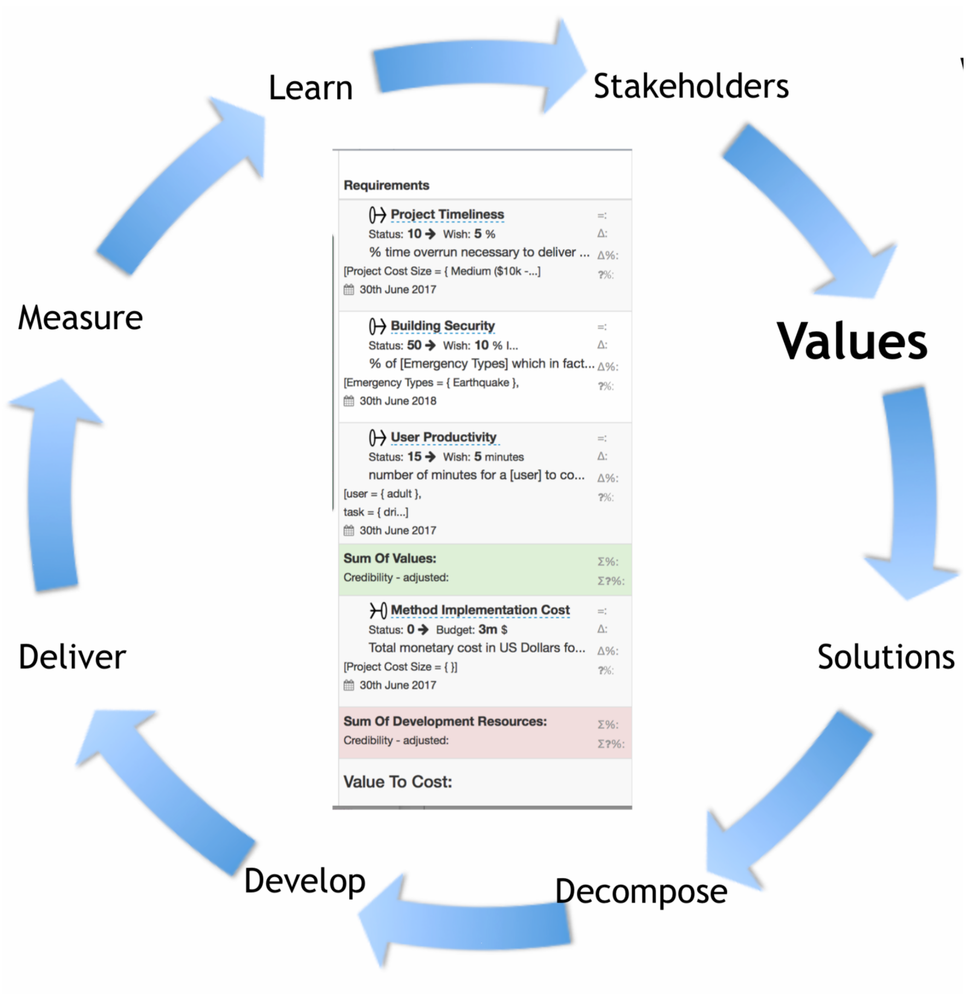
Value Design

How to get the Qualities you need to win and succeed, using advanced design thinking



Requirements

Project Timeliness
Status: **10** → Wish: **5** %
% time overrun necessary to deliver ...
[Project Cost Size = { Medium ($10k -...]
30th June 2017
=: Δ: Δ%: ?%:

Building Security
Status: **50** → Wish: **10** % I...
% of [Emergency Types] which in fact...
[Emergency Types = { Earthquake },
30th June 2018
=: Δ: Δ%: ?%:

User Productivity
Status: **15** → Wish: **5** minutes
number of minutes for a [user] to co...
[user = { adult },
task = { dri...]
30th June 2017
=: Δ: Δ%: ?%:

Sum Of Values: Σ%:
Credibility - adjusted: Σ?%:

Method Implementation Cost
Status: **0** → Budget: **3m** $
Total monetary cost in US Dollars fo...
[Project Cost Size = { }]
30th June 2017
=: Δ: Δ%: ?%:

Sum Of Development Resources: Σ%:
Credibility - adjusted: Σ?%:

Value To Cost:

By
Tom Gilb
Copyright 2019-2021
tom@Gilb.com,
www.Gilb.com
Version 9 September 2019,
LeanPub edit 250721

Title Page 'Value Design' Overview.[1]

This book is for professionals who need to go beyond conventional design thinking methods, in order to get far better results; to get high values and qualities, at low resource costs, for non-trivial systems.

Here are some subjects we cover:

- Being exceptionally clear about design requirements
- Being very systematic and objective about the multiple effects, on qualities and costs of *all* design ideas
- Systematic evaluation of uncertainty, risks, dubious sources
- Decomposition of design components into smaller, implementable, components
- Numeric *prioritization of implementation*, of design components
- *Presentation* of design options: value for money wrt risks
- Digital tools for design
- *Creative* Design using discipline and clarity
- Scaling up to national and international scales, and complexity
- Value-Based Decision making
- Value Delivery
- All systems: products, services, IT systems, organizations, political planning, NGO planning, health planning, ... anything
- Systematic *weakness analysis* with conventional design methods.
- *Stakeholders:* a more realistic basis for design than 'users' alone. UX becomes SX.
- The Architecture Week: top level overview, and a quick start for results.
- Dynamic Design to Requirements
- Organizing the Design Process

[1] The Logic of Design: Design Process Principles. Tom Gilb, 2015, Paper. http://www.gilb.com/dl857 (reference 3)

Introduction to Value Design

Environment ->   Stakeholders ->   Requirements ->   Design ->   Result
<—————————- <————————-   <—————————   <—————————————

