Planguage (Expansion of the SGEP)

Turning ambiguous goals into measurable outcomes through evidence, not promises

Tom Gilb

2026-01-18T09:00:00Z

How to Use this Booklet

This reference serves different needs. Choose your path:

For Quick Introduction (15-20 minutes)

→ Executive Summary
→ Planguage in 60 Seconds
→ Value Planning: Deliver → Measure → Adapt
→ Impact Estimation Tables: Compare Before Building

For First Week Adoption (1-2 hours)

→ Planguage in 60 Seconds
→ Getting Started: The First Week
→ A Worked Example
→ Tested Large Language Model Prompts

For Comprehensive Understanding (3-4 hours)

→ Read sequentially from Executive Summary through Conclusion
→ Pay special attention to Impact Estimation Tables
→ Study the Worked Example

For Reference & Tools

→ A Basic List of Planguage Keywords
→ Credibility Scale
→ Twelve Tough Questions
→ Key Conventions

For AI-Assisted Application

→ How to Benefit from Planguage if One Doesn't Want Details
→ Tested Large Language Model Prompts
→ Use any LLM: "Find ambiguities in <your text> using Tom Gilb Style SQC"

For Executives

→ Executive Summary
→ The Big Idea of Value Planning
→ Why This Works: The Competitive Edge
→ Management Role
→ Management Responsibility

For Product Teams

→ Planguage in 60 Seconds
→ Value Planning Cyclical Steps with Plan-Do-Study-Act
→ A Prioritization Technique – Impact Estimation Table
→ Portfolio Optimization: The Insight Most People Miss
→ Getting Started: The First Week

For Change Agents & Coaches

→ Enable a Bill of Rights for the People Doing the Work
→ Management Responsibility

→ Being a Supporter
→ Learning Loops

Choose your starting point based on your immediate need. All paths eventually connect.

Executive Summary

The Business Case for Planguage

Coherent Communication

The written form is stable, not subject to fading memory, and includes necessary details. It can be systematically improved and changed. It can be reviewed and quality-assured against high enough standards. It can be referenced in any discussion, meeting, or presentation. It can be read and reread at the individual's pace. It provides a legal record and can be referenced in contracts and legal cases. The written form allows detailed interrelationships between planning elements to be mapped rigorously. It can easily be communicated widely geographically, independently of people's schedules.

Coherent Decomposition or Experimentation

For readability, when this booklet says 'solution-option', assume some or all of: outcome-options, solution-options, experiment-options, designs, architectures, or 'strategies' in Tom Gilb's parlance (not to be confused with strategy as defined in the SGEP). (SGEP/Strategy)

Document the primary relationships between critical objectives, solution-options, and other ideas, upwards, downwards and sideways.

If one uses an estimate or a sizing, provide evidence and sources to reduce the risk of bad decisions. When concerned with the credibility of the estimates, document the evidence and sources and use that to rate the level for each estimate. One does this for the entire package of solution-options for all goals in a management summary. Most initiatives are not so special. Whatever one is doing usually has reference cases somewhere; look for them and look at the issues they had.

Estimate the ± range for best and worst-case impacts of the solution-options. Connect Impact Estimation Tables at different related levels of planning **vertically (bottom-up and top-down) and horizontally**.

Evolutionary Delivery of Stakeholder Value

Dr. W. E. Deming told Tom Gilb, about 1983 in London, that the PDSA cycle goes on forever 'as long as there is competition.' So, there is a subtle point:

- Conventional notions like the 'end of an initiative' do not apply: this is a competitive process.
- If the value delivery cycle, an experimental cycle, is short (prefer 'one week'), the posit here is that it does not matter where one enters or exits the cycle (adaptation to PDSA). Any convenient point will be OK.
- Beginning with "big requirements" is usually less valuable than getting experience of a base set first and then setting 'requirements' in an evolutionary manner. Favor 'desirements.' "Desirement" is a coined term used in the Scrum community to contrast with traditional "requirements." It emphasizes that Product Backlog Items express hoped□for outcomes or hypotheses about value, not guaranteed, must□have specifications.
- Maybe "Just get started." Cynefin® [71] provides a leadership compass.

> Plan solution-options to be twice as effective as the goal needs and half the cost of the budgeted resources. But do not commit to actually implementing more solution-options, or more fidelity than needed.

The Big Idea of Value Planning

The 'big idea' is that:

- If management focuses on the results in the form of measurable and critical objectives and uses these results as a constant filter on all technology, then the wrong technology cannot easily emerge or survive. This could avoid a "solution looking for a problem" pattern.
- The right technology has a better chance, an opening, to be discovered and invested in before it is too late. At the very worst, when it becomes clear that no suitable technology is available or known to one to date, one can change overly optimistic objectives and budgets to become more realistic.
- Give up early: give up seeking impossible objectives or impossible deadlines.
- We can either await improved technology in the future or provoke those inventions into reality, as the most outstanding entrepreneurs consistently do (Jobs, Edison, da Vinci, etc.).

Planguage & Value Planning: Evidence-Informed Product Delivery

The Core Problem

> **100 words (out of say 300) per page are typically ambiguous.**
>
> Terms like "better," "faster," "available" mean different things to different people. This ambiguity causes misaligned teams, wasted effort, and failed products. Most organizations can't answer: "How will we know if we've succeeded?"

A Working Solution: Planguage + Value Planning

- **Planguage** is a keyword-driven language that quantifies the qualitative—turning vague goals into measurable objectives.
- **Value Planning** is evolutionary delivery: ship vertical slices weekly or more frequently, measure actual outcomes, let evidence reshape your malleable goals.
- Together, they create **clarity** (everyone understands the target) and **agility** (adapt based on measured results, not assumptions).

How This Works

- Focus on value deliver, not cost-cutting
- Improve one small friction point at a time
- Measure visible benefits within weeks

What to Do First

- Fix one low controversy overhead issue
- Adopt a small corrective action
- Realize a measurable improvement quickly

What Management Will See

- Faster decisions, faster delivery
- Reduced operational noise
- Clear evidence before any expansion

Why This Is Low Risk

- No reorganization announcements
- No headcount targets
- No irreversible commitments

How Success Expands

- Benefiting teams request further improvements
- Learning guides next steps
- Cumulative gains emerge quietly

Management Role

- Protect the experiment
- Judge by evidence, not promises
- Authorize continuation only after visible results

How to Benefit from Planguage if One doesn't want Details

Planguage was designed for machines, so it's a fabulous AI LLM prompting language.

- ChatGPT Gilb Bot – https://tinyurl.com/GilBotGPT
- Grok Gilb Bot – https://tinyurl.com/GilBot; much as I like to avoid Grok, from my testing of these two bots, it's much better than ChatGPT. I wish there was a bot on Perplexity and Claude.
- Try the following prompt for example: using the stakeholder information and other information at `https://evolved.institute`, quantify the value proposition and provide an impact estimation table of the offerings and format it so it can be copied and pasted into a Word document.
- Then ask the LLM to "redteam" its answer. After it responds then say, "fix it."
- But one doesn't need bots really. Beside, bots go out of date and why get limited? Tom's work is so engrained that all LLMs know about his work. See prompts at the end of this document. Or use the above prompt. I tested it on Perplexity and Claude.

Planguage in 60 Seconds

Instead of: *"Improve login speed"*

Write this:

TAG: LoginSpeed

SCALE: Median seconds from app launch to account view

METER: Firebase Analytics, measured on 4G networks

STAKEHOLDER: End Users (Priority: Critical)

PAST [2024-Q4, London]: 8.2 seconds

GOAL [Release 1]: 2.0 seconds

CONSTRAINT: Must be ≤ 3.0 seconds

TOLERABLE: 3.5 seconds (avoid user abandonment)

WISH: 1.0 seconds

What one gains:

- **Unambiguous clarity**: Everyone knows exactly what “better” means
- **Measurable progress**: One can track whether one is succeeding
- **Range of outcomes**: Goal (2.0s), acceptable (3.0s), failure (>3.5s)
- **Evidence requirement**: Past performance grounds expectations

Value Planning: Deliver → Measure → Adapt

The Weekly Cycle (with adapted Plan-Do-Study-Act)

PLAN (2 hours max)

1. Define top 3-5 measurable objectives (use Planguage for critical ones)
2. List solution-options that could achieve them
3. Estimate impacts using Impact Estimation Tables (see below)
4. Pick the highest value/cost option with decent credibility

*
*DO** (rest of week)

- Deliver one complete **vertical slice** (not horizontal layers)
- A vertical slice = end-to-end feature users can use
- Example: simplified login flow (UI + backend + data) that real users try

*
*STUDY** (30-60 minutes)

- Measure actual outcomes against one’s Planguage scales
- Compare reality to estimates (Did LoginSpeed actually improve to 2.0s?)
- Spot side effects (Did abandonment rate change unexpectedly?)

*
*ACT** (30 minutes)

- **Persevere** (it’s working, keep going)
- **Pivot** (switch to different approach based on evidence)
- **Stop** (kill it—this isn’t delivering value)

![image](
In parallel with PDSA, cultivate the work environment and climate (hence the gardening metaphor – a watering can for the plants)

Impact Estimation Tables: Compare Before Building

> Don't guess which solution is best. **Estimate impacts on multiple expectations or limits simultaneously**.

Expectations, Limits </th> <th style="text-align: center;"> <blockquote> <p>Goal </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Solution Option:
 Biometric Login

Solution Option: Progressive Load </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Solution Option:
 Cached Authentication

<strong>LoginSpeed [sec]

2.0

1.5 ← 2.5

Source: RefCase.CompetitorA

Credibility: 80%

2.0 ← 2.8

Source: ABTest

Credibility: 85%

1.2 ← 2.0

Source: TeamGuess

Credibility: 40%

<strong>DevEffort [days]

≤40

25 ← 35

Credibility: 60%

10 ← 15

Credibility: 80%

5 ← 8

Credibility: 85%

<strong>SecurityPass [%]

≥99.9

95 ← 99

Credibility: 50%

99.9

Credibility: 95%

90 ← 95

Credibility: 65%

How to read:

- **1.5 ← 2.5** = best case 1.5 seconds, worst case 2.5 seconds
- **Credibility** = based on a clear scale (later in this booklet) from 0 to 1 in decimals, confidence in estimate (often multiplied by 100 to get a percentage 0-100%). Low credibility = experiment first, don't commit.
- Scan **across rows** to compare solutions on one objective
- Scan **down columns** to see a solution's complete profile

Decision: Progressive Load wins for Week 1 (high credibility, meets goals, low cost). One can test Cached Auth in Week 2 if needed.

The Living Table

- Week 1: Table filled with estimates (impacts, costs) and reference cases
- Week 2: Deliver Progressive Load, measure actual results
- Week 3: Update table with measurements (e.g., credibility jumps to 95%)
- Week 4: Use evidence to decide next move

Why This Works: The Competitive Edge

Traditional Approach

Write comprehensive requirements → Build everything → Hope for correct prediction → Discover 4+ months later it's wrong

Value Planning Approach

Write initial objectives → Deliver weekly vertical slice → Measure actual outcomes → Let evidence reshape objectives → Repeat

Key differences:

- **Objectives emerge from evidence**, not speculation
- **Give up early** when data shows a goal is unrealistic (in weeks, not months)
- **Right solutions get discovered** because one measures what works

- **No “end”** to product development—continuous evolution while there’s competition or it makes sense (other ideas might become relatively more valuable over time)

![image](

Bias for sourced quantified evidence with credibility scores per solution option

Assume Resources means effort and money in this case. People are not resources.

Core Principles

Quantify Critical Objectives Only

Don’t Planguage everything. Reserve it for objectives where someone could die (pharma, aerospace) or the business could die (revenue, security, scalability). Use plain language for the rest.

Decompose by Value, Not Scope

Don’t slice work as “database layer, then API, then UI.” Slice as complete user experiences: “simplified checkout flow that 100 users can try this week.”

Plan to Target, But Don’t Commit to Excess

Design solutions with a safety margin that could deliver on target (perhaps imagining reaching **2x the goal** at **half the budget)**. But only build what measurements prove one needs. Adaptive delivery allows one to “trim the tail.”

Team Credibility Matters More Than Keeping to Commitments Exactly

A rough estimate with high credibility (based on reference cases) beats a precise estimate that’s pure guesswork.

Credibility Scale (0-10):

- 0-3: Guess or distant analogy (experiment first)
- 4-6: Some relevant measurements (proceed with caution)
- 7-9: Proven in your organization (high confidence)
- 10: Solid long-term experience on this exact project (bank on it)

Results Filter Solution Options within Constraints/Limits and Capacity (throughput ranges)

If a solution can’t prove its impact on measurable objectives, **don’t build it**. This principle automatically:

- Kills wrong solutions quickly
- Creates space for unexpected innovations
- Reveals unrealistic goals before they waste months
- Is a forewarning for the risk associated with options that have low credibility scores

Getting Started: The First Week

Monday (2 hours):

1. Pick the #1 product improvement goal
2. Write it in Planguage (TAG, SCALE, METER, PAST, GOAL, CONSTRAINT)
3. List 3 solution options that could achieve it
4. Create simple Impact Estimation Table (estimate impacts, sources, credibility)

Tuesday-Thursday (3 days):

1. Deliver thinnest possible vertical slice of highest-value option
2. Get it into users' hands (even if just 10 users)

Friday (2 hours):

1. Measure actual outcomes (was the GOAL hit? close?)
2. Update the Impact Estimation Table with measurements
3. Decide: persevere, pivot, or stop
4. Plan next week's slice based on evidence

Repeat forever (or until competition ends or it no longer makes sense to do so).

The Bottom Line

In competitive markets, organizations that **deliver better value faster and adapt more efficiently to market needs** win. Planguage provides the language for clarity. Value Planning provides the approach for adaptiveness at speed.

Write down measurable objectives. Deliver weekly vertical slices. Measure what happens. Let evidence reshape the malleable objectives. That's the rhythm that keeps one competitive.

Based on the work of Tom Gilb, pioneer of evolutionary delivery and author of Competitive Engineering. For full methodology: Value Planning: Practical Methods for Measuring, Understanding and Delivering Value.

Introduction

Tom Gilb's work was manually adapted in this booklet for the current times. Readability is a major factor in not calling out all adaptations. Some LLM prompts are included at the end of this booklet. For a summary of Tom's work that it true to the essence of Tom's work according to Tom Gilb, see *Value planning: Practical methods for measuring, understanding and delivering value* [493].

Plan Do Study Act (PDSA) was popularized by W. Edwards Deming and is also adapted in this booklet.

