Experiences of Implementing a Value-Based Approach

PASI OJALA
Hintanmutka 17 A 6, 90650 Oulu
FINLAND
pasiojala10@yahoo.com

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.” [3]

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 [3]. This is rather surprising because Value Engineering (VE) as a theory already has a forty- to fifty-year history [5]. It has been used in several different situations in several different industrial branches [8].

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) [7]:

$$\text{Value} = \text{Benefit} - \text{Cost} \quad (1)$$

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) [9]

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. [9]

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.[4] 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. [4]

Tallon et al. have confirmed that process-level impacts can be used to measure IT business value. [19] 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 [4, 9, 19]. 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 [3, 4, 9, 19]. 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?" [6]

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.[13] 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 [1,7,20], 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 [18]. Therefore, the phases included to VE process include orientation, information, creativity, evaluation, development and presentation.

4 Activity-Based Costing (ABC)

Theoretically, one could claim that it is not enough simply to admit, like Boehm [2], that when process costs determine product costs, cost effectiveness is analyzed and controlled only in regard to the software development process as Boehm has encouraged [2]. This is because products are the outputs of processes and, according to Kaplan & Cooper [12], for instance, there is a clear dependency between process and product costs. Products use processes and therefore, in Activity-Based Costing (ABC), costs are allocated to products as well. If product prices do not cover all production costs, the company suffers losses.

Product pricing is also significantly more difficult if we do not know the quantities of materials, the amount of working time and other resources we have used in production, and how much these items have cost us. We can rather easily find ourselves in a situation in which we price products based on our beliefs, and we hope that customers will buy them. When we do not know which product has generated which particular cost for us, we cannot tell which products are the most profitable either. In this kind of situation there is the danger that some products may look profitable and other ones may not, so we end up cutting the manufacturing quantities of the most profitable products, rather than of the products which cause the greatest losses.

Activity-Based Costing is based on rationalizing and strengthening cost accounting at the process as well as the product level. More precisely, ABC enables expenses to be driven using cost drivers, first in activities and processes, and then in products, services, and customers [12]. Naturally, the more use products make of processes, the more costs are allocated to them, and this is taken into account in product pricing too. According to Jones [11], the ability to measure all activities associated with software production, not just coding, has led to the concept of activity-based studies. He continues by saying that activity-based cost analysis can be much more accurate than other estimating methods that lack any internal structure or granularity.

Jones [11] claims that “the ability to measure all activities associated with software production, not just coding, has led to the concept of activity-based studies”. Ooi et al. [15] note that many companies nowadays use large integrated systems, and because of this there is a clear need for accurate estimation and appropriate allocation of actual development and implementation costs to users. Ooi et al. [15] also present an ABC approach to estimating and recovering software development and implementation costs. They also address two main problems faced currently by many organizations: “1) inaccurate estimation of project resources and 2) incorrect allocation of actual resource costs to projects. Inaccurate estimation occurs because there is often no systematic organizational procedure for learning from previous estimation errors. The ABC approach enables organizations to track actual resource consumption by development activity for each project and provides a basis for variance analysis of estimated and actual costs. This facilitates learning that is specific to the organizational context.”

Ooi [15] point out that: “Many organizations apply a single charge-out rate to the number of man-days consumed by the project. This does not recognize the vastly different unit costs of different types of IT resources and penalizes the simple project that may be using relatively less skilled and hence less expensive development resources. The ABC approach explicitly recognizes different resource pools and allocates the different costs to various development and implementation activities.”

Even though the dependency between process and product cost is obvious, some people still think that in a world of Total Quality Management (TQM) and customer satisfaction, financial and cost measurement may be unimportant. For example, a group of professors [14] argue against the costing system: “… when cost accounting data of any kind does not help, “better” cost accounting data will not help either … Several companies had even gone beyond tinkering with their cost accounting systems. These companies cut the Gordian knot by finding simple, innovative, yet bold solutions to the performance measurement problem.”

