Value-Aware Approach to Management of Innovative Software Products and Services

Aleksandra Karimaa

***Abstract*—The success of software products and services is greatly driven today by the process of innovation. Innovation encourages creation of unique first-of-the-kind products and services. The value and potential of such products can be difficult to assess and manage. The ambiguity increases the risk of taking strategically wrong decisions and investments concerning the product portfolio development. This contributes to business condition of the company, especially in case of so common today one product companies.**

This work introduces the concept of value-aware approach to product and service management, reviews applicable methods and comments their usability in context of development and management of innovative software products and services.

***Index Terms*—Product management, software, value assessment, value monitoring.**

I. Motivation

Awareness of the value potential in company's products and services is an essential asset of well managed company. Product value assessment defines product place on the market and targets customer base and possible competitors by defining price range acceptable for the customer in given business model. This in turn contributes to product vision and strategy and determines possible development directions. Ambiguous value of product is likely to have negative impact on product development increasing the risk of wrong investments as it is challenging to decide which features are worth to develop if the business value of current solution is not known [7].

Product value assessment is critical phase of many financial processes and provides a base for reliable decision making and profitability-oriented product and company management.

All consequences of above have great impact on value of not only product or product portfolio but also the value of company itself. Therefore, the ability to assess value of the product is one of the most important challenges for innovative software companies especially in case of portfolio being based on one type of product.

The value-aware management relies on value monitoring. It enables management decisions to be evaluated according to their estimation of value-adding capability. The advantage of value-aware product management is a possibility to see beyond traditional methods focused on comparing production cost with profit forecast.

The value-based management is focused on delivering the product which satisfies the needs of strategically important markets, which means addressing uncertainties of: value of the market for the product and the market share the product can capture. These have to be compared against possible cost, and their financial measures to select suitable (business profitable) development strategy scenarios.

The ultimate goal for value-aware management is to understand relationship between product value and business itself and to create the model of the product value with helps to estimate financial value of the product when investigating different strategy scenarios.

Manuscript received December 20, 2012; revised February 10, 2013. This work was supported in part by the Business and Innovation Development department of University of Turku.

The author is with the Business Innovation and Development at Turku University, Turku, Finland (e-mail: aleksandra.karimaa@utu.fi).

II. Challenges and Goals of Software-Based Product Value Assessment and Monitoring

The subject of product value assessment is complex. One could also challenge if is worth effort - very often product decision are taken based on manager experience and certain assumptions, such as market trends or possible interest from customer side. However, this might be very difficult and not accurate method when it comes to innovative one-of-the-kind products and services

A. Quantitative Measures of Qualitative Value

Despite the fact, the (business) value of the product has a direct impact on company financial condition; its definition is rather informal and not precise. The quantification of value is challenging due to its qualitative nature and its quantitative measurements are often not accurate.

In general, software-based products below to the group of so called information goods. The value of such good lies in the value they present to their customers, and therefore it is focused around subjective attributes. The most common methods of accessing the product value are focused on end-user or customer valuation and do not reflect well the real market value and potential of the product. Furthermore, most of the methods used for customer experience are not sufficiently reliable- the most popular being customer survey could be easily challenged for lack of accuracy or lack of explanation of motivation behind customer choices.

B. Objective and Subjective Value

Despite the fact only some of researched value attributes could be completely quantifiable and measured, the qualitative attributes shall not be omitted and the value assessment shall research both objective and subjective values of the product. The objective values include the value of production which reflects the cost of the software production. The cost is proportional to so called production

effort (assuming the resources are available and utilized), which in turn is proportional to size of the software, processes and tools, personnel skills and other factors. The idea of this assessment to provide mathematical, statistical and possibly empirically verified method to model such effort and cost estimation before the project is executed (or even decision about its execution is taken).

The subjective values include product quality, user needs, the context of offering, and similar. Despite the fact, the character of such values is rather qualitative the assessment shall provide the means of their measuring or evaluation and deliver the value preferably (but not necessarily) in given currency. The subject of measuring such values will be explored in later chapters concerning the software and product quality as well as product fulfilling certain markets needs and user expectations.