Deemphasis

To clarify, in this booklet, there is a deliberate de-emphasis of the following:

- Any notion of context-free recipes, as "copy and paste" is generally not recommended by competent people, e.g.:
  - Any notion of "Planguage is the solution, what's your problem?" (or opportunity)
  - Any notion of "Scrum Guide Expansion Pack is the solution, what's your problem?" (or opportunity)
  - Any notion of Planguage + EVO/PDSA being the only viable continuous emergent strategy options
- A common misunderstanding of complexity, e.g., the prevalent mix-up of complex work (where expertise is valuable yet insufficient) and complicated work (where expertise is sufficient or can be relatively easily attained)
- **Any notion of Big Design Up Front**; while stewing in the problem/opportunity space is valuable, feedback loop speed is key
- Any notion of yearly to quarterly planning; given human nature, it leads to big batch thinking, no matter how short the delivery cycle
- The use of Specification Quality Control [484] [485] apart from finding critical ambiguities in selections of text
- The use of Tom Gilb's Software Metrics book [494]
- The use of EVO; as adapted PDSA is sufficient (and better supported) for most scenarios
- **Fixed goals**; instead, there is a bent toward tangible-outcome-oriented malleable goals in a direction of travel
- **Any notion of Plan Do Study Act (PDSA) as a fixed loop**; PDSA is adapted here so one can enter or exit the loop at any point

Emphasis

And it adds explicit emphasis to:

- Discovery work interlaced with the problem space, delivery, and value realization
- Failure demand, building on the theory surrounding side-effects that is already mentioned in Tom's work
- "Tidying up the garage", improving team capability and the work climate while delivering valuable work

'I Understand'

There is a difference between:

- I can read a request, and my vendor or team can read it the same way.
- I can read a request, and I really understand it.
- I think I understand a request, and I probably do understand the way the writer intended it.
- I read it and understood a request, and I do not know what vital specs are missing.

Whatever about uncertainty, how many people ever met a manager who would publicly argue that their critical goals, strategies, or objectives should be ambiguous and unclear? But are the critical goals ambiguous and unclear – the critical goals they write and the goals one reads?

One hundred words are typically ambiguous per page of 300 words! "Weasel words" are used such as "better," "improvement," and "available." Even if the goal is wrong (but later adapted), there is generally some merit in being clear about it.

Reasons to be Ambiguous or Unclear

There are reasons to be ambiguous or unclear, some valid, some not so valid. Here are some sarcastic or cynical ones:

1. I want to be able to declare success and meet any deadline; if goals are unclear, I can express my interpretation.
2. I want to sell an unproven fad, perhaps a solution looking for a problem.
3. I want to hide my ignorance, incompetence, or negligence.
4. I am worried that others will look for clarity, and that's a lot of work.
5. I don't know how to be clear and lack the motivation to find out how.
6. The stakeholders don't know what they need or want (labelled "it") anyhow, struggle to articulate it, or I struggle to extract it.

Here are some other reasons:

1. If I quantify non-critical expectations, limits, or objectives, team members "may lose the will to live."

2. To reduce tension, I want to approach the real objective obliquely through an indirect objective. Because I am not trusted, if I declare my true intentions directly and publicly, my words might get twisted or I might "scare the horses."

3. I don't grasp the problem, opportunity or stakeholder's "struggling moment" <u>yet and I'm urgently trying to figure out how.

    1. I need to decide whether to stay in the problem/opportunity space.

    2. And I need to embrace that people (more often) struggle to articulate what is wanted.

    3. I should not stay in this space for too long; there is tension to solve problems or capture opportunities sooner.

Generally, treat fuzziness in stakeholder expectations or limits as a defect. It's clearer to be explicit about fuzziness by using <*Fuzzy Brackets*>. Planguage use is best kept for the most critical expectations, needs, or objectives. Think someone could die (Pharma, Space travel, aircraft design), or the financial health of the organization could die. Critical is in the eye of the beholder. Planguage's sister, Specification Quality Control, is not in the scope of this booklet.

Ambiguity

What happens if an objective is ambiguous? It will often get misunderstood and misapplied. **What happens if an objective is unclear and cannot be properly tested for intended delivery?** One cannot prove that it was carried out correctly. There is something one can do but few have been informed as to how, and fewer have done anything about it.

Stakeholders (including but not limited to customers) often struggle to articulate what they want or why. That said, there is still merit in clarifying what they think they want. Keep plan-quality measurements simple, low cost, inspiring, not demotivating (on the assumption people are already motivated), and direct. Realistic weekly feedback is better, cheaper, and faster.

Yes, there are times when a problem or opportunity needs to be tackled obliquely and where clarity would not be one's friend. And the opportunity or problem should be treated as a hypothesis rather than a fact. But in many cases, there is merit in being clear about the direction of travel, even if it's wrong. Value Planning, with further emphasis on parallel safe-to-fail experiments and an adapted Plan-Do-Study-Act, can provide the feeling: "I don't know where we're going, but I know how to get there" [492].

Value Planning through Plan Do Study Act

Many decompose goals by scope. Try decomposing by value and prioritizing for a slice of value delivery next week using Value Planning. Tom Gilb uses Evo but is also ok with Plan Do Study Act (PDSA). As PDSA is more straight forward and better known (if less specific), we will use that instead of EVO, albeit adapted PDSA. Outside-in feedback-loops are key to learning and adaptation.

Planguage and Value Planning are like a marriage made in utopia. They fit together like hand and glove. They can apply to items at different levels of granularity, e.g., a winning aspiration (vision, mission, purpose), a (product, sub-organizational, or corporate) strategy (e.g., history, diagnosis, problem or response, deliberation on where to play and how to win [231], what needs to be true for this to work, success criteria), a North Star, an outcome-oriented Objectives & Key Results (OKRs), directions of travel, goals, 'epics,' initiatives, experiments, features, 'user stories,' 'use cases,' 'job stories,' 'jobs to be done,' etc. Each of those examples can suffer from a lack of unambiguous clarity. We often read or hear the same words but extract different meanings. To spend efforts more wisely, I suggest one focuses the use of Planguage for the most critical of the above.

In some contexts where the cost of getting something wrong is high (think aviation, space exploration, a high-speed rail system, energy, or fixing a health service), ignorant, ambiguous fuzziness will almost certainly cost billions. Combined with an evolutionary approach, teams can signature-detect the value; think heat-seeking missile.

Planguage Summary

Planguage is a keyword-driven language whose name is derived from summarizing the words 'planning' and 'language' in one neat expression. In many contexts, the cost of getting the wrong results could put lives or the organization at risk. Think of the UK post office scandal or Boeing's troubles 2018-2025. While vagueness is welcome for discussion, details should eventually become fit for coherence, purpose, context, and use. Its primary benefits are quantifying the qualitative and improving communication for complicated ideas (where expertise is enough) and borderline complex ideas (where expertise is valuable yet insufficient for progress toward malleable goals).

Planguage Benefits

Ease of Use

Planguage can be effectively taught to individuals and groups in only a few hours. With a small amount of follow-up mentoring and a catalog of examples, the results can be pretty good. One significant company still well known today used it (and, perhaps, still does) in engineering, quality assurance, marketing, and program management. There are tweaks in this booklet that re-emphasize some key points that were there all along in Planguage and Value Planning, to reduce the risk of negative disruption over the long-term, and there also some adaptations that Tom Gilb agreed to.

Extensibility

Planguage is designed to be extensible and customizable to fit local and contextual needs. This includes adding keywords and its rich structure, which allows it to create and label statements, collections, and other internal structures for reuse. These properties have made Planguage more valuable.

Prevention of Thinking Gaps

One of Planguage's most potent benefits is its ability to prevent omissions when quantifying qualitative statements. Because keywords are applied for all the important dimensions, users of Planguage are more likely to include necessary information. Users praise its ability to bring issues to light through its separate, and consistent treatment of the important dimensions of quantification.

There are usually many possible levels of achievement. The question is not whether a system is consistent or performant but how consistent or performant. Planguage excels at expressing these ideas by using multiple levels of achievement by allowing for the elicitation of the best-recorded level of performance, the goal level, the tolerable level, and the level below which financial or political failure occurs.

A Basic List of Planguage Keywords

Planguage paints a detailed picture of success, survival, and failure, allowing for informed decision-making.

TAG – A unique, persistent identifier

GIST – A short, simple description of the concept contained in the Planguage statement

STAKEHOLDER – A party affected by the "requirement," "desirement," or objective

CONSTRAINT – Limits to operate within

SCALE – Quantification units/range (think cubic units of home gas)

METER – Device for quantification (think utility meter for home gas or electricity)

STATUS – The intent of Status, is real time right now, Not past, not future. One can note the exact instant where one took a Status and can save them as a time series. But they are then by definition a series of Past instance, and the measurement was taken then.

BENCHMARK - A benchmark is a specified reference point, or baseline. There are two main types: scalar and binary benchmarks.

A scalar benchmark is normally defined using the benchmark parameters {Past, Record, Trend}.

PAST – Previous results

TREND – A historical trend based on a range or extrapolation of data

RECORD – The best-known results to date

TOLERABLE – The minimum level to avoid failure (not catastrophic, more like a level during a period where performance is inadequate)

GOAL - a primary numeric target level of performance will *reasonably satisfy* stakeholders, a commitment

WISH – A desirable level of achievement, what the stakeholder thinks they need which if we can and choose to do it becomes a Goal (committed wish)

STRETCH – A stretch goal; Stretch is by definition greater than Goals (settled agreed committed Level), and there is no commitment, yet (or it would be a Goal), there is just the fact that we have noted to desire, assume there is, a reason or justification, and we will return to evaluating (if we have resources, and technology to do it at all, and if we will prioritize it; in which case it become a Goal)

DEFINED – The official definition of a term

AUTHORITY – The person, group, or level of authorization

And METER can be broken down further into:

METHOD – The method for measuring to determine a point on the Scale

FREQUENCY – The frequency at which measurements will be taken

SOURCE – The people or department responsible for making the measurement

REPORT – Where and when the measurement is to be reported

Planguage Example

TAG NPS

GIST Improve Net Promoter Score to what is deemed good by NPS experts

AMBITION Segment leading NPS in the region.

STAKEHOLDER Product Manager

CONSTRAINT {current moment, end-to-end-customer-satisfaction-with-product}

SCALE (of measure) Net Promoter Score

METER (for feedback) high to low NPS range previous 180 days

STATUS [USA across 50 states, 1<sup>st January 2026] 5

TOLERABLE >0

FAIL – starving <0

SURVIVAL – hungry but alive – 0

GOAL 20

STRETCH 30

WISH 50

PAST [2025]: -20 <- Marketing Report [February, 2026]

TREND -30 |<-| to +10 |–>| last 18 months

RECORD 10

DEFINED https://en.wikipedia.org/wiki/Net_promoter_score

AUTHORITY Market Insights team

**

**

!\[image\](Planguage Limits (left hand side) and Expectations (right hand side)

In essence, Planguage offers unambiguous clarity about a multiplicity of expectations and limits.

Arrows indicate scalar limits/constraints or goals, Exclamation Mark indicates failure.

After all, some qualities (like security) are almost infinite, and people tend not to have infinite budgets.

Value Planning

> !\[image\](Officially, the traditional approach has its place, although I have not needed to use it since 2004. Incremental is insufficient for emergent strategy unless it includes assessments of result feedback and telemetry, and adapts based on it. Value Planning is evolutionary. Scrum as described in the Scrum Guide Expansion Pack is evolutionary. But let's take a journey back in time before Scrum. Let's go back in time.

Value Planning Cyclical Steps with Plan-Do-Study-Act

If Planguage helps to establish clarity, an adaptive approach is needed to deliver slices of value, using clarity and trade-offs as a guide. Tom Gilb's approach is EVO. Value Planning is the use of EVO normally. An alternative approach to Value Planning is Plan Do Study Act (PDSA). Let's investigate PDSA.

Plan

Identify Critical Stakeholders

Pinpoint critical, as in the most important or urgent, stakeholders among all stakeholders (including but not limited to internal politicians, customers, consumers, and users) and their pivotal expectations, limits, needs, "jobs," "struggling moments," and wants (collectively in Planguage called 'values') to form the foundation of objectives, `<u>in relation to one's constrained environment`, e.g., medical systems in England and Wales.

Define and Quantify Objectives

Based on critical scalar parameters, quantitatively establish the top ten stakeholder value objectives, detailing the who, for whom, when, where, and how. Create built-in quality, ideally competing (with each other for a bit of fun) to find errors/omissions in objectives or solution-options. It's cheaper to mend holes now but try to strike a balance; **reduce risk with an evolutionary approach**. Therefore, with some exceptions, **avoid a big design or big plan up front that delays feedback.**

Solution-Options

Identify a set of possible solution-options (one's current understanding of how one plans/hopes to achieve objectives) that can efficiently deliver the planned value levels of objectives (effectiveness at lowest costs / highest speed for delivery). Design-to-Attribute (e.g., Design-to-cost or Design-to-date) is often a more cost-effective alternative to productivity increases. Stop playing silly games like how much or when. Instead, specify and design to the constraint/limit (in an emergent way) based on the "art of the possible."

Prioritize

If there is only one solution-option there is no need to prioritize; the priority is already decided. Otherwise, evaluate and estimate the timeliness and cost-effectiveness of each critical solution-option and the associated costs (using Impact Estimation Tables as we'll see later) to help decide which solution-options one should experiment with or deploy as soon as possible. But assess each option with a credibility score to go into battle forewarned if the options are unproven or untested for the context.

Credibility of the impact estimates and cost estimates follow a scale:

1. Wild guess, no credibility
2. We know it has been done somewhere
3. We have one measurement somewhere
4. There are several measurements in the estimated range
5. Several measurements relevant for the use case (e.g. qualifier values)
6. Several relevant measurements obtained using a reliable method
7. Have used the same solution-option previously in the organization
8. Reliable in-house measurements of same solution-option
9. In-house measurements of same solution-option correlate to external sources
10. Have previously used this same solution-option on this initiative and measured it
11. Have solid, contract-guaranteed, long-term experience of this solution-option on this initiative

Do – Generically Show Value Delivery This Week

Optionally, decompose. If one decomposes, then:

- Decompose and evaluate a critical set of the multitude of expectations/limits/needs/"jobs-to-be-done"/"struggling-moments"/wants (values) and associated costs to get a more accurate picture of any solution-option's effectiveness-at-lowest-long-term-costs (efficiency).
- Decompose a solution-option into small, delivery (or interlaced discovery-to-delivery) steps.
- Ideally sequence within throughput ranges and with relative cost-effectiveness for incremental delivery.

