How to Quantify Quality: Finding Scales of Measure

Tom Gilb

Norway
Tom@Gilb.com

Abstract. Quantification is key to controlling system performance attributes. This paper describes how to quantify performance requirements using Planguage – a specification language developed by the author. It discusses in detail how to develop and use *tailored* scales of measure.

1 Finding and Developing Scales of Measure and Meters

The basic advice for identifying and developing scales of measure and meters (practical methods for measuring) for scalar attributes is as follows:

1. Try to re-use previously defined Scales and Meters. *Examples [1], [3].*

2. Try to modify previously defined Scales and Meters.

3. If no existing Scale or Meter can be reused or modified, use common sense to develop innovative home-grown quantification ideas.

4. Whatever Scale or Meter you start off with, you must be prepared to learn. Obtain and use early feedback, from colleagues and from field tests, to redefine and improve your Scales and Meters.

1.1 Reference Library for Scales of Measure

'Reuse' is an important concept for, sharing experience and saving time when developing Scales. You need to build reference libraries of your 'standard' scales of measure.

Tag: <assign a tag name to this Scale>.

Version: <date of the latest version or change>.

Owner: <role/email of who is responsible for updates/changes>.

Status: <Draft, SQC Exited, Approved>.

Scale: <specify the Scale with defined [qualifiers]>.

Alternative Scales: <reference by tag or define other Scales of interest as alternatives and supplements>.

Qualifier Definitions: <define the scale qualifiers, like 'for defined [Staff]', list their options, like {Nurse, Doctor, Orderly}>.

Meter Options: <suggest Meter(s) appropriate to the Scale>.

Known Usage: <reference projects & specifications where this Scale was actually used in practice with designers' names>.

Known Problems: <list known or perceived problems with this Scale>.

Limitations: <list known or perceived limitations with this Scale>.

J. Filipe, B. Shishkov, and M. Helfert (Eds.): ICSOFT 2006, CCIS 10, pp. 27–36, 2008.

Remember to maintain details supporting each 'standard' Scale, such as Source, Owner, Status and Version (Date). If the name of a Scale's designer is also kept, you can probably contact them for assistance and ideas. Here is a template for keeping reusable scales of measure.

Example: This is a draft template, with <hints>, for specification of scales of measure in a reference library. Many of the terms used here are defined in Competitive Engineering [2] & [3]. See example below for sample use of this template.

Tag: Ease of Access.

Version: 11-Aug-2003.

Owner: Rating Model Project (Bill).

Scale: Speed for a defined [Employee Type] with defined [Experience] to get a defined [Client Type] operating successfully from the moment of a decision to use the application.

Alternative Scales: None known yet.

Qualifier Definitions:

Employee Type: {Credit Analyst, Investment Banker, …}.

Experience: {Never, Occasional, Frequent, Recent}.

Client Type: {Major, Frequent, Minor, Infrequent}.

Meter Options:

Test all frequent combinations of qualifiers at least twice. Measure speed for the combinations.

Known Usage: Project Capital Investment Proposals [2001, London].

Known Problems: None recorded yet.

Limitations: None recorded yet.

Example of a 'Scale' specification for a Scale reference library. This exploits the template in the previous example.

1.2 Reference Library for Meters

Another important standards library to maintain is a library of 'Meters.' Meters support scales of measure by providing practical methods for actually measuring the numeric Scale values. 'Off the shelf' Meters from standard reference libraries can save time and effort since they are already developed and are more or less 'tried and tested' in the field.

It is natural to reference suggested Meters within definitions of specific scales of measure (as in the template and example above). Scales and Meters belong intimately together.

2 Managing 'What' You Measure

It is a well-known paradigm that you can manage what you can measure. If you want to achieve something in practice, then quantification, and later measurement, are essential first steps for making sure you get it. If you do not make critical performance attributes measurable, then it is likely to be less motivating for people to find ways to deliver the necessary performance levels. They have no clear targets to work towards, and there are no precise criteria for judgment of failure or success.