C. Value Attributes

Different types of offering focuses on different product values. If the same product is offered as off-the-shelf product the focus of the value assessment will be different from the case when the product is part of complex system offering. In the first case the focus will be put in the aspects that can "encourage" product volumes, minimize the production costs and simplifies the maintenance and support. In the second case the value of the product will be high if the product has capacity to flexibly adjust to support given system scenario, the interfaces are open and standardized (which enables multiple vendor cooperation and healthy competition), the cost of integration are relatively low or any other aspects creating one-of-the-kind product offering.

Moreover, the product value depends also on assessment recipients. If the product value assessment is done to estimate development budget the outcome of such exercise might be different than it would be in case of preparing marketing material for the product or preparing the product pricelist by manager or product vendor. First one will be focused on quantitative values such as number of line codes, man-days, or money cost of new features; the second will concentrate on rather qualitative aspects such as competitive market demands and needs or target customer expectations; the third one will most likely benchmark the product features against features (usually it is considered as quantitative measure if the features are similar in different products) and price of similar products existing on the market (quantitative measure) taking into ac-count production cost vs. profit margins and strategic decisions to maintain or change given market share.

III. Value-Aware Production

The assessment and measurement of the production value provides the knowledge about the non-marginal value of the product. Non-marginal product value can be used by managers in process of business decision making.

The ultimate goal of value-aware production is to develop the methods of value monitoring and control that are the most suitable for specific product to be delivered. It means, addressing the formality of user requirements, the process and its acceptance by the end user, and finally the challenges of current management practices that must be addressed.

A. Agility of Management

Value-based production management is focused on human factors. The efficiency of implementation of value-aware production relies greatly on agility of software development methods and processes and their interaction with human perception of values that is Value Based Software Engineering (VBSE) practices. VBSE describes management practices which use "risk considerations to balance software discipline and flexibility and to answer other key "how much is enough?" questions [3].

Most of VBSE management practices are in fact defined well under the common term of project management for agile software development. Scrum is a management method [8], which was invented to address the problem of so called requirements churn. Requirements churn is very typical problem for management of complex software - during a project the customers are very likely to change their requirements. Scrum focuses on maximizing the team's ability to deliver quickly and respond to emerging requirements. Scrum relies on communication between all team members involved in the product creation.

Scrum is a process skeleton that contains sets of practices and predefined roles. Scrum practices capture very well the dynamics and final value of the product.

B. Development Effort Management

Kanban management method introduced by David J. Anderson [1] improves the efficiency of development effort. The general idea is to display the queue of work, the individual tasks and their delivery for participant involved in development project. The visualization of tasks management has informational role and provides monitoring of value of production, which in turn enables management and control of such value.

C. Production Quality Management

Extreme Programming practices include two-person team programming (quality), code review practices and high focus on unit testing. The focus of this method is to monitor and improve software quality and address the problem of requirements churn while keeping short development cycles.

Agile Modeling method improves end-user/customer value-management by encouraging customers' participation in development modeling. Agile Modeling replaces UML –based modeling methods as not understandable for customers involved in development projects.

D. Life Cycle Management

Dynamic systems Development Method (DSDM) and similar Rapid Application Development (RAD) describe a general approach to management of software production lifecycle where the cost, quality and time are fixed and development tasks are categorized using MoSCoW prioritization model (Must, Should, Could and Won't Have). The product deliverable is adjusted to meet the given time constraint.

Feature Driven Development method is focused on management of requirements specification phase by splitting feature development requirements and software modeling into multiple phases and describing feature-oriented check-point for each of the phases.

E. Resource Management

The resource scheduling and practices of time reporting are one of the most crucial for efficient factors when it comes to production cost estimation as they provide the measurement tools to validate the estimates. The difference between estimated production cost and the final cost might come not only from wrong selection of estimation method but also from imperfect process of measuring the effort and distributing the effort within the team.

Typical approach to time or effort reporting is measurement of time spent by the team member on actual production process (design, coding and testing). McConnell [6] proposes to extend the reporting to include the non-production activities, such as holidays, sick leave days and support activities. The result of this operation will be effort measurement which better reflects availability and actual use of the production effort. This approach for improving time reporting system is continued further by Joel Spolsky [9], who claims the effort is utilized not only on production and non-production activities but it is also affected by other factors such as frequent interruption of normal tasks or non-job related activities performed (such as discussion over football game, surfing or longer coffee break) which are to some extend unavoidable. He has proposed the method called Evidence-based Scheduling where the measurement (so time reporting) is limited and is focus only on reporting job milestones. The advantage of this method is it more accurately reflecting the actual state of resources and used effort and therefore it can be trusted as a way of effort measurement.

