
Other Stakeholders	MMI	-	0	0	0
	AFDSO	0	0	0	-
	ODM	0	0	0	0

Exhibit 31. *Example Excerpt:* Project Stakeholders/Objectives Summary Matrix

Candidate Objectives	Stakeholder Interests											
	Connell			Morro	Webonautics		MasKara		Persona		Perri	MMI
	Collaboration	Gov't Security Standards	Combat graphics perceptions	Attract best technical people	Wide audience	Distrib system	Consistency with gov't work	Reduce production costs	High-profile hit	Make a splash to attract talent	Partner with MMI	Get contract with CDC
Produce platform independent library for WEB and CORBA	!	0	+	+	!	!	+	!	-	+	X	X
Library support for vertical market medical law	0 0	+ 0	- -	0 0	- -	- -	- -	- -	+ +	+ +	! X	! X
Make all persona products conform to PC look and feel standards	0	0	-	-	+	-	X	X	-	-	+	+
Provide capability not currently available from COTS systems through API	0	0	!	+	+	0	0	0	0	+	0	0

Exhibit 29. *Example Excerpt*: Project Objectives/Stakeholders Interests Matrix

Concept Glossary

I would appreciate hearing your reactions, suggestions, difficulties understanding me, objections, references and links. After each chapter!
tom@Gilb.com

PS
Feel free to send to friends, and post link on social media, or to use in whole or part in training, lectures, blogs, your own books, papers, slides with main link
https://tinyurl.com/StakeholderBook

2021 Full Glossary,

http://www.gilb.com/DL386.

See ref. [36] for more

Core Stakeholder Concept Glossary [24]

Stakeholder: anything *within* a system boundary, or externally able to *influence* a defined system, from which we can derive our *own* system requirements, in order to manage successful long-term operation of ‘our’ defined system.

See 1.0 Overview for definitions of many terms intros definition

Stakeholder: (Short Version) : **Any potential requirement source.**

Any: anything, incl social, legal, motivational, hardware, software, data, environments. Any *system* element or relation, even outside our local system boundaries.

Potential: we do not yet know, without analysis and engineering, if the potential requirement can be accepted by our current effort, as feasible, economic, cost-effective and prioritized.

Requirement: any performance attribute (incl. all qualities), constraints, resource budgets, or deadlines, which we will need to consider in our project’s design and implementation or operation.

Source: any person, group, inanimate source such as law, contract, policy, rule, plan, or natural or political or social phenomena we can access for analysis of potential requirements.

Specification Maturity: various types, detail, approval, and commitment for a specification object, such as a Stakeholder, or a Requirement. From a simple Tag to detail, reviewed, quality controlled, approved, committed, implemented.

Stakeholder Engineering: the analysis of stakeholders, and their multiple changing requirements as a multi-dimensional quantified dynamic discipline.

Systems Engineering: a systematic discipline for large and complex systems, which tries to consider absolutely all factors and technologies that can affect a successful outcome of the project and the lifetime operations of a system.

For much larger concept definition sources see [4, 24] and

http://concepts.gilb.com/dl985

2021 14 June version