3 Practical Example: Scale Definition

‘User-friendly’ is a popular term. Can you specify a scale of measure for it?
Here is my advice on how to tackle developing a definition for this.

1. If we assume there is no ‘off-the-shelf’ definition that could be used (in fact there are, see [1] and [3]):

- Be more specific about the various aspects of the quality. There are many distinct dimensions of quality for user-friendly such as ‘user acceptance’, ‘user training’, ‘user errors made’, ‘user customization’ and ‘environmentally friendly’. List about 5 to 15 aspects of some selected quality that is critical to your project.
- For this example, let’s select ‘environmentally friendly’ as the one of many aspects that we are interested in, and we shall work on this below as an example.

2. Invent and specify a Tag: ‘Environmentally Friendly’ is sufficiently descriptive. Ideally, it could be shorter, but it is very descriptive left as it is. Let’s indicate a ‘formally defined concept’ by capitalizing the tag.

Tag: Environmentally Friendly.

Note, we usually don’t explicitly specify ‘Tag:’, but this sometimes makes the tag identity clearer.

3. Check there is an Ambition statement, which briefly describes the level of requirement ambition. ‘Ambition’ is a defined Planguage parameter. More parameters follow, below.

Ambition: A high degree of protection, compared to competitors, over the short-term and the long-term, in near and remote environments for health and safety of living things.

4. Ensure there is general agreement by all the involved parties with the Ambition definition. If not, ask for suggestions for modifications or additions to it. Here is a simple improvement to my initial Ambition statement. It actually introduces a ‘constraint’.

Ambition: A high degree of protection, compared to competitors, over the short-term and the long-term, in near and remote environments for health and safety of living things, which does not reduce the protection already present in nature.

5. Using the Ambition description, define an initial ‘Scale’ (of measure) that is somehow quantifiable (meaning – you can meaningfully attach a number to it). Consider ‘what will be sensed by the stakeholders’ if the level of quality changes. What would be a ‘visible effect’ if the quality improved? My initial, unfinished attempt, at finding

a suitable ‘Scale’ captured the ideas of change occurring, and of things getting ‘better or worse’:

Scale: The % change in positive (good environment) or negative directions for defined [Environmental Changes].

My first Scale parameter draft, with a single scalar variable.

However, I was not happy with it, so I made a second attempt. I refined the Scale by expanding it to include the ideas of specific things being effected in specific places over given times:

Scale: % destruction or reduction of defined [Thing] in defined [Place] during a defined [Time Period] as caused by defined [Environmental Changes].

This is the second Scalar definition draft with four scalar variables. These will be more-specifically defined whenever the Scale is applied in requirement statements such as ‘Goal’.

This felt better. In practice, I have added more [qualifiers] into the Scale, to indicate the variables that must be defined by specific things, places and time periods whenever the Scale is used.

6. Determine if the term needs to be defined with several different scales of measure, or whether one like this, with general parameters, will do. Has the Ambition been adequately captured? To determine what’s best, you should list some of the possible sub-components of the term (that is, what can it be broken down into, in detail?). For example:

Thing: {Air, Water, Plant, Animal}.

Place: {Personal, Home, Community, Planet}.

Thing: = {Air, Water, Plant, Animal}.

Place: Consists of {Personal, Home, Community, Planet}.

Definition examples of the scale qualifiers used in the examples above. The first example means: ‘Thing’ is defined as the set of things Air, Water, Plan and Animal (which, since they are all four capitalized, are themselves defined elsewhere). Instead of just the colon after the tag, the more explicit Planguage parameter ‘Consists Of’ or ‘=’ can be used to make this notation more immediately intelligible to novices in reading Planguage.

Then consider whether your defined Scale enables the performance *levels* for these sub-components to be expressed. You may have overlooked an opportunity, and may want to add one or more qualifiers to that Scale. For example, we could potentially add the scale qualifiers ‘… *under defined [Environmental Conditions] in defined [Countries]…*’ to make the scale definition even more explicit and more general.

