May 2, 2006

AGILE

Now What?

ABSTRACT

Agile Software Methods are a great relief and a big step in the right direction compared to traditional development methods. Agile Software Methods biggest weakness is in it's complete lack of systematic control towards satisfying Product Quality and Stakeholder Value Requirements.

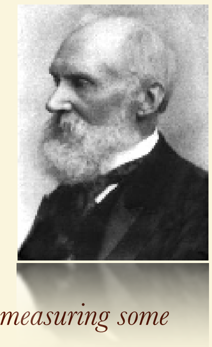
As a result of lack of control, Agile methods are typically only used for smaller less-critical projects. The challenge is rooted in the software industry's dangerously limited view of requirements. This paper aims to shed some light on different types of requirements, requirements that are better suited for controlling both small and large projects systematically. It will go on to suggest some simple principles that the Agile community can adopt to regain systematic control, while still being Agile. If you choose to follow these simple principles you will be much better armed to tackle not only small projects, but also to expand the use of Agile principles to any size project. Most of the SW industry is suffering with unhappy customers, bad quality, huge delays and often project cancellations. By using these principles our customers are achieving huge improvements, happy customers, no delays. I don't expect you to take my word for it, but please investigate it.

This paper is directly inspired by work authored by Tom Gilb, some parts are co-authored with Tom.

Author Bio.

Kai Gilb has been an independent consultant, teacher and author, since 1992. He mainly works with multinational clients; helping improve their organizations, and their systems engineering methods.
He is working on his book entitled 'Evo'.

Qualities -What are they? Why and How to specify them?

> **Do you know something about your trade?**
>
> 
>
> *"In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it.*
>
> *I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it;*
>
> *but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind;*
>
> *it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be."*
>
> *Lord Kelvin, 1893*

A product quality describes 'how well' the function of a product is experienced by another product or a person. Quality is experienced as a variable, from colder to warmer, from difficult to easy, this variation can, and I think normally should, be described quantitatively.

One quality, describes one of many attributes of a function. There seems to be an infinite number of qualities related to any products. Product Quality Requirements should focus on the qualities that enable the product to satisfy the Stakeholder Values, and that are critical for the success or failure of the project.

Even though all quality levels are created by a particular 'means', quality attributes can be articulated independently of any particular means (solutions, designs, architectures).

In order to find the best Solutions to satisfy a set of Product Quality Requirements, the Qualities should be separated from the Solutions (to better understand why, see the use of IET later in this paper.)

The Product Qualities that the Stakeholders desire, are mostly independent of the Solutions needed to satisfy them. When soliciting requirements, do not allow the Stakeholders to require particular technical Solutions, they should require levels of Product Quality, Stakeholder Values and Price. Leave it to engineering to find the optimum Solutions to satisfy the set of requirements.

A particular quality might need to be described in terms of a 'complex' concept, that is, consisting of multiple elementary quality dimensions. User-friendly, for example, can be broken up into User-friendliness.Learn, User-friendliness.Intuitive etc. Love is a many-splendored thing too, like many high level requirements!

Quality levels can be measured in practice, even during development of a product. When you measure the critical product qualities during development, you can learn from real time experience, and consequently steer the project towards success. If you don't, you cannot act on emerging facts of your own reality! If you have quantified the critical qualities of your project, you can to some degree trade off between different product qualities, and also trade off between development resources and product qualities. Many critical product qualities, like reliability and security, have exponentially expanding costs as they get closer to perfection. If you want to push the state of the art of a quality closer to perfection, you must be willing to pay the exponentially expanding costs.

Two chairs, same Function, different Product Qualities, and different Development Resource Budgets.

Stakeholder Values - What are they? Why and How to specify them?

We can describe a product by specifying the set of its functions, qualities, and solutions. Similarly we can describe a stakeholder with Stakeholder Functions - what they do, Stakeholder Values - how well they need or want to do something, and with Solutions - how they do it, (hopefully their solutions are or will become our products).

What the Stakeholder wants to do. (Stakeholder Functions)

How well the Stakeholder wants to do it. (Stakeholder Values)

How the Stakeholder does what she does so well. (Systems with Product Functions, Product Qualities and Solutions)