The theory that advocates the elimination of financial measures for front-line employees and replaces financial measures with measures of quality and timeliness, apparently assumes that “if the operational measures are good, the result is on-time delivery and product-line budget cost. If operational measures are bad, these results should not occur.” [12]. In other words, good operational measures are both necessary and sufficient for achieving good financial performance, so employees do not need to see and consider cost-based measures [12]. Nanni et al. [14] state that "Accounting-based prformance measures become less useful as the following situations occur:

1. The market environment in which the firm competes becomes more dynamic; and
2. The level of management at which performance is being measured becomes closer to physical activities."

Kaplan & Cooper [12] consider that in the statement above, the authors clearly believe that cost signals are, at best, distracting and disruptive to front-line employees in innovative, quick response, high quality environments. It is, however, implausible to suggest that if we have a business in a more dynamic environment, we would not be interested in minimizing costs at all levels and at the same time maximizing profits as well. In a dynamic environment employees as well as managers can own the company, and they are sincerely interested in minimizing costs and maximizing profits as well.

Another academic, Johnson [10], argues that companies should focus on the most important information concerning what it takes to keep customers satisfied, and states that cost and financial information are not relevant for this purpose: "Always keep information that is used to control operating activities separate from the financial information that is used for planning and budgeting." Johnson seems to be fully against sharing financial information with employees involved in customer-satisfaction activities; he states: "Whatever form management accounting takes … never again should it be seen as a tool to drive people with measures." Why, then, do many companies have bonus salary systems based on this financial information? Why do employees get stock options based on financial results not on measures? Most probably because the owner of the company is willing to reward them based on results not measures.

According to Kaplan & Cooper [12], "these academic critics claim that financial measures should not coexist with non-financial measures to promote employee learning and improvement activities. Their quite explicit recommendation is that in a world where quality, responsiveness, and customer satisfaction are primary, organizations should direct their front-line people towards achieving excellence along these dimensions. They assume that companies that improve quality, reduce cycle and lead times, and keep customers satisfied will be rewarded with both low costs and high profits. If this view is correct, indeed there would be little apparent need for financial measurements for front-line employees. They should just focus on quality and time. Low costs and expense control will inexorably follow. And there should not be no space for employees to enhance learning and improvement activities." In Johnson's environment, employees do not know which products are the most profitable. They try to improve all processes based on the assumption that if processes are randomly improved, costs will be lower and customers happier. Quite often this applies to capability-maturity -based assessments methods too, which focus on improving processes and assume that the products will get better too.

Kaplan & Cooper [12] point out that: "Facts, unfortunately, occasionally intrude on academic theories, because many companies have made enormous commitments to improving quality and satisfying customers, yet still had disappointing, if not disastrous, financial performance." This is quite obvious. If the company concentrates only on quality and timeliness and forgets costs many things can happen. For example, workers may not have any idea what kind of work gives the best financial benefits to the company, and most of the working time they will neglect these tasks. Furthermore, to improve timeliness, managers may add resources, such as workers or machines, so that there is always the capacity to handle new orders, but perhaps no orders will come in, and as a result costs will rise significantly. It is also always possible that to improve quality, managers run machines at low speeds to avoid errors, which creates a need for more labor to run the machines and is in conflict with timeliness and possibly also with financial profit maximization by cutting costs.

As an addition to Activity-Based Costing we should also note that costs need to be calculated for the entire process, product or service life cycle. If only certain costs are taken into calculation in a certain period, cost accounting does not give an accurate view of the situation. Dell'Isola [8] has defined life cycle costing (LCC) as follows: "It is the process of making an economic assessment of an item, area, system, or facility by considering significant costs of ownership over economic life, expressed in terms of equivalent costs. The essence of LCC is the analysis of the equivalent costs of various alternative proposals. To ensure that costs are compared on an equivalent basis, the baseline used for initial costs must be the same as that used for all other costs associated with each proposal, including maintenance and operating costs."

5 Value Assessment for Processes and Products

There are four ways to enhance a standard software process assessment using VE [16, 17, 18]. 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.

6 Value Assessment for Processes and Products: Company A

6.1 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.