Scale qualifiers (such as …*‘defined [Place]’*…) have the following advantages:

- they add clarity to the specifications
- they make the Scales themselves more reusable in other projects

• they make the Scale more useful in this project: specific benchmarks, targets and constraints can be specified for any interesting combination of scale variables (such as, ‘Thing = Air’).

7. Start working on a ‘Meter’ – a specification of how we intend to test or measure the performance of a real system with respect to the defined Scale. Remember, you should first check there is not a standard or company reference library Meter that you could use.

Try to imagine a practical way to measure things along the Scale, or at least sketch one out. My example is only an initial rough sketch.

Meter: {scientific data where available, opinion surveys, admitted intuitive guesses}.

This Meter specification is a sketch defined by a {set} of three rough measurement concepts. These at least suggest something about the quality and costs with such a measuring process. The ‘Meter’ must always explicitly address a particular ‘Scale’ specification.

The Meter will help confirm your choice of Scale as it will provide evidence that practical measurements can feasibly be obtained on a given scale of measure.

8. Now try out the Scale specification by trying to use it for specifying some useful levels on the scale. Define some reference points from the past (Benchmarks) and some future requirements (Targets and Constraints). For example:

Environmentally Friendly:
Ambition: A high degree of protection, compared to competitors, over the short-term and the long-term, in near and remote environments for health and safety of living things, which does not reduce the protection already present in nature.
Scale: % destruction or reduction of defined [Thing] in defined [Place] during a defined [Time Period] as caused by defined [Environmental Changes].
============= *Benchmarks* =================
Past [Time Period = Next Two Years, Place = Local House, Thing = Water]: 20% <- intuitive guess.
Record [Last Year, Cabin Well, Thing = Water]: 0% <- declared reference point.
Trend [Ten to Twenty Years From Now, Local, Thing = Water]: 30% <- intuitive. "Things seem to be getting worse."
============ *Scalar Constraint* ==========
Fail [End Next Year, Thing = Water, Place = Eritrea]: 0%. "Not get worse."
=============== *Targets* ===================
Wish [Thing = Water, Time = Next Decade, Place = Africa]: <3% <- Pan African Council Policy.
Goal [Time = After Five Years, Place = <our local community>, Thing = Water]: <5%.

If this seems unsatisfactory, then maybe I can find another, more specific, scale of measure? Maybe use a ‘set’ of different Scales to express the measured concept better? See examples below.

Here is an example of a single more-specific Scale:

Scale: % change in water pollution degree as defined by UN Standard 1026.

Here is an example of some other and more-specific set of Scales for the ‘Environmentally Friendly’ example. They are perhaps a complimentary set for expressing a complex *Environmentally Friendly* idea.

Environmentally Friendly:

Ambition: A high degree of protection, compared to competitors, over the short-term and the long-term, in near and remote environments for health and safety of living things, which does not reduce the protection already present in nature.

--Some scales of measure candidates – they can be used as a complimentary set --

Air: Scale: % of days annually when <air> is <fit for all humans to breath>.

Water: Scale: % change in water pollution degree as defined by UN Standard 1026.

Earth: Scale: Grams per kilo of toxic content.

Predators: Scale: Average number of <free-roaming predators> per square km, per day.

Animals: Scale: % reduction of any defined [Living Creature] who has a defined [Area] as their natural habitat.

Many different scales can be candidates to reflect changes in a single critical factor.

Environmentally Friendly is now defined as a ‘Complex Attribute,’ because it consists of a number of ‘elementary’ attributes: {Air, Water, Earth, Predators, Animals}. A different scale of measure now defines each of these elementary attributes. Using these Scales we can add corresponding Meters, benchmarks (such as Past), constraints (such as Fail), and target levels (such as Goal) to describe exactly how Environmentally Friendly we want to be.

Level of Specification Detail. How much detail you need to specify, depends on what you want control over, and how much effort it is worth. The basic paradigm of Planguage is you should only elect to do what pays off for you. You should not build a more detailed specification than is meaningful in terms of your project and economic environment. Planguage tries to give you sufficient power of articulation to control both complex and simple problems. You need to scale up, or down, as appropriate. This is done through common sense, intuition, experience and organizational standards (reflecting experience). But, if in doubt, go into more detail. History says we have tended in the past to specify too little detail about requirements. The result consequently has often been to lose control, which costs a lot more than the extra investment in requirement specification.

