Combining Capability Assessment and Value Engineering: a New Two-dimensional Method for Software Process Improvement

Pasi OJALA
Nokia ltd.
Oulu, Finland

ABSTRACT

During the last decades software process improvement (SPI) has been recognized as a usable possibility to increase the quality of software development. Implemented SPI investments have often indicated increased process capabilities as well. Recently more attention has been focused on the costs of SPI as well as on the cost-effectiveness and productivity of software development, although the roots of economic-driven software engineering originate from the very early days of software engineering research.

This research combines Value Engineering and capability assessment into usable new method in order to better respond to the challenges that cost-effectiveness and productivity has brought to software companies. This is done in part by defining the concepts of value, worth and cost and in part by defining the Value Engineering process and different enhancements it has seen to offer to software assessment.

The practical industrial cases show that proposed two-dimensional method works in practise and is useful to assessed companies.

Keywords: Software, Software process improvement, Value, Worth, Cost, Value Engineering and Capability assessment.

1. INTRODUCTION

General understanding concerning software process improvement is that the quality of the software development process has a close relationship with the quality of the resulting software [8]. Krasner [11] points out that: "In a mature software organization, the following holds:

- Quality is defined and therefore predictable
- Costs and schedules are predictable and normally met
- Processes are defined and under statistical control".

For several decades companies have aimed at software process improvement, SPI, [22] using different capability-maturity -based assessments which are based on capability-maturity models like CMM or BOOTSTRAP (introduced in appendixes) as a basis for discovering necessary improvement initiatives.

Although capability-maturity -based assessments have been recognized as a viable means of increasing the quality of the software development, they are perhaps not complete. They have been criticized as expensive, disruptive, infrequent and inflexible [1, 2, 6, 10]. Herbsleb et al [7] have stated that they also often encourage too much bureaucracy.

In addition, methods like BOOTSTRAP, the Capability Maturity Model (CMM) and ISO standards seem to have weaknesses in recognizing costs, values and business needs, since these aims have not been considered to be the most important ones when developing these methods [20]. For example, the BOOTSTRAP method does not support process assessment and improvement work by justified and defined cost calculations and accounting. Therefore, if a company is, for example, interested in focusing assessment on processes where there are a lot of costs and value creation is therefore low, the BOOTSTRAP method does not give support in finding these areas and improving them by focusing the assessment and using cost or value as criteria. Nor do the other methods which have been used in its development, such as the Capability Maturity Model (CMM), ISO or several other standards.

However, although the assessment methods discussed do not seem to be complete, interest in their development is increasing [24] and some researchers are also examining alternative possibilities for assessing software processes [2, 14]. From this study's point of view, a method, which does not take into account costs and values created in processes is perhaps not complete enough for business purposes and for taking business needs into account.

This criticism has strengthened the author's view that capability-maturity -based assessment methods lack proper cost and value calculation characteristics, which provides a motivation for this study as the importance of such characteristics seems to be increasing. The author also sees that many companies are in a situation where they would like to make use of methods which can point out their most capable processes and identify how much value these processes provide to justify for example, SPI initiatives or see whether software development is cost-effective.

In this study, capability, defined in detail in each capability-maturity model for its own use, and value are seen as equally important. There are mainly four reasons for this. Firstly, if the capability of the process is high, it will probably cause high product quality, and if the value is also high, the situation is under control, because the company is acting economically and is creating value and high product quality with capable processes. Secondly, problems arise if a process's capability is low, which will also probably cause low product quality and if a process's value is low, the low quality products will not be produced economically either. Thirdly, problems arise when the company is creating high quality products with high capability processes, which are not cost-efficient and do not create value either. In the long term this is not economical and will endanger the future of the company. Fourthly, problems arise when the company is producing low quality products with low capability processes even if they would create value and be

cost-efficient, because in the long term customers might not be happy about buying low quality products even if it would be economical to the company. (Figure 1)

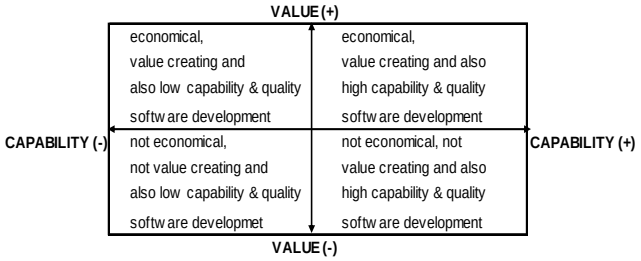

Fig. 1. Capability and value based process evaluation. [17, 18, 19, 20]

Here value, cost and worth concepts are adopted from Value Engineering (VE) (synonymous with the terms Value Management and Value Analysis) which is professionally applied, function-oriented, systematic team approach used to analyze and improve value in a product, design, system or service and process – methodology for solving problems and/or reducing costs while improving performance/quality requirements. By enhancing value characteristics, VE is understood to increase customer satisfaction and value to investments. In enhanced assessment is seen that VE can be used also in improving the value and optimizing the life cycle cost of a process and its practices. [17, 18, 19, 20]

2. THE CONCEPT OF VALUE

Value is a goal for many activities what we do. However, as a concept it is a large, complex, and abstract subject, and simply defining it requires a great deal of thought as Shillito and De Marle [23] have stated. However, defining it as a concept helps to understand it more precisely.