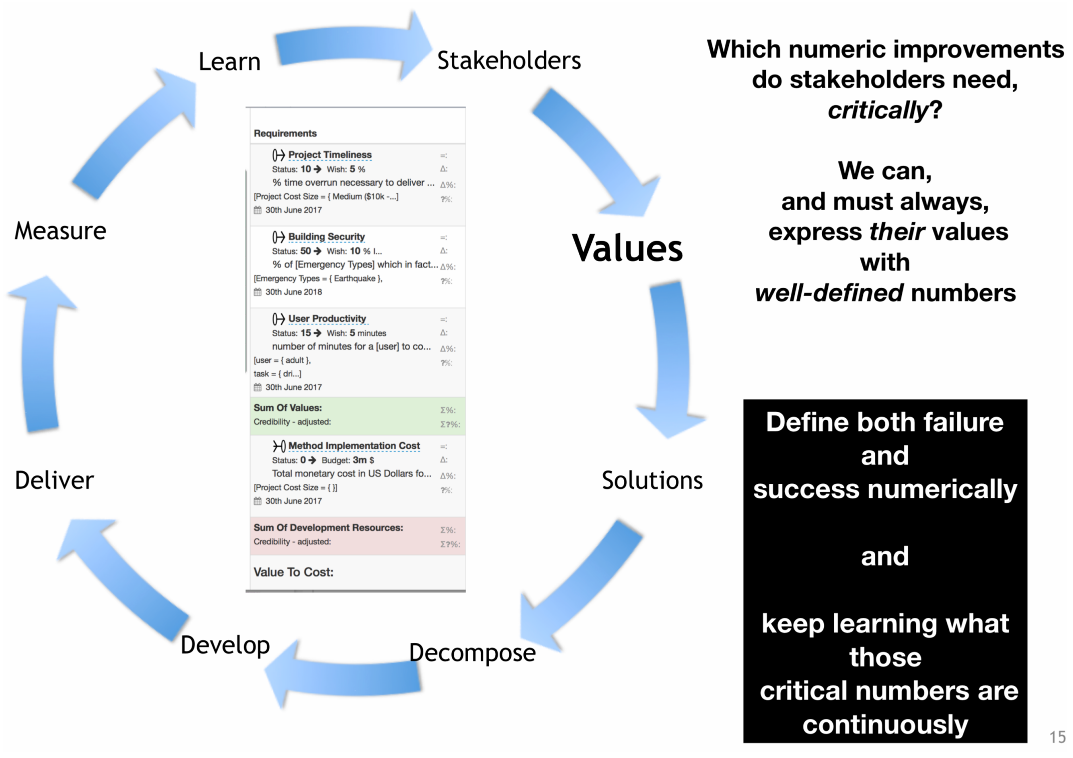
FIGURE: IDENTIFYING YOUR ENVIRONMENT HELPS IDENTIFY YOUR STAKEHOLDERS

The design process can begin *anywhere*, including with a ‘bad design result’. But there is an ideal logical sequence, if you are starting from scratch.

We can start with a description of the environment. What, in the environment, is clearly related to our design problem? What in the environment, is potentially or possibly related; what is clearly outside scope or influence?

This *environment* step helps us to identify our critical stakeholders.

The ‘stakeholder identification’ helps us identify *their* Critical values, and the *values* drive the design process (Solutions, Decompose, Develop, Deliver, Measure, and Learn)



Requirements

Project Timeliness
Status: 10 → Wish: 5 %
% time overrun necessary to deliver ...
[Project Cost Size = { Medium ($10k -...]
30th June 2017

Building Security
Status: 50 → Wish: 10 % l...
% of [Emergency Types] which in fact...
[Emergency Types = { Earthquake },
30th June 2018

User Productivity
Status: 15 → Wish: 5 minutes
number of minutes for a [user] to co...
[user = { adult },
task = { dri...]
30th June 2017

Sum Of Values:
Credibility - adjusted:

Method Implementation Cost
Status: 0 → Budget: 3m $
Total monetary cost in US Dollars fo...
[Project Cost Size = { }]
30th June 2017

Sum Of Development Resources:
Credibility - adjusted:

Value To Cost:

Which numeric improvements do stakeholders need, *critically*?

We can, and must always, express *their* values with *well-defined* numbers

Define both failure and success numerically and keep learning what those critical numbers are continuously

FIGURE: THE 'EVO' CYCLE. STAKEHOLDER VALUES DRIVE THE DESIGN PROCESS AND KEEP THE DESIGN PROCESS ON TRACK TO DELIVERING REAL VALUE TO STAKEHOLDERS IN PRACTICE.

The book in a sound bite :

"Design must deliver value efficiently."

The book in a page: The Evo[2] Process.

Value Requirements (12,VR)[3]

1. Clarify your environment: critical-stakeholders' territory
2. Identify your critical stakeholders: the ones that can make or break your project
3. Identify their critical values
4. Quantify and clarify your critical values: what degree of values do you expect the design to deliver to stakeholders
5. Identify design constraints: legality, political, cultural, policy, other plans

[2] Evo meaning Evolutionary, or Efficient Value Optimization is the name of our total system project management process, of which Design is a key component. See more in book references.

[3] We will summarize, but not detail, the requirements process here. So VR is a reference to the Value Requirements book drafted Primo July 2019.

6. Identify design resource-limitations: time, money, operational costs for example.

Top-Level Design

7. The Project Startup Week: an architecture overview
8. Identification and prioritization of top-level architecture
9. Decomposition of top-level architecture into design components.

Value Delivery Cycle[4]

10. The Evo Value delivery steps (about a week, or 2% of total project budget)
    1. Select your highest priority value, and the most-critical scale-parameter attributes.
    2. Find a design component which will deliver the most value-for-resources to your priority requirement.
    3. Ready the design component for delivery: integration to the existing system.
    4. Deliver the design component to the real system
    5. Measure the results (values and costs) of the design increment
    6. If results are negative, attempt design improvement, and redeliver.
    7. If results ok then repeat this value delivery cycle until ‘done’.

Project Completion

11. When all value requirements are reached, or when critical resources are used up. Stop.

[4] We will not focus on project management itself, but mainly on the incremental delivery aspects which involve measuring design, design costs and adjusting design, on a step by step basis. Dynamic Design to Cost.

TABLE OF CONTENTS

‘Value Requirements’, a summary of the VR book

The details of how to gather and specify Value Design-Requirements are in other sources (VR, VP, VE, CE) and the reader is recommended to study one of them for depth. But *this* book is focussed on **Design**, so we will only give a short summary of ‘*design requirements’* here.

From the point of view of the designer, we need the design requirements to be specified, as *clearly and completely* as possible. One misleading or missing requirement can ruin the entire design. As the car bumper-sticker, I saw in London once, said, with good British understatement. “One Nuclear Bomb, can ruin your whole day”.

We have here a process of determining design requirements, which improves your chances of getting your critical requirements, into much better shape (clarity, detail) than less-formal processes.

Design gets a good start if the design requirements are high quality.