4 Language Core: Scale Definition

This section discusses in more detail the specification of Scales using qualifiers.

The Central Role of a ‘Scale’ within Scalar Attribute Definition. The specified Scale of an elementary scalar attribute is used (re-used!) within all the scalar parameter specifications of the attribute (that is, within all the benchmarks, the constraints and the targets). In other words, a Scale parameter specification is the heart of a specification. A Scale is essential to support all the related scalar parameters: for example Past, Record, Trend, Goal, Budget, Stretch, Wish, Fail and Survival. (As these parameters specify the levels using the Scale.)

Each time a different scalar level parameter is specified, the Scale specification dictates what has to be defined numerically and in terms of Scale Qualifiers (like 'Staff = Nurse'). And then later, each time a scalar parameter definition is read, the Scale specification itself has to be referenced to 'interpret' the meaning of the corresponding scale specification. So the Scale is truly central to a scalar definition. For example '*Goal [Staff = Nurse]* 23%' only has meaning in the context of the corresponding scale: for example '*Scale: % of defined [Staff] attending the operation',* Well-defined scales of measure are well worth the small investment to define them, to refine them, and to re-use them.

Specifying Scales using Qualifiers. The scalar attributes (performance and resource) are best measured in terms of specific times, places and events. If we fail to do this, they lose meaning. People wrongly guess other times, places and events than you intend, and cannot relate their experiences and knowledge to your numbers. If we don't get more specific by using qualifiers, then performance and resource continues to be a vague concept, and there is ambiguity (which times? which places? which events?).

Further, it is important that the set of *different* performance and resource *levels* for different specific time, places and events are identified. It is likely that the levels of the performance and resource requirements will differ across the system depending on such things as time, location, role and system component.

> Decomposing complex performance and resource ideas, and finding market-segmenting qualifiers for differing target levels is a key method for competing for business.

Embedded Qualifiers within a Scale. A Scale specification can set up useful qualifiers by declaring embedded scale qualifiers, using the format 'defined [<qualifier>]'. It can also declare default qualifier values that apply by default if not overridden, 'defined [<qualifier>: default: <User-defined Variable or numeric value>]'. For example, […default: Novice].

Additional Qualifiers. However, embedded qualifiers should not stop you adding any other useful additional qualifiers later, as needed, during scale related specification (such as Goal or Meter). But, if you do find you are adding the same type of parameters in almost all related specifications, then you might as well design the Scale to include those qualifiers. A Scale should be built to ensure that it forces the user to define the critical information needed to understand and control a critical performance or resource attribute. This implies that scale qualifiers serve as a checklist of good practice when defining scalar specifications, such as Past and Goal.

Here is an example of how locally defined qualifiers (see the example in the Goal specification) can make a quality specification more specific. In this example we are also going to show how a requirement can be made conditional upon an event. If the event is not true, the requirement does not apply.

First, some *basic definitions* are required:

Assumption A: ***Basis*** *[This Financial Year]: Norway is still not a full member of the European Union.*

EU Trade: ***Source****: Euro Union Report "EU Trade in Decade 2000-2009".*

Positive Trade Balance: ***State*** *[Next Financial Year]: Norwegian Net Foreign Trade Balance has Positive Total to Date.*

The Planguage parameters {Basis, Source, & State} are in ***bold*** *text for readability of this example.*

Now we apply those definitions below:

Quality A:

Type: Quality Requirement.

Scale: % by value of Goods delivered that are returned for repair or replacement by consumers.

Meter [Development]: Weekly samples of 10,

[Acceptance]: 30 day sampling at 10% of representative cases,

[Maintenance]: Daily sample of largest cost case.

Fail [European Union, Assumption A]: 40% <- European Economic Members.