Allow time to innovate and time to think:

- Not much innovation arises from 100% "resource utilization" [<u>36].
- Even if tight spots of thinking time can give rise to innovation, a lot of innovation arises during unstructured time slots.
- Allow time to cooperate, collaborate, think, and innovate/invent.
- Don't go so close to the wire. Allow slack time, slack energy, and slack money also for the unexpected.

The main idea here is to get results and enable early learning from early feedback.

Study

Inspect the qualitative and quantitate evidence of outcomes and side-effects. Study product telemetry and result feedback. Be careful with indictive reasons: the conclusion is a good guess, but it's never guaranteed; it could be wrong if new cases appear. Be careful with abductive reasoning: one sees a clue and chooses the most likely explanation, even though other explanations are possible. Be extra careful with deductive reasoning: think of this as starting with a rule and applying it to a case; if the rule is true and one applies it correctly, one's conclusion must be true. In the complex space, where expertise is valuable but insufficient for progress toward malleable goals, deductive reasoning is the less frequent.

Be intentional about recognizing and estimating the positive and negative side effects of solution-options on other value objectives and constraints / limits to ensure higher-quality emergent (continuous) strategy and empirical planning – better done with data than without (it's sometimes ok for a sensible coherent group to guesstimate also).

Act / Adjust / Adapt – Let Reality Foster Humility

It's decision time:

- Persevere: continue with trying or doing; but be careful as most people fall for the 'sunk cost fallacy'
- Pivot: switch to another option (SpaceX excel at this)
- Stop: call it a day, a brave call and sometimes the right option; this does not happen often enough

And

- Amplify something that's working well, and
- Dampen what's not working so well, having really tried

Rinse and repeat… Plan, Do, Study, Act (intentionally skipping steps as needed)

![image](

An evolution from the thought-leading works of W. Edwards Deming et al, inspired by works by Walter A. Shewhart et al.

*Cultivate the work environment and work climate in parallel to PDSA, perhaps with parallel PDSAs.

-

![image](

Adaptations:
1). Enter or exit at any point,
2). Cultivate the work environment and climate (hence the gardening metaphor – a watering can for the plants), and
3). Sometimes replace "Act" with "Adjust" or "Adapt"

![image](

The goalposts should move if evidence tells one the goalposts are in the wrong place.

This is why goalposts and goals should be malleable in product development.

Through intentional pause and reflection, avoid execution bias [210] - the human tendency to persevere when one should pivot or stop.

A Prioritization Technique – Impact Estimation Table

It feels like very prioritization argument is broken. There is an approach that has aged like whiskey (or whisky depending on where it's from). It's the core decision-making tool in Value Planning. An **Impact Estimation Table** is a structured matrix that estimates how different outcome/architecture/design/solution/experiment ideas impact the specified objectives using absolute measurable values.

The Basic Structure

Think of it as a comparison table:

- **Columns** = Design ideas being considered (plus a "Do Nothing" baseline)
- **Rows** = The measurable objectives from Planguage scale definitions
- **Cells** = Estimated impact using the same units as the objectives

Here's what makes it powerful: **one use absolute scale values, not percentages.**

Why Absolute Values Matter

Wrong approach: "Solution A improves login speed by 60%"

Problems with percentages:

- 60% improvement from what baseline? 10 seconds? 2 seconds? The meaning changes drastically
- Can't compare across different objectives
- Hides whether the goal is met

Gilb's approach: "Design A reduces login speed from 8.2 seconds (current) to between 2.5 and 1.5 seconds (worst to best case)"

Why this works:

- Clear baseline (8.2 seconds currently)
- Clear target (the goal is 2.0 seconds)
- Clear estimate range (1.5 to 2.5 seconds)
- Easy to see if the goal is met (yes, even worst case of 2.5 beats the 3.0 goal constraint)

Connection to Planguage Scale Definitions

IETs don't stand alone. Each row references a **fully defined Planguage scale** that specifies exactly what one is measuring and what one cares about.

Example scale definition:

Scale [LoginSpeed, Seconds]:

Meter: Median time from app launch to account display, measured via Firebase Analytics

Past [CurrentApp, London, 4G, Q4.2024]: 8.2 seconds

Goal [Release1, Global, 4G]: 2.0 seconds

Constraint [Release1, Global, 4G]: 3.0 seconds

Wish [Release1, Global, 4G]: 1.0 seconds

Stakeholder [EndUser]: Priority = Critical

Stakeholder [ProductManager]: Priority = High

What this tells:

- **Meter:** Exactly how to measure it (no ambiguity)
- **Past:** Current performance with context (where, when, conditions)
- **Goal:** The target (2.0 seconds)
- **Constraint:** Acceptable limit (must be better than 3.0 seconds)
- **Wish:** Nice-to-have (1.0 would be amazing but not required)
- **Stakeholder:** Who cares and how much

The IET uses these same numbers and units. Every row in the IET corresponds to one Planguage scale definition.

The Four Critical Elements

Each cell in an IET can contain the following pieces of information (more or less):

1. Impact Range Using ← Notation

Format: 1.5 ← 2.5 (for LoginSpeed where lower is better)

What this means:

- Best case: 1.5 seconds
- Worst case: 2.5 seconds
- The **arrow points toward the worse value**

For objectives where lower is better (time, cost, defects):

- 1.5 ← 2.5 means best case 1.5, worst case 2.5

For objectives where higher is better (revenue, security score, satisfaction):

- 90 ← 99 means worst case 90, best case 99

Why ranges? Because honest estimates acknowledge uncertainty. Single-point estimates are false precision and are prone to the ‘flaw of averages’ [234].

2. Source or Evidence

Where did this estimate come from?

- **RefCase.ProductName.Date** = Reference case from similar product
- **Measurement.WeekN** = Actual measured data from delivery
- **TeamEstimate** = Professional judgment from the team
- **ABTest.CompetitorX** = A/B test results
- **VendorSpec** = Vendor’s published specifications
- **Current** = Current measured performance
- **NoChange** = This design doesn’t affect this objective

3. Authority

Who has the expertise to make this estimate?

Format: Authority: Smith.ProductDeveloper or Authority: SecurityScrumTeam

This identifies who made the estimate and has the knowledge to defend it. Different authorities have different credibility for different types of estimates (Scrum has a bias toward estimates from people who do the work as they're often closest to the latest information).

4. Credibility Rating (0-100%)

How confident is the authority in this estimate?

Low credibility doesn't mean "bad estimate"—it means "we should test this before committing significant effort or resources."

Credibility of the impact estimates and cost estimates follow a scale:

1. Wild guess, no credibility
2. We know it has been done somewhere
3. We have one measurement somewhere
4. There are several measurements in the estimated range
5. Several measurements relevant for the use case (e.g. qualifier values)
6. Several relevant measurements obtained using a reliable method
7. Have used the same solution-option previously in the organization
8. Reliable in-house measurements of same solution-option
9. In-house measurements of same solution-option correlate to external sources
10. Have previously used this same solution-option on this initiative and measured it
11. Have solid, contract-guaranteed, long-term experience of this solution-option on this initiative

An adaptation of credibility is to multiply the above by 10 (or treat the above as 0, 0.1…0.9, 1.0 and multiply by 100 to get a percentage.

A Simple Example First

Let's start with two objectives to see how this works. We're comparing three design ideas for faster mobile banking login:

Expectations, Limits </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Past </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Goal </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Solution Option:
 Do Nothing

Solution Option: Biometric </p> </block-quote> </th> <th style="text-align: center;"> <block-quote> <p>Solution Option:
 Progressive

<strong>LoginSpeed [Sec]

8.2 [2024.Q4]

2.0

8.2 Source: Current

Credibility: 95%

1.5 ← 2.5

Source: RefCase.CompetitorA

Authority: Smith

Credibility: 80%

2.0 ← 2.8

Source: ABTest.CompetitorB

Authority: Jones

Credibility: 85%

<strong>DevEffort [Days]

0

≤40

0

Source: Current

Credibility: 100%

25 ← 35

Source: TeamEstimate

Authority: TechLead

Credibility: 60%

10 ← 15

Source: TeamEstimate

Authority: TechLead

Credibility: 80%

How to read this:

The "Past" column shows current performance:

- LoginSpeed is currently 8.2 seconds (measured Q4 2024)
- DevEffort is 0 (we haven't built anything yet)

The "Goal" column shows the targets:

- LoginSpeed must reach 2.0 seconds
- DevEffort must stay within 40 days budget

The "Do Nothing" column shows what happens if one doesn't build anything:

- LoginSpeed stays at 8.2 seconds
- DevEffort stays at 0 days

The design idea columns show estimated impacts:

Biometric Login:

- LoginSpeed: Best case 1.5 seconds, worst case 2.5 seconds (meets goal even in worst case)
- Source: Similar project at a Competitor bank
- Credibility: 80% (pretty confident based on internal reference cases)
- DevEffort: Best case 25 days, worst case 35 days (meets budget)
- Credibility: 60% (less confident on effort estimate)

Progressive Loading:

- LoginSpeed: Best case 2.0 seconds, worst case 2.8 seconds (worst case misses the goal of 2.0, but might be acceptable)
- Credibility: 85% (very confident based on internal A/B tests)
- DevEffort: Best case 10 days, worst case 15 days (well within budget)
- Credibility: 80% (confident on effort)

What this reveals:

Scan across the LoginSpeed row: Biometric looks faster, but Progressive has higher credibility and costs less effort.

Scan down each column: Progressive has balanced performance—good speed, high credibility, low effort. Biometric is faster but more expensive and the effort estimate is less reliable.

The Full Table: Real Decision Complexity

Real decisions involve multiple objectives and trade-offs. Here's what a complete IET looks like:

Impact Estimation Table: Mobile Banking Login Redesign

Date: 2025-01-15

Authority: ProductTeam

Context: Global rollout, primarily 4G networks, iOS and Android

Expectations, Limits </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Past </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Goal </p> </blockquote> </th> <th style="text-align: center;"> <blockquote> <p>Solution Option:
 Do Nothing

<strong>Solution Option: Biometric

<strong>Solution Option: Progressive

<strong>Solution Option: Cached

<strong>LoginSpeed [Sec]<Median time,app launch to account view

8.2

[London,4G,2024.Q4]

2.0 Constraint: ≤3.0

8.2 Source: Current Credibility: 95%

1.5 ← 2.5 Source: RefCase.CompetitorA.2024 Authority: Smith Credibility: 80%

2.0 ← 2.8 Source: ABTest.Competitor Authority: Jones Credibility: 85%

1.2 ← 2.0 Source: TeamEstimate Authority: Brown Credibility: 40%

<strong>Abandonment [%]

Users who start login but don't complete

12% [London, 4G, 2024.Q4]

3% Constraint: ≤5%

12% Source: Current Credibility: 95%

4 ← 6 Source: UXStudy.Bio.2023 Authority: Smith Credibility: 70%

6 ← 8 Source: ABTest.Competitor Authority: Jones Credibility: 75%

7 ← 10 Source: TeamGuess Authority: Brown Credibility: 35%

<strong>DevEffort [Days] Development and testing effort

0

Constraint: ≤40

0 Source: Current Credibility: 100%

25 ← 35 Source: TeamEstimate Authority: TechLead Credibility: 60%

10 ← 15 Source: TeamEstimate Authority: TechLead Credibility: 80%

5 ← 8 Source: TeamEstimate Authority: TechLead Credibility: 85%

<strong>SecurityPass [%] Passing security audit criteria

99.9% [Audit 2024.Q4]

≥99.9%

99.9% Source: Current Credibility: 95%

95 ← 99 Source: VendorSpec Authority: SecTeam Credibility: 50%

99.9 Source: NoChange Authority: SecTeam Credibility: 95%

90 ← 95 Source: SecAudit.2023 Authority: SecTeam Credibility: 65%

How to analyze this table:

Read across each row to compare design ideas on one objective (but also take with a pinch of salt):

- **LoginSpeed:** Cached looks fastest (1.2-2.0 sec) but credibility is only 40%. Biometric is second (1.5-2.5 sec) with better credibility (80%). Progressive is safe (2.0-2.8 sec) with highest credibility (85%).
- **Abandonment:** Biometric wins (4-6%) and meets the constraint (≤5% in best case). Others don't meet the goal reliably.
- **DevEffort:** Cached is cheapest (5-8 days), Progressive is moderate (10-15 days), Biometric is expensive (25-35 days). All meet the ≤40 day constraint.
- **SecurityPass:** Progressive maintains current security (99.9%) with high confidence (95%). Biometric and Cached both show security risks with lower credibility.

Read down each column to see a design's complete profile:

- **Biometric:** Fast, reduces abandonment, but expensive and uncertain security (50% credibility). Mixed profile.
- **Progressive:** Moderate speed improvement, doesn't solve abandonment well, low cost, no security risk, high credibility across all estimates. Balanced and safe.

- **Cached:** Looks great on paper (fast, cheap) but terrible credibility (35-40%) and security concerns. This is a "test first" candidate.

  Key insight: One is not looking for the "winner." One is looking for the **minimum set of designs needed to meet all critical goals.** Yes, sometimes there is more than one goal!

How to Read Impact Estimation Tables

Impact Estimation Tables are **decision tools**, not predictions.

1. Read Vertically (Values First)

Each row answers: *"How well does this solution-option help us move this value toward its goal?"*

If a critical value has no strong impacts, the solution-option set is incomplete.

2. Read Horizontally (Solution-Option Personality)

Each column answers: *"What kind of solution-option is this really?"*

- Few rows → focused, specialist solution-option
- Many rows → systemic, architectural solution-option

3. Ignore Precision, Look for Direction

The numbers are:

- Explicit
- Debatable
- Replaceable by data later

What matters is **relative impact**, not false accuracy.

4. Never Add Columns Blindly

Solution-option impacts:

- Interact
- Overlap
- Sometimes cancel out
- Always check **dependencies and synergies**.

5. Tie Every Cell to Cost and Risk

A high-impact cell is meaningless unless:

- Cost is known
- Risk is explicit
- Learning is planned

6. Use Tables to Design PDSA Experiments

Good solution-options:

- Deliver value early
- Reduce uncertainty
- Earn the right to invest more

Bottom Line

If it can't be shown in an Impact Estimation Table, it isn't ready for board-level decision-making.

Portfolio Optimization: The Insight Most People Miss

Here's where IETs become truly powerful: **one doesn't have to pick just one design idea.**

Traditional thinking: "Which solution is best? Let's pick one and build it."

Gilb's insight: "Which **combination** of designs meets all goals at minimum cost?"

