Experiences of Implementing a Value-Based Approach to Software Process and Product Assessment

PASI OJALA
Nokia ltd.
Hintamutka 17 A 6, 90650 Oulu
FINLAND

Abstract: - This study tries to advance and define the concepts, principles and practical methods of a value-based approach, involving definition of the essential concepts of value, cost and worth in software development. These definitions originate from the Value Engineering (VE) method, originally applied and developed for the needs of the production industry. Therefore, for the purposes of this study these concepts are firstly justified and secondly defined. In order to study and evaluate the value-based approach, a method called value assessment is developed and used when assessing processes and products. The results of industrial case show that even though there is still much to do in making the economic-driven view complete in software engineering, the value-based approach outlines a way towards a more comprehensive understanding of it. For industrial users value assessment seems to give a practical help for handling cost and profitability related challenges.

Key-Words: - Software process and product improvement, assessment, value, worth, cost and Value Engineering.

1 Introduction

The fundamental goal of all good design and engineering – to create maximal value added for a given investment – has become vital for many software companies. Boehm notes that there are many dimensions in which “value can be assessed”, from monetary profits to the solution of social problems. He continues: “Software economics is the field that seeks to enable significant improvements in software design and engineering through economic reasoning about product, process, program, and portfolio and policy issues.” [1]

Past work in the software engineering field has focused largely on costs, not on benefits, thus not on value added; nor are current technical software design criteria linked clearly to value creation [1]. This is rather surprising because Value Engineering (VE) as a theory already has a forty- to fifty-year history [3]. It has been used in several different situations in several different industrial branches [5].

Perhaps the slow approval for Value Engineering in software engineering has something do with the business’s maturity. When companies do not see value improvement as a priority, they focus mainly on “technical design.”

Grady has outlined a model for the value of software development process improvement. This model is made from the development organization’s point of view and can be presented as follows (1) [6]:

\[ \text{Value} = \text{Benefit} - \text{Cost} \]

Grady emphasizes that his model offers four benefits as answers to common questions. Will an improvement in process:
- Get you more/better products? (Product Capability)
- Get you products sooner? (Time to Market)
- Get you products to meet commitments? (Timeliness)
- Help make your products long-lasting and easy to evolve? (Product Evolution) [6]

Grady also notes that Product Capability, Time to Market, Timeliness and Product Evolution affect our decisions about how much benefit (or worth) we get. In relation to costs, the more Development, Rework or Knowledge Recovery we do the more we create costs in our processes. [6]

Grady’s model is based on improving value in software processes. It ties together software processes and products, but it neglects value in software products. Boehm also sees that there has been progress over the years in integrating some value-oriented perspectives into software engineering.[2] However, he states that these approaches have generally been treated as “add-on band-aids” to baseline software engineering principles and practices including:
- Requirements engineering
- Architecting, design and development
• Verification, validation, planning and control
• Risk, quality and people management
• Principles and practices. [2]

Tallon et al. have confirmed that process-level impacts can be used to measure IT business value. [11] According to their value-based model of IT, business value can be derived from the impact of IT on processes and inter-process linkages within the value chain. Using their model they have explained IT business value at the organizational level.

Grady, Boehm and Tallon et al. see value from a process point of view [2, 6, 11]. This is a good start but not necessarily enough. In successful software engineering we should take the product point of view into account more clearly, because a product is usually what the customer is buying, and the more worth they see in it the more income the company gets. All companies which are able to calculate value (Value = Worth/Cost) for their manufactured components have a more powerful tool at their disposal, because the product value will represent the customer’s point of view more clearly. In the product value formula, “Worth” describes how much worth all product components give to the customer and “Cost” what amount of resources the company has used as costs in order to make the components in question.