Since the assessment was to include product, process and accounting system evaluations, there was a wide range of material to be analyzed. It included documents such as:

- Strategy plans
- Project plans
- Process descriptions
- Budget principles
- Cost accounting principles and calculations
- Invoice approval and handling procedures
- Financial statements
- Personnel task lists
- Infrastructure definitions
- Selling agreement templates
- Personal development discussion templates
- Architectural descriptions
- Product requirement definitions
- Product testing plans
- Platform definitions and manuals
- Configuration management plans
- Quality plans and reports
- Accounting system definitions and user manuals

6.2 Information

The product assessed was a typical software product. It was developed and tested in the R&D department of Company A. Personnel in the sales department, together with requirement engineers and technical personnel, defined the requirements for different product versions. Based on the available resources, these requirements were mapped to new product versions. The product was developed using the standard C language. The overall goal of product development was to produce different versions of the product for different operating system environments, using the same base code. Unfortunately, this was not possible because, in practice, there was always a small part of the product that had to be coded separately for each operating system environment. In practice, the product was a collection of sub-products that were selected by the customer to provide the software combination required.

There were more than 20 platforms for which the product was made. Company A had divided the supported operating systems into three categories: 1) easy; 2) difficult; 3) very difficult. For category one operating systems, the product was prepared every time the version or a product was changed. For category two, a new package was prepared twice a year, and for category three, the package was created on demand. Company A received income from product license fees and maintenance service fees. There were two different licenses. A runtime license was used when the product was used as is, and a development license when it was used as a part of the customer's product.

Company A had a strong interest in analyzing priorities 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. Based on a discussion of requirement- and component-level cost accounting, it was agreed that component-level cost accounting presented more challenges in implementation than requirement-level accounting. This was due to the fact that, based on customer meetings, salesmen had a fairly good awareness of how customers prioritize and allocate worth to requirements, but at the architectural, component level, customer was lacking enough experience to understand technical product structure enough and worth allocation was therefore not as easy. Customers did not necessarily know how the product was classified into components, or what each of these components did. Nor was Company A interested in giving precise component structures to customers for worth-calculation purposes. Therefore, component-level worth calculation was beyond the scope of this assessment, although Company A underlined it as an interesting challenge for evaluating value inside the company.

Based on a discussion in the information phase, it was noted that Company A did not have platform-level cost accounting, which would follow costs for the more than 20 platforms. Therefore it was not possible to use actual cost, worth or value information for each platform separately. However, the assessor pointed out that if Company A was interested in basing the assessment strongly on estimates, using existing cost information as a pool, it would be possible, using estimation, to allocate those costs to architectural components as well as to all platforms. Since Company A was more interested in gaining actual, rather than estimated cost information, it was decided that product-focused value assessment would only be carried out on three high priority operating systems, and Company A would create accounting identifiers for each platform after the assessment. Based on this information it was decided that value assessment would be performed as well as possible and the company would create accounting identifiers to follow costs and income for each platform later. Product-focused assessment was implemented for the following operating systems:

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

Company A had participated in a capability-maturity -based assessment in the past. The results of the capability-maturity -based assessment were available when collecting information. Also available were all process descriptions written in Company A. 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. Therefore, it was decided that the driving force of value assessment for processes should be based on the company's own process descriptions rather than on a reference model. The processes selected for value assessment included:

- Architectural design
- Design
- Code Implementing
- Testing

When comparing any reference model to the selected processes it is however possible to note that they represent the typical processes presented in these models as well.

6.3 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.

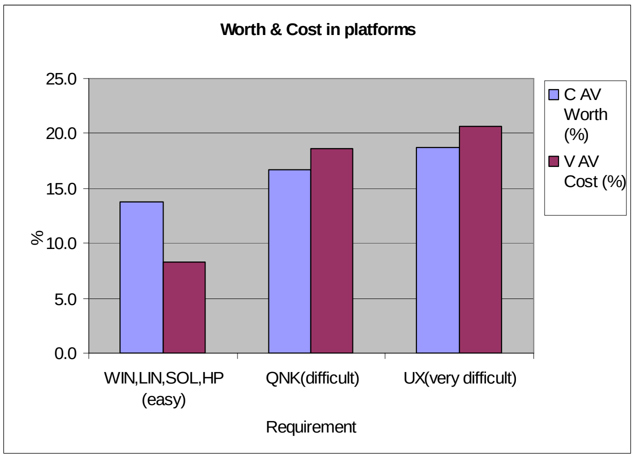

Figure 1: Worth & Cost in platforms (AV=average, C=customer, V=vendor)