Terzakis at Intel used our requirements processes and reported 230% overall project engineering productivity process improvement (VP, VR). That is evidence of a good start and the effectiveness of these methods. Try to find credible evidence like that from any other methods.

Environments

Your design environment is a very small grain of sand on the beach. If you can narrow in on *your* environment, two things happen.

You eliminate too-general consideration of irrelevant and less-important considerations. Eliminate over-design.

You become aware of stakeholders, and critical needs that you might otherwise have forgotten, at early stages of design. These stakeholder-, and stakeholder-need, oversights can lead to delays, failures, extra costs, and less-successful systems.

A simple example: Our Environment: EU, Voting Citizens, all EU Laws and Regulations.

Stakeholders

Stakeholders decide design requirements. We design to please stakeholders. ‘Stakeholders’ is a very broad category encompassing all sources of design requirements. In particular the stakeholder category is much broader than the popular ‘user’ and ‘customer’ category.

In particular, and many international standards miss this point, there are non-human stakeholder, or ‘design requirements sources’. Such as laws, regulations, plans, policies, standards, customs and budgets.

It is essential that the designer is aware of all *critical* sources of *critical* requirements: because by definition a *failed critical requirement* can kill your entire project.

There are the following generic categories of stakeholders:
1. Individuals, like a user, customer
2. Groups of People, like Steering Committee, Trade Union
3. Victims: like neighbors, poor families, protesters
4. Weak Victims: like diseased, employees, trainees
5. Antagonists: like enemies, competitors, criminals
6. Defenders of Weak Victims: like Councils, NGOs, Free Legal Services, Climate `Change Campaigners
7. Non-Human Requirement Sources: Like laws, regulations, standards, policies, contracts.

You should normally be able to identify 30 to 100 potentially-critical stakeholders, just by brainstorming, or using previous experience.

What you fail to specify and analyze early, if it is critical, and you do a bad job with it, provides problems which will confront you later (loss of market, loss of reputation, inability to give aid effectively).

The *art* here, is reducing *late discovery* to a minimum. It pays off to do so because of high downstream costs.

Stakeholder Values

Then, for each stakeholder, try to identify their needs.

This simple ‘Critical Need’ question will decide if their needs are ‘critical’ for **us**.

If we totally fail to deliver this value,
or respect this constraint on time,
might our project possibly be seriously or completely damaged?

If not, then it remains a valid stakeholder *need*, but it cannot become a design *requirement* for *your* design process.

We cannot make everybody happy about everything. We all have limited resources, and thus we need to prioritize our design requirements, to do good where it *counts*.

Design Requirements

Before you can safely ask the ‘critical need’ question (above), you should probably clarify the requirement well enough, so that you can better discover, that it is *really clearly critical*.

So for example:
if the need is formulated as ‘Safety’ then you are going to always get a false positive: ‘yes safety is critical’

But if you formulate the requirement more carefully, like
No more injured or dead during construction, than maximum in last 10 years, for any class of construction worker.

Then that better-formulated requirement can be accepted as critical, but the badly-defined need ‘safety’ will not be used, and will not trigger a knee-jerk positive reaction.

Value Requirements

1. ‘Value’ requirements are the values of the *stakeholders*, which *we have decided* are worth designing to achieve.
   Many of the stakeholder values will be rejected by us as *design* requirements, or because their values are not worth it, or not critical for our top-level critical purposes, or not possible to deliver, or not prioritized high enough.
2. Value requirements include all qualities concepts (*how well* a system functions) and other values like ‘saving resources’, ‘reducing annual maintenance costs’.
3. Value Requirements are always ‘**variables**’. We can always use adjectives like *better* or *worse*.

The consequence is that we can always define Value Requirements **quantitatively**, ie **numerically**: and therefore very **clearly** for design purpose, and delivery-control purposes.

We *can* specify a numeric value requirement like this:

We want to detect 95% of hackers within 5 seconds.

Which is a lot clearer than the vague 'we want high security'.

But, at a **more-advanced level** we are going to specify our critical requirements more like this:

<u>Security</u>:

Scale: % [Coping] with [Attackers] within a [Time Limit].

Past 2019, 5%, Coping = Detect, Attackers = Hackers, Time Limit = 5 Seconds

Tolerable 2025, 25%, Coping = Aprehend, Attackers = Hackers+Thieves, Time Limit = 1 hour.
Goal 2025, 45%, Coping = Aprehend, Attackers = Hackers+Thieves, Time Limit = 1 hour.
Goal 2030, 95% Coping = Aprehend+Charge, Attackers = Hackers+Thieves+Employees, Time Limit = 1 hour.

This is what I call defining a design requirement both 'quantitatively', and with 'structure'.

The '**structure**' above is:

1. All requirements and designs have a **<u>Unique Name Tag ('Security')</u>**, so we can reuse that set of statements which the Tag references.
2. We quantify our 'quality value' with one-single well-defined and well-structured **Scale**.
   That Scale has a number of [Scale Parameters], so that we can identify and specify any interesting conditions, attributes, and dimensions of our Environment.
3. **Past** statement is a *benchmark*, not an actual requirement. Benchmarks give us background information, to help us understand our proposed requirements better. "Are we actually *planning* to be much better?"
4. **Tolerable**: is a required 'worst acceptable case', a scalar constraint, for the defined set of [Scale Parameters] defined in the Tolerable statement itself. The design must at least be sufficient for this level, and these [Conditions].
5. **Goal**: statement is a project-committed, target level, we are required to aim for, with our design. Notice we can have any useful set of such requirement levels for different times and conditions.

These requirement specification methods, are part of a planning (and Design) language I invented[5], and we continue to develop, called **Planguage**. It is detailed in my writings and those of Kai Gilb (**<u>www.Gilb.com</u>**). See books: 'Value Requirements' (VR, VP, VE, CE and more).

Constraint Requirements

Constraint Requirements are *not* primarily what we are *aiming **for***, which is to 'deliver stakeholder value'.

Constraint Requirements impose defined and specified *limits* on our designs.

[5] the decades-long development of Planguage, has had many individual and client contributors. Very close, for decades, have been Kai Gilb and Lindsey Brodie (CE, PoSEM, PhD, Middlesex University)

Constraint Requirements are of three main types:

Binary Constraints: you must do something, or you must *not* do something else.

Scalar Constraints: like the Tolerable-level specification just above ('deliver at least this much value')

Resource Constraints: typically resource limitations with stated Budgets for time, people, money: in the short term (building an improved system), or the long term (operating an improved system through its lifetime).