In Ancient Greece people believed that certain primary or essential principles existed in our environment. According to Shillito and De Marle [23] these indwelling principles gave value to the items they inhabited. Furthermore ethics contained " the good", religion, "the holy", and aesthetics, "the beautiful." When the indwelling principle was present, the object had value, when it was absent, the object was worthless. Religion was perverted when it lost its holiness. Art was degrated when it lacked beauty.

Even from historical point of view the Greeks idea of sprits dwelling in the rocks, water and other objects around us is old, it still influences to our thinking. We see quite often value dwelling in a product. Engineers and economists seem to think quite often that value is a feature that a product or a feature has. Good example of this thinking is the way how modern advertising see these features. When putting petrol to car we put "tiger to tank" and the engine of car when running sounds like cats mourning. However, Shillito and De Marle [23] have seen value more than just a property of matter. They keep value as a force that governs our behaviour. In their opinion we need to discard the anthoropomorfic concept of value and examine this force.

Even value as a force seems to be more variable than other forces it can be measured from several point of views. According to Shillito & De Marle [23] good consumer value exists when desired product is costs little and performs well. As a formula (1), customer value is presented as follows:

$$Customer\ value = \frac{performance}{price} \qquad (1)$$

The value of a product to a retailer may differ significantly from the value for a customer. This is because retailer is primarily interested in financial return from his investments [23]. Retailers are interested in quick profits and products that they can sell quickly. Furthermore, retailers want to achieve maximum income with minimum investment. Therefore value formula for retailer differs from value formula to customer. Retailer's value formula shows that instead of calculating a cost-benefit ratio, retailers estimate the sales revenues they would expect to receive from selling their products. Retail product value is calculated using formula (2): [23]

Retail value = unit sales (unit price – unit cost) (2)

The manufacturer of a product for a retailer or a customer uses also own formula for value calculation. This is because, his interest is slightly different than others. Compared to retailer the customer has more capital costs related to manufacturing plant and tools. This forces manufacturers also to keep manufacturing going on most of the time. Because manufacturer is also interested in about the return what he will get for the money, his formula (3) for determining manufacturing value is the following: [23]

$$Manufacturing\ value = \frac{[Customer\ and\ retailer\ benefits\ (1\ to\ n)\ ] + profit}{costs} \qquad (3)$$

Even Value can be defined from different point of views we can state that Value (V) is a measure usually in currency, effort or exchange or on a comparative scale, which reflects the desire to obtain or retain an item, service or ideal. Cost (C) is the price paid or to be paid. It can be divided into elements and to some extent functions (or processes). Worth (W) is defined as the least cost to perform the required function (or process) or the cost of the least cost functional equivalent. Using a formula Value Engineering defines Value as follows: V=W/C and when emphasizing the characteristics of worth Value has often been defined using a formula (4): [4, 5, 21, 23].

$$Value = \frac{Function + Quality}{Cost} \qquad (4)$$

Where:

Function	=	The specific work that a design/item must perform.
Quality	=	The owner's or user's needs, desires, and expectations.
Cost	=	The life cycle cost of the product.

Practically in Value Engineering value is the most cost-effective way to reliably accomplish a function that will meet the user's needs, desires, and expectations. More specifically, function represents the work that should be done and quality represents the needs, desires and expectation how this should be done. On the other words Function+ Quality defines what is the worth of examined item to customer. If customer has higher expectations, the worth is higher to him and if he has lower

expectations the worth is lower. As well as, when quality rises worth, same happens with functions, because if customer wants to list more work to be done with product, the amount of functions rises which rises worth as well. [20]

This applies also to manufacturer's software processes. If a process has more tasks what it has to be able to perform, there are more functions and, therefore worth rises. On the other hand if manufacturer's desires and needs to processes are in a higher "capability level" so that process is a better quality process, also worth rises. Practically, customer does not have often interest to software processes and therefore, it is not often worth to examine value of processes from customer point of view. However, if customer is buying for example testing services, his interest to testing process and its value to him might be quite clear. [20]

In general product is often seen as an output of using processes. Therefore, it is possible to claim that it is not enough to assess only processes but products should be assessed as well. This results that value should be examined from both point of views especially from product point of view, because it is interesting to customer and manufacturer as well.

As a conclusion it can be seen that value has a close relationship with cost. This is inevitable, because if it's expected more functions to be done with a process and expectations do not become lower, costs of running a process are higher. The same logic applies as well to software product. If it is expected that a software process will perform more work the costs become higher. If the expectations on functions, how product should perform are more strict, are costs rising again. [20]

3. THE VALUE ENGINEERING PROCESS

Nowadays, VE methodology is widely known and accepted in the industry. It is an organized process with a history of improving value and quality. The VE process identifies areas in which unnecessary costs can be removed, while assuring that quality, reliability, capability, and other critical factors will meet or exceed the customer's expectations.

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. [20]

In practice, the improvements developed 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. A wide range of companies and establishments have used VE effectively, to achieve their continuous goal of improvement in the decision-making process.

VE has been combined with several other methods, such as TQM and ABC. Most likely this direction has strengthened VE's interpretation as an "Engineering" method, rather than an "Analyzing" method (VA), as well. [20]

This study categorizes VE phases in three main classes: pre-study (tasks carried out before the value study), value study, and post-study. These phases are considered appropriate since they constitute independent areas of VE, emphasizing the preparation aspect, the independent study aspect and the post-study aspect performed after the other phases. The detailed VE process in study includes following phases: [20]

PRE-STUDY
- Orientation

VALUE STUDY
- Information phase

Function Analysis phase
- Creativity phase
- Evaluation phase
- Development phase
- Presentation phase

POST STUDY
- Monitoring & implementing phase

The structure mainly follows the lines which Lawrence Miles outlined at General Electric and later refined for the purposes of the Department of Defense. However, since the significance of function analysis as a separate phase has been presented in nearly all recent VE process descriptions, it is presented as a separate phase. This is also seen as justified, because it provides a clear structural way to analyze the value of different functions which are later defined more precisely to be processes, practices and products or their components. Finally, the flexibility of function analysis is considered necessary in order to combine capability and value assessment together, as discussed later in this study. Furthermore, the creativity phase is considered as a separate phase since it questions the earlier VE phases and, using this information, produces alternatives for the subsequent ones. The inclusion of this process in its own phase, rather than combined with other phases, highlights the importance of such comprehensive investigative work. [20]

4. VALUE BASED ENHANCEMENTS TO SOFTWARE ASSESSMENT

There are four ways to enhance a standard software process assessment using VE [19, 20]. 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 recommendation is to define VE process as own process cluster so that it is possible to find out how mature the company is in Value Engineering process [19, 20].

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. If company has implemented also a normal capability assessment, after Value Assessment it knows both value and capability of each assessed process and has a significantly better start for its process improvement work. [17, 18, 19, 20]

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 practises which are then examined from cost, worth and value point of views using VE process. [17, 18, 19, 20]

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. Partially, the product improvement ideas are reflected also to process development work, because in this enhancement, product is seen as an output of processes. [17, 18, 19, 20]

5. EXPERIENCES OF INDUSTRIAL ASSESSMENTS USING VALUE ENHANCED ASSESSMENT

The four enhancement possibilities were experimented by enhancing ISO 15504 conformant software process assessment methodology called BOOTSTRAP and performing it in practical assessments. The evidence from performing practical industrial assessments shows that the new enhanced assessment model works in practice. Capability levels can be defined for Value engineering processes as well as cost, worth and value to all processes in existing process model. However, when calculating cost, worth and value practical problems might arise if a company does not have sophisticated cost accounting system that helps to assign costs to processes, practices and product components. [20]

Assessment in Company A

In the first capability-maturity -based assessment including VE processes, new points arose. First, it was observed that people were performing value analysis and improvement actions at the SPU (Software Producing Unit) as well as at the project level. Mostly these actions were planned on many occasions but often they were not improved at all. It was rather surprising to see that human resources-related functions were planned with the most precision, and included improvement planning, whereas many technical functions were lacking systematic value improvement actions. The capability-maturity -based assessment including VE process was seen as usable help for assessed company. The highest capability levels, company achieved in phases related to value data collection and lowest in phases related to its evaluation and development. Secondly, the assessed company liked the capability assessment of the VE processes because it combined technical and economical personnel in improvement planning, as well as took customer needs into consideration.

In practical terms, the capability-maturity -based assessment for VE processes worked well in Company A. In the final assessment meeting, the discussion seemed to bring up several possibilities for the improvement of quality, customer satisfaction, and the reduction of production cost. However, the discussion was at a purely conceptual level, because costs were not calculated and the exact amounts of costs were not presented. It seemed that the first enhancement was outlining a way to future ones, and supported the usefulness of the value assessment.