Looking at the table:

- **Progressive alone** doesn't solve abandonment (best case 6%, goal is 3%)
- **Biometric alone** is expensive (25-35 days) and has security uncertainty
- **Cached alone** has low credibility—too risky to commit

Portfolio approach:

Week 1-2: Implement Progressive (10-15 days)

- Meets LoginSpeed constraint (2.0-2.8 sec vs ≤3.0 goal)
- No security risk
- High credibility
- Cost: 10-15 days

Week 3: Run small experiment with Cached (1 day to prototype)

- Test if the 1.2-2.0 sec estimate is real
- Test security implications
- If credibility increases and security passes, consider implementing
- Cost: 1 day experiment

Week 4-5: If abandonment is still above 5%, implement Biometric (25-35 days)

- But first, get better security estimate (talk to vendor, run security audit)
- Only commit if security credibility improves
- Cost: 25-35 days (if needed)

Total potential cost: 10-15 days (Progressive) + maybe 1 day (Cached test) + maybe 25-35 (Biometric if needed) = 36-51 days

But one starts delivering value in Week 2 with Progressive, and one only commits to Biometric if one actually needs it to meet the abandonment goal.

The principle: "Plan solution-options to be twice as effective as the goal needs and half the cost of the budgeted resources. But do not commit to actually implementing more solution-options than needed."

Translation: Design ideas that could deliver 2x what one needs, cost half what was budgeted. But one only builds what one actually needs based on measured results.

Multi-Level Planning: IETs Connect Vertically

IETs exist at multiple planning levels and connect to each other:

Strategic Level (CEO perspective):

Objective: Market Share in Mobile Banking

Design Idea: Mobile App Relaunch Initiative

Impact on Market Share: 5% ← 8% increase

Investment Required: $500K ← $800K

Tactical Level (Product Owner perspective - decomposes "Mobile App Relaunch"):

Objective: LoginSpeed, Abandonment, Security

Design Ideas: Biometric, Progressive Loading, Cached Credentials

Implementation Level (Product Developers' perspective - decomposes "Biometric Login"):

Objective: Integration Time, License Cost, iOS/Android Support

Design Ideas: FaceID SDK, TouchID SDK, Third-Party Library

Each level's design ideas become objectives at the next level down. This traces strategy to implementation and ensures every technical decision supports business goals.

How It Evolves: From Guesses to Evidence

The IET is a living document that evolves weekly:

Week 1: Create initial IET

- Lots of "TeamEstimate" and "TeamGuess" sources
- Credibility ranges from 35% to 85%
- Mix of reference cases and educated guesses

Week 2: Implement Progressive Loading (highest credibility, meets constraints)

- Deliver complete vertical slice
- Measure actual LoginSpeed: 2.3 seconds
- Measure actual Abandonment: 7.5%
- Measure actual DevEffort: 12 days

Week 3: Update the IET

The Progressive column now shows:

LoginSpeed: 2.3 seconds

Source: Measurement.Week2.London.4G

Credibility: 95%

Abandonment: 7.5%

Source: Measurement.Week2.London.4G

Credibility: 95%

DevEffort: 12 days

Source: Measurement.Week2.Actual

Credibility: 100%

What changed:

- LoginSpeed met goal (2.3 vs 2.0 target, within acceptable constraint of 3.0)
- Abandonment didn't meet goal (7.5% vs 3% goal)—we need another design

- Effort was accurate (12 days, within estimated 10-15 range)
- All credibility jumped to 95-100% because these are measurements, not estimates

Week 3 decision: Run Cached experiment (1-day, low risk)

Week 4: Cached experiment results:

LoginSpeed: 1.8 seconds (good!)

Source: Measurement.Week3.Prototype

Credibility: 90%

SecurityPass: 92% (failed security audit)

Source: Measurement.Week3.SecurityTest

Credibility: 95%

Decision: Don't implement Cached—security failure is unacceptable. Move to Biometric but get security clarification from vendor first.

Week 8: Most columns show "Measurement.Week-X" sources with 90%+ credibility. One is making decisions based on evidence, not hope.

Connection to Decision-Making

The IET makes six critical things visible:

1. Which options (for outcomes, solutions, experiments, or designs) meet which goals?

Scan across rows: Does the option's estimated range satisfy the goal/constraint?

- Progressive LoginSpeed (2.0-2.8 sec) vs Goal (2.0 sec): Meets goal in best case, meets constraint (3.0 sec) in worst case □
- Biometric Abandonment (4-6%) vs Goal (3%): Meets goal in best case □

2. Which estimates are risky?

Scan for credibility below 60%:

- Cached has 35-40% credibility = don't commit, run experiment first
- Biometric SecurityPass has 50% credibility = get better data before committing

3. What evidence is missing?

Look for "TeamGuess" or low credibility:

- Cached Abandonment: "TeamGuess, 35%" = we're just guessing, need research or test

- Biometric SecurityPass: "VendorSpec, 50%" = vendor claims aren't reliable, need independent audit

4. What are the trade-offs?

Read down columns and compare:

- Biometric: Great user experience, questionable security, expensive
- Progressive: Moderate everything, high confidence, safe choice
- Cached: Looks cheap and fast, but low confidence and security concerns

5. Can we combine designs to meet all goals?

Look for complementary designs:

- Progressive meets LoginSpeed, doesn't meet Abandonment
- Biometric meets Abandonment, has security concerns
- Combination strategy: Progressive first, then address abandonment if still needed

6. Should we commit or experiment?

Decision matrix:

- High credibility (≥80%) + meets goals = commit and build
- Low credibility (<60%) + small cost = run experiment first
- Low credibility (<60%) + large cost = get better estimates before deciding
- Doesn't meet critical constraints = eliminate option

Gilb's Foundational Principle

> "If you cannot estimate the impact of a design idea on your critical objectives, you should not implement it."
> – Tom Gilb

The IET enforces this discipline through its structure:

No vague claims allowed:

- Not "this will be faster"
- Instead: "1.5 to 2.5 seconds based on RefCase.CompetitorA.2024"

No unmeasured assertions:

- Not "users will love it"
- Instead: "Abandonment 4% to 6% based on UXStudy.Bio.2023"

No invisible uncertainty:

- Not “about 2 seconds”
- Instead: “1.5 ← 2.5 (best ← worst case)”

  No unattributed estimates:

- Not “we think it’ll take 30 days”
- Instead: “25 ← 35 days, TeamEstimate, TechLead, Credibility 60%”

  If one cannot fill in all these fields with actual values, sources, and credibility ratings, one doesn’t understand the design idea well enough to build it. **Start with a small experiment instead.** The IET makes ignorance visible. And visible ignorance can be fixed with evidence. Invisible ignorance just leads to expensive failures.

A Story from Medium Corp That Was Not Dissimilar to a Pre-Scale-Up

For a particular context, the objectives in over-simplified terms were:

- Improve the Software Development process from 0 to 30 Net Promoter Score (NPS) by six months’ time
- Delivery Value Faster by 50% by six months’ time
- Prepare for Scale-up with 50% reduction in queues by six months’ time
- Reduce % incidents of people wrong on the wrong thing by 50% by six months’ time

“As Rome wasn’t built in a day” and due to some things, that I learned about the context, I offered the following over-simplified solution-options, detailed elsewhere:

- Use either Agile Kata, Kanban, Flight Levels, Value Planning, or LeSS (or a combination of one of more of these options)
- Change the approach to sizing and forecasting, inspired by rightsizing in Kanban (work items under a maximum size, not the same size, for knowledge work at least)
- Adopt a specific flow measurement tool to enable a focus on work item aging at dailies

For the specific context, I (and others) struggled to figure out the best option(s) to try first. Work peers notice that “I don’t have a thing”, as in preferred ways regardless of context. I have some deal-breakers though, let’s call them “anti-things.” Still, after using impact estimates, costs estimates, and credibility estimates of each of those, the impact estimation table gave me (and others) food for thought. Like any prioritization technique, use it to inform decision making, not drive it; context is king.

A Large Language Model could produce the following based on the above text

> **Context:** Development Process Improvement Initiative
> **Timeline:** Target completion by six months from {real date}
> **Method:**Tom Gilb's Impact Estimation Analysis
>
> I cheated with percentages here. Sometimes I cheat using money ranges for impacts and effort using: $-$$, $$-$$$, $$$-$$$$, $$$$-$$$$$, $$$$$-$$$$$$, $$$$$$-$$$$$$$. And that's better than nothing, as long as I get started and collect more data later.

Values within Objectives,
 Expectations, Limits (by six months' time)

<strong>Solution Option: Agile Kata

<strong>Solution Option: Kanban

<strong>Solution Option: Flight Levels

<strong>Solution Option: Value Planning

<strong>Solution Option: LeSS

<strong>Solution Option: Rightsizing

<strong>Solution Option: Flow Tool

Total

Dev Process NPS
 0 → 30 NPS </td> <td> 30% </td> <td> 20% </td> <td> 15% </td> <td> 10% </td> <td> 15% </td> <td> 5% </td> <td> 5% </td> <td> 100% </td> </tr> <tr> <td> Value Delivery Speed +50% Faster

15%

25%

10%

20%

10%

15%

5%

100%

Scale-up Readiness
 -50% Queues </td> <td> 10% </td> <td> 30% </td> <td> 20% </td> <td> 5% </td> <td> 15% </td> <td> 15% </td> <td> 5% </td> <td> 100% </td> </tr> <tr> <td> Right Work Focus -50% Wrong Work

25%

15%

25%

30%

5%

5%

5%

100%

Total Impact Score

80

90

70

65

45

40

20

410

Credibility (0-1.0)

0.7

0.8

0.6

0.7

0.5*

0.8

0.9

> **Credibility Scale:** 0.0 (pure guess) to 1.0 (proven with evidence in the context); it's not enough to be generally credible—- context is king.

> > **Critical Assumptions & Risks**

——

- Agile Kata: **Assumes team buy-in and consistent practice.** High impact on NPS due to continuous improvement culture. Agile Kata accepts the matrix but is still impeded by it.

- Kanban: **Well-proven for flow optimization**. Strongest for queue reduction and speed improvements. Kanban accepts the matrix but is still impeded by it.

- Flight Levels: Best for continuous strategy alignment (reducing wrong work) and **requires organizational willingness for transparency**. Flight Levels accepts the matrix but is still impeded by it.

- Value Planning: **Critical for prioritization but needs strong product ownership/management capability.** Value Planning accepts the matrix.

- LeSS Risk (*): Low appetite for deep systemic change, hence, LeSS has a higher failure risk. Consider a pilot test first but not so small as to be useless.

LeSS flips the product development organization toward Theory Y (over The-

- Synergy Effects: Combining Kanban + Flow Measurement + Right-sizing creates multiplicative benefits.

>
Implementation Order: Consider starting with high-credibility, quick-win options (Flow Tool already installed and ready for use, Kanban) before complex transformations but beware of an excessive focus on tooling and process.
> >

- **Priority Bundle:** Kanban + Agile Kata + Flow Measurement Tool (covers 70% of impact with high credibility)
- **Phase 2:** Add Flight Levels and Value Planning for strategic alignment

- **Caution:** LeSS implementation should be carefully evaluated given context-specific risks
- **Quick Win:** Flow Measurement Tool can be implemented immediately with minimal disruption

- **Evidence Basis:** Industry benchmarks, similar transformation case studies, expert judgment
- **Review Cycle:** Re-evaluate impacts after 3 months of implementation

---
-
Note: Estimates are context-dependent and should be adjusted based on organizational specifics ### Analysis of the Thinking Triggered by that Real Impact Estimation Table

---

> Given the organization's experience with flow metrics, the Impact Estimation Table tells me that more regular use of Flow Metrics tooling, particularly focusing on work item aging, was the most efficient option (effec-

Planguage and Value Planning offer a path to higher profitability

> Planguage and Value Planning offer a path to higher delivery profitability while offering:

- Techniques for professionally challenging the viability and feasibility of ideas, and

- People doing the work to the right to clarity on the problems they’re being asked to solve or the opportunities they’re being asked to capture.

> If you don’t know where you’re going you’ll end up somewhere else > – Yogi Berra > > So many people and LLMs like to solve problems or opportunities they don’t yet understand. So, let’s talk about clarity, ideally co-created clarity. It’s good to

###
Co-
create
or
Pro-
vide
Clar-
ity

> Executives have an active role in fostering timely humane effectiveness in the enterprise. Start with Why (2) for whom and attempt to quantify what the whom values. The quantification of expectations and limits is

> With Evolutionary Value Optimization (EVO) or Plan-Do-Study-Act (PDSA), unambiguous clarity is dynamically achieved—not just through early attempts at clarity, but ongoing feedback, adaptation, and close stakeholder (including

Enable a bill of rights for the people doing the work

> I have estimated that we spend a lot of time and money for each objective or key result and too frequently screw things up. If defining an objective or key result in forty lines instead of one line solves that problem, then that is a small

- Know precisely what is expected of them,
- Clarify things with colleagues anywhere in the organization,
- Initiate clearer definitions of objectives and solution-options,

-
Get objectives presented in measurable, quantified formats,

- Change their objectives and solution-options for a quantified improvement in performance against evolving stakeholder expectations and limits,
- Try out new ideas for improving communication,

- Fail when trying, but after really trying, should kill their failures quickly,
- Challenge constructively higher-level objectives and solution-options,

- Be judged objectively and humanely on their performance against measurable objectives (and how they were delivered) and
- Offer constructive help to colleagues to improve communication.

> Please stop the “everything on one slide” nonsense; such executive behavior can cause major accidents (“Death by PowerPoint” – 433). Even Amazon uses six-pagers. While “one-pagers” (as in literally one page, with no supplementary information

Strive for Unambiguous Clarity

>
Aim for unambiguous clarity on objectives (and key results), focusing on the ends rather than the means.
> >
****
> >
UNAMBIGUOUS: The Objectives will be *unambiguous* to the Intended Readership—one *interpretation: the right one only.*

Clarify Desired Outcomes

> A desired outcome is an intended, measurable change in participants’ behavior, knowledge, skills, status, or functioning that often results directly from activities or outputs. Desired outcomes equate to near- to mid-term changes that one

---

- Type: Marketing Director Objective.
- Supports: Corporate Profitability.
- Stakeholders: Marketing, Production, Distribution.

- **Potential Solution-options: 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' safety regulations might differ slightly. <- Legal Department.

> **Consider the “Mafia Offer:”** If one could obtain the objective without using that solution-option, is that OK, or would one prefer to do the solution-option even if one does not achieve the objective?

> >

Here is a different example;

- Type: CTO Level Corporate Objective.
- Scale: Annual frequency of access to our Domain Database % improvement over the previous year.

- Goal [Annually] 10% or more than the previous year.
- Tolerable [Annually] 5% or more than the previous year.
- Constraints:
  - C1: access must be for real internal needs.