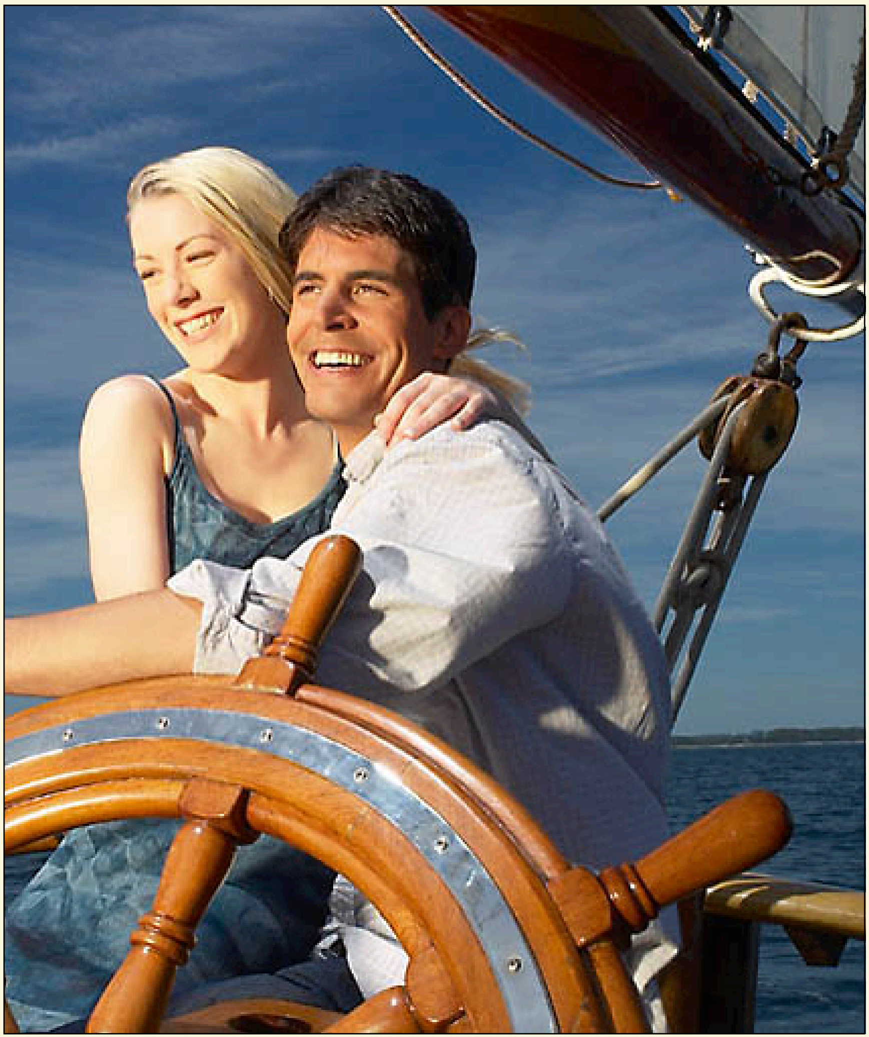
To understand the product functions and product qualities of our product, we had better understand what our stakeholders do (Stakeholder Functions) and where they want improvements (Stakeholder Values). We can then customize our products to satisfy the Stakeholder Values.

Success is not developing product functions. Success is when Stakeholder Values have been satisfied.

The Principle of: You don't compete on Function, you compete on Quality!

A 'buyer' will choose among several products that all have similar function. Their choice will be the product, that within their budgets, has the qualities that best satisfies their Stakeholder Values.

- Kai Gilb



Stakeholder Function: *What the Stakeholder wants to do. (Sail a boat)*

Stakeholder Value: *How well the Stakeholder wants to do it. (feel safe, have fun, feel free, relax)*

Product Function: *What the product must do. (float on water, moved by wind)*

Product Quality: *How well the product must do it. (MTBF, max wind, style, cost of ownership, ease of control)*

FIRM - case study -part 1: Product Quality Specification.

First a look at how FIRM (Future Information Research Management) specify their requirements and some initial results FIRM achieved in their first 6 month cycle of using Agile Evo.

Here is one of 25 Product Quality improvements.

Usability.Productivity:

Quantification **Scale: Time for the system to generate a survey.**

Past level [release 8.0] **7200 sec**.

Status level [release 8.0]**15 sec.**

That is an incredible

48000%

improvement to a product quality that is critical for FIRM's stakeholders.

MORE HIGHLIGHTS

Name tag	Quantification Scale	Past	Status
Usability.Productivity	Time to set up a typical specified Market Research-report.	65 min	20 min
Usability.Productivity	Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness	The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Confirmit Web Services without any user documentation or any other aid.	15 min	5 min
Workload Capacity.Runtime.Concurrency	Maximum number of simultaneous respondents executing a survey with a click rate of 20 seconds and an response time < 500 milliseconds, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical].	250 users	6000

How to build on Agile methods to make them winners, not just hacking with a nice name.

Agile without Requirements have its place;-)

You are using Agile methods, you are freed from stale bureaucratic waterfall type methods, now what?

Understand your stakeholders, their values, and how your product will best satisfy these Stakeholder Values. Learn how to describe your product with Product Qualities, and Product Functions, without mixing in Solutions.

Keep the Solutions separate from the requirements, so you get freedom to choose and to engineer a set of Solutions that satisfy the complete set of requirements, from all the Stakeholders. We can use tools like Impact Estimation Tables (shown later) to quantitatively help us choose the solutions, and to choose the next development cycle (Evo Cycle) that give us the most benefits for our limited development resources.