The assumption that the value of the software as the cost of its production is proportional to the effort of production is valid if and only if the resources are available for given job and there is no idle resources waiting for job tasks. Classic scheduling methods might be sufficient for project of small size. However, multiple factors encourage applying more intelligent resource utilization tools can be found, these include: multiple multi-person development teams working on different development projects simultaneously, software modularity with modules interdependencies, uncertainty of correctness of job estimation caused by adaptation of new technologies, and tools.

Some software valuation methods also offer the mechanisms for resource utilization. PERT (Program Evaluation and Review Technique) method offer task scheduling mechanisms with parameters of effort and time to deliver. The method was created to handle the schedule planning for complex systems production [5]. The method allows scheduling paths of tasks, which paths can be dynamically routed depending on available time, resources and required quality to be achieved. One of the advantages of this method is that it can handle minimal effort reporting in the development team.

IV. Quality Assessment and Management

Let us consider the following case: two systems are under investigation. In system no.1 the investor decided to support the strategy expanding the list of software features to increase the market share. Shareholders of system no.2 business are choosing to concentrate on improving the quality of the existing product. When comparing two systems it can be noticed that in the offering or solution quoting phase system no.1 will be more attractive. However, if the implementation of the features has poor quality there is a significant risk there maintenance cost will increase (proportionally to market share), the customers will be lost or /and the owners are legally forced to deliver the correctly implemented functionality including possible fees for the project delays.

Considering the above scenario it can summarize that the general motivation towards assessing and managing software quality as a combination of two factors: cost and risk- both being crucial for business well-being.

The management of cost is an obvious motivation- profitable business is healthy business. Additionally, the software business is rather mature. It means the COTS soft-ware products are available even for high end market so far occupied by specialized and customized software. It created aggressive price-based competition in the market and therefore the cost management becomes of major importance. In general it should motivate the focusing on quality; however it can be noticed that the maturity of the market and price-based competition creates pressure on delivering new functionality. In this situation, the careful consideration of prioritization between quality and new functions should be done with special focus to be paid to the risk management.

The risk management is very important part of development of software product. Especially safety-critical software developers shall pay special attention to the risk management. The risks can also involve environmental disasters, political, military or social risks. The above concludes the risk management should receive special attention when evaluating the need and value of software and system quality.

Additionally, the implementation of quality management can lower the cost of production. It is especially the true in case of complex. The low quality of the software means the risk of software bugs and errors is higher. If the system is complex, it has many different customers using different sets of software functions and therefore the probability the bugs are discovered is high. The correction of such bugs is much more expensive (and might be even impossible) in later phase of software development. The development team in company overlooking product quality is likely to spend its work-force on correcting the old bugs instead of developing new functions.

However, the main motivation for implementing the quality management is to add the value to the product for both manufacturer and the consumer. Moreover, especially for safety-critical application the quality defines the product and becomes very important product differentiator. Quality, by having impact on efficiency of the processes, has also big impact on the value of the business itself especially if the quality driven thinking is present through the organization.

Often the quality is referred as measure of excellence similarly to value being a measure of worth. In this case the idea behind assessment of value of quality shall help to answer questions on how much will customer pay for certain level of quality. Such assessment should also display the

relation between cost and the quality. It is also worth to under-line that the quality in itself does not provide any value if the product itself does not carry any- the assessment of quality for the product which does not have economical sense would not make sense.

Traditionally, the system quality is defined by two relatively separate types: functional and structural quality. The quality of the software product intends to describe how well the product is fulfilling its purpose and functional requirements (software functional quality), and how well it follows general quality rules of software creation (structural qualities). Altogether, the quality assessment should verify the software is fulfilling general expectation towards such software, such as the level of availability, security, expected efficiency, etc. The focus of quality assessment is provide the arguments for management and control of the product to provide product with certain level of system attributes and system functionality while maintaining cost efficiency critical for business perspective. It is partially ensured by both structural and functional quality control but it required special attention in case of software product.

The functional quality is measured in the process of system testing – the functionality of the software is tested against test specifications and according to the test book. In case of software functional quality the assessment of value behind the quality will try provide the arguments to answer the question: what level of software testing is the best from product value perspective. On one hand, the more testing is done the more systems bugs are find and the software is better. On the other hand, testing increase the cost of software production and it does not guarantee the software is bug-free (just minimizes the risk of encountering the bug at the customer side). The problem of cost efficient functional quality testing is well visible on the example of testing of complex software products. The test book for such system does not exhaust all possible system capabilities. The time and resources are not unlimited, and therefore either the representative test cases are selected (which might be subjective) or only the new functionalities are tested (which might cause problems with backwards compatibility and support of existing functions). The structural quality assessment evaluates the product in terms of following general design rules (for the software of the type) including coding (including the specifics of the language), software architecture, etc.

A. Defect Count and Rework Cost

The simplest approach is typical representation of the manufacturer view for product assessment and it answers the question of defect count (number of defects detected during development and operation) and rework cost. Defect count is measured as number of defects per 1000 lines of code and it is a measure the quality of produced work.

Rework cost is the cost of fixing these defects and it includes development rework cost and operation rework cost. Development rework cost is the rework cost for work happening before a product is released. It is a measure of development efficiency. Operation rework cost is the rework cost for work happening after release- when a product is in operation. This is a measure of the delivered quality.

B. ISO/IEC 9126 Assessment Metrics

ISO/IEC standards define the industrial standards for quality assessment by defining quality models and proposing the metrics for measuring the quality.

ISO defines two types of quality models for evaluation purposes. The first describes metrics for measuring product internal and external quality, where the internal quality describes the general quality characteristics of the software design and the external quality characteristics are the ones which are visible externally when the software is used as part of a system. The second quality model defines metrics for so called “quality in use” which evaluate product quality evaluation in context of specific use.

V. Value in Intellectual Property

The subject of intellectual properties and their value has been covered by multiple publications and it is in general a topic of high interest. Most important discussion is the estimation of value of intellectual properties (IP and understanding the role the value of IP plays in value of the business and the value of the company itself.

Stefano Zambron and Thomas Steward [10] claim, in case of the companies rich in intangible assets traditional financial statement (where reporting value of intangibles is voluntary) are not sufficient to reflect the real value of the company – there is a difference between market value and book value of such company. It is believed to be a case especially in case of companies creating and delivering intangible assets, such as software. Therefore, it is recommended for software manufacturers to assess and report the value of software intellectual assets. The recommendation is especially relevant for software and solution manufacturers on software market, which is very dynamic market in terms of the number of formed, acquired and closed business, but also in terms of innovative solutions and adapt technological novelty.

Another motivation is more general. The software is an asset which can is costly to produce, and very cheap to reproduce and distribute. This means there is high risk of business loss on the manufacturer side when the specifics of software intangibility and the value behind it are overlooked.

The motivation towards software IP value, its assessment and management can be explored further. Baruch Lev [2] claims the success of innovation and effective commercialization depends on smart management of intangible capital and its in-corporation in corporate processes and value creation. It means the IP in the company cannot be successfully managed without knowing the value of such IP and without assessment of such value and its attributes.

It has been known problem- very often (it depends on the country) the legal protection of IP is not well applicable to the software. Typical mechanisms existing in property laws have not been made to protect the software but to handle intangibility of modern media (see music copyrighting, etc). The laws when applied to software and soft-ware creation processes arise many controversies. In case of the software it

is not easy to identify which aspects of program shall be protected as being IP, which processes, what software modules and software deliverables (such as libraries, code, compiled programs).

In principle the value lies mainly in software behavior. However, the behavior itself cannot be protected by traditional methods of legal protection (copyright).

Also, the application of patents is controversial. The process of software creation is innovative, rather than inventive which means it is not easy to define which part of the process or the product could be a subject of patent. It also means the most of the value (in principle the patent should protect the value) is not identified and secured against duplication. Other way would be protection by trading secrets, but then it is not easy to use it as market differentiator – it could not be advertised as it would be secret.

The process of value assessment for company intellectual properties is a part of IP (Intellectual Properties) audit. As a result of this process each IP is turned into financial model according to selected International Valuation Standards, such as Financial Accounting Standards Board (FASB), Uniform Standards of Professional Appraisal Practice (USPAP), International Valuation Standards Committee (IVSC) (50 Countries) or International Financial Reporting Standards (IFRS)

In general, there are many different approaches to IP valuation where the most popular with their methods and metric will be described below.

A. Cost-Based IP Valuation

Cost- based IP valuation determines the value of IP based on the cost of purchasing, or producing asset of similar utility. It means the value behind e.g. innovation is the value of its functional substitute. The most common example of such valuation is evaluating software as an alternative against original functionality product, for example software-based movie rental services against movie rental stores. If such invention would be evaluated as a subject of IP valuation its value will be estimated based on the cost of professional video rental store business.

In general value assessment of this type can be done based on history (how new is the approach of providing movie rental software services), replacement cost (how much it cost to replace the online video rental with another way of providing movie rental functionality), replication cost (how much cost to produce the function of similar functionality). .

B. Market Supply and Demand-Based IP Valuation

The main idea behind this value assessment is that IP is a normal product which is a subject of market rules. This principle reflects the value of IP as the result of supply and demand combination. Practically, the idea boils down to the value estimation based on price estimation on which the transaction between the buyer and the seller will take place. In the case of assessing IP-value of particular software features the manager would have to investigate the customer motivations towards buying such features on the given market in the given date. Most probably the initial valuation is needed to be adjusted later based on the feedback from the buyers. This method presents a dynamic approach for software IP value assessment. .

C. Income-Based IP Valuation

These methods originate from assumption that IP value of intangibles is bringing wealth. In its most basic approach the value assessment is done based on estimation of future financial benefits (gross revenues and profits) coming from the particular IP item. These methods are applicable if the future economical benefits are possible to be identified and estimated precisely.

One of such methods is 25 Percent Rule which assess the value based on projected gross revenues. The 25 percent rule assumes the customer interested in particular IP item (usually software license or patent) would be willing to give up 25 percent of profits for the right to use a patented technology or device, keeping 75 percent. In addition to the fact the proposed proportions are not well applicable to high competition software market this method has also other disadvantages. It does not reflect changes on investment and risk so it is not well applicable to dynamic and emerging markets. Also it does not reflect different go-to-market and sales approaches (and their cost consequences) visible in today's software market. It is not well applicable in the environment of many alternative and competing technologies where proposed values and the level of commitment is too high cost for the potential user of IP. In summary, the method is considered being not reliable in case of non-matured software markets where the future value of the innovation is not known.

Another variant of Income-based method assume such valuation should be done based on income currently generated by given IP item. This, in turn requires small dynamics on the income generated by given software feature, so it is applicable only for the features existing long enough on the market. It also requires the feature to be well present on the market as well as well patented. This model is visible in software market for IP items such as coding algorithms.

In case of algorithms IP the Industry Standard Royalty Rate assess the value of the IP item based on rates in past transactions in an industry.

D. Real Options

The real options assessment method assumes the value of IP is proportional to the risk of compromising the IP (so in case of complex software - copying the feature). It assumes the value of the IP decrease in time. The method is based on complex mathematical method. It is used mainly in world of major software system players for IP expected to present significant value and high risk category and for high-risk strategic projects evaluation.

E. Competitive Advantage Valuation (CAV)

Competitive Advantage Valuation is a process of identifying business value of the IP by specifying market value of the functionality where IP are identified.

The method is performed in steps. First the IP (for example patent) is associated with product, the product is described by with set of competition parameters (such as accuracy, portability, etc) and then the IP is associated with these parameters. Second, contribution of the IP to the competitive

advantage is calculated. The calculation is done by comparing the product to average substitute product. Additionally, the IP within the product are compared against each other to calculate so called relative competitive ad-vantage. In third step the net value of the product is calculated and the fraction of this value is attributed to relevant functionality. Forth step is dividing further the value of the functionality by assigning fraction of net value to each IP based on proportions calculated in previous steps. In the last step the IP value is adjusted to reflect the IP risks.

The method is a good alternative for complex methods (such as Real Options) when manual assessment is preferred (such as in situation when IP value assessment is done for need of software development planning). It is applicable for complex development projects where multiple IP might be involved [4].

VI. User Value

The product value assessment is not reliable is it does not take into account the value perception of the end-user. The users' expectations towards the software product and the level of satisfaction with current solution are present in process of software production in form of customer requirements.

The customer aspect of product value is the most important – the value should have direct impact on price of the product and the customer(s) is there to decide if the valuation was correct. The customer also represent (to a lesser or greater extent) the requirement of given market sector and therefore should (indirectly) define the product strategically addressed for this sector. The interests of customers and the end users might vary depending on the product environment. Networking of different systems (which is natural trend in modern technologies) emphasizes the values of end user, such as ergonomics. The profitability aspects of product development put stress rather on customer.

The structure of this work might suggest the user value is yet another aspect of product value to analyze. However, it is indirectly or directly present in every valuation method. In general, product value assessment methods are reflecting user-values if they provide the means of measuring the functionalities and therefore measuring user's view. The simplest approach to measure user value in product delivery is to assess the amount of passes functional test cases. Such assessment is typically part of product deliveries and presented as a FAT (Functional/Factory Acceptance Test).

In order to efficiently design and deliver the product fulfilling customer expectations the cooperation with the user should be present already in very early phases of product creation. When traditional software creation methods might be suitable for some type of development, typically they don't give much attention to the user outside the phase of précising and freezing the requirements. In this work the author argues the advantages of agile software creation methods. These methods bring the attributes of user value into the product in every phase of product lifecycle. They are also great help in customizing management practices to reflect the care of user-values.

VII. Conclusion

The main motivation for improving product creation and management model is to improve the product or products profitability. This work gives an overview of product value assessment methods applicable in management of software products. Author believes the work provides a good base for indentifying present market value of software product and assessing the value of such products in future..

Future considerations for the topic of software value assessment should reflect the economic changes brought by modern technologies where significant value of the product originates not from its scarcity but its popularity. The best example can be presented by valuation of Facebook Company where the utility factor of the product is minimal, the product does not necessarily satisfy critical user's needs but the company value originates from information sharing and customer database.

Acknowledgment

Author would like to thank prof. Pasi Malinen and prof. Antti Paasio for valuable comments and support for this work.

References

[1] L. Baruch, *Intagibles: Management, Measurement and Reporting*, New York, U. S. A: Brookings Institution Press, 2001.
[2] D. J. Anderson, *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*, New Jersey, U.S.A: Prentice Hall, 2003.
[3] B. Boehm and L. Huang, "Value-based software engineering: A case study," *Browse Journals & Magazines Computer*, vol. 36, issue. 3, pp. 33-41, 2003.
[4] T. Hagelin, "Competitive advantage valuation of intellectual property assets: A new tool for IP managers," *IDEA: The Journal of Law and Technology*, vol. 44, 2003.
[5] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar, "Application of a technique for research and development program evaluation," *Operation Research*, vol. 7, no. 5, 1959.
[6] S. McConnel, *Software Project Survival Guide*, Washington, U. S. A: Microsoft Press 1997.
[7] T. Patton, "Ambigious business value harms software products," *IEEE Computer Society*, vol. 25, no.1, 2008.
[8] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, New Jersey, U.S.A: Prentice Halls, 2002
[9] J. Spolsky, *Joel on Software: And on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, Designers and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity*, Berkeley, U.S.A: Joel Spolsky, 2004.
[10] R. D. Stewart, *Intellectual Capital*, U.S.A: Doubleday, 1997.

A. Karimaa was born in Cracov on 20 March, 1978. She holds MSc degree in electronics and telecommunications from Silesian University of Technology, Gliwice, Poland (2004), BSc in information technology from Turku Polytechnic, Turku, Finland (2006) and MBA from Turku School of Economics, Turku, Finland (2011) The author's major field of study should be lower-cased.

She is has dozen years of professional experience in software technology industry, including several years of experience in product management in video technology products. Her publication topics include software product creation, software complexity, security, and new solutions evaluation. Her research interests include digital video technologies, product management and business development in technology sector.