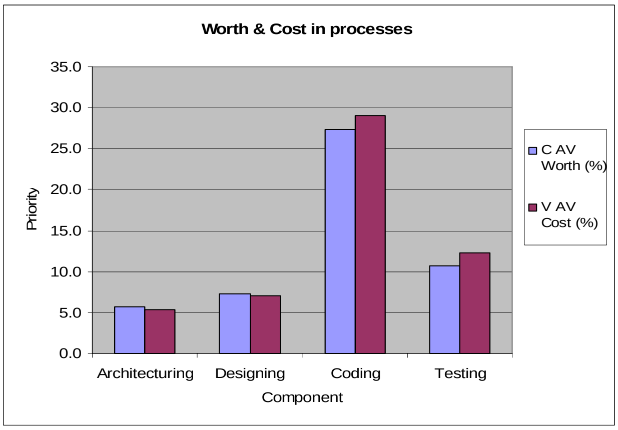

Figure 2: Worth & Cost in processes (AV=average, C=customer, V=vendor)

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.

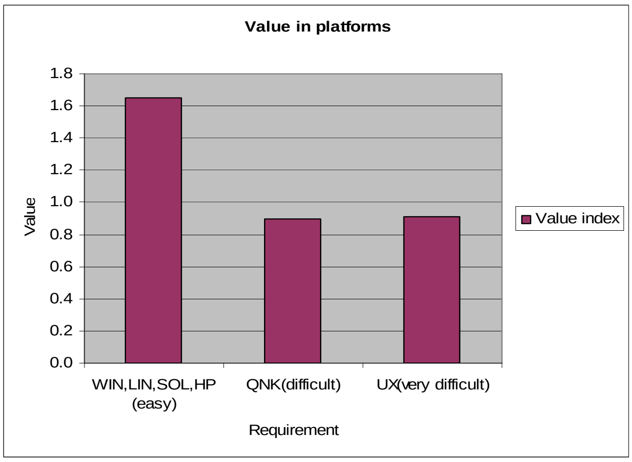

Figure 3: Value in platforms

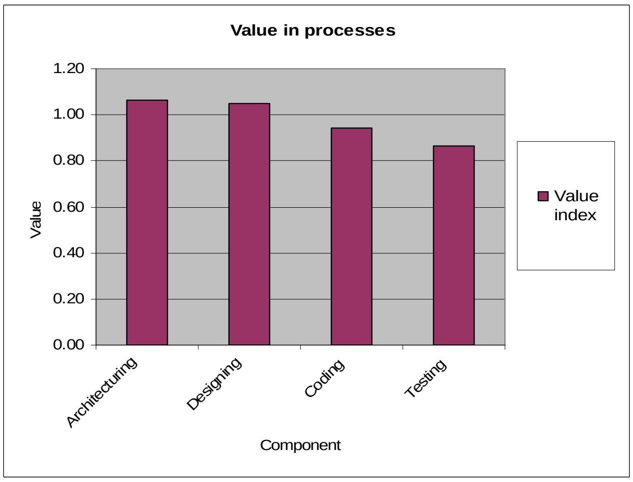

Figure 4: Value in processes

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.

6.4 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.

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.

6.5 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.

6.6 Development

Product-related value

During the development phase, benefit analysis, a data package, an implementation plan and a presentation to top-level management were all prepared. 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. Some customers had stated that they did not have technical knowledge of “easier” platforms and they would need significant support in moving to these platforms. 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. There was no need to put together a comprehensive technical data package for this proposal, since it was not a question of developing new technical solutions. The implementation plan included a program for a one-year period, in which the necessary actions were described.

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. Working time in coding, module testing and security testing was analyzed quantitatively as well. Empirical analysis of the history data showed that coding was done reasonably quickly, but all testing-related activities took a significant amount of time, and many designers did not have a clear understanding of what security testing should include. 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.
- The test manager should 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. In relation to these value improvement proposals, it was recommended that working hours should be tracked and reported monthly in order to follow how the situation was proceeding.

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
- Defined process practice.