In a successful business relationship both the customer and the manufacturer points of view need to be taken into account. If both feel that trade is worth doing, it is usually done. This applies to processes, products, and services. It is not also enough to show that a certain process or product is more valuable than another. To be effective, the software process and product improvement itself should also give value, which is neglected in Grady’s, Boehm’s and Tallon et al.’s research [1, 2, 6, 11]. If the value of the improvement itself is not taken into account properly, the justification for the management’s use of resources for improvement work continues to be poor. Capers states that “comparatively little solid, empirical data is being published on three important topics:

• What does it cost to improve software processes?
• How long will it take to make tangible improvements?
• What kind of value can be expected in terms of better quality, productivity, or user satisfaction?”[4]

According to the outlined theory of successful software engineering, we need to concentrate equally on quality, timeliness and value. At the product level the customer evaluates our products from the quality point of view to find out how good the product is, from the worth point of view to evaluate how he would benefit from using it, and from the timeliness point of view to evaluate whether he needs it at this particular moment. If we have effective processes producing reliable, good quality products at a reasonable cost and at the right moment, we form our business on a healthy basis.

2 Value-Based Approach

Using the framework presented by Koskela & Huovila, the value-based approach is understood in this paper as a process.[7] The main principle of this process is to eliminate value losses in software development, products, processes and SPI. It uses economic-driven tools, which are based on economic studies including, for example, the areas of cost estimation, cost calculation (for example ABC and life cycle costing) and investment calculation. The value-based approach prefers calculating costs instead of estimating them, and also considers software development and SPI as investments, on which it is possible to spend too much money. In practice, the value-based approach takes care that the customer requirements are met in the best possible manner, ensuring quality, timeliness and value in products as well as in processes, over their entire life cycle. In particular, the aim of ensuring quality connects it to the other methods aiming for quality improvement.

The value-based approach also indicates a clear dependency between the process and products. It sees that we need to develop and optimize process activities so that processes produce the products needed. Furthermore, it sees that we must analyze products in order to reveal problems in processes and develop processes from the product point of view as well. This is vitally important, especially for companies respecting customer opinions and aiming to optimize costs in their processes, because the customers are the ones paying for the products and product-related services, and companies have to allocate all costs to products to be able to price them. The happier the customer is, the more worth he sees in buying the products from us. It is also clear that when we know our process and product costs, worth and value, our ability to estimate, budget and control future risks will improve significantly.

3 The Value Engineering Process

Even though there are several definitions in the literature for the VE process, they all have similarities. Generally, they state that VE collects and analyzes value-related information, to create new ideas using the analyzed results and to evaluate and
further develop them into a meaningful package, with the reduction of costs or the increase of worth and improvement of value as ultimate goals.

In practice, the improvements developed in VE process are the result of recommendations made by a multidisciplinary team representing all the parties involved in the subject studied, and led by a facilitator. Development ideas are systematic efforts to improve the value and optimize the life cycle cost of a function or facility. It is vitally important that the VE team has technical as well as cost-accounting knowledge. This paper uses VE process as presented in earlier research by Ojala [10]. Therefore, the phases included to VE process include orientation, information, creativity, evaluation, development and presentation.

4 Value Assessment for Processes and Products

There are four ways to enhance a standard software process assessment using VE [8, 9, 10]. The first possibility includes an addition of defined VE process into the existing process models of used capability assessment method (for example in CMMI or SPICE).

The second possibility covers Value Assessment for processes defined in used process model. The main idea of this enhancement is to run through all defined VE phases and as part of it calculate costs, worth and value for each assessed process existing in used process model.

The third possibility includes Value Assessment for processes without process model. The purpose of this enhancement is to find out from company’s own defined process descriptions all process practices which are then examined from cost, worth and value point of views using VE process.

The fourth possibility includes Value Assessment of a product. This enhancement examines Value of product components and requirements and reveals value improvement possibilities in them.

5 Value Assessment for Processes and Products: Company A

5.1 Background

Value assessment was implemented in Company A in fall 2005. Because company did not know whether its cost accounting would be able to provide the necessary cost data for all processes and product components, one purpose of the assessment was also to help to give information on how to build a cost accounting system for tracking process and product costs using identifiers.

During the first meeting, the assessor explained the purpose, content and plan of the value assessment to the personnel who were to be interviewed. The definition value=worth/cost was discussed, and it was seen as extremely important to find out which components of the product gave the best value to the vendor without neglecting customer needs. Since there were several customers for the product in question, it was not possible to include all customers in the assessment. Therefore, Company A decided to base worth calculations on ideal production costs, which represented the cheapest way of building a product or running a process.