Systematically learn from every development cycle, and apply what we learned to the next cycles.

Finally, we can keep fighting the bureaucrats at every stage. To plan a project as outlined above, can all be done on one single page, if you choose, use up to 10 pages, but you should not need any more.

Seven Agile Principles: to avoid bureaucracy and give creative freedom

(Kai's version of Tom Gilb's Ten Agile Principles)

1. Control projects by using a few quantified critical business (not technical) results. 1 Page total ! (not with stories, functions, features, use cases, objects, ..).

2. Align your project with your financial sponsor's interests!

3. Give developers freedom, to find out how to deliver those results.

4. Estimate the impacts of your solutions, on your quantified Product Quality goals (IET). Select solutions with the best impacts on the requirements compared to their development cost. Do the profitable ones first.

5. Decompose the workflow, into weekly (or 2% of budget) time boxes.

6. Change anything (Solutions, Architecture, Designs, Processes, Requirements etc.), as needed to satisfy the critical quantified Stakeholder Values and Product Qualities.

7. Involve the stakeholders, every week, in actually using increments, measuring their improvements, and in setting quantified goals.

Seven Agile Values

(Kai's version of Tom Gilb's Ten Agile Values)

Simplicity:

1. Plan for and focus all development activities, on delivering real stakeholder values.

Communication

2. For every weekly delivery cycle, estimate & communicate expected development costs and results, quantitatively.

Feedback

3. Deploy to stakeholders, in their environment, real improvements, every week.

4. Measure all critical aspects of the deployment.

5. Analyze deviation from estimates.

Act!

6. Change plans and actions to reflect weekly learning.

7. Next week, again deliver high priority Stakeholder Values. Don't wait!

Impact Estimation Tables (IET)

An Impact Estimation Table (IET) uses Requirements from one level and evaluates the Solutions at the next level (i.e. choose technical Solutions based on the Product Quality Requirements.) It can give significant help in prioritizing and choosing Solutions.

In our simple IET example, we evaluate Solution A, B or C against a set of Product Quality requirements.

Solution A is estimated to get us 10% of the way from our Past level, towards our Goal level, on our Usability Requirement scale. Solution A is estimated to get us 10% further away (negative side effect) from our Security Goal, and 50% closer to our Performance Goal.

Solution A is estimated to consume 20% of our Development Resources.

When we divide the Total Product Quality on the Consumed Development Resources, we can see that Solution C gets the most overall Product Quality, compared to Development Resources consumed.

A SIMPLE IET

	Solution A	Solution B	Solution C
Usability	10%	10%	0%
Security	-10%	20%	50%
Performance	50%	50%	-10%
Total Product Quality	50%	80%	50%
Development Resources	20%	10%	5%
Prod. Quality / Dev. Resources	2,50	8,00	9,98

IET = Priority

Choice, Team Communication,
Progress Tracking, Estimations

FIRM Paradigm Shift

"With EVO, our requirements process changed.
Previously we focused mostly on function requirements.
We realized that it's the product quality requirements that really separate us from our competitors."