In terms of platforms, it was decided that all income and costs would be allocated to project and product (platform) numbers. It was decided that each project would have four numbers and each product, two numbers. Using the new identifiers it was possible, for example, to get income, cost and profit reports for each project and product separately. As a final improvement proposal, all process descriptions were reviewed and work practices defined for time-keeping purposes. Since Company A had the software necessary to implement these changes, 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. At the company level, it was calculated that the system would pay for itself in an even shorter time, with the ability to price more products more profitably.

6.7 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.

6.8 Strengths and weaknesses of value assessment

Company A was satisfied with the value assessment. The assessment began with an initial meeting, in which the phases of value assessment were discussed. The auditor presented the content of each assessment phase, and the decision to carry out process and product assessment was made. In practice, problems arose only when it was noted that Company A did not have a proper cost accounting system in place. Even though in this assessment the focus was on both processes and products, the value assessment process seemed to take place with no additional problems. By using both approaches at the same time, the company can gain an even more powerful value assessment tool for the formulation of improvement proposals. Because, in this assessment, capability-maturity -based assessment results were available throughout the whole assessment, it was easier for all participants to make improvement proposals which took into consideration both capability and value.

The assessment results support value-based approach to software engineering in several ways. The results show that there exists a practical need to enhance the scope of software engineering in a value-driven direction. It is easy to draw this conclusion, since Company A showed an interest not only in value assessment itself, but also in building up a cost accounting system for process practices and product components. The results also show that Company A needed a two-dimensional assessment, which evaluated both processes and products. Therefore, the theoretical claim that process-focused assessment alone is not enough to start improvement was justified. As well, capability-maturity -based assessment results formed a good basis for value assessment, even though Company A did not consider them adequate for starting expensive improvement work in the software engineering area. The assessment results also justify the use of Activity Based Costing (ABC) in improving the software engineering area. This can be justified, because Company A wanted to build a two-level cost accounting system, which took into account processes as well as products; this is one of the main ideas behind ABC. Finally, the results also gave several more reasons for using estimation methods in the software engineering area. In Company A’s situation in particular, where there was no cost accounting system in place before the assessment, estimation methods were required in order to find cost-reduction and value-improvement areas. The theoretical claim that value should be considered an important part of process and product assessment was also justified, because actual costs alone would not have been enough to show Company A which areas needed improvement the most. This is due to the fact that "cost" only measures how expensive software implementation is, not how much someone would pay for it.

According to Company A the assessment process worked as planned, and the phases from creativity to presentation were also useful in combining value and capability-maturity -based assessment results. From Company A's point of view the information collection phase collected enough information for the next VE phases. It also provided opportunities to discuss the needs of the company. The most significant result of the information phase was perhaps that Company A already knew that it needed a better cost accounting system which would justify the areas in which process- and product-related improvement should be done. Cost estimation alone was not seen as enough for these purposes even estimations were made using the main cost-driving variables, such as working hours, from the time-keeping system. Actual costs were clearly preferred to estimated ones. Company A's top-level management also agreed that the previous capability-maturity -based assessments neglected two important points of view concerning software engineering. They did not take the product and business points of view into account sufficiently. Instead, they assumed that money is "always" given to process-related improvements if capability is low, even if there is no guarantee that these investments will ever pay back the costs incurred.

The importance of actual cost, worth and value (rather than estimates) was considered to be so great that the representatives of Company A wanted to postpone the full value assessment further, until the cost accounting system was working properly. However, even the focused assessment showed that Value Assessment has a significant place when improving software product profitability in relation to software process improvement.

Generally, all the assessment results found are reliable. The reliability of the results was also improved significantly because the assessor interviewed several people and went through the same questions with all of them. The interview results were also compared to existing written material to check that they matched. Since the assessor had also passed the Value Analyst exams, he had the necessary skills to interpret the findings.

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

Further writings in this area should present more discussion on the theoretical basis of value-driven software engineering. This is due to the fact that value is a rather complex phenomenon, and the Value Engineering point of view is only one way to see the situation.

Clearly, further empirical evidence of industrial value assessments is also needed. In the first place, these experiences are needed in order to carry out an exhaustive discussion of the advantages and disadvantages of the suggested value assessment process and method. In practical terms, it would also be interesting to clarify how improvement plans have changed when using value information. Do software companies still mostly use process assessments? Do they become more interested in product assessments? Is value assessment used as a practical tool for industrial assessments? Does value information play a more important role when formulating software process- and product-related improvement proposals? Are there conflicts when deciding whether to use value or capability information primarily, when forming improvement proposals?