Constraint Requirements will force the designer to 'Design to Cost'. That means to stay within the budgets specified. This implies that the designer is conscious of the resources needed for a design.

It also implies that the designer can become aware of the **efficiency** of their design: the Values Delivered in relation to the resources needed.

This was a summary of the Value Requirements book, 2019. (VR)

And now on to our main subject, Value Design.

What is 'Design'?

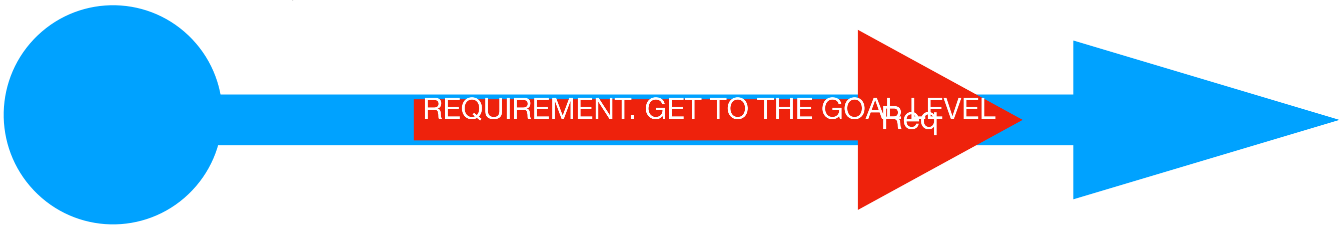
A design is
a suggested solution to the problem specified
by the design requirements.
It might not turn out to be valid, or as efficient as alternative solutions.

A perfect valid design specification would

- deliver all value requirement levels within their deadlines
- While not exceeding any resource budgets
- And not exceeding any other specified constraints

FIGURE: A VALUE REQUIREMENT USUALLY IMPLIES IMPROVING A VALUE IN THE DIRECTION OF A

REQUIREMENT. GET TO THE GOAL LEVEL



TARGET OR CONSTRAINT LEVEL. FROM LEFT SIDE OF RED ARROW, TO TIP OF RED ARROW. ALONG A DEFINED VALUE SCALE (BLUE ARROW)

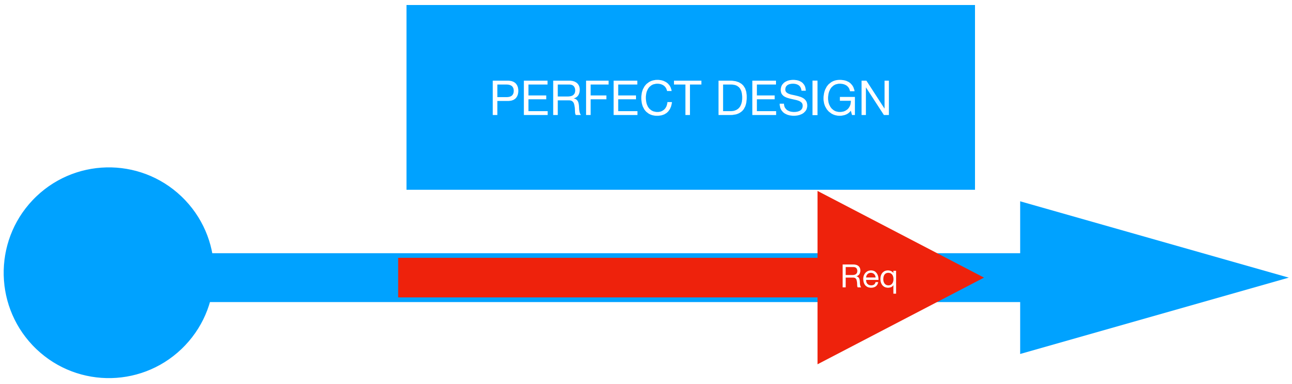


FIGURE: A 'PERFECT' DESIGN (THE RECTANGLE IS OUR SYMBOL FOR A DESIGN) WHEN IMPLEMENTED, WOUND DELIVER THE REQUIRED LEVEL OF VALUE, WITHIN ITS DEADLINE. BUT DESIGN CONSTRAINTS MIGHT CAUSE THE DESIGN 'HYPOTHESIS' TO BE INVALIDATED.

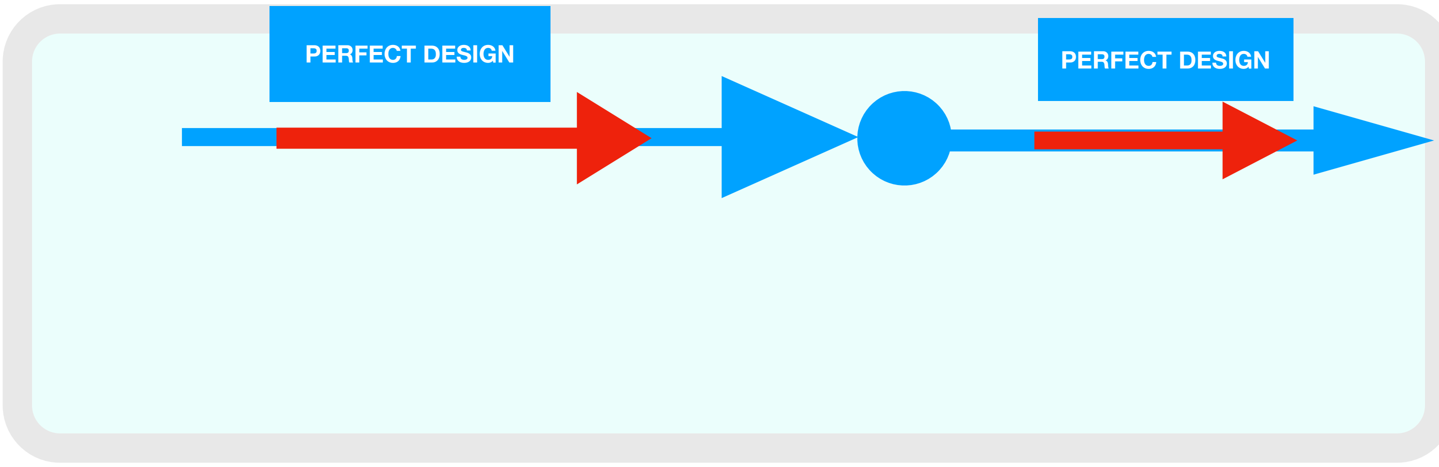


FIGURE: IF THE DESIGN DOES NOT EXCEED LIMITED RESOURCES (LEFT RED ARROW), AND STAYS WITHIN ALL OTHER CONSTRAINTS (GREY LINES), IT IS VALID.