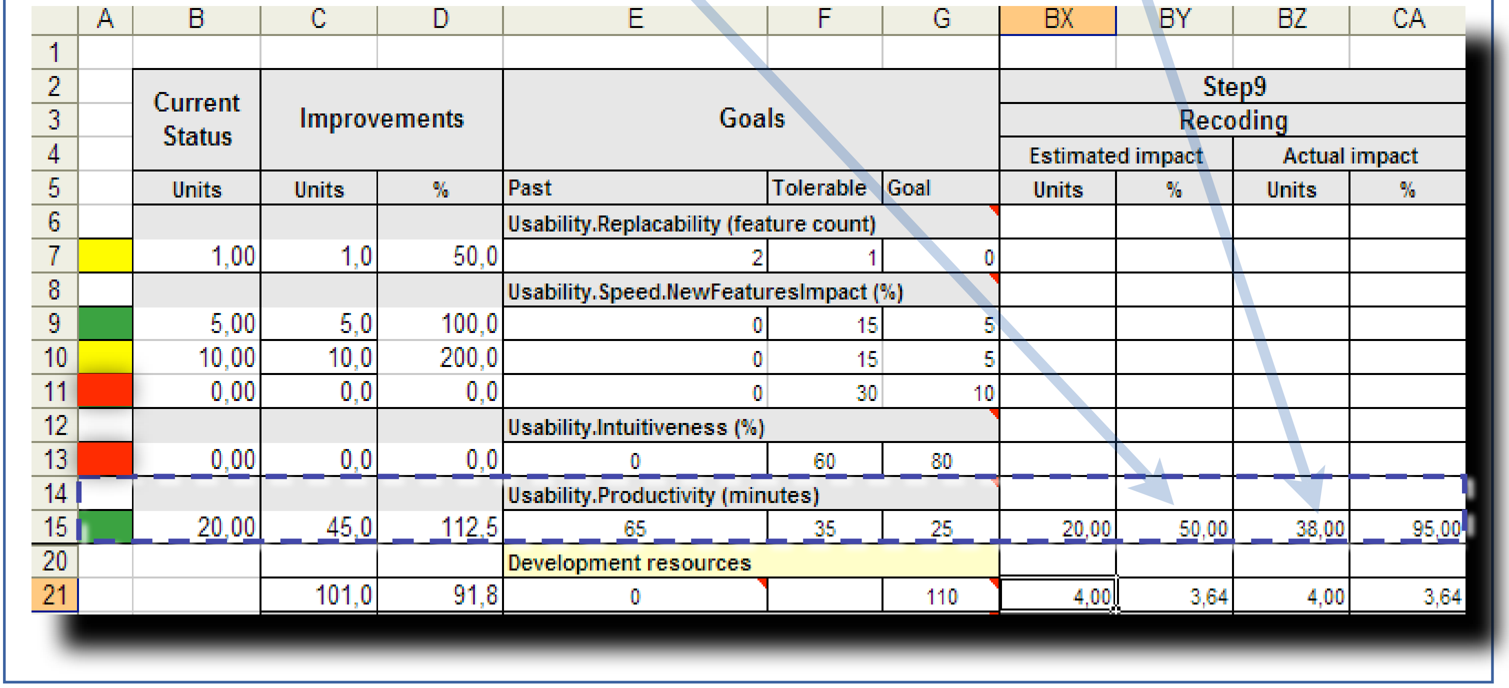
Trond.Johansen@firmglobal.com
Head of Project Management, FIRM A/S

> *"It looks like you have listened to the people that actually use the software daily and aimed to make it easier for them ... "*
>
> *- FIRM customer*

FIRM - case study -part 2: Evo

How FIRM uses Evo IET to choose their next Evo Cycle.

In the Evo IET below, FIRM estimates and tracks the impact that "Step9 - Recoding" has on their requirements. Their "Estimated impact" for "Step9" is that it will satisfy 50% of their Usability.Productivity requirement. The "Actual impact" shows that "Step9" actually fulfilled 95% of the requirement, it shaved off 38 min.



	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7	🟨	1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9	🟩	5,00	5,0	100,0	0	15	5				
10	🟨	10,00	10,0	200,0	0	15	5				
11	🟥	0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13	🟥	0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15	🟩	20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

A Simple Agile Process Description, that 'rule'!

(Kai's version of Tom Gilb's Ten Simplest and Best Agile Project Method)

1. From all the key stakeholders, gather and get agreement with them on the top few (3 to 20) most critical Stakeholder Values that the project needs to deliver. Give each Stakeholder Value a reference name (a tag) quantification scale and set a Goal level .

2. Determine, and get agreement from key stakeholders, about which Product Qualities the product must have to satisfy the Stakeholder Value Goals. Give each Product Quality a reference name (a tag) quantification Scale and set a Goal level .

3. Define 1 to 3 Budgets for your most limited Development Resources (for example; time, people, money, and equipment).

To navigate a boat, you measure where you are, your direction, monitor the weather and constantly adjust your course accordingly to arrive at your destination.
The same goes for agile evo projects.

Agile Project Management Policy

(Kai's version of Tom Gilb's Agile Management Policy)

All people involved in managing, developing, testing, delivering and maintaining the system, will be judged exclusively on the progress made towards achieving of the quantified requirements
versus the amounts of development resource budgets used.

The project team is empowered to do anything legal and ethical to deliver the quantified requirements within the development resource budgets. This includes finding their own work process and their own solutions, designs, tools etc..

The team will be paid and rewarded for benefits delivered in relation to cost.

4. Use an Impact Estimation Table (see FIRM case study above) to evaluate if the product, with the selected product qualities, will satisfy the Stakeholder Values. If point 4 seems too advanced, don't do point 4.

5. Find and evaluate Solutions suited to create a product with the specified quality levels. Consider using an Impact Estimation Table.

6. Divide the Solution into weekly evolutionary deliveries, that each give improvements towards Goal levels.

7. Develop and deliver a next week's partial-requirement delivery. Measure actual progress towards the Goal levels. Learn from every delivery cycle (what works, what does not work, our understanding of technology or stakeholder values, etc.), and change anything (Stakeholder Values, Product Qualities, Solutions, next Evolutionary Cycle, etc.) accordingly.

8. After each delivery cycle, write a one page report to project sponsors, content: 1. actual progress and 2. quantitative estimates for next cycle.

9. When all Goal levels are reached: 'Claim success and move on'

That's it, good luck!-)

- Kai Gilb