More research should also be carried out in evaluating VE techniques and selecting the most suitable ones for assessment work. In this development it is clear that value assessment for processes and products, and capability-maturity - based assessment including VE processes, all need partially differing tools, templates, checklists and techniques. In this paper not all these use cases have been examined.

The usefulness of Activity-Based Costing (ABC) and other costing methods could also be studied further. Several companies have adopted ABC, but a significant number of them are still running other cost accounting methods as well. More information is also needed in order to make a decision concerning which estimation method would work best with value assessments.

Since value assessment claims to take the customer into account in decision-making, it would also be relevant to conduct an analysis of customer interest in participation in assessment. Are customers more satisfied with projects where value assessment takes place and they can formulate their requirements together with the vendor? Does customer participation in value assessment help the vendor to implement and manage the project?

References:

[1] Bellos, V., Leopoulos, V & Sfantsikopoulos, M., On the state of the art of cost estimation and cost uncertainty analysis pp. WSEAS Transactions on Computers, Oct, 2003, pp.1188-1196.

[2] Boehm, BW., Software Engineering Economics. New Jersey, Prentice-Hall. 1981

[3] Boehm, B.W., Software Economics: A Roadmap. Future of Software Engineering. Limerick, Ireland. ACM, 2000. pp. 319-343.

[4] Boehm, B.W., Value Based Software Engineering: A Case Study. IEEE Computer Society, 2003, pp. 33-41.

[5] Brown, J., Value Engineering. A Blueprint. Portland (Oregon), Book News, 1992.

[6] Capers, J., The economics of software process improvement. Software Challenges, Jan, 1996, pp.95-97.

[7] Crespo, J., Sicilia, MA & Cuadrado, JJ., On the Use of Fuzzy Regression in Parametric Software Estimation Models: Integrating Imprecision in COCOMO Cost Drivers. WSEAS Transactions on Systems, Jan, 2004, pp. 96-101.

[8] Dell' Isola, A., Value Engineering: Practical Applications…for design, Construction, Maintenance & Operations. Kingston (MA), R.S Means Company Inc, 1997.

[9] Grady, R.B., Successful Software Process Improvement. New Jersey, Prentice-Hall, 1997.

[10] Johnson HT., Beyond Product Costing: A Challenge to Cost Management's Conventional Wisdom. Journal of Cost Management, 1990, pp. 15-21.

[11] Jones C Activity-based software costing. Software Challenges, May, 1996, pp. 103-104.

[12] Kaplan R & Cooper R., Cost & Effect. Using Integrated Cost Systems to Drive Profitability and Performance. Harward, Harward Business School Press, 1998.

[13] Koskela, L., Huovila, P., "On foundations of Concurrent Engineering in Construction CEC'97. London, 3-4 July. London, The Institution of Structural Engineers, 1997, pp. 22-23.

[14] Nanni AJ, Dixon JR & Vollman TE., "Strategic Control and Performance Measurement." Journal of Cost Management, summer, 1990, pp. 34-35.

[15] Ooi G, Soh & Lee PM., An Activity Based Costing Approach to Systems Development and Implementation. Proceedings of the International Conference on Information Systems. Helsinki, Finland, 1998, pp.341-345.

[16] Ojala, P., Enhancing Software Process Improvement Using Value Engineering. Working Paper. Series B. University of Oulu, 2000.

[17] Ojala, P., Combining Capability Assessment and Value Engineering: a BOOTSTRAP example. Proceedings of the Profes 2004. 5th International Conference. Kansai City, Japan, April, 2004.

[18] Ojala, P., Implementing a Value-Based Approach to Software assessment and Improvement. Doctoral dissertation. University of Oulu, 2006.

[19] Tallon, P., Kraemer, K., Gurbaxani, V.A., Value-Based Assessment of the Contribution of Information Technology to Firm Performance. Working Paper ITR-119. University of California, 1999.

[20] Xia, W., Capretz, LF & Ho, D, A Neuro-Fuzzy Model for Function Point Calibration. WSEAS Transaction on Information Science and Applications, Jan, 2008, pp. 22-30.