- C2: domain knowledge must be produced by our corporation or competitors.
- C3: all outside knowledge must be legally acquired and usable.
- Justification: Corporate Transparency Policy.

- Source: Legal Dept.

> Take Note:

- The guidance using specified constraints rather than the explicit choice of solution-option.

- ‘Tolerable’ and ‘Justification’ are also forms of constraint, meaning restrictions on the choice of solution-option.

> Try to foster focus for teams on the measurable results they produce.

- Avoid discussions on their direct solution-options (one's 'means objectives'): that is the team's business.

- Ask if there is anything at all that keeps them from meeting the goal level on time; remove barriers and document any constraints they experience on their ability to produce results for future discussion.

- Challenge teams to use their imagination, creativity, wisdom, intelligence, and all possible sources to find the smartest and best solution-options. Test whether all rewards and recognition for one's support team are aligned with achieving the objectives; work

- Empower the teams to do the work by giving them clear unambiguous problems to solve but also be honest about the constraints and the levels of delegation. Comply with the *bill of rights*.

Discover or Deliver Solution-Options

> On occasion, solution-options are equal in overall efficiency (defined here as effectiveness at lowest long-term costs) regarding their effects on multiple goals and resource (or effort) consumption.

> But imagine a situation where

- Adjust to (perceived) lower-long-term-cost solution-options.
- One will assume that all initial cost estimates should be corrected.

- One will estimate all impacts on goal levels for each solution-option increment and then measure and record observed costs, together with the solution-option details and examples.
- One will learn fast.

- One will consider re-designing the solution-option (find a [perceived] lower-long-term-cost one!) and try again if necessary.
- One will not waste time by planning solution-options that violate known constraints.

- One will know the constraints and invent ways around them or try to remove them – which is also invention.

> But some observations of many organizations and management planners tell us that:

- Managers and planners need to be trained (in business school and elsewhere) to decompose by value.

- Managers and planners do not practice decomposition by value in conscious, observable ways, with corresponding documentation for their process and thinking (there is no policy to do so).

- Managers and planners make excuses for why it isn't possible to divide solution-options by value (but in a debate with Tom Gilb one will often discover that the excuses are invalid).

- The question is not if one is going to have a value stream flow; the question is if one knows how to decompose solution-options of all kinds to get the value flow early, frequently, and in small increments (like weekly).

> For example:
> Solution-option: *AI removal of questions in call centers. [Country, City, Product Line, Service Level]*

- A1 [Country = UK, City = London, Product Line = Magic, Service Level = None] 50% of Planned revenue.

- A2 [Country = USA, City = LA, Product Line = Magic Version2, Service Level = 24 Hour Help]30% of Planned revenue.

- A3 [Country = Norway, City = Oslo, Product Line = Magic Version 2, Service Level = {24 Hour Help, Norwegian Language}] 15% of planned revenue.

> There is a subtle difference between decomposition to build and decomposition to deliver value. One should be highly interested in these things, listed below, with each instance of decomposition:

- The ability to deliver some real, measurable value, however small, in a single value delivery cycle (suggest weekly) implies that the stakeholders, who are going to receive the value, can do so.

- If weekly value delivery cycles sound like a tight squeeze, consider Lean Startup (474), which reported experimenting with value changes 60 times a day or more!

- That means one should be able to acquire, construct, test, integrate, and field trial all within a single (weekly) cycle. If this sounds strange or difficult, check out the DORA report (279, 280).

- Suppose one decides to decompose to these small-value delivery cycles. Suppose management is clearly behind the idea; suppose one wants it to happen. Suppose one learns the methods in Value Planning. Suppose one has imagination

- It would be best if one was not interested in decomposition like this:

- Decomposing to small steps of construction or acquisition without the immediate consequent ability to deploy to stakeholders and measure the benefits delivered or not delivered.

> The risks of failing to reach objectives within constraints are many and messy. All planners need to identify risks, prevent risks, discover early, and mitigate risks.
> >
Solution-option *A. [Country, City, Target, Product Line, Service Level]*

- *A1 [Country = UK, City = London, Product Line = Magic, Service Level= None] 50% of Planned revenue.*

- A2 [Country = USA, City = Los Angeles, Product Line = Magic Version2, Service Level = 24 Hour Help]30% of Planned revenue.

- A3 [Country = Norway, City = Oslo, Product Line = Magic Version 2, Service Level = {24 Hour Help, Norwegian Language}] 15% ± 5% ?? of planned revenue. <- German Sales Planner

> Comments on the example: - Product development and value realization are often less predictable than the weather.

- The decomposition of solution-option A into A1, A2, and A3 helps define a realistic and helpful definition, or ‘subsets’ of solution-option A.

- One does not risk understanding that any other options are included or planned, yet these could have been intentionally decomposed into high-risk and lower-risk sets.

- A3 shows signs of being a bit 'special' because it has a set of Generic Qualifiers (Country, City), limiting consideration.
- But also permit us to ask which valid combinations have yet to be planned.

-
The 15% ± 5% reduces the chance that anyone will expect, or assume, 15% exactly.

———
- There is a risk that the reality will inform one that a German and a Sales Planner estimated for Norway. This is a warning that there may be a risk of irrelevant competence.

—
-
The ?? clearly warns that the estimate should not be taken seriously. There is a risk it is very wrong. One could argue that is always the case anyhow, but it does no harm to spell out extra uncertainty.

- ‘Norwegian Language’: The capital letters ‘N’ and ‘L’ signal this is a ‘formally defined’ term somewhere. If it is not formally and adequately defined, there is a risk that the specification will be misunderstood.

-

Hint: there are at least 4 Norwegian languages. Norway’s state language is Norwegian (with two official written standards, Bokmål and Nynorsk); Sámi (North, Lule, South) and Kven are recognized Indigenous/minority languages, with Sámi co‑official with Norwegian

- The Statement Tags (solution-option A & A1 & A2 & A3) permit us to have a single tagged ‘master’ planning element, independent of updates, avoiding the confusion of multiple versions in multiple plans and presentations. All plans

---

> ![image](style="width:6.11655in;height:3.44048in"
> alt="© 2024 Tom Gilb A diagram of strategies A, B, and C. Like a Venn diagram, Strategy does not fit within Constraints' specifications. Strategy B is within two of three constraints. Only Strategy a is within all of them. Figure

1. Investigating and reinvestigating the ‘problem space’ or ‘opportunity space’ now and again.

2. Ensuring telemetry and monitoring are in place to discover what people are using.

3. Doing interlaced discovery-delivery supported by frequent delivery, ideally continuous delivery.
4. Requesting feedback and tuning value accordingly.

Foster Evolutionary Value Delivery

> Agree on critical quantified expectations and limits:

- Allow individuals on the team or initiative to distill their own professional critical value dimensions concerning a single objective—this often takes 5 to 30 minutes.

- Then, charge them, as a responsible team, to come up with a list of a maximum of 10 dimensions that they can agree on for the purpose of the initiative—this often takes one hour.

—— - Point out that they cannot be sure what they agree to until each dimension is defined, for example, with a scale of Measure—this often takes thirty minutes for each of the 10.

- They should not ‘agree’ at the ‘tag’ level but at the level of clear definitions of the name of the concept. Refrain from assuming people attribute the same meaning to a word.

>
Respect Subjective Values:

-
One will note and respect the subjective values of one's critical stakeholders while trying to understand their values as wholly and objectively as possible.

- Why? So that one can deliver value to one's stakeholders and satisfy their needs.

- <u>To minimize dissonance between actions and words, try appealing to the expectations and limits that represent the self-interest of the key stakeholders; Tom Gilb believes this to be the reason his work sticks.

Experimentation

> The clear advantages of ‘experimentation,’ of trying out an solution-option and measuring its effects, are:

-
It encourages the team to think through all system-level "requirements" (people, motivation, integration with the current system, etc.)—all the things needed to make value real.

- It gives more credible data about all the multiple needs, “jobs,” or wants (values) delivered and the costs incurred.

- It moves one towards value objectives and improves team experience, confidence, and pride.
- Some mainly pontificate in meetings while others deliver the goods and get on with it.

- One could adjust plans to fit pockets with Dynamic Design to Cost (take the desired cost as a constraint, and PDSA toward it).

-
When evidence is lacking about potential value, experiment with AI, discover what not to deliver, and delivery.

- Consider parallel safe-to-fail experiments to improve candor and improve elapsed times.
- Try out the winners of rough forecasts in a quick, rough way.
- Go with the results.

- ‘Tidy the garage’ as ‘bad stuff scales.’ After cleaning up, and after trying all options to improve value delivery without scaling, only then scale up if needed. The first rule of scaling is not to scale.

- And measure again. Dump losing ideas fast but not so fast not to give them a real chance.

> ![image](style="width:6.12361in;height:3.44444in"
> alt="© 2024 Tom Gilb Strategy A - more cost than Strategy B for a Function, both strategies have "resource remaining" Strategy A - more impact Estimate for Strategy A than Strategy B Performance Gap in Strategy B Fig-

Decompose by Value

A Word of Caution: Big System Replacements

> Tom's view: Always assume that value delivery increments can best be delivered using the 'old system' as a base. > > Tom has a point; system replacements tend be massive failures. My work included a leverage point rescue,

###
Evolve
Rev-
olu-
tions

- Try not to listen to seductive arguments about ‘building a great new future system’ replacing an old system all at once. There is no proof of that argument, and the result is often much worse.

- It's generally better to make teams prove they know what they are doing in practice early. Make them confront measurable reality and learn, learn, and learn early.

- The most successful revolutions are made by a series of mountain goat steps of change. One works towards larger long-term goals, one practical step at a time, as sure-footed as mountain goats. It seems to be how some highly successful organiza-

> ![image](style="width:6.19861in;height:3.48663in"
> alt="© 2024 Tom Gilb Not decomposition for this: Build 1..4 then deploy and measure More Like This: Emergent strategy Create and deploy increment or experiment Get feedback Tweak Realize value Rinse Repeat Figure

Evaluation of Priorities

> Management is about making choices. It is best to make choices when one recognizes limits to time, capability, capacity, and money. Another word for ‘choices’ is prioritization: choosing one thing ahead of or instead of another. I pre-

- What scope of measurement would be roughly able to give satisfactory feedback for the purpose at a cost that seems reasonable?

- To declare that one fully intends to measure results in practice.
- To remind one to build the measurement costs into the cost estimates for the solution-options.

- To trigger a discussion, in meetings and reviews, about whether the METERs (as informal as they might be, in initially specified detail) would be acceptable concerning accuracy/credibility, costs, and cultural/legal acceptance.

> Additional computations to clarify risks:

- The uncertainty factor is introduced, and the least optimistic end of the range for objectives and costs is used. 60 ± 40 = 20, 20 is the ‘least optimistic’ end.

- The credibility score (0.0 none, to 1.0 perfect) is multiplied by the estimate to get a more pessimistic estimate. Example: 0.5 credibility x 60% = 30%.

-
The Credibility score is based on factual evidence for the estimate and the source of those facts.

- Impacts on more long-range goals (some years from now) for the same objective are considered, not just shorter ranges (this year, initial deadlines).

- Final prioritization is based on the initial experimental measures of 2 or more competitive solution-options being applied in parallel to find a winner based on actual results, not just estimates.

> Automatic priority computation is most useful when looking at several scenarios, and:

- There is a frequent need to re-prioritize (like weekly for each initiative, or many initiatives)

- Many factors (e.g., 10 objectives and 10 solution-options) need to be evaluated simultaneously. Avoid over-simplification. In critical industries, there is a multiplicity of expectations and limits that need to be satisficed.

- There is a need for transparency in decision-making: accountability to boards, steering committees, electorates, press, media, and the public; the side effect is one ensures that various initiatives have a sound approach to their decision-making.

> It’s like asking “What if?” so that even a ‘losing solution-option’ proponent sees why their ideas lose and maybe accepts defeat gracefully or fights back constructively.
>
> > It is the (first) priority to survive, not to die or disappear. Then, it is

- **TOLERABLE:** survival, borderline, threatened with death constantly.
- **OK:** absence of pain and discomfort, but much improvement desired

-
GOAL: quite satisfied, successful, only marginally helpful desire to improve this
-
STRETCH: a level better than the Goal but with marginal value for some stakeholders and instances if the price is right.

- **IDEAL:** a theoretical state of maximum perfection, not attainable, not sustainable in practice, or not economical. Like 100% availability 24/7. Is there infinite capacity or budget?

>
One might plan for a range, a comfort zone, or a 'landing zone' rather than an exact number.
![image](

Generally, beyond the world of Planguage, evidence can be qualitative or quantitative, but Tom Gilb strongly leans toward quantitative. Tom even did a TedX talk on how to quantify love! [189]

Source refers to where that evidence was collected. One could also use Authority which means an authoritative source.

> *Credibility expresses the strength of belief in and hence validity of, information. Within Impact Estimation, credibility is usually assessed for the evidence and sources supporting each specific impact estimate. Credibility is ex-*

![image](
A simplified version of the previous image. Assume Resources means effort and money in this case. People are not resources.

Health warning: estimates are just estimates. Some argue one is multiplying a question mark(impacts) by a question mark (credibility) and dividing by another question mark (costs). There is an element of truth to that. However, when one continually re-

Consider money or value points for value. But for effort, consider estimated numbers of items (but not subtasks) to deliver the goal.

Coherent Quality Management

> A ‘quality’ describes ‘how well the system functions;’ focus on ‘how well.’ > – Tom Gilb > > What is common practice with ‘quality’ planning? If one was cynical:

———

- One does not define the many qualities one is concerned with at all (one just says the naming words, like ‘better security’)

- One, even less frequently, quantifies qualities so that they can be seriously planned for (like a 99.9% chance to detect hacking')

- One doesn’t even have a decent definition of ‘quality’, but one uses the word often (‘improved quality’).

- Qualities are not the only values that need to be managed in ‘values objectives.’ Tom Gilb’s experience was that they are among the most critical for most stakeholders (ask stakeholders what they most want to improve and expect 70%

- Non-quality values can be, for example, costs, time, and work capacity of systems, and one is pretty good at quantifying them. ![image](

Here the expectations for the quality “Service Availability” are clarified.

Risk Management

> Some basic concepts:

-
Risk: any factor that could result in a future adverse consequence. It combines {(Threat or Attack) & Mitigation}.

- **Threat:** something (person, thing, situation, circumstances) that can cause some defined class of initiative failure or lack of success.

- **Attack:** this happens when the threat becomes real, emerges, and potentially does or can do some damage to a system.