1. Basics for Valid Design: ways to checkout a design.

Validating that the design will deliver enough value.

The Value Requirement levels specified (Goal 2030, 95%... in example above) is the driving force of the design process. We read it, and *one* of our *design problems* is 'defined' by that statement.

We need to ask ourselves, **which design will get me to 95% by the year 2030** (including the other [conditions] detail, included in our problem specification)?

If we have no design ideas at all, and neither does anyone else we communicate with; then we might possibly have a problem spec, for which there is no 'known' design solution.

If we some ideas at all, then the process needs to get more-disciplined.

More Design Discipline: some questions to ask about your designs.

1. Has anyone ever actually reached such a level (95%) with any design at all?
2. Has anyone ever reached such a level within the deadline (2030, X years from when we start)?
3. Has anyone ever reached such a level using the design 'hypothesis' (= we *think* it might work) which we have suggested?
4. Have people *regularly* reached such a level, using our design hypothesis?
5. How costly would it be to try out (pilot, prototypes, experiment) with the design hypothesis and see how it works for us, before committing to it, on a longer-term, larger-scale basis?
6. Is the design idea so *generically specified*, that it in fact could include both winning designs, and *failures*?
7. Would a sub-supplier agree to a no-cure no-pay contract, based on actual value delivered, with '95%' (the Goal) or better, giving a double bonus?

The answers to these questions will guide you into pursuing or abandoning the design idea, and if necessary looking for other better ideas.

Understanding that we probably have the required resources to use the design. Can we afford it? Are there cheaper ideas?

Let's assume you have one or more design candidates that are acceptable, in terms of the questions above. And let us assume all candidates look roughly as good as any other.

So they *might* deliver the value levels you require. But can you afford them? And is any option much cheaper or faster than the others?

We can ask the following questions about the design options, in order to pick a 'resource winner':

1. is the design specified in enough detail, that we can hope to estimate costs roughly ? (Order of magnitude, or maximum). Vague ideas have a very broad 'cost range'.
2. Do we know any resource information (time, people, money) at all about any previous uses of our design options, by anyone anywhere?
3. Can we get a sub-supplier to give us a fixed price, fixed-delivery-time contract, for the design options?

These questions will help you point to a likely, cost-effective ('efficient design') candidate.

In some situations, that might be enough to go ahead and try the promising-designs out.

In other situations you would be gambling, too much of someone else's money and lives; so you might like some even-more-advanced design-methods for those cases.

Making sure that the design does not violate any other specified constraints.

It is not enough to have a cost-effective design.

It would be nice if it were ‘legal’ too!

There are a number of other considerations, we call them ‘design constraints’, that your design needs to satisfy.

You can ignore them, and no one will bother you for a while. But recent Facebook and Google publicity is a reminder of the price of arrogance.

Design Constraints need to be specified and acknowledged, as valid. Management needs to lead and ask questions. Have we identified the constraints? Are the designers aware that we need to respect them 100%, or we are in legal/financial/reputation trouble?

Stakeholder analysis will give you a lead to many design constraints. Hard experience reminds you of others. Did your organization learn anything specific, from previous projects that had problems? Is this experience embedded in reusable checklists of design constraints that apply in most of your projects? Try to lay your hands on that checklist!

Well here is a generic beginners list, but you need lists for your specific domain!

DESIGN CONSTRAINT CHECKLIST
1. Take another look at your 50+ stakeholder list. Do they have any potentially critical design constraints? Would they accept any design (even if it killed Whales, with plastic bags?)
2. Do not limit yourself to what the stakeholders would demand up front. What would they say if later confronted with that fact that ‘you chose plastic over wood’ ? Would you prefer to surprise them later?
3. Contract conditions
4. Laws: council, national international
5. Regulations: health, finance, transport safety, building, employment
6. Organizational Policy, Partner Policy
7. Local culture, environmentalist, equality, consensus, anti-corruption
8. Public Opinion, media, respect

2. How <u>not</u> to evaluate a design.

Everybody is using that design

Popularity is not a good reason for you to use a design. Your only valid reason is that the design will credibly deliver the value required. The popular design might be good enough for other people's value levels.

But maybe *your* value requirements are different, and *your* design needs to be different.

Maybe they have all chosen a bad design, because it was well marketed, and they are getting bad results; or getting results that they do no realize are so bad, since they have no clear numeric idea of the level they need, or the level they got.

The design is effective for X-ility

The design is the best for security, or some value of interest.

This might, or might not be true. Is there serious evidence to prove that? Or is it just a subjective opinion, by someone who does not even know what 'serious evidence' means?

But even if it is, the best, there are other considerations before we make a design commitment.
1. What negative side-effects does it have on other critical values, like Usability?
2. What does it cost, now and later, in money and time? Does it steal all the budget?
3. Does it violate any of our constraints? Is it legal in all our markets?

In other words, such an argument ('best') is insufficient to lead to adoption of the design.

The Powers That Be have asked for a particular design.

The 'Power' might be a big potential customer, your boss, your biggest client, the contract, even the 'law'.

However, that does not automatically mean you blindly and unquestioningly use that design.

Why not?
1. The design might be obsolete in many ways, replaced by better ones
2. The design might simply be their way of telling you what their values are, but not really intended as an unconditional demand from them to you.
3. The the design might have negative side effects that they are unaware of, and would not accept.
4. The design might cost far more sooner or later, than the Powers know about and would accept.
5. The Power behind the Power might have changed: a net Chief Technical Officer in power, might like to be consulted on their ineffective predecessors bad decisions.

We can treat the design request with respect. But we need, to be 'design responsible', to be the professional designers they employ; to shed light on it, both for the Power, and for our own reputation and responsibility's sake.

Consequently we need to do the following Design Request analysis.

1. Are there any clearly superior alternatives now available, which might have been overlooked or have recently emerged?
2. What are the measurable expected side effects of this design request on any and all value requirements?

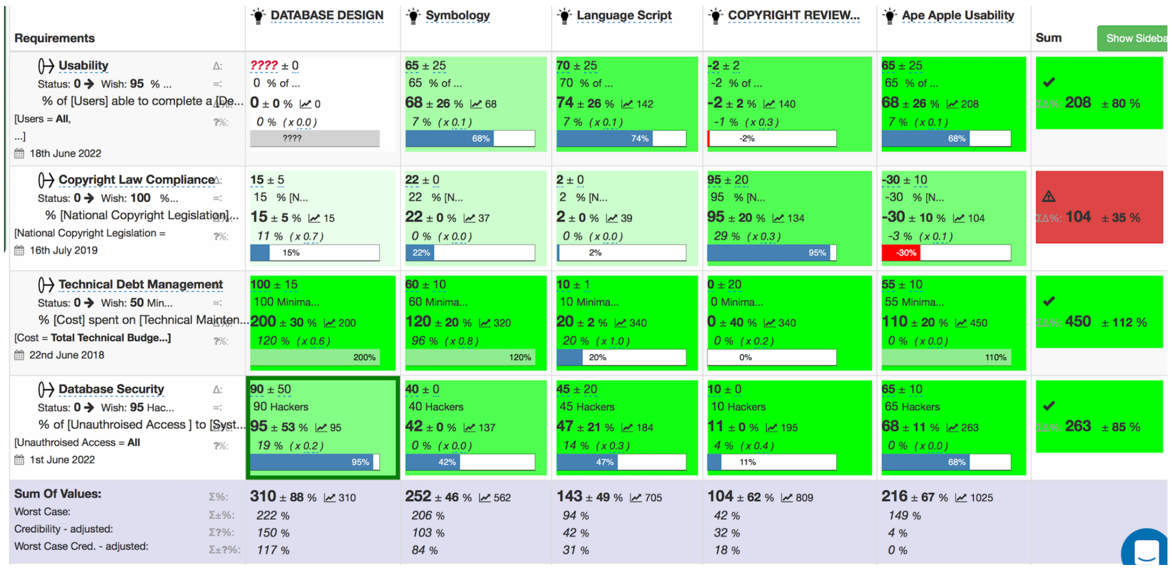
3. What is the estimated consumption of our budgeted resources (time, money, maintenance costs, conversion costs, training, ...)
4. Are there for the design request any potential violations of legal or other constraints, in any potential area of deployment?

A written analysis should be submitted to the Power, with a recommendation about what to do, and requesting their written answer. Put the monkey on their back, or the monkey is still on yours.

We always use that design

Well that is fairly safe. A conservative option. But the times they are a changing, and a designer is responsible to their client to be aware of interesting new options, and to at least point them out.

A table comparison of the options would be one way to explain the differences, and to give the



Requirements		DATABASE DESIGN	Symbology	Language Script	COPYRIGHT REVIEW...	Ape Apple Usability	Sum	Show Sideba
Usability Status: **0** → Wish: **95** % ... % of [Users] able to complete a [De... [Users = **All**, ...] 18th June 2022	Δ: =: Δ%: ?%:	???? ± 0 0 % of ... **0** ± **0** % 📈 0 *0 % (x 0.0)* ????	65 ± 25 65 % of ... **68** ± **26** % 📈 68 *7 % (x 0.1)* 68%	70 ± 25 70 % of ... **74** ± **26** % 📈 142 *7 % (x 0.1)* 74%	-2 ± 2 -2 % of ... **-2** ± **2** % 📈 140 *-1 % (x 0.3)* -2%	65 ± 25 65 % of ... **68** ± **26** % 📈 208 *7 % (x 0.1)* 68%	✓ ΣΔ%: **208** ± **80** %	
Copyright Law Compliance Status: **0** → Wish: **100** %... % [National Copyright Legislation]... [National Copyright Legislation = 16th July 2019	Δ: =: Δ%: ?%:	15 ± 5 15 % [N... **15** ± **5** % 📈 15 *11 % (x 0.7)* 15%	22 ± 0 22 % [N... **22** ± **0** % 📈 37 *0 % (x 0.0)* 22%	2 ± 0 2 % [N... **2** ± **0** % 📈 39 *0 % (x 0.0)* 2%	95 ± 20 95 % [N... **95** ± **20** % 📈 134 *29 % (x 0.3)* 95%	-30 ± 10 -30 % [N... **-30** ± **10** % 📈 104 *-3 % (x 0.1)* -30%	⚠ ΣΔ%: **104** ± **35** %	
Technical Debt Management Status: **0** → Wish: **50** Min... % [Cost] spent on [Technical Mainten... [Cost = **Total Technical Budge...]** 22nd June 2018	Δ: =: Δ%: ?%:	100 ± 15 100 Minima... **200** ± **30** % 📈 200 *120 % (x 0.6)* 200%	60 ± 10 60 Minima... **120** ± **20** % 📈 320 *96 % (x 0.8)* 120%	10 ± 1 10 Minima... **20** ± **2** % 📈 340 *20 % (x 1.0)* 20%	0 ± 20 0 Minima... **0** ± **40** % 📈 340 *0 % (x 0.2)* 0%	55 ± 10 55 Minima... **110** ± **20** % 📈 450 *0 % (x 0.0)* 110%	✓ ΣΔ%: **450** ± **112** %	
Database Security Status: **0** → Wish: **95** Hac... % of [Unauthroised Access] to [Syst... [Unauthroised Access = **All** 1st June 2022	Δ: =: Δ%: ?%:	90 ± 50 90 Hackers **95** ± **53** % 📈 95 *19 % (x 0.2)* 95%	40 ± 0 40 Hackers **42** ± **0** % 📈 137 *0 % (x 0.0)* 42%	45 ± 20 45 Hackers **47** ± **21** % 📈 184 *14 % (x 0.3)* 47%	10 ± 0 10 Hackers **11** ± **0** % 📈 195 *4 % (x 0.4)* 11%	65 ± 10 65 Hackers **68** ± **11** % 📈 263 *0 % (x 0.0)* 68%	✓ ΣΔ%: **263** ± **85** %	
Sum Of Values: Worst Case: Credibility - adjusted: Worst Case Cred. - adjusted:	Σ%: Σ±%: Σ?%: Σ±?%:	**310** ± **88** % 📈 310 *222 %* *150 %* *117 %*	**252** ± **46** % 📈 562 *206 %* *103 %* *84 %*	**143** ± **49** % 📈 705 *94 %* *42 %* *31 %*	**104** ± **62** % 📈 809 *42 %* *32 %* *18 %*	**216** ± **67** % 📈 1025 *149 %* *4 %* *0 %*		

Powers a real and responsible choice.