The main problem presented by Company A was that there was no real understanding of all the product environments and their profitability. In other words, Company A did not know which operating systems gave the best value for the product and the company. Neither was it sure where the development and maintenance effort should have been focused. Some processes were attached to value assessment, because Company A saw that they were closely related to product development, and value information was needed for them as well.

5.2 Information

Company A had a strong interest in analyzing priorities, cost and worth in its product requirements and architectural product components for further product development work. However, when planning the assessment it was considered obvious that Company A does not have cost accounting system for architectural components, and simple estimation, not based on real calculated cost, was not considered to be good enough. Therefore it was decided that value indexes would be calculated for the prioritized requirements and component-level assessment would be postponed to the following year, when cost accounting would be able to produce the necessary component-level cost information. The value calculations for product platforms were done using estimates for following operating systems based on development difficulty:

- Windows, Linux, Solaris and HP (easy)
- QNK (difficult)
- UX (very difficult).

The assessor explained the idea of calculating value indexes for processes, both with and without a reference model. Company A preferred the idea of not using a reference model, because the personnel was dedicated more to the activities defined in the company’s own process descriptions than to those defined in an external reference model. The
Processes selected for value assessment included architectural design, design, code implementing and testing.

5.3 Function Analysis

All representatives of Company A found it easy to assign costs for the selected three product categories, because Company A followed all these costs every day. As well Company A had a system for following hours worked at the process and process practice levels.

Since Company A did not want to make precise cost and worth information public, the following figures are percentages. Worth is calculated using the ideal cost that Company A has defined for the platform or process, and costs are based on real costs. Figure 1 illustrates the calculated percentual worth and cost for platforms.

Figure 1: Worth & Cost in platforms (AV=average, C=customer, V=vendor)

Figure 1 shows that the platform category in which products are considered the easiest to implement creates more worth than costs. Generally, the situation is in control in this platform. However, in the platforms considered “difficult” and “very difficult”, the opposite situation is true. Therefore, in these platforms, the situation is not in control. When discussing these results with Company A, they explained that the amount of sales in the “easy” platform were significantly higher, and therefore the production costs were divided over larger amounts of sales than in the two other platforms. As an explanation, representatives of Company A also stated that employees had more experience of creating products for the “easy” platforms.

The worth and cost calculation results of value-related process assessment can be seen in Figure 2. Design and architectural design seem to be in balance with worth expectations. Company A uses the same level of resources for them as the worth that customers expect to get from them. However in coding and testing, the situation is not the same. Customers do not assign as much worth to them as they cost, which indicates a clear need to cut costs in these activities to improve value.

Figure 2: Worth & Cost in processes (AV=average, C=customer, V=vendor)

After calculating worth and cost, the assessor calculated value indexes. Platform-level value indexes (Figure 3) indicated that the easiest platforms produce the greatest value. Since the value indexes for the other platforms are below 1.0, these platforms do not produce as much money as they cost. Generally, it was recommended to Company A to avoid using a lot of resources on this kind of products where value is below 1.0. However, it was also advised that if the Company A wanted to move into new markets, it might occasionally be necessary to create poor value for a certain time. In Company A’s situation, this was not the case.

Value indexes for processes (Figure 4) clearly show that Company A creates most value in design and architectural design. However, Company A should start to look for value improvement possibilities mostly in coding and testing. These processes create more costs than worth.

Figure 3: Value in platforms

Figure 4: Value in processes
5.4 Creativity

Since value determination had been performed for both products and processes, it was decided that both aspects would also be brainstormed. In addition, it was decided that the requirements for a new cost accounting system would also be discussed. All participants were asked to list product-related improvement proposals first, process-related improvement proposals second and cost accounting-related improvement proposals third.

The main ideas were classified in three categories, and included:

- **Products:**
  - Someone should be responsible for discussing a move to easier platforms, with customers using “difficult” and “very difficult” platforms.
  - The company should announce that it will no longer make products for “difficult” platforms.
  - The company should not implement all new features in platforms which it considers “difficult”, and some features should be implemented significantly later.

- **Processes:**
  - The project managers and testing manager should organize a workshop in which the most time-consuming work practices would be listed.
  - Designers came to code inspections without preparation, and there was no clear procedure for how to act in inspections.
  - Module and security testing were part of each designer’s responsibilities, but they felt that they did not have enough training to implement these test procedures.
  - There was no nominated testing engineer for each project, and testing engineers used a great deal of time learning about the new project before testing.

