

A. Aurum and C. Wohlin, "A Value-Based Approach in Requirements Engineering: Explaining Some of the Fundamental Concepts", Proceedings 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, editors: P. Sawyer, B. Paech, and P. Heymans, LNCS 4542, pp. 109 - 115, Trondheim, Norway, June 2007.

# A Value-Based Approach in Requirements Engineering: Explaining Some of the Fundamental Concepts

Aybüke Aurum<sup>1,2</sup> and Claes Wohlin<sup>2</sup>

<sup>1</sup> School of Information Systems, Technology and Management,  
University of New South Wales, Sydney 2052 Australia  
aybuke@unsw.edu.au

<sup>2</sup> School of Engineering, Blekinge Institute of Technology,  
PO Box 520, SE-372 25, Ronneby, Sweden  
Claes.Wohlin@bth.se

**Abstract.** Today's rapid changes and global competition forces software companies to become increasingly competitive and responsive to consumers and market developments. The purpose of requirements engineering activities is to add business value that is accounted for in terms of return-on-investment of a software product. This article introduces some of the fundamental aspects of value by borrowing theories from economic theory, discusses a number of the challenges that face requirements engineers and finally provides a model that illustrates value from business, product and project perspectives.

**Keywords:** value based approach, requirements engineering, business strategy, technical decisions, alignment.

## 1 Introduction

Increasing global competition, dynamic market needs and new technologies are some of the challenges that software companies face today. There is an incredible pressure on these companies to achieve and sustain competitive advantage. To remain competitive in an era of increasing uncertainty and market globalization it is important to focus on the value of different customers and markets when developing products. This article addresses a value-based approach in requirements engineering (RE) when creating, measuring and managing product value through requirements selection for a software release, and hence providing quality to the end-user.

Adding value is an economic activity that has to be taken into account from a software business perspective. Value is created when a company makes a profit. The critical success factor for software companies is their ability to develop a product that meets customer requirements while offering high value that provides increased reassurance of market success [2] and [4]. Since the ultimate aim for a software company is to maximize value creation for a given investment, it is essential to understand the relationships between technical decisions and the business strategy that drives the value [5]. Boehm argues that ([4], chap 1) software engineering (SE) is

largely practiced in a value neutral setting, i.e. every requirement is considered equally important, even though not all requirements are equal. Furthermore, there is often a mismatch between the decision criteria used by software developers and the value creation criteria of organizations in which software is developed [5]. Hence, a value based approach in RE promotes

- Alignment of technical decision with business strategy
- Sustaining competitive advantage by increasing the business and customer value
- Multiple perspectives involvement in creation of product, project and business value.

It is important that requirements engineers understand value creation for a software company while also taking into account the customer's perspective. This article a) introduces some of fundamental aspects of value, b) provides a model to illustrate value creation for a software company; c) highlights different value perspectives in the context of RE activities and provides a preliminary discussion on this topic.

## 2 Background

### 2.1 History of “Value” Concept

While philosophers and ethicists used to define value as a normative approach to separate right from wrong in the 17<sup>th</sup> century, the basic concept of value in economic theory can be traced back to the 19<sup>th</sup> century. However, the concept of a value-based approach in software development was not used until the late 1990s.

John Stuart Mill, who had a strong influence on economic theory, defines the concept of value, at a very high abstract level, in terms of *use* and *exchange value* [12]. A *use value* is what the customer is willing to pay for the product, and an *exchange value* is the market value of the product. As Mill's definition of value dominated economic theory in 19<sup>th</sup> century, the term “value-adding” became very popular in the early 20<sup>th</sup> century.

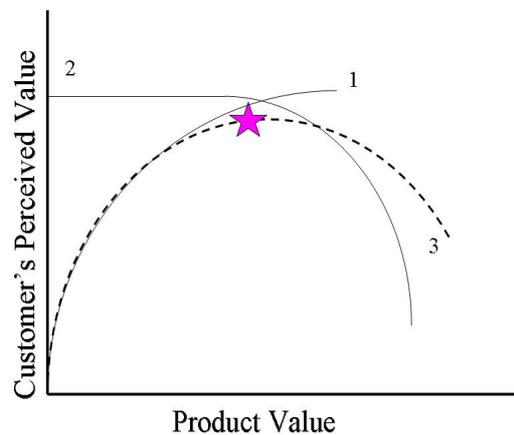
In the early 20<sup>th</sup> century, the focus in product development was the product itself (product-oriented approach) and customer value was seen as being integrated in the product. A shift in economic theory started after World War II, in the late 1960s. By the end of 1980s the focus of product development was placed on the relationships between the customer service and customer needs. This approach was based on the notion that value was related to long-term relationships between the customer and the company (customer-oriented approach). Value was created in cooperation with the customer where the customer was an active participant in value creation activities [10] and [13].

In late 1990s, the concept of the value-based approach in SE was introduced in the context of decision-making about product lines [7], managing investments in reusable software [9] and software economics [5]. Since then the value-based approach has attracted both software practitioners and academics and leading them to integrate value considerations in existing and emerging software principles and practice [4].

## 2.2 Defining Value

In economic theory value constructions are built based on customers' satisfaction, loyalty and re-purchasing behavior [10]. In the context of software development, by borrowing the ideas from economic theory, we believe the following fundamental aspects of value have importance to software developers.

- **Product value:** This is the market value of the product (*i.e. exchange value*) and related to the product, and is influenced by the quality attributes of the software product. The value of a product increases in direct proportion to its advantage over competitive products and decreases in proportion to its disadvantages [1].
- **Customer's perceived value:** This is the benefit derived from the product and is a measure of how much a customer is willing to pay for it. A customer's perceived value (*i.e. use value*) is influenced by his/her needs, expectations, past experience and culture. It is defined as  $perceived\ value = perceived\ benefits / perceived\ price$ , where the perceived benefits and the perceived price are both measured relative to competing products [6] and [14].
- **Relationship value:** This is created through the social relationships between the software company and the customer. It exists through the product and customer's perceived value.



**Fig. 1.** Customer's Perceived Value and Product Value Relationship (adapted from [6])

It is important to understand the relationship between customers' perceived value, and the time and money spent on product development [6] (see Fig 1). A customer views a purchase as a bargain, if the *customer's perceived value* > *perceived price* of the product. If the *price* > *product cost* then the software company makes a profit on their sale. If the customer's perceived value is assumed to be equal to the technical performance of the product (which is the traditional approach in software development), then the perceived value continues to increase, up to a certain point, as more time and money are spent on product development [6] (curve 1 in Fig 1). At the same time, the product price will increase, because of the more time and efforts put

into product development. As a result the customer will not be able to afford the product; he/she will start looking for similar products within the market or will decide to wait until the product price will go down. In turn, this will cause a decline in customer's perceived value (curve 2 in Fig 1). Hence, it is important to understand at which point additional effort (on product development) is not worth to marginal improvements as it will effect product level decisions [6]. Allocation of this point (star on curve 3 in Fig 1) will be strongly influenced by customers' perceived value and other products within the existing market. In the context of incremental product development, this point needs to be re-calculated, for every requirements selection process, when a new release is about to be made as the customer's perceived value is subject to change due to their varying expectation, needs and past experiences.

### 3 Challenges to Requirements Engineers

Although companies put a great amount of effort in their product development process into increasing customer's perceived value, determining *how* and *when* value is created, measured and managed is still a challenge to software developers.

A value-based approach supports the alignment of decisions at business, project and product level with the aim of maximizing business value while maintaining a profit for a given investment. By following this argument, we expect that a company needs to create, measure and manage value from business, product and project perspectives. In other words the following value perspectives are importance to software developers as illustrated in Fig 2:

- **Value for business perspective:** Business value to Software Company which stems from product sale.
- **Value for product perspective:** Product value to Software Company which stems from Customer and Market requirements.
- **Value for project perspective:** Project value to Software Company stems from project budget/timing/delivery etc.

Fig 2 illustrates the relationship between the value perspectives, Software Company and Customer. The objective of this model is to show where the value needs to be created measured and managed. It is important to note that as the software company aims to maximize their business value through their product sale and related to its project, in the same way, customers' aim to maximize the value for their own business through the product purchase. The model also shows the relationship value between the company and the customer which is formed through the product buy/sale transactions between these two entities. Fig 2 also includes some additional factors that influence the value creation/measurement/management for both the software company and the customer, i.e. Competitor and Market.

It is important to note that there are some other factors that have an affect on value creation such as economic movement and social culture which are not illustrated in this model as it gets more complicated. The intention of the model in Fig 2 is to mainly address the value perspectives from a software company point of view and provide a guideline to practitioners to give them an idea about where the value

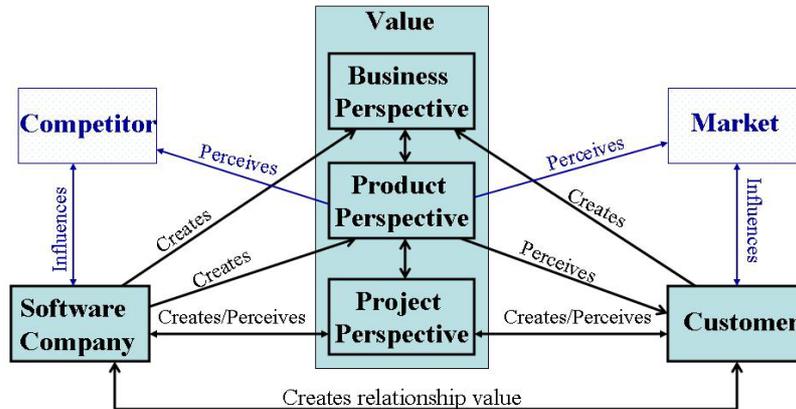


Fig. 2. Software Company-Value-Customer Triangle

needs to be created, measured and managed while making sure that the product, project and business level decisions are aligned and different value perspectives are involved in the decision making process.

#### 4 Discussion

A value-based approach is about linking strategy, measurement and operational decisions. Unfortunately there is no “one size fits all” model for software developers that shows *when* and *how* to create, manage and measure for value.

Value creation in software development is not a one-off event rather it is an iterative approach. It is supported by aligning product, project and business level decisions throughout the development process [2] and [3]. This requires that software developers firstly consider customers’ requirements, business requirements and technologic opportunities when making decisions. Secondly, they need to have a sound understanding of both technical and business implications of decisions that have been made throughout the development process. Thirdly, it is essential to understand the business dynamics that drive software development in terms of cost, time, and product quality as well as how software processes and products interconnect.

A customer buys the product not only for its price but also for other reasons, such as the lifestyle it creates for them. For example, George Jensen, or Efva Attling (designers from Denmark and Sweden) manage to create lifestyles around their products. Value creation strategies are highly contextual. Companies basically adopt one strategy that best suits to their circumstances and that is successful within the context of their business environment. An example of this is the Sony-Ericsson and Siemens-Nokia marriages for their mobile phone products.

Measuring for value is always crucial for a software company. There are metrics used to measure technical performance. In many cases, regrettably, technical performance metrics mismatch (or disconnected) the business strategy that drives the value in software development. Hence, alignment of key performance metrics with strategic objectives is crucial.

An effective management of the product development process contributes to sustainable competitive advantage for software companies. Managing for value requires sound understanding of company structure, business objectives, market and product strategy as well as the social culture of the company to manage for value.

## 5 Conclusion

Favaro [8] points out that the purpose of the requirements process is to add business value. This is a big challenge for requirements engineers because they are used to operating in a value neutral setting in the past. As global competition forces companies to become increasingly competitive and responsive to consumers and market developments, ongoing discussion in SE indicate that a value-based approach makes all the difference to product success. It puts the requirements engineer in the position of managing requirements in a way that allows the company to take advantage of the strategic business opportunities.

We believe that the alignment of technical decisions with business strategy continues to be a challenge as requirements engineers, product managers and IT managers operate at different levels. Product quality, its performance and product/project cost control (short or long term) will remain important, but the attention must be refocused on flexibility, creativity and timing. Hence,

- It is necessary to provide timely feedback between business and technical level decision makers and to support communication between them
- It is crucial that software developers put more effort into expressing the technical decisions as a business case while adding value to the product at hand. In the same way, management should have a good understanding of the internal structure of the company, the product and operation level decisions.

This article is set to provide some preliminary discussion on value aspects of RE inspired by the importance of understanding the terminology and the concepts that we borrow from economic theory. We are currently in contact with practitioners from software industry to evaluate the model in Fig 2, and we are conducting industrial studies in several countries including Sweden, Germany, Australia and China.

## References

1. Alwis, D., Hlupic, V., Fitzgerald, G.: Intellectual Capital Factors that Impact of Value Creation. In: 25th Int. Conf. Information Technology Interfaces, Cavtat, Croatia, pp. 411–416 (2003)
2. Aurum, A., Wohlin, C., Porter, A.: Aligning Software Engineering Decisions. *International Journal on Software Engineering and Knowledge Engineering (IJSEKE)* 16(6), 795–818 (2006)
3. Aurum, A., Wohlin, C. (eds.): *Engineering and Managing Software Requirements*. Springer-Verlag, Heidelberg (2005), ISBN 3-540-25043-3
4. Biffi, S., Aurum, A., Boehm, B., Erdogmus, H., Grunbacher, P. (eds.): *Value-Based Software Engineering*. Springer, Heidelberg (2005), ISBN 3-540-25993-7
5. Boehm, B.W., Sullivan, K.J.: Software Economics: A Roadmap. In: *Proceedings of The Future of Software Engineering Conference*, pp. 319–343 (2000)

6. Browning, T.R.: On Customer Value and Improvement in Product Development Processes. *Systems Engineering* 6(1), 49–61 (2003)
7. Faulk, S.R., Harmon, R.R., Raffo, D.M.: Value-Base Software Engineering: A Value-Driven Approach to Product-Line Engineering. 1st International Conference on Software Product-Line Engineering, Colorado (2000)
8. Favaro, J.: Managing Requirements for Business Value. *IEEE Software*, pp. 15–17 (2002)
9. Favaro, J.: Value-Based Management and Agile Methods. In: Proceedings of 4th International Conference on XP and Agile Methods (2003)
10. Heinonen, K.: Reconceptualizing Customer Perceived Value: The Value of Time and Place. *Managing Service Quality* 14(2/3), 205–215 (2004)
11. Henneberg, S.C., Pardo, C., Mouzas, S., Naude, P.: Value Dimensions and Strategies in Dyadic Key Relationship Programmes: Dealing with Dualities. In: Proceedings on the 21st IMP Conference, Rotterdam (2005)
12. Mill, J.S.: *Principles of Political Economy with Some of Their Applications to Social Philosophy* (First published in 1848). Winch, D. (ed.) Harmondsworth, Penguin (1970)
13. Storbacka, K., Lehtinen, J.R.: *Customer Relationship Management: Creating Competitive Advantage through Win-Win Relationship Strategies*. McGraw-Hill, New York (2001)
14. Weinstein, A., Johnson, W.C.: *Designing and Delivering Superior Customer Value: Concepts, Cases, and Applications*. St. Lucie Press, Boca Raton, Florida, USA (1999)