FIGURE: A VALUE DECISION TABLE. DESIGNS ARE TOP COLUMNS ('DATABASE DESIGN') AND THEY ARE RATED AGAINST 4 VALUES, AS TO % ABILITY TO MEET THE VALUE WISH ON TIME.

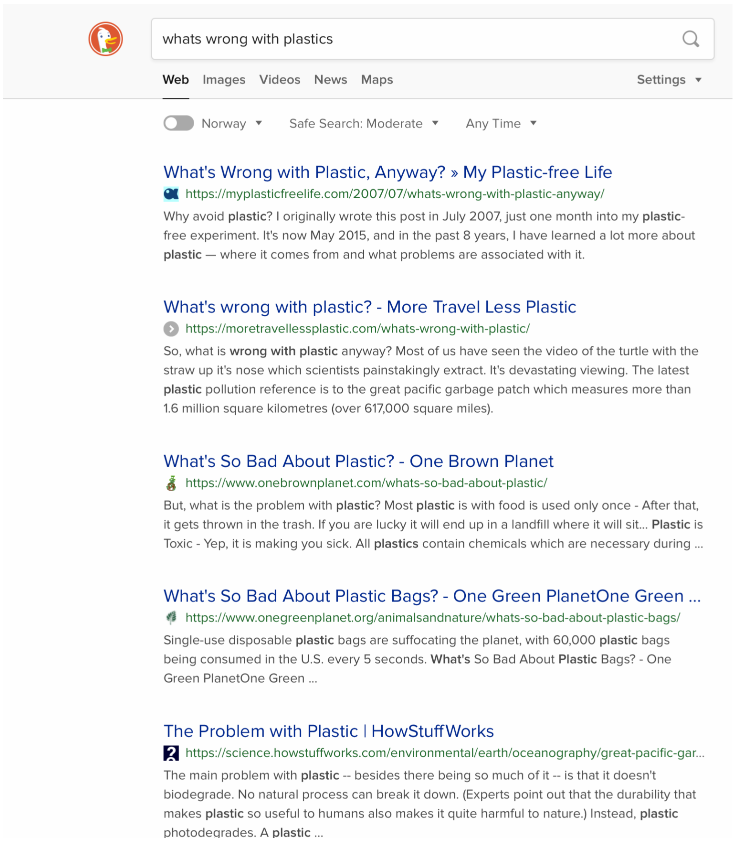
WHICH DESIGN DO YOU THINK IS THE BEST OVERALL OPTION ? LOOK AT THE 'SUM OF VALUES' NUMBER. BUT CONSIDER UNKNOWNS ('????') <- TECHNOSCOPES.

More about this design evaluation tool later in the book.

We do not know of any other design

There is, you should safely assume, always some better design out there. If you cannot think of near alternatives even, then you can also assume your design competence is low.

To avoid being found out as incompetent, you need to find someone who has better ideas. The simplest and cheapest way to do that initially is an internet search.
Look for criticisms of the 'one design' you have, which will probably contain suggestions for better designs.


whats wrong with plastics

Web Images Videos News Maps Settings

Norway Safe Search: Moderate Any Time

What's Wrong with Plastic, Anyway? » My Plastic-free Life
https://myplasticfreelife.com/2007/07/whats-wrong-with-plastic-anyway/
Why avoid **plastic**? I originally wrote this post in July 2007, just one month into my **plastic**-free experiment. It's now May 2015, and in the past 8 years, I have learned a lot more about **plastic** — where it comes from and what problems are associated with it.

What's wrong with plastic? - More Travel Less Plastic
https://moretravellessplastic.com/whats-wrong-with-plastic/
So, what is **wrong with plastic** anyway? Most of us have seen the video of the turtle with the straw up it's nose which scientists painstakingly extract. It's devastating viewing. The latest **plastic** pollution reference is to the great pacific garbage patch which measures more than 1.6 million square kilometres (over 617,000 square miles).

What's So Bad About Plastic? - One Brown Planet
https://www.onebrownplanet.com/whats-so-bad-about-plastic/
But, what is the problem with **plastic**? Most **plastic** is with food is used only once - After that, it gets thrown in the trash. If you are lucky it will end up in a landfill where it will sit... **Plastic** is Toxic - Yep, it is making you sick. All **plastics** contain chemicals which are necessary during ...

What's So Bad About Plastic Bags? - One Green PlanetOne Green ...
https://www.onegreenplanet.org/animalsandnature/whats-so-bad-about-plastic-bags/
Single-use disposable **plastic** bags are suffocating the planet, with 60,000 **plastic** bags being consumed in the U.S. every 5 seconds. **What's** So Bad About **Plastic** Bags? - One Green PlanetOne Green ...

The Problem with Plastic | HowStuffWorks
https://science.howstuffworks.com/environmental/earth/oceanography/great-pacific-gar...
The main problem with **plastic** -- besides there being so much of it -- is that it doesn't biodegrade. No natural process can break it down. (Experts point out that the durability that makes **plastic** so useful to humans also makes it quite harmful to nature.) Instead, **plastic** photodegrades. A **plastic** ...

Try the search key words: “what's wrong with xxxxx'

FIGURE: A SEARCH FOR ALTERNATIVE IDEAS.

The supplier recommends this design

Sometimes suppliers have really valid reasons for recommending certain products or services. In which case we should be able to ask questions and get good answers:

1. Exactly why are you recommending this design to us?
2. Are there better alternatives, if we pay more, from you or others?
3. Can you give us some references from people who have used your recommendation
4. Are there any weaknesses with your recommendation that you do not think we care about ?
5. What would you recommend if we said, we do not care about the initial cost as long as we get lifetime good quality from it?
6. Are you willing to give a total refund guarantee if we are unhappy with the design?
7. How many years have you been using that design?
8. Have you recommended this because we asked for a low bidder fixed price?

You and I know that a supplier, might have unspoken reasons for recommending a design. And for not telling us about better alternatives. Forgive them for that. You might act the same in their shoes.

But there are many ways of finding out if there are some designs you might actually like better, even at a price, or maybe even a lower price.

Our architect has specified this design

There are many kinds of 'architects'. Some have proper qualifications and ethics. Some do not.

The architects are a type of supplier, and deserve the same analysis as for suppliers above.

Prioritize: to decide sequence of activation.