- **Cost accounting:**
  - Accounting identifiers should be created to follow costs in all platforms and in main practices.
  - Reporting schedules, and templates should be created for cost accounting and value-monitoring needs.
  - The working hour tracking system should be improved, to include all value creation-related areas.

5.5 Evaluation

During the evaluation phase all the ideas presented were analyzed and evaluated. It was decided that there was no need to create weighted criteria in prioritizing improvement proposals. It was proposed that all of the ideas should be implemented, except the one suggesting that the company should announce that it would no longer support all platforms. This idea was not widely supported because it was considered to be against the company’s strategy and customer service principles.

5.6 Development

**Product-related value**
According to the benefit analysis, product-related benefits would be achieved if customers changed their platforms from “difficult” or “very difficult” platforms to easier ones. Some customers had already indicated that this would be possible in the near future, but Company A had not been active in supporting it. It was estimated that within a one year timeframe, 60 percent (AV=average, C=customer, V=vendor) of customers could change platform, to an “easy” one. It was further estimated that if not all the new, minor improvements were implemented, the costs involved in “difficult” and “very difficult” platforms would decrease by 25 percent. The total cost savings were estimated at around 50 percent.

**Process-related value**
In terms of the process-related improvement proposals, project managers and the testing manager organized workshops with their teams to discuss the most time-consuming work practices. Based on these workshops it was noted that:

- Designers came to code inspections without preparation, and there was no clear procedure for how to act in inspections.
- Module and security testing were part of each designer’s responsibilities, but they felt that they did not have enough training to implement these test procedures.
- There was no nominated testing engineer for each project, and testing engineers used a great deal of time learning about the new project before testing.

The value assessment team collected history information about code inspections, analyzed all information quantitatively, and formed an understanding of which kind of inspections lasted longest and shortest. After collecting and analyzing the supporting data package, the participants generated improvement proposals related to processes:

- Each design should be inspected by another designer, who should send design comments, before the inspection, to the project manager, who acts as a chairman in inspection meetings.
• Security testing should be given to test engineers, who have a better understanding of it.
• The test manager should organize module test training for designers and nominate test engineers for each project.
• Testing plans should be inspected by a test team before testing.

It was estimated that the proposed improvements would reduce coding costs by 10 percent over a one-year period. In testing, the cost reduction was estimated at around 15 percent.

**Cost accounting system**
The third selected value improvement area included the cost accounting system. Since Company A already had appropriate cost accounting software, it was considered possible to use it for the required cost accounting purposes. It was decided that the financial manager would create identifiers in the cost accounting system for each platform and defined process practice. It was calculated that it would take one person one week to implement the identifiers and train the needed bill approvers in the new practices.

5.7 **Presentation**
The results of this value assessment for processes and products, including cost accounting system improvement opportunities, were presented to the top-level management. Since the proposed improvements only reduced costs, the top-level management decided to put them into use.

Company A was satisfied with the results of value assessment. However, they announced that since there was no proper time-keeping and cost accounting system in place before the assessment, a new assessment, using the new information, would be carried out in the following year.

6 **CONCLUSIONS**
The value-based approach to software engineering appreciates the clear dependency between process and product. It helps in developing and even optimizing process activities, while ensuring that processes still produce the services and products needed. It analyzes products to reveal problems in processes, and develops processes from a product point of view. This is vitally important, especially for companies who respect customer opinions and aim to optimize costs in their products and processes. Customers pay for products and services, and companies have to allocate all costs to products to be able to price them. The happier the customer is, the more worth he will see in buying a given product. It is also evident that when we know our process and product costs, and worth and value, our ability to estimate, budget and control future risks will increase significantly.

Perhaps the most significant risk of drawing false conclusions regarding to the presented case study is in understanding the ideal cost that the company had defined for products and processes. This does not necessarily represent the average opinion of all customers well enough, since it is based on the company's own estimate. The use of ideal cost is perhaps even riskier when analyzing the products, because customers usually have a clear opinion of their worth. In the case of processes, the company's own estimates of worth are perhaps more valid, since the customer does not usually see all processes as their main interest for “buying”, whereas the company wants to manage them efficiently.

References: