

Agile Value Scoring: Harnessing Change to enhance Customer Value

S M Kripanidhi

Overview

In this paper I am presenting a simple and practical two-stage numeric subjective feedback mechanism “Value Scoring”, which can be used in any Agile development method to improve actual value delivery to your stakeholders. We have used it in real projects, and the results are very satisfying.

Background

Software development was initially focussed on delivering lines of code to meet requirements. As complexity and size of software projects grew, there was a shift towards creative problem solving, and towards engineering paradigms.

This led to a series of challenges of the assumptions that formed the basis of traditional software engineering process frameworks and standards.

To address this change in the software solution development paradigm, Tom Gilb (www.gilb.com), first conceived his ‘Evo’ model in the late 60s, which was later formally published in 1976 [1]. The Evo model was to better enable management of such ‘solution ideation’ and ‘systems engineering’ in development of software solutions, evolutionarily.

The Evo model integrated the concepts of incremental and iterative software development with the PDSA approach (plan-do-study-act) [12] that mandated a feedback and learning cycle at the end of each iteration, and caused a cycle of continuous improvement, incrementally.

This resonates with what we today study as empirical evolution in ‘complex adaptive systems’ [10].

The Evo Model for software development offers a comprehensive management framework for empirical evolution of a software solution by a cross-functional team working collaboratively with customers, striving to

incrementally evolve the right solution (one that provides value) for their customers, through active iterative feedback, learning and change.

The fast changing market and technology scenarios made traditional sequential models of software development seriously handicapped in their ability to deliver value for money to buyers of software solutions [11].

With a focus on ‘customer value’ and ‘software economics’, the emerging agile methodologies initially in part inspired by Tom Gilb’s Evolutionary Model-Evo [13], have become a popular alternative to traditional software development.

The key focus of agile principles [2] is to enhance value delivery to customers by:

- (a) Development teams actively collaborating with customers to better understand their value perceptions, so that they can meet customer expectations correctly.
- (b) Focussing on communication, relationships and motivation to enhance people productivity in an empowered, self-organizing, cross functional, collaborative team environment
- (c) Harnessing change to enhance value for customers
- (d) Creating an empirical process environment that yields continuous improvements through iterative feedback, learning and change.

While in theory, the rationale behind each of the concepts in the ‘agile value system’ is practical and sound; its ability to deploy successfully, and work in organizations with a traditional mindset, has seen some roadblocks.

Understanding “customer value” in a software development context has been the main shortcoming for decades [11]. Since customer value is a key imperative in agile, lack of this understanding can make all other agile concepts unproductive.

I am attempting to resolve this key problem by a simple set of proven practices that agile teams can implement in their agile projects, to significantly enhance their ability to anticipate and understand customers’ value perceptions correctly and to be able to deliver the right value at the right time to their customers, incrementally. This will hopefully give the agile customers, a higher return on their investments.

What is ‘Customer Value’?

The simplest definition of Customer Value I found was to say:

Value is what the ‘customer’ enjoys paying for

Sometimes the customer concept is extended from ‘paying customer’, to ‘the receiver of a defined service or product; the next group in a value chain’. It is

our belief and that of for example Gilbs' and others [14] that the well-intended 'customer' (not just 'user') concept is a serious mistake. The realistic and applicable concept is 'stakeholder', and there are many of them – dozens! 'Stakeholder Valued Value' should be the realistic objective of development.

'Customer Value' is also a very complex and volatile term to get a tangible definition that can hold good universally. What the customer values and how much value they perceive in a product or service can change with time and context. Different customers can attach different value perceptions on the same product or service at different points of time.

When I am hungry, I may attach a higher value for food that I eat in a restaurant while I may not attach a higher value to the same food at the same restaurant when I am not hungry or not in a mood to eat food. I may also perceive higher value for the food I ate in the restaurant, if the service at the hotel was pleasant. Aspects of convenience, responsiveness, assurance through past good experiences, looks and feel of the service environment also affect customer value perceptions [3]

Value perceptions of customers are affected by what they expect from the product or service they buy. Therefore in 'Service Quality Management' it is said that "anticipating, understanding, meeting and exceeding customer expectations" **is** delivery of 'Customer Value'.

In the software development context we always go by what the customer says he wants. This is treated as the explicit requirement specifications. Many times even when software is delivered exactly as per specifications, it may not deliver value. In requirements engineering it is noted that "many times customers do not know what they want" [4]. Therefore the requirements engineer has to explore, evolve, elicit, elaborate, understand and validate the integrity of customer requirements during a study. The customer perceives and realizes value only when he gets what he truly needs. But this may not be what he conveys as his requirements.

So, we have to understand, and close the gap between, what the 'customers say they want', and 'what they expect to get for the money they pay'. If they get what they expect, and it serves their purpose, they realize or perceive Value in it. If they get what they said they had wanted, but it does not serve their purpose or meet their expectations, they do not perceive or realize Value in it. [5]

Harnessing Change to enhance Customer Value

Incremental and iterative software development is said to flatten the '*cost of change curve*' in software development. [6] In agile, we go one step further and harness change to enhance customer value. [2]

How is this done? Here is a storyline I use to illustrate this concept.

John is a tall handsome guy and is very particular about what he wears. He spends a lot of time in the malls finding the exact kind of shirts and pants he aspires to wear. One day he walks into a boutique and is scanning the shirts on display. Finally he finds one that closely matches what he wanted. He tries it on and decides to buy it as he finds the price reasonable.

The boutique salesman tells him that he has to provide a feedback on the purchase before he leaves.

The feedback session goes thus:

Salesman: "Did you get exactly what you wanted, Sir?"

John: "Sort of yes, may not be the perfect shirt I was looking for".

Salesman: "What would you want changed in this shirt?"

John: "The colour of it...I wanted this kind of blue (he points to another shirt) but that one is not my size. I also wish that this shirt did not have the pocket. I don't like pocket on my shirts".

Salesman: "Can you wait for a day. I will have a shirt made exactly of the kind you desire, if you please. It will cost you 15% more than this shirt. I can have that delivered to your home tomorrow".

John: "That would be great. At least I will get what I wanted". (John pays the money in full and leaves).

Next day at the same time, a courier delivers his new shirt and he finds it exactly as expected. Within the parcel, he also finds a letter saying that he is welcome to call them if he is not satisfied with the shirt or wishes to have it further modified anytime later. The boutique also requests feedback on the shirt delivered to John.

A week later the salesman calls up John to find out if he got his shirt right. He also offers to customize it for him further, if John finds a need for it.

John: "Thanks buddy. The shirt seems good but it would have been better if it had a formal collar and white buttons instead of these dark blue ones. That would have made it real close to the perfect one I aspire for".

Salesman: "Thank you Sir. I will surely have this done. If I can collect your shirt tomorrow at your residence, I will have it modified and deliver it to you the next day. It may cost you another 15%".

John: "Hey! That's great. I am truly enjoying your service. Go ahead and do this for me".

The shirt is collected, modified and returned to John the next day and its charges are paid for by John. He also thanks the salesman in ensuring that he got a wonderful shirt, exactly what he wanted.

A week later the salesman calls up John again for feedback on his shirt. This time John tells him that he is very happy with the shirt and it is near perfect.

The salesman coaxes him again if he wished something was better.

John: “I have one more point, if this shirt was a little longer, it would have tucked into my trousers better. It tends to slip out of my waist sometimes though it not a major problem”.

Salesman: “I would love to have it modified for you to a few inches longer. I will also shape it up a little so it stays tucked well. I can do that for you for a small fee”.

John: “Wonderful. That would be really really great”.

Finally John gets a shirt that he always wished for that ultimate occasion. The perfect one! He had no other shirt that so perfectly matches what he had always aspired for. He loves going back to this boutique for all his new buys. He says he finds real “value for money” only in this shop.

That was the story.

But in hindsight, can we use the same concept to delight our customers in an agile project?

Yes, we can .

How to build ‘Customer Value’ in Agile?