—— -

Damage: anything contrary to plans, including lack of performance, value delivery for objectives, or increased unexpected resources (time, money), capability, or capacity. Or any negative consequence of an attack on a

- **Mitigation:** any act, pre-planned or not, which reduces or eliminates the damage caused by a threat happening.

> One should assume all plans are ‘risky theory’ until proven otherwise:

- Notwithstanding complexity, management needs to be somewhat fanatical about the policy that ‘they expect, without fail, early measurable improvements in the direction of critical objectives or “jobs to be done;” and continued progress

- Staff need to be trained and coached to do these things. ### Clean-room

> The major innovation in Cleanroom (there were many) was the small 2% (monthly for years) value delivery steps. The 2% value delivery steps act together with people examining (measuring) real impact, actual cost, cumula-

Learning Loops

> **Single loop learning** focuses on correcting errors and improving efficiency within established policies, routines, or goals. The core question is, '**Are we doing things right?'** For example, when an outcome is unsatisfactory,

Management Responsibility

> Dr. Juran (Quality Control Handbook, Juran Institute) [490] made the point in one of his many books that the most significant single thing he learned from his teacher (also Deming's teacher), Walter Shewhart, the originator of Statistical Pro-

- Focus on the top-level critical ‘few’ objectives.
- Pick solution-options that have a large number of good impacts on many critical objectives at the same time.

- Prioritize immediate short-term, next week, value delivery steps to get the highest total value for the time and resources available.
- **Consider all costs, not just development costs.**

- Pick the solution-options with the best values-for-costs in the 'worst-worst' case.
- Make sure that supplier contracts are based on value for payment.

> ![image](style="width:4.94949in;height:2.78402in"
> alt="© 2024 Tom Gilb Iceberg Above the water line: ACQUISITION Below the water line:Documentation Training Test Penalties Maintenance Repairs Upgrades Installation Software Modifications Obsolescence Tool Infras-

Being a Supporter

> Here are some things one can do in planning and with solution-options to help support teams deliver value:

- Guide that management’s job is to remove any barriers to delivering value; ask them frequently what their current barriers are.

- Visit teams so often and in such an intentional way that the truth emerges [10].
- Sit down with real people and try to find out how one can help them.

- Continuously listening and acting on supporting team’s experiences:
- Continuously asking:
- “What could I do to help deliver value better?”

- “Has management taken the need for improvements seriously?” or

- “If you were in my shoes, what is the most useful change you would make to help people get more value delivery?”

- Experiment with discovering and removing barriers.
- Make a list of the three worst obstacles to progress.

- Ask one of the teams or initiatives to list the five worst barriers to their current progress.

- Figure out how to remove one of them, remove them, and tell how it was (already) done. Rinse and Repeat. ## Caution ### Keep a Written List of Stakeholders

> Keep a written list of top stakeholders with every critical objective and every critical solution-option. Here is why one should keep the stakeholders list visible:

- One is really concerned with the critical stakeholders.
- These are the ones that can make or break an initiative.

- If a critical stakeholder is forgotten in Planning, there is a big danger that one will be unaware of them (unless one gets lucky, and another type of stakeholder accidentally covers the forgotten stakeholder's needs).

- One will not include them in one's objectives or constraints.
- One will fail to deliver them.

\-
The critical stakeholders will react and threaten the initiative, like the potential buyer who does not buy.

> If one was to analyze why initiatives fail so frequently, to some degree or other, one would probably find, at the root, a lack of stakeholders’ needs or ‘jobs to be done’ to be taken seriously in the planning for one reason (existence, pri-

Conclusion: From Planning to Value

> In competitive markets, the organizations that win are those that deliver value faster and more precisely than their rivals. Planguage and Value Planning provide the discipline to help make this happen.

The Foundation: Write It Down

> The power starts with something deceptively simple: writing down the ‘desirements’—what one desires the product to achieve—in clear, measurable terms. Not vague aspirations like “improve user experience,” but specific targets like “reduce checkout time

Beyond Writing: Making Connections

> Planguage demands that one maps the relationships between objectives and solution options. For every estimate, one must provide evidence and sources. For every impact, one calculates (or guesstimates but be transparent if doing

How ‘Desirements’ Emerge from Evidence > Consider a mobile banking app. One might start with this desirement: #### Week 1

> Scale [LoginSpeed]:
>
> Ambition: Time from app launch to account view < 3 seconds
>
> Meter: Median time measured via analytics
>
> Past [CurrentApp]: 8.2 seconds @ 2024-Q4
>
> One delivers a vertical slice—a complete, thin fea-

Week 2 - Desirement evolves based on evidence

> Scale [LoginSpeed]: > > Ambition: Time from app launch to account view < 2 seconds > > Meter: Median time measured via analytics > > Past [CurrentApp]: 8.2 seconds @ 2024-Q4 > > Past [SimplifiedFlow]: 4.1 seconds @ Week1 deliv-

Week 3 - Further evolution

> Evidence shows the 2-second target is unrealistic on 3G networks (60% of users). The desirement evolves again, now differentiating by network conditions and focusing on perceived speed through progressive loading. This is adap-

Continuous Value Delivery: The Competitive Edge

> Here's where Value Planning diverges from traditional product development. As Deming told Tom Gilb in 1983: the improvement cycle "goes on forever as long as there is competition." There is no "end" to product development—only continuous

- **Start anywhere.** If one’s delivery cycle is short (one week is ideal), it doesn’t matter where one begins. Jump in, deliver a vertical slice, measure the results, adapt.

- **Learn before one finalizes desirements.** Rather than writing comprehensive ‘desirements’ upfront, deploy a baseline version first. Let real user behavior and measurable outcomes shape the next ‘desirements.’ They emerge from evidence,

- **Give up early.** When evidence shows a desirement is unrealistic or a deadline impossible, revise immediately. Don't wait six months to discover what one week could have revealed.

The Big Idea: Let Results Filter Solutions

> At its core, Value Planning operates on one powerful principle: measurable objectives must filter every technology decision. > > When one makes this real—when every solution must prove its impact on specific, measurable goals—

———
- **Wrong solutions die quickly.** That “innovative” technology that doesn’t move the metrics? It can’t survive when results are visible weekly.

——-

Right solutions get discovered. When one focuses on outcomes, one creates space for unexpected solutions. The best answer might not be the one that was planned.

— **Unrealistic ‘desirements’ become obvious.** If no available technology can hit one’s targets, one would know within weeks, not months. One could then revise the ‘desirements,’ wait for better technology, or invest in creating it.

A Different Way of Working

> Planguage and Value Planning aren't just techniques for writing better 'desirements.' They represent a fundamental shift from predictive planning to evidence-based evolution.
>
> > Traditional approach: Write comprehensive requirements

The Path Forward

> In competitive product development, uncertainty is permanent. Customer needs shift. Technologies evolve. Competitors adapt. The organizations that thrive don't eliminate uncertainty—they learn to navigate it through short cycles, clear measurement,

About the author: Tom Gilb

> Tom Gilb (born USA 1940, Hired at IBM 1958) is the author of 10 paper books, and over 25 digital books. and hundreds of papers, on “requirements,” design, project management and related subjects. His 2005 book ‘Competitive Engineer-

A worked example Process Improvement Goal: Continuous Security Feature Delivery

TAG: SecProd.CycleTime.2026.Q1

GIST: Transform from dysfunctional large-batch delivery to competent Scrum practice, where security expertise enables rapid, small-batch production releases to enable result-feedback and value confirmation or disconfirmation.

STAKEHOLDER:

- **Product Owner:** Senior Product Manager, Compliance Platform
- **Scrum Team:** 8 Developers (including security engineers)

-
Key Stakeholders: VP Product, CISO, enterprise customers **SCALE:** Calendar days from Sprint Backlog selection to Done (deployed to production)

“Sprint Backlog selection” DEFINED: Timestamp when status changes to “In Sprint”

“Done (Output)” DEFINED:

1. Code merged to main branch
2. Automated security scan passed (zero vulnerabilities CVSS ≥9.0)

3. Integration test suite passed (≥95% coverage)
4. Deployed to production Kubernetes cluster (namespace: production)
5. Feature flag enabled for ≥1 user (if applicable)

6. Monitoring shows ≥99.5% success rate over 24 hours
7. Stakeholder verified functionality in Production (going beyond the normal Scrum requirement of “releasable”)

8. Design documentation and support documentation updated
9. No rollback required within 48 hours

"Calendar days" DEFINED:
- Includes weekends and holidays

- Calculation: (End timestamp - Start timestamp) / 86400 seconds + 1 (rounding up to next integer) **METER METHOD:** Start event: Kanban board status “In Sprint” timestamp

1. **End event:** All 9 Done criteria met, and Acceptance Criteria met
2. **Calculation:** (End timestamp - Start timestamp) in calendar days

3. **Aggregation:** Rolling 4-6 Product Backlog Items - calculate P50 (50<sup>th percentile), P85 (85th percentile), P95 (95th percentile) > **FREQUENCY:**

- Real-time dashboard updates (within 5 minutes of Product Backlog Item reaching Done)
- Daily Scrum: Team views current Work In Progress age

- Sprint Review: Distribution of completed Product Backlog Items
- Sprint Retrospective: Outlier analysis

> **SOURCE:**
- Automated: Jira API → Grafana dashboard

- URL: https://kanbanguides.org
- Validation: Scrum Master spot-checks three Product Backlog Items weekly
- Transparency: Team room TV + Slack channel

REPORT:

- Cycle time distribution histogram (0-60 days, 5-day buckets)
- Trend line: P50 over last 12 Sprints
- Outlier table: PBIs >P85 with reasons

PERFORMANCE LEVELS PAST (Q4 2024, Sprints 22-25):

- **Sample:** 10 security feature PBIs
- **P50:** 42.0 calendar days
- **P85:** 54.0 calendar days
- **P95:** 59.0 calendar days
- **Range:** 28-61 days (ratio: 2.18x)

TREND (Last 4 quarters):

- Q1 2024: P50 = 47d, P85 = 58d
- Q2 2024: P50 = 45d, P85 = 57d
- Q3 2024: P50 = 43d, P85 = 55d
- Q4 2024: P50 = 42d, P85 = 54d
- Slope: -1.25 days/quarter (too slow)

---

*

*RECORD:**
28.0
days
(OAuth
SSO,
Sprint
19,
with
ex-
ter-
nal
consultant)
**TOLERABLE
(Min-
i-
mum
ac-
cept-
able
-
Ba-
sic
Scrum
com-
pe-
tency):**
-
**P85
≤
7.0
cal-
en-
dar
days**
-
**Mean-
ing:**
85%
of
PBIs
Done
within
half-
Sprint

——

- **Current status: FAILING by 7.7x** (54d vs. 7d)
- **Trigger:** If P85 >10d for 2 consecutive Sprints → Scrum Master intervention

GOAL (Good Scrum performance for the context):

- **By date:** 2026-09-30
- **P50 ≤ 3.0 days**
- **P85 ≤ 5.0 days**
- **P95 ≤ 7.0 days**
- **Persistence:** Sustain 4 consecutive Sprints

——

- **Business value:** Enables bi-weekly releases; unblocks $400K ARR

* *WISH (Approaching continuous delivery):*

- **By date:** 2026-12-31
- **P50 ≤ 1.0 day**
- **P85 ≤ 2.0 days**

- **Deployments:** ≥15 per Sprint
- **Business value:** “Real-time threat response” positioning; $2M ARR opportunity

STRETCH (Amazon-style continuous deployment):

- **By date:** 2027-06-30

———
-
P50
≤
0.17
days
(4
hours)
-
P85
≤
1.0
day
-
Commit-
to-
production:
≤2
hours

- **Requires:** Microservices architecture ($200K, 6 months), mob programming with AI, or AI agent Product Developers supervised by the human Product Developers

CONSTRAINTS

Operational Quality:

- CONSTRAINT.1: Zero critical vulnerabilities (CVSS ≥9.0)
- CONSTRAINT.2: Deployed to production (not staging/QA)
- CONSTRAINT.3: Monitoring confirms operation (≥99.5%, 24hrs)

Flow:

- CONSTRAINT.4: WIP limit = 6 PBIs for the entire Scrum Team including carry-over
- CONSTRAINT.5: PBIs in Sprint Planning ≤5 days each

- CONSTRAINT.6: PBI >7 days triggers decomposition (using vertical slices such as stakeholder relevant examples or acceptance criteria)
* *Budget:**
- CONSTRAINT.7: ≤$15K per Sprint

-
CON-
STRAINT.8:
Prod-
uct
Owner
spend-
ing
au-
thor-
ity
within
limit
DEFINED
“Security
Fea-
ture
PBI”:
-
Au-
then-
tica-
tion,
au-
tho-
riza-
tion,
en-
cryp-
tion,
mon-
itor-
ing,
vul-
nera-
bil-
ity
fixes,
com-
pli-
ance
re-
port-
ing

- Excludes: Maintenance, performance (unless security), general bugs - Tagged: Jira label security-feature * *"WIP (Work in Progress)" for CONSTRAINT.4:**

- **Count:** All Product Backlog Items (PBIs) in “In Progress” status in Jira
- **Includes:** Carry-over from previous Sprints (PBIs started but not yet Done)

- **Limit:** Maximum 6 PBIs simultaneously for entire 8-person team
- **Measurement:** Count at Daily Scrum
- **Enforcement:** Team should finish existing work before starting new PBIs if at limit

- **Rationale:** Limits context switching; encourages finishing over starting; ~0.75 PBIs per Developer
* *“Decomposition Capability”:* **Current (Sprint 25):**
- Team articulates 1 technique

- 20% of PBIs decomposed reactively - 0% enter Sprint Planning ≤5 days * *Target (Goal state):* - Team articulates ≥5 techniques - 100% of PBIs >5 days decomposed proactively

- ≥80% complete within Sprint started

“Feature Flag”:

- Technology: LaunchDarkly or similar
- Default: off for new flags
- Rollout: Progressive (1%→5%→25%→50%→100%, ≥7 days)
- Rollback: <60 seconds via dashboard

---

* *Effort Levels:**

- **Low:** ≤16 person-hours, ≤2 days, ≤$500, reversible ≤4 hours
- **Medium:** 17-40 person-hours, 3-5 days, $501-$5K, reversible 1-2 days
- **High:** >40 person-hours, >5 days, >$5K, difficult to reverse

Credibility Scale (Tom Gilb):

- **0:** Wild guess, no credibility
- **1:** We know it has been done somewhere
- **2:** We have one measurement somewhere
- **3:** There are several measurements in the estimated range

- **4:** Several measurements relevant for the use case
- **5:** Several relevant measurements obtained using a reliable method
- **6:** Have used the same solution previously in our organization

- **7:** Reliable in-house measurements of same solution
- **8:** In-house measurements correlate to external sources
- **9:** Have previously used this same solution on this initiative and measured it

- **10:** Solid, contract-guaranteed, long-term experience of this solution on this initiative

AUTHORITY Product Owner: Orders Product Backlog; spending ≤$15K/Sprint; works on decomposition with Product Developers and stakeholders

Product Developers: Self-manage; accountable for decomposition skills; create Sprint Backlog; manage WIP as team

Scrum Master: Coaches on decomposition; makes flow metrics visible; challenges PBIs >5 days; helps team maintain WIP limit; removes impediments with the support of other change agents

VP Product: Sets Tolerable; approves >$15K; rolling wave roadmaps laced with uncertainty ### Adapted Impact Estimation Table (columns and rows switched)

Context: Sprint 26, Day 3. Current P85 = 54 days (7.7x worse than Tolerable 7 days). Need experiments to reach Tolerable quickly.