Procedure: a specified sequence of activities for a defined purpose.

Process: a continuous, repetitive procedure with a possible ending when complete.

Quality: How Well a function functions. Often ending in '-ility'

Requirement: a stakeholder-desired future system state, which can be tested for presence, or measured for degree: but which might be impossible to deliver in practice.

Resource: any attribute which might be consumed, might be limited, and might be needed to build or maintain a system. Money, time, people, dominate, but many other resource concepts are potentially useful, such as image, qualities, functionality, space.

Risk: a risk is something that can go wrong. An opportunity is by contrast, something that can 'go right', get better. (+140819 tg)

Risk Dimensions: The risk dimensions are all critical *values*, and all critical *resource* limitations. The term 'Critical' by definition, is a value or cost area, which is negative, to the potential extreme of total system failure, as a result (+ 140819 tg)

Rules: a standard in Planguage which specified the recommended way to do, or not do, a specification of any kind. Failure to follow a rules is classified as a specification defect.

Scale (of Measure): a Parameter which defines a Value or Cost scale of measure, for reuse and reference when specifying Benchmarks, Scalar Constraints, and Targets. It does NOT specify a measurement process, that is for the Meter or Test parameter

Scale Parameter: a dimension, announced in [Square Brackets] in the middle of a Scale specification. It is defined using a {set of Conditions}. This device permits quite detailed Modelling of a system, and allows decomposition of problems so that ctitical Conditions can be prioritized. Example: [Sex]

Scale Parameter Conditions: a set of named conditions which belong to a defined Scale Parameter. Example [Sex] = {Male, Female, Other, Unspecified, Unknown, Multiple}.

Source: the named origin: a person, group, stakeholder, document, or URL of some immediately-previous specifications in a Parameter Specification. The purpose is to enable QC, give credibility, lend authority.

Spec, Specification: a written planning item in Planguage: Requirements, Designs, Analysis, Project Plans, presentations.

Specification Object: a set of Planguage Parameter statements, comprising a meaningful unit of informations, typically a requirement, a design, or sets of these.

Specification Owner: a person (or group) which has undertaken responsibility, by name, for the update and maintenance of a specification object, such as a requirement, a design, or a table.

Stakeholder: an entity; human, organizational, or document, from which we can derive needs, demands, resource limits, constraints, and any form of information, which can be acknowledged as our potential project requirements, and specified formally and clearly as a requirement. A 'requirement source'.

Status: a numeric update of the incremental progress of a Scale Level as we incremental deliver a system design components and measure progress towards our requirement levels.

Standards: best accepted practices for developing and maintaining systems. These include, Rules, Procedures, Exit Levels, Concept Definitions, Templates, Scales of measures, and even App conventions.

Target: a level of Value that we are aiming to reach. It includes Wish, Goal, Stretch.

Trend: a Background Benchmark level, which estimates the future of that level. Useful for pointing our Value degradation, or potential competitor future levels of Performance.

Use Case: a written graphic description of how a system element might be used in practice. In Planguage it can be covered by using an appropriate Scale Parameter. Example: [Uses] : {Register, Delete, Update}.

User: a person who personally and physically interacts with a system.

User Story: a requirement statement in the format: Stakeholder + Requirement + Justification. This is roughly at the level of an Ambition Level, and can replace Ambition Level as a starting point for formulating a more detailed Planguage requirement.

ValPlan: ValPlan.net is the URL of an App released for sale May 2019 by Gilb International AS. It is based on Planguage and the Competitive Engineering book.

Value: value is perceived stakeholder

Value Analyst: analyzes stakeholder needs, and priorities, and selects critical, or possibly critical, needs and specified them as requirements, at least at the 'Wish' level (potential Goal requirement).

Value Architect: A person or team, who sits at the Apex of the system, and synchronizes all ongoing efforts in order to get maximum necessary value for available resources. Manages to top critical values, and the top level design architecture.

Value Contracting: this contracting can be done at both internal levels and external suppliers, and basically motivate suppliers to deliver value and get rewarded for it.[35]

[35] Value Planning, Chapter 8 Delegation Outsourcing Contracting
https://www.dropbox.com/sh/eubo1zkvybl2q8k/AAD6cUUIOqco0aTPK2OZx6gua?dl=0

Value Design Builders: people and organizations who build, physically, logically or organizationally, any design component or related activity.

Value Designer: a generic (all possible design areas) designer (or team) who undertakes to identify possible design components to reach a Value Requirement level, on time. To research them as to all side-effects and costs, documenting such facts in the design object and corresponding Value Tables. The Value Designer might hand over exploration of a design idea to a Specialist.

Value Director: the person, or group, responsible for focusing on the Value Delivery, and reporting to a steering committee or Board about the plans and accomplishments to date in Value Delivery.

Value Engineering Specialist: a designer with a narrow speciality (usability, security, performance, organizational improvement, AI) who is updated on the state of the art, and has a good international network of people and sources to find good specialist designs.

Value Policy: this is the written policy that gives clear guidance to the Value process, from organizational management. Perhaps Chief Technical Officer level.[36]

Value Process Manager: a person or team responsible for getting a best possible value stream flowing from the other people involved. Sort of like old project manager, except they are focussed on the Values/Costs numbers, not building stuff. They

[36] a considerable body of suggested Policies (over 100) is in every sub-chapter of Value Planning book. https://www.gilb.com/store/2W2zCX6z

allocate resources (money, time), and assign people to specialist tasks.

Value Quality Control: these people carry out Specification Quality Control of specifications, to make sure the Defects Per Page is economically low enough before Exit to any other process. They are also responsible for measuring value levels and costs after incremental implementation. They will check that designs are in fact implemented as specified by suppliers for Exiting to integration delivery.[37]

Value Suppliers/Sub-contractors: internal and external to the organization people or organizations who undertake a specific responsibility, for construction, implementation, organizing, designing, or any other activity needed to deliver value.

[37]Value Planning Chapter 10 Quality Management
https://www.dropbox.com/sh/vjwybhqfxrvctk7/AAAdabECBSo5x-tSOI85R-1da?dl=0

Edit notes for Value Design book. All comments to tom@gilb.com
Start 260719, complete draft 1 of ms. 060819 19:38

070819 added sound remark to each chapter

090919 added ref 3 Logic of Design, corrected page 39 delivery to deliver

250721 LeanPub edition.
Edit cover
Edit 2019 books in refs to Leanpub