In an agile project, the teams build and deliver customer value incrementally and iteratively on a PDSA approach (Plan-Do-Study-Act). In the ‘concept ideation phase’ [7] of a solution development lifecycle, no one knows what will meet customer expectations upfront. The customers may be as ignorant of what they may really need, as are developers. So here is how we build ‘Customer Value’ in Agile.

Harness Change to Co-create Value (*John’s boutique style*)

At the end of each iteration, agile teams have an iteration review with the customers. They show the customers what they built in the iteration and seek acceptance from the customers on each story. They demonstrate the running of each acceptance criteria, story by story, as was specified in the story cards / product backlog.

The customers now get what they *said* they wanted in the stories. However, this may not be exactly what they *expected*. And unless they get what they really, but perhaps implicitly, expected, and it really works for them, they will not realize genuine value from it.

Specifications are met; but expectations may not have been met.

How do we close this gap? To do this we have to understand the reasons why developers are unable to meet real *customer expectations* although they are able to meet *customer requirements specifications* consistently [9]

Here is the ‘Value Scoring’ practice that we can follow to bridge this gap.

After the iteration demo and review, we will seek an additional set of feedback from our customers. On each story that they accepted (all tests passed), we ask them to rate their level of satisfaction on a 10 point scale, as to what extent each story met their real expectations and served their purpose.

This will reveal the customers' hidden aspirations. The customers might also learn what more they may need to serve their purpose better, only *after* using what we give them. They might like to have the *features built with better performance and quality levels so that the features or functions are more useful to them* [8]. They expected to have such improvements delivered with the current functionality, although they did not realize that before.

Without performance or quality improvements, the features we gave them, may not meet their *original purpose* (solution to their problem) for which they wanted it in the first place. Note that many stories ('functions', features, use cases) are not core business functionality. They are actually 'design' intended to improve some qualitative aspect of the system, like usability or security. The delivery may simply tell that the design was not as effective as they hoped.

The team's goal in an iteration is to have customers rate every story to a minimum score of 9 out of 10 in having met their expectations fully, and fitness for its purpose. For all the stories that the customers rate less than 9, the teams should explore how they could enhance its value.

For each story rated less than 9 by the customers, the team poses a question to the customers:

“What should we have done differently or additionally in this story to be able to get 10 out of 10 from you on this story?”

The Customer is usually happy narrating his woes, or clarifying his expectations, point by point. The team notes down each point as a new wish-list and considers adding them as new stories in the Product Backlog, or at least considers what to do and if it is worth it. Teams may iterate one story through several iterations to get 10 out of 10. But they are enhancing Customer Value every iteration, as their score on the feedback improves with each new iteration of the story, and the Customer is hopefully enjoying paying for it through those iterations.

I was very happy to hear what the developers said after this practice was piloted in their agile projects for over a couple of iterations: [anc]

- We always wanted to save money for our customers. We never knew that they could enjoy paying us more for the changes they wanted.
- Our customers are learning what they truly need, by trial and error through these iterative sessions and they are feeling very happy about it.
- We can now tell you why they want, what they say they want.

- They enjoy giving us their satisfaction rating and love to suggest what we might do differently to get a better rating. These sessions seem to be the most useful and enjoyable for all of us.
- We never knew how little we knew about our customers and their needs. These sessions helped us understand our customers' expectations better.
- With the introduction of this new practice of customer feedback in iteration demos, we now see more customer stakeholders are attending it. Earlier we had only one customer attending our demo sessions, now more customer representatives come in to give us their wish-lists during demos. They seem very enthusiastic about this new approach.

And the **customers** too were raving about it. They said: [anc]

- The teams are now more responsive to our real needs
- They are now better able to anticipate what we expect from each story.
- Our collaboration with the team has significantly improved over the last few iterations, after we set up this practice.
- We collect feedback-rating from actual users of the features and give it to the team, along with their new set of wish lists, with a delay of one iteration.
- We never imagined that we could change, refine and learn so much more about our own true needs. This was a good R&D for us in our product feature planning phase.
- The teams were always keen to prove that they were getting all the stories right without any bugs. Now they are more empathetic towards what we say we want in terms of changes to those features.
- This is how we should build software. Only a few features but we get them to work perfectly, exactly the way we want, through several iterations of changing and refining the same feature.