Recommended Solution Option for next PDSA loop

Option H: Combined Workshop + Feature Flag Deployment
Rationale:
- **Highest potential impact:** Could reach Tolerable (P85 ≤7d) in Sprint 26
- **Zero budget:** Fits within CONSTRAINT.7

- **Addresses root cause + enabler:** Skill gap + psychological barrier
- **Credibility 4:** Several relevant measurements exist externally, though one hasn't tried this combination

-
Low risk: Reversible in 1 day, $0 cost

Credibility caveat: At credibility 4, this is based on external evidence, not our own. One should treat Sprint 26 as a measured experiment to raise credibility to 6-7 for future use.

Alternative if seeking higher credibility: Option E (Spike) has credibility 6 (we've done spikes before) but provides no direct cycle time improvement. Could do spike first to de-risk Option H.

Alternative if budget available and willing to accept lower credibility: Option D (External Coach) at credibility 4 provides expert guidance, $1,500 investment.

Tested Large Language Model (LLM) Prompts using Planguage

> Tom Gilb’s work is so established and clear, that it is ideal as a LLM prompting language. Here is the top 20:
> >
> > **1. Stakeholder & “Requirements” Analysis**
> >
> > Identify the top stakeholders and for each specify their critical performance “require-

11. Specification Quality Control (SQC) Checklist

Review this specification against Planguage quality rules and identify major defects: Are all critical "requirements" quantified? Are all <fuzzy> terms defined? Are sources cited? Are assumptions stated? Is every "requirement" testable? Are constraints (FAIL, SURVIVAL) specified? Are stakeholders identified? Rate defect density as major defects per page.

12. Multi-Context "Requirements" with Qualifiers

For this "requirement," specify different levels for different contexts using qualifiers: [Time: Year1 vs Year3], [Place: USA vs Europe vs Asia], [Event: Normal Load vs Peak Load vs Crisis], [User Type: Novice vs Expert], [Configuration: Small vs Enterprise]. Quantify target levels for each context.

13. Dependency & Relationship Mapping

For each "requirement," document: Depends On: [other "requirements" needed first], Supports: [higher-level objectives this enables], Impacts: [requirements affected by this], Conflicts With: [requirements that compete with this], Priority: [relative importance score], Justification: [why this matters quantifiably].

14. Design-to-Requirement Traceability

Create a traceability matrix showing which design ideas [Design.X] are intended to satisfy which performance "requirements" [Requirement.Y]. For each Design→Requirement link, estimate the expected impact (% improvement) and identify any "requirements" not yet addressed by any design.

15. Resource Budget Specification

Specify resource constraints using: RESOURCE.Development.Time: Scale: [months], Budget: [max acceptable], Target: [planned], Spent-To-Date: [current]. RESOURCE.Development.Cost: Scale: [$], Budget: [max], Target: [planned]. RESOURCE.Operations.Cost: Scale: [$/month], Goal: [target ongoing cost]. Include burn rate and forecasts.

16. Value-Focused Thinking & Prioritization

For each stakeholder, identify their value hierarchy: What are their fundamental values? How do they measure success? What are they willing to pay for? Quantify the economic value of each performance improvement using formulas like: Value = [improvement %] × [affected users] × [$ per user per year].

17. Trade-off Analysis Between Competing "Requirements"

Identify competing "requirements" (e.g., Speed vs Accuracy, Cost vs Quality). For each trade-off, specify: the performance attribute pair, current position on the trade-off curve, desired position, what designs move us in desired direction, and the estimated cost of improvement. Create a trade-off matrix.

18. Exit Condition & Definition of Done

For this evolutionary step or project phase, specify measurable exit conditions: Performance.Exit: [specific attribute levels that must be achieved], Quality.Exit: [defect rates or SQC scores required], Stakeholder.Exit: [approval criteria], Budget.Exit: [must be within X% of target]. Define how each will be measured and verified.

19. Defect Prevention & Root Cause Analysis

For specification defects found in SQC, analyze: Type: [(critical) ambiguity, missing “requirement,” unmeasurable, etc.], Root Cause: [why it occurred], Prevention: [rule or process change to prevent recurrence], Learning: [what this teaches us]. Update your specification rules based on patterns found.

20. Continuous Improvement Metrics & PDSA

Define process improvement metrics using Plan-Do-Study-Act cycles: Plan: [what improvement we’ll try], Do: [implement on small scale], Study: [measure results - defect rates, cycle time, value delivered], Act: [adopt, adapt, or abandon]. Track trends over multiple cycles: Past: [baseline], Goal: [target improvement], Actual: [current measurement].

Key Conventions

This booklet does not mean to impose terms or definitions on the reader. The authors respect rights of the readers and their need to define things, in any useful way for them. The authors also respect right or readers to rename any terms. The authors needed to take a position on concepts and terms to communicate and develop their ideas.

Stakeholder

- A stakeholder is an entity, individual, or group interested in, affected by, or impacting inputs, activities, and outcomes. Stakeholders have a direct or indirect interest inside or outside the organization, its products, or services.
- Examples of stakeholders include customers, users, vendors, influencers, managers, colleagues, leaders, and governance, AI, and the law. Inanimate stakeholders such as the law and AI are ignored at the peril of the team doing the work and other stakeholders.
- Some stakeholders have more impact or are more impacted than others, and each can favor different factors.
- Supporting stakeholders, known as Supporters, are stakeholders who influence the organization’s workflows, processes, systems, products, services, and work

environment; they do so to improve consistency with adaptiveness.

- Supporting Stakeholders should offer support that is appreciated by those doing the work.
- Value creation often requires effective, constructive collaboration with stakeholders.
- Depending on the size of the organization, examples of supporting Stakeholders include colleagues, managers, subject matter experts, marketing, HR, finance, procurement, etc.

Value

- Stakeholder value refers to any perceived need that a stakeholder (including customer) considers important, which the team delivers.
- However, stakeholders may not always be aware of what could be valuable to them.
- Observation or evidence could intentionally or unintentionally surface value and influence priorities.
- As new information arises, potentially valuable items should be identified, inspected, refined, and adapted.
- Value remains an assumption until confirmed by evidence, such as observation or measurement of outcomes.
- Value is the fulfilment of expectations, needs, "jobs," or wants from a stakeholder's perspective. It can encompass qualities, costs, constraints, or performance. It can be discovered or be previously unknown to the stakeholder(s).
- Examples include meeting the needs of customers, end-users, organizations, or even environmental considerations.
- Potential value refers to output that remains as inventory until validated by feedback.

Risk

- A risk is any factor that could result in a future adverse consequence.
- Risk exposure remains unpredictable even as time elapses; anticipation is key.
- Risk exposure can include market risk, problem-solution fit, product-market fit, technology, signal detection, responsiveness, compliance, remediation, poor trade-off decisions, etc.; adaptiveness supports active risk mitigation.
- For example, adaptiveness encourages a reduction in the distance between stakeholders who present problems or opportunities and the people solving with the problems or opportunities, the Product Developers, by keeping objectives focused and delivering value quickly and frequently.

Result Feedback

- Result feedback is evidence, ideally both quantitative and qualitative, that might result from changes to the product or environment; it contributes to stakeholder value, effort, resources, or costs.

Side Effect

An impact by a design idea, on any "requirement" attribute, other than the direct impact(s) one primarily intended.

Notes:

- Side effects can be evaluated at a design stage and/or observed at an implementation stage, or even operational or decommissioning stage.
- Conventional usage of 'side effect' implies 'negative effects,' but positive side effects can be just as likely, and just as interesting!
- Side effects can be of the following categories:
  - 'Intended or unintended': 'Intended' means that one has chosen the design because one knew about and valued those particular side effects;
  - 'Known or unknown': 'Known' means one was aware of the existence and possibly the levels of the side effects. 'Unknown' means one was not initially aware of the side effects, but may have become aware of them at some later stage of considering the design (such as in testing, in a review or in operation);
  - 'Negative, neutral or positive'

Evolutionary delivery

A 'high-value-first' approach toward the desired goals, and seeking to obtain, and use, realistic, early feedback. Key components of evolutionary delivery include:

- Frequent delivery of system changes (steps) or experiments (sometimes multiple in parallel, ideally safe-to-fail)
- Steps delivered to stakeholders for real use or experiments run to gather learnings toward stakeholder value
- Feedback obtained from stakeholders to determine next step(s)
- The existing system is used as the initial system base
- Small steps (ideally between 2%–5% of total initiative financial cost and time)
- Steps with highest value and benefit to cost ratios given highest priority for delivery
- Feedback is used 'immediately' to modify future forecasts and "requirements" (John breaks out in a rash when "requirements" are mentioned, as he prefers problems to solve, opportunities to capture or "desirements") and, also to decide on the next step

- Total systems approach ('anything that helps')
- Results-orientation ('delivering the results' is of prime concern), but John would say also how results are delivered

Unambiguous Clarity Defined

Words in documents are often undefined which adds to confusion or misinterpretation. `<u>So, let's define unambiguous clarity.`

Goal Attribute: Unambiguous Clarity

Tag: *CLARITY*

Definition: Unambiguous clarity *means a goal is so clear that everyone understands it in the same way, with no chance for misunderstanding or mixed interpretations. The wording leaves no room for guessing or assumption.*

Scale Unit: *Percentage of stakeholders who provide written interpretations of the goal that exactly match the official definition.*

Meter: *Survey all Scrum Team members and key stakeholders, asking them to write what they believe the goal means. Compare their answers to the official goal statement. Calculate the percentage that are the same.*

Verification Authority: *The Scrum Master or Product Owner checks the results. An independent reviewer (for example, a Product Developer) occasionally double-checks consistency.*

Progression Scale

- ***Tolerable:*** *At least 70% of stakeholders write matching interpretations of the goal.*
- ***Goal:*** *At least 85% of stakeholders write matching interpretations of the goal.*
- ***Wish:*** *At least 95% of stakeholders write matching interpretations, with any differences limited to spelling or grammar, not meaning.*
- ***Stretch:*** *100% of stakeholders write matching interpretations, showing total, error-free shared understanding.*

How to Achieve:

- *Use clear, concrete, and simple language in all goal statements.*
- *Test new goal statements by having the team write or discuss what they think the goal means, then resolve any confusion.*
- *Update and review goal wording during each planning or Sprint Review to ensure ongoing clarity.*

Why it Matters: *In Scrum, having clear goals keeps the Scrum Team focused, makes progress visible, and avoids wasted work or conflict. With unambiguous clarity, everyone knows exactly what the Scrum Team is working toward, leading to more reliable results and a better team culture.*

Plan Quality Defined

But let's clarify 'plan-quality:'

Goal: Plan□Quality Measurement

Tag: PLANQUAL

Definition: *Plan□Quality quantifies the effectiveness, efficiency, and value produced by the planning process as expressed through measurable artefacts such as scheduling adherence, prioritization accuracy, and backlog stability.*

Stakeholders: *Manager, Chief Product Officer, Scrum Team*

Scale Unit: *Defined under each Quality Attribute*

Meter: *Primary data from Jira Flow Analytics, Audit Log Summaries, and Stakeholder Feedback Surveys stored in Confluence.*

Verification Authority: *A Scrum Team member validates internal data collection; an independent reviewer audits sampling accuracy and reproducibility each quarter.*

Quality Attribute: Simplicity

Definition: Ease and speed of collecting and defining plan□quality metrics.

Scale: Percentage of plan□quality measures definable and collectable within a fixed time budget per iteration.

Meter: Randomly sample ten or at least 30% of active metrics (whichever is greater) using a documented random□number generator. Measure the time from data□source access to final metric entry using screen□recorded sessions.

Progression:

Benchmark – 50% collectable within 15 minutes (June 2025 baseline)

Tolerable – ≥ 60% within 12 minutes

Goal – ≥ 80% within 10 minutes using standardized templates

Stretch – ≥ 90% automated capture within 5 minutes or less

Owner – Scrum Team

Quality Attribute: Economy

Definition: Minimizing the cost of maintaining and operating the measurement system.

Scale: Total quarterly cost, using weighted mean staff rates (£60/hour for Product Developers, £75/hour for Managers) plus software licensing.

Meter: Derived from Jira time-tracking logs tagged PLANQUAL and verified quarterly financial reports.

Progression:

Benchmark – £300 per quarter (FY2025-Q3)

Tolerable – ≤ £250 (manual workflows optimized)

Goal – ≤ £150 (semi-automated collection)

Stretch – ≤ £100 (fully automated with negligible additional cost)

Owner – Scrum Team

Quality Attribute: Motivation

Definition: The degree to which plan-quality metrics inform real decision-making.

Scale: Percentage of metrics recorded as evidence in decision logs that result in backlog or planning adjustments.

Meter: Count unique metric IDs cited in Jira change logs or meeting notes where decisions reference data outcomes.

Progression:

Benchmark – 40% metrics influencing planning decisions (FY2025-Q3)

Tolerable – 60% referenced monthly

Goal – 80% referenced weekly

Stretch – 95% influencing at least one Sprint decision per cycle

Owner – Scrum Team

Quality Attribute: Directness

Definition: Reliance on objective artefact-based data instead of perceptions or estimates.

Scale: Direct Data Ratio = percentage of metrics based primarily (≥60%) on automated or traceable digital artefacts.

Meter: Regular review of data lineage and input sources by the Scrum Team.

Progression:

Benchmark – 50% direct artefact data

Tolerable – 60% validated system data

Goal – 70% automated digital sources

Stretch – 85% digital artefacts (manual entry ≤ 15%)

Owner – Scrum Team

Quality Attribute: Feedback Frequency

Definition: The elapsed time between data collection and delivery of analyzed insights to decision-makers.

Scale: Feedback Interval, in business days (excluding weekends and holidays, UTC-adjusted).

Meter: Difference between the Sprint closure timestamp in Jira and publication timestamp of the Plan-Quality report.