Goal [EU and EEU members, Positive Trade Balance]: 50% <- EU Trade.

Some of the user-defined terms used here (like EU Trade) are more fully defined in the example above this one.

The Fail and the Goal requirements are now defined partly with the help of qualifiers. The Goal to achieve 50% (or more, is implied) is only a valid plan if ‘Positive Trade Balance’ is true. The Fail level requirement of 40% (or worse, less, is implied) is only valid if ‘Assumption A’ is true. *All* qualifier conditions must be true for the level to be valid.

5 Principles: Scale Specification

Here is a set of principles to help summarise the key points about Scales – see below.

The Principle of 'Defining a Scale of Measure'

If you can't define a scale of measure, then the goal is out of control.

Specifying any critical variable starts with defining its units of measure.

The Principle of 'Quantification being Mandatory for Control'

If you can't quantify it, you can't control it[1].

If you cannot put numbers on your critical system variables, then you cannot expect to communicate about them, or to control them.

The Principle of 'Scales should Control the Stakeholder Requirements'

Don't choose the easy Scale, choose the powerful Scale.

Select scales of measure that give you the most direct control over the critical stakeholder requirements. Chose the Scales that lead to useful results.

The Principle of 'Copycats Cumulate Wisdom'

Don't reinvent Scales anew each time – store the wisdom of other Scales for reuse.

Most scales of measure you will need, will be found somewhere in the literature, or can be adapted from existing literature.

The Cartesian Principle

Divide and conquer said René – put complexity at bay.

Most high-level performance attributes need decomposition into the list of sub-attributes that we are actually referring to. This makes it much easier to define complex concepts, like 'Usability', or 'Adaptability,' measurably.

The Principle of 'Quantification is not Measurement'

You don't have to measure in order to quantify!

There is an essential distinction between quantification and measurement.

Be clear about one thing. Quantification is not the same as Estimation and Measurement.

"I want to take a trip to the moon in nine picoseconds" is a clear requirement specification without measurement."
The well-known problems of measuring systems accurately are no excuse for avoiding quantification – Quantification allows us to communicate about how good scalar attributes are or can be – before we have any need to measure them in the new systems.

[1] Paraphrasing a well-known old saying.

The Principle of 'Meters Matter'

Measurement methods give real world feedback about our ideas.

A ‘Meter’ definition determines the quality and cost of measurement on a scale; it needs to be sufficient for control and for our purse.

The Principle of 'Horses for Courses'[2]

Different measuring processes will be necessary for different points in time, different events, and different places.[3]

The Principle of ‘The Answer always being ‘42’’[4]

Exact numbers are ambiguous unless the units of measure are well-defined and agreed.

Formally-defined scales of measure avoid ambiguity. If you don’t define scales of measure well, the requirement level might just as well be an arbitrary number.

The Principle of ‘Being Sure About Results’

If you want to be sure of delivering the critical result – then quantify the requirement.

Critical requirements can hurt you if they go wrong – and you can always find a useful way to quantify the notion of ‘going right; to help you avoid doing so.

6 Conclusions

This paper has shown how Planguage specifies performance scales of measure, and how such scales can be used to define benchmarks, targets and constraints. Further discussion and additional information about Planguage can be found in ‘Competitive Engineering’ [3].

References

1. Gilb, T.: Principles of Software Engineering Management, p. 442. Addison-Wesley, Reading (1988), ISBN 0-201-19246-2 (See particularly page 150 (Usability) and Chapter 19 Software Engineering Templates)
2. Various free papers, slides, and manuscripts on, `http://www.Gilb.com/`
3. Gilb, T.: Competitive Engineering, p. 474. Elsevier Butterworth-Heinemann, Amsterdam (2005), ISBN 0-7506-6507-6 (An Indian edition is also available)

[2] ‘Horses’ for Courses is UK expression indicating something must be appropriate for use, fit for purpose.

[3] There is no universal static scale of measure. You need to tailor them to make them useful.

[4] Concept made famous in Douglas Adams, The Hitchhiker’s Guide to the Galaxy.