This practice is also very useful for the teams to understand customer value perceptions in a context. Over time, this understanding gives them the ability to anticipate and understand their customers' expectations better. With such iterative value creation, the teams learn more about their customers and the customers too learn and discover more options for their own true needs. This mutual empathy, responsiveness and assurance enables creation and delivery of true value for customers in agile, collaboratively.

It also nurtures the true value system in agile software development and achieves customer delight evolutionarily.

I will be happy to receive your feedback and comments on this innovative 'Value Scoring' practice that significantly helps us within agile teams, to deliver real value to customers in agile projects. You can reach me at kripa@binaryessentials.com

Acknowledgement

I was deeply inspired by an article titled “**The Forgotten Side of Quality**” by **Jeff Patton** which was published in the weekly column of Sticky Minds in October 2007. Subsequent to this we had interesting discussions on this topic in agile discussion forums that finally led me to innovate and hone this practice for over a year now, in its current form, with the agile teams I coach.

The original article by Jeff Patton can be found at the following URL:
<http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=COL&ObjectId=12836>

References

[anc] These are real verbatim anecdotes that I recollect from conversations I have had as an agile coach with the team members and the customers in agile projects during team retrospectives. These were responses from them when asked if this process helped them at all and if so, how did it help?

[1] Chapter 10, Page 211 from :- Agile and Iterative Development: A Manager’s Guide by Craig Larman, Addison-Wesley Professional, 2003, ISBN-10: 0131111558

[2] Agile Principles at <http://agilemanifesto.org/principles.html>

[3] SERVQUAL – an Introduction to the SERVQUAL theory can be found at <http://en.wikipedia.org/wiki/SERVQUAL>

[4] Chapter 6 , Page 110, from:- Determining Project Requirements by Hans Jonasson, CRC Press, 2007, ISBN 1420045024

[5] VALiD (Value in Design) comprises the main findings of a three-year joint industry and academic research study called “Managing Value Delivery in Design.” The work, which was funded by UK Government and Industry, sought to increase customer satisfaction through a better, shared understanding of appropriate value systems (for the project process, the product and its performance) and standardised mechanisms that map and measure the flow and delivery of value within the design solution. More details at <http://www.valueindesign.com/principles/principles.htm>

[6] Examining the Agile Cost of Change Curve by Scott W Ambler <http://www.agilemodeling.com/essays/costOfChange.htm> . More detailed explanation on this can also be found in the book : Agile Modelling by Scott W Ambler, John Wiley & Sons Inc, ISBN 0-471-20282-7

[7] What happens in a Concept Ideation Phase is described briefly here: <http://www.problemistics.org/courseware/toolbook/ideation.html>

[8] To understand how the same feature built differently can add more value to the customers I would strongly recommend this article: “Measurable Value

with Agile by Ryan Shriver” published in Overload 89, A Magazine from ACCU– Feb 08 issue. Accessible at <http://accu.org/var/uploads/journals/overload89.pdf>

[9] 5G-Model for Stakeholder Value Engineering by S M Kripanidhi, Nov 2006, Binary Essentials accessible at <http://www.binaryessentials.com/cve.pdf>

[10] A description of what is Complex Adaptive Systems in the context of Agile Development can be found at <http://www.targetprocess.com/blog/2008/11/software-development-is-complex.html>

[11] Standish Group Chaos Report net.educause.edu/ir/library/pdf/NCPO8083B.pdf

[12] The PDCA Cycle described at <http://en.wikipedia.org/wiki/PDCA>

[13] Chapter 10 on Evo : Agile and Iterative Development: A Manager’s Guide by Craig Larman, Addison-Wesley Professional, 2003, ISBN-10: 0131111558 and Craig Larman’s paper co-authored with Victor R Basili, titled “Iterative and Incremental Development – a brief History” published in IEEE Computer Magazine, June 2003 issue.

[14] Agile, now what ? by Kai Gilb, May 2006 accessible at http://www.gilb.com/tiki-download_file.php?fileId=30