Progression:

Benchmark – 14 calendar days (current cycle)

Tolerable – ≤ 10 business days

Goal – ≤ 7 business days (aligned with weekly cycles)

Stretch – ≤ 3 business days (continuous automated dashboards)

Owner – Scrum Team

Quality Attribute: Sampling Effectiveness

Definition: Accuracy and reproducibility of plan-quality insights derived through statistical sampling.

Scale: Confidence interval width, measured at 90% confidence.

Meter: Monthly audit following these steps:

1. Randomly select at least 30 plan artefacts.

2. Analyze using three AI classifiers (Perplexity, Claude, and Grok).

3. Compare outputs for semantic consistency (cosine similarity ≥ 0.9).

4. Human Quality Reviewer rechecks 10% of samples to confirm ≥ 90% agreement.

Progression:

Benchmark – ± 25% margin (Q3 2025)

Tolerable – ± 15% (sufficient for trend monitoring)

Goal – ± 10% (week-to-week reliability)

Stretch – ± 5% (predictive trend accuracy)

Owner – Scrum Team

`<u>Or more simply in plain English:`

Goal: Plan-Quality Measurement (PLANQUAL)

What is Plan-Quality?

Plan-Quality measures how good and useful the planning process is, using clear results such as whether the schedule is followed, priorities are set correctly, and the backlog is stable.

Who is Involved?

- *Manager*
- *Chief Product Officer*
- *Scrum Team*

How is it Measured?

- *Each quality area (attribute) has its own measurement unit.*
- *Data comes mainly from Jira Flow Analytics, logs, and stakeholder surveys.*
- *The Scrum Master checks the data, and a separate Product Developer with Quality Reviewer skills examines a sample each quarter to make sure everything is accurate and repeatable.*

Quality Attributes

Simplicity
How quickly and easily plan-quality data can be collected and defined.

How it's measured:

- *Look at a random sample of ten or 30% of current metrics (whichever is more).*
- *See how long it takes (from getting the raw data to entering the metric) using screen recordings.*

Targets:

- *50% collectable in 15 minutes (current standard)*
- *At least 60% in 12 minutes (acceptable)*
- *At least 80% in 10 minutes with templates (target)*
- *90% automated in 5 minutes or less (best)*
- *Responsible: Product Owner*

Economy
How much it costs to keep the measurement system running.

How it's measured:

- *Look at the total cost each quarter, including wages and software.*
- *Uses Jira logs and quarterly finance reports.*

Targets:

- *£300 per quarter (current standard)*
- *£250 if manual work is optimized (acceptable)*
- *£150 with some automation (target)*
- *£100 with full automation (best)*
- *Responsible: Product Owner*

Motivation
How much the plan-quality data actually affects real decisions.

How it's measured:

- *Look at the percentage of metrics mentioned in decision logs that lead to changes.*
- *Count unique metric IDs referenced in change logs or meeting notes.*

Targets:

- *40% influencing decisions now (current standard)*
- *60% referenced monthly (acceptable)*
- *80% referenced weekly (target)*
- *95% influencing at least one Sprint decision per cycle (best)*
- *Responsible: Scrum Master*

Directness
How much the data depends on hard evidence instead of personal opinion.

How it's measured:

- *Calculate the percentage of metrics based mainly (at least 60%) on automated or traceable digital sources.*
- *Regular checks are done by a Product Developer.*

Targets:

- *50% from direct artefact data (current standard)*
- *60% from validated system data (acceptable)*
- *70% from automated sources (target)*
- *85% from digital artefacts, with a maximum of 15% entered by hand (best)*
- *Responsible: Product Developers*

Feedback Frequency
How long it takes to deliver analyzed insights to people making decisions.

How it's measured:

Measures the time (in business days) between the end of the Sprint (in Jira) and when the report is published.

Targets:

- *14 days (current standard)*
- *10 days (acceptable)*
- *7 days with weekly cycles (target)*
- *3 days or less with continuous dashboards (best)*
- *Responsible: Scrum Master*

Sampling Effectiveness
How accurate and repeatable the plan-quality findings are when using samples.

How it's measured:

- *Every month, pick at least 30 plan artefacts at random.*
- *Analyze using three different AI tools.*
- *Compare the results (need them to agree at least 90% of the time).*
- *A human checks 10% of samples to confirm over 90% agreement.*

Targets:

- ± *25% margin now (current standard)*
- ± *15% margin for trend monitoring (acceptable)*
- ± *10% margin for week-to-week reliability (target)*
- ± *5% margin for predicting future trends (best)*
- *Responsible: Product Developers*

So, is this is too much work? Are the critical goals for the context clear? While plain English can be clear, the structure of Planguage communicates ideas more clearly to AI systems. Ask a favorite Large Language Model (LLM) "find the (critical) ambiguities in the following text using Tom Gilb Style Specification Quality Control (SQC): <paste text here>" Then say, "fix it with SQC and Planguage" or "fix it." There will be a lot of auditing and editing to do on the responses. But after carving away all unnecessary complicatedness then (close to) unambiguous clarity will help one to frame "work as problems to solve or opportunities to capture." Planguage is often more succinct than plain English.

References

[10] Vodde, B. and Larman, C. (2023) Go see, LeSS. Available at: https://less.works/less/management/go-see (Accessed: April 3, 2023).

[36] Kniberg, H. (2014) The resource utilization trap, YouTube. Available at: https://youtu.be/CostXs2p6r0 (Accessed: April 3, 2023).

[71] Cynefin.io, V. (2022) Cynefin wiki, Cynefin.io. Cynefin.io. Available at: https://cynefin.io/ (Accessed: April 4, 2023).

[102] Daniel S. Vacanti, Inc, Orderly Disruption Limited (2020) The official Kanban Guide., Kanban Guides. Available at: https://kanbanguides.org/ (Accessed: April 5, 2023).

[126] Reinertsen, D.G. (2023) Books | Reinertsen & Associates, The Principles of Product Development Flow. Available at: http://reinertsenassociates.com/books/ (Accessed: April 5, 2023).

[168] Gilb, T. (2016) Value planning, Leanpub. Leanpub. Available at: https://leanpub.com/ValuePlanning (Accessed: April 7, 2023).

[169] Gilb, T. (2018) Vision Engineering (free version for top executives with real examples), ResearchGate. Available at: https://www.researchgate.net/publication/384600479_VP_Top_Level_Vision_Engineering_not_master_6MB_MAY18_2017 (Accessed: October 5, 2025).

[173] Gilb, T. (2021) 12? Twelve tough questions, Leanpub. Leanpub. Available at: https://leanpub.com/12ToughQuestions (Accessed: April 7, 2023).

[189] Gilb, T. (2013) Quantify the un-quantifiable: Tom Gilb at TEDx Trondheim, YouTube. Available at: https://youtu.be/kOfK6rSLVTA (Accessed: April 8, 2023).

[210] Marquet, L.D. (2020) Leadership is language: The hidden power of what you say and what you don't. New York, NY: Portfolio/Penguin.

[211] Deming, W.E. (2018) Out of the crisis. reissue. Cambridge, MA: MIT Press.

[231] Martin, R.L. (2022) A New Way to Think: Your Guide to Superior Management Effectiveness. Boston, MA, USA: Harvard Business Review Press.

[234] Savage, S.L. (2012) The flaw of averages. London, England: Wiley & Sons.

[288] Flyvbjerg, B. (2023) How big things get done. London, England: PAN MACMILLAN.

[292] Gilb, T. (2005) Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering

Using Planguage. Amsterdam, NL: Butterworth-Heinemann. Available at: <u>https://www.researchgate.net/publication/237129623_Competitive_Engine
(Accessed 27 December 2025)

[299] Deming, W.E. (1952) in Elementary principles of the statistical control of quality; a series of lectures. 2nd edn. Tokyo: Nippon Kagaku Gijutsu Remmei, pp. 5–5.

[325] Cagan, M. (2024) Transformed: Moving to the Product Operating Model. Hoboken, NJ, NJ: John Wiley & Sons, Inc.

[329] Edwards Deming, W. (2014) Red Bead Experiment with Dr. W. Edwards Deming, YouTube. At: https://youtu.be/ckBfbvOXDvU?si=yRV7Tt JWhcSKNpOT (15 November 2023).

[330] Edwards Deming, W. (2023) PDSA cycle, The W. Edwards Deming Institute. Available at: https://deming.org/explore/pdsa/ (Accessed: November 15, 2023).

[421] McGregor, D. and Cutcher-Gershenfeld, J. (2006) The human side of Enterprise. New York, NY: McGraw-Hill.

[432] Flyvbjerg, B. (2022) ‘Heuristics for Masterbuilders: Fast and frugal ways to become a better project leader’, SSRN Electronic Journal – https://ssrn.com/abstract=4159984 [Preprint]. doi:10.2139/ssrn.4159984.

[434] Keeney, R.L. (1998) Value-focused thinking: A path to creative decision-making. Cambridge, MA: Harvard University Press.

[474] Gilb, T. (2014) Lean Startup Vs Evo - Krakow 2014, ResearchGate. Available at: https://www.researchgate.net/publication/399110798_Lean_Startup_slides_Originally_held_for_2014_ACE_Conference_Poland?channel=doi&linkId=694fcebb0c98040d4823129e&showFulltext=true (Accessed: December 27, 2025).

[475] Mills, H. 1980.The management of software engineering: part 1: principles of software engineering. IBM Systems Journal19, issue 4(Dec.):414-420. At https://trace.tennessee.edu/utk_harlan/5/ (Accessed: March 30, 2024).

[476] Mills, Harlan D.; Dyer, M.; and Linger, R. C.,“Cleanroom Software Engineering”(1987). The Harlan D. Mills Collection. At http://trace.tennessee.edu/utk_harlan/18 (Accessed: March 30, 2024).

[477] Mills Generally. At https://trace.tennessee.edu/utk_harlan/ (Accessed: March 30, 2024).

[478] Quinnan, R.E. (1980) ‘The management of software engineering. Part V: Software engineering management practices’, IBM Systems Journal, 19(4), pp. 466–477. Available at: https://trace.tennessee.edu/cgi/viewcontent.cgi?article=1004&context=utk_harlan (Accessed 27 December 2025).

[479] Gilb, T., Graham, D. and EDITOR Finzi, S. (1993) Software inspection. Harlow, Essex, England: Addison-Wesley. Available at https://www.researchgate.net/publication/397439528_Software_Inspection_Pdf_1993_84MB (Accessed: December 27, 2025)

[480] Practical Purposeful Creativity. Paper, T Gilb. This paper is written as an invited contribution to a book "Creativity, Innovation, and Cooperation" (Springer) and a special issue of "AI & Society: the Journal of Human-Centred Systems and machine Intelligence". The editor is Robert C. Muller Published around 1992. At https://www.researchgate.net/publication/391438759_Innovative_CreativIty_Book_copy_August_7_edit_2018_2153 (Accessed: October 5, 2025)

[481] Gilb, T. (2016) Planguage and innovation: How metrics drive ideas. IBM Brno, Czech Republic / GilbFest 2015, conference talk. Available at: https://www.researchgate.net/publication/399111233_Planguage_Innovation_MASTER_Talk (Accessed 27 December 2025).

[482] DEEPER PERSPECTIVES ON EVO DELIVERY Chapter 15 in (1988) Principles of Software Engineering management "Deeper Perspectives on Evolutionary Delivery" plus a page extra of quotations from Agile Gurus crediting it as inspiration for them, and it being first. At https://www.researchgate.net/publication/380874956_Ch_15_Deeper_perspectives_on_Evolutionary_Delivery_later_2001_known_as_Agile_in_Gilb_Principles_of_Software_Engineering_Management (Accessed: October 5, 2025).

[483] Tom Gilb: "Competitive Engineering" (2005) Chapter 10: Evolutionary Project Management. Chapter 8, Specification Quality Control. For a summary see [484] [485] [486]. At https://www.researchgate.net/publication/288741383_Competitive_Engineering (Accessed: October 5, 2025)

[484] Specification Quality Control "Agile Specification Quality Control: Shifting emphasis from cleanup to sampling defects", in Testing Experience, March 2009. See also Chapter on decomposition in Competitive Engineering book [292]. At https://www.researchgate.net/publication/344267092_Chapter_8_Specification_Quality_Control (Accessed: October 5, 2025).

[485] The McDonnell-Douglas Case Study of SQC and Engineering Improvement Case DAC Inspection 1988-89. ROI 4.5:1. Also contains Philips (UK, detail in (479) as case study) and Boeing SQC client cases. Tom Gilb and Kai Gilb did Boeing (Renton) after McDonnell-Douglas success and recommendation (in better times when quality was still the top concern). At https://www.researchgate.net/publication/383877080_Using_'Evo'_to_Rapidly_deliver_measurable_improvements_to_Aircraft_Design_Engineering_Drawing_QC (Accessed: October 5, 2025).

[486] Terzakis, J. (2013) 'The impact of requirements on software quality across three product generations', Proceedings of the 2013 21st IEEE International Requirements Engineering Conference (RE 2013), Rio de Janeiro, Brazil, 15–19 July, pp. 284–289. Available at: https://www.thinkmind.org/download.php?articleid=iccgi_2013_3_10_10012 (Accessed 27 December 2025).

[487] Dan Ariely, The Upside of Irrationality. The unexpected benefits of defying logic at work and at home. ISBN 978 0 00 735478-8, Harper 2010

[488] Shewhart, W.A. (2003) Statistical Method from the Viewpoint of Quality Control (Dover Books on Mathematics). Mineola, NY: Dover Publications Limited.

[489] Deming, W.E. (2021) Knowledge of variation, The W. Edwards Deming Institute. Available at: https://deming.org/knowledge-of-variation/ (Accessed: March 31, 2024).

[490] Juran, J.M. and Gryna, F.M. (1988) Quality Control Handbook. New York and London, NY, USA and England: McGraw-Hill.

[492] Tim Ferriss (2022) The tracker's maxim: "I don't know where we are going, but I know exactly how to get there." YouTube, 17 February. Available at: https://www.youtube.com/watch?v=yDHm7lGArRU (Accessed: 26 October 2025).

[493] Gilb, T (2023) Value Planning. Practical Tools for Clearer Management Communication. Available at: <u>`https://www.researchgate.net/publication/39911`
(Accessed December 27, 2025).

[494] Gilb, T. (1976) Software metrics. Lund: Studentlitteratur AB. Also published as: Gilb, T. (1977) Software metrics*. Cambridge, MA: Winthrop Publishers. Available at: https://www.researchgate.net/publication/235978133_Software_Metrics (Accessed 27 December 2025).*