Assessment in Company B

The value assessment in Company B was based on focused evaluation of both processes and products. This approach was taken because Company B was interested in gaining experience of both kinds of value assessments. At first it did not know whether its cost accounting would be able to provide the necessary cost data for all processes and product components. Based on this, 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.

The results show that there exists a practical need to enhance the scope of software engineering in a value-driven direction as Company B 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 B 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 B did not consider them adequate for starting expensive improvement work in the software engineering area.
The assessment results also encourage to use two-level cost accounting system, like Activity Based Costing (ABC) which takes into account as well process as product costs.
According to Company B 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 B'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 B 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 B'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 B 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.

Assessment in Company C

The product assessment was considered to be significantly more effective than the process assessment. Company C did not consider capability-maturity -based assessment to be as good as product-focused assessment for its purposes. Perhaps this was due to the fact that Company C was rather small, and all improvement initiatives were expected to give advantages as soon as possible. It saw capability-maturity -based assessment to outline areas where improvements should be carried out, but did not see it to give justifications of the business advantages of doing so.

The product-focused assessment had several strengths. It was seen to give more customer-oriented improvement proposals than process assessments and product-related improvement was the language that the customer understood and was in a way "buying". Company C saw also that when the assessment is undertaken together with the customer, it can keep the customer more satisfied, which is a good basis for business. As well it emphasized that if value assessment is done in the planning phase of a product, it is cheaper for any company than making changes after several months of development work.

The results also show that Company C had a need for two-dimensional assessment, which evaluated both processes and products. Capability-maturity -based assessment results formed a good basis for value assessment in Company C. By using them, it was possible to gain an understanding of the capability of the processes producing the product in question.

The assessment results for Company C also support the use of Activity Based Costing (ABC) in improving the software engineering area. When discussing cost, worth and value it seemed clear that these should be calculated for processes as well as products, which is the purpose of ABC. Since in Company C the product was expected to include several new features in the future, life-cycle costing methods also had to be used to foresee and understand product costs over time.

Generally, all the assessment results in this assessment 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. had the necessary skills to interpret the findings.

6. CONCLUSION

Please do not enumerate the pages in your article. We will do that as part of the printing process. In conclusion, 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. Furthermore, 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 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. Value assessments seem to have a place in the software engineering field.

7. REFERENCES

[1] T.B. Bollinger and C. McGowan, A Critical Look at Software Capability Evaluations. IEEE Software 8(Jul), 1991, pp. 25-41.

[2] M. Campbell, Tool Support for Software Process Improvement and Capability Determination: Changing the Paradigm of Assessment. Software Process Newsletter 4 Fall, 1885, pp. 12-15.

[3] R. Cooper amd R. Slagmulder, Target Costing and Value Engineering. Productivity Press, Portland Oregon, 1997.

[4] L.W. Crum, Value Engineering: The organized search for value. Longman, London, 1971.

[5] A. Dell'Isola, Value Engineering. Practical Applications…for Design, Construction, Maintenance & Operations. R.S.Means Company Inc. Kingston MA, 1997.

[6] J-N. Drouin, The SPICE Project. Elements of Software Process Assessment and Improvement. K. El Emam & N. H. Madhavji (eds.) Los Alamitos (CA). IEEE Computer Society, 1999, pp. 45-55.

[7] J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes & M. Paulk M, Capability Maturity Model. Communications of the ACM 40(6), 1997, pp. 30-40.

[8] W.S Humprey, Managing the Software Process. Reading, Addison Wesley, MA, 1989.

[9] D.L. Johnson & J.G. Brodman, Tailoring the CMM for Small Businesses, Small Organisations, and Small Projects. In: El Emam K & Madhavji NH (eds) Elements of Software Process Assessment and Improvement. Los Alamitos (CA). IEEE Computer Society, 1999, pp. 237-257.

[10] J. Kelly and S. Male, Value Management in Design and Construction. E & FN Spon, U.S, 1993.

[11] H. Krasner, The Payoff for Software Process Improvement: What it is and How to get it. Elements of Software process assessment and improvement. K. E1. Emam & N.H. Madhavji (eds). Los Alamitos, Calif. IEEE: Computer Society , 1999.

[12] P. Kuvaja, BOOTSRAP 3.0 – A spice Conformant Software Process Assessment Methodology. Software Quality Journal, 8. Kluwer Academic Publishers, Netherlands, 1999, pp. 7-19.

[13] P. Kuvaja, A. Bicego, and A. Dorling, SPICE: The software process assessment model, Measuring Training Based Process Improvement. September 11-12. Proc. ESI_ISCN '95 Conference, Vienna Austria, 1995.