A Study in Market-Driven Requirements Engineering

Carina Alves[1], Silvia Pereira[1], Jaelson Castro[1,2]

[1]*Centro de Informática, Universidade Federal de Pernambuco*
{cfa,scp,jbc}@cin.ufpe.br

[2]*Instituto Trentino di Cultura - ITC*
Istituto per la Ricerca Scientifica e Tecnologica - IRST
Trento-Povo, Italy
jaelson@itc.it

Abstract

Packaged software is recognised as an important segment of software market. Market-driven software development consists of developing packaged software to a market rather than to a specific client. It has been argued that the requirements engineering process for market-driven software differs from customer specific software. This paper presents a study on market-driven requirements engineering. We present ten hypotheses we have collected from the literature in this field. We discuss the implications of these hypotheses and identify areas to conduct further research.

1. Introduction

Market-driven software development has gained increasing importance in software engineering research and practice. Market-driven software companies focus on developing, packaging and selling software products to entire markets rather than developing bespoke software to specific clients [7]. By following this strategy, companies have the prospect to reach larger consumer markets, generate increased revenues and growth opportunities. A fundamental challenge faced by these companies is that they have to battle against several competitors for market share. Key goals of market-driven software companies are satisfying customer needs at the right time and at adequate price. To gain competitive advantage, companies must achieve an optimum balance among these goals instead of maximising a single one [15].

Market-driven development is also referred in the literature as packaged software development [32] or Commercial-Off-The-Shelf (COTS) software development [28]. Several authors have examined the characteristics and problems involved in market-driven development. A consensus among them is that development of market-driven software products produces a major impact on established software engineering practices [9][17][23][28]. We are particularly interested to examine the impact market-driven software development brings to the requirements engineering (RE) process.

We are conducting a study that focuses on current practices and challenges faced by market-driven companies in Brazil. This paper reports preliminary results of this study. We investigated the relevant literature in the field of market-driven requirements engineering. Several academic and industrial surveys have been published in this area. We decided to present the results of our study following the same spirit as Basili's renowned papers COTS-Based Systems Top 10 list [4] and Software Defect Reduction Top 10 List [5]. In the following section, we present a list of 10 hypotheses on market-driven software development characteristics and current practices. We have collected information from several sources. For each hypothesis, we discuss technical and organizational implications.

2. Market-Driven Software Development Top 10 List

Hypothesis 1

Time to market is a key strategic objective of packaged software companies.

Source

According to [9][15][23][32], time to market is a crucial concern that market-driven software companies have to meet. Time to market is important during all stages of product lifecycle from its launch through each successive release. During the initial stages, suppliers aim to establish their products in the (sometimes incipient) market. In later stages, suppliers have to retain and expand their market share [32].

Implications

Karlsson [15] suggests that a key goal of suppliers is to develop and release products to the market at the right time “not too early, not too late”. Late arrivals may cause loss of competitive advantage or even complete elimination of market share. On the other hand, if the product arrives too early, the market may not be ready for the product. Sometimes, arriving in the market a bit later is more beneficial than being a first comer. In [2], we have discussed the importance of understanding the dynamics of emerging technologies and identifying the different stages that a technology achieves dominance in the market.

Hypothesis 2

The biggest challenges to growth are management and marketing related instead of technical problems.

Source

Results from a survey conducted in software product companies in Finland show that the most critical challenges faced by studied companies are management and marketing problems [12]. According to this study, these are key areas to focus in order to successfully manage the growing business. Interestingly, technological challenges seem to be less critical than business related ones.

Implications

An increasing number of software product companies begin as small entrepreneurial start-up businesses [32]. These companies generally have highly qualified technical developers but with limited business and managerial expertise. As a result, organizational problems such as, lack of skilled sales and marketing personnel and high staff turnover, have considerable impact to these companies.

Hypothesis 3

Requirements are generally invented by developing companies.

Source

Before the first release of the product, there is no real group of stakeholders [9]. There are only potential customers and users of the product. This means that requirements for COTS products are not always elicited from customers; instead, they are proposed, invented, or designed by the developing organization [28]. Requirements are initially derived from strategic business goals and market opportunities perceived by suppliers [32]. After the product is made available in the market, consumers are better identified and their requirements more easily understood.

Implications

Traditional elicitation methods assume direct contact between requirements engineer and stakeholders [26]. However, software products are developed for a diverse and sometimes unknown market. This situation requires great imagination from developers. Maiden [21] suggests the use of explorative, combinatorial and transformational reasoning to stimulate creative thinking.

Hypothesis 4

Requirements are rarely documented.

Source

In custom specific software development, the requirements document act as a contract between the parties [31]. However, given that in market-driven software development the customer is not yet known until the product is released in the market, there is no need of contract. As a result, there is no need of a formal requirements document and requirements are seldom written [16]. Another important factor supporting this hypothesis is that several packaged software suppliers are small-to-medium enterprises (SMEs) which have not yet developed a formal document-based RE culture [32].

Implications

The lack of a formal requirements document will limit the ability to trace information obtained during the development process. Requirements traceability is particularly helpful to support change management and reuse of software artefacts. Thus, organizations that do not follow a strategy to document requirements may not be able to benefit from requirements traceability.

Hypothesis 5

Requirements selection and release planning are crucial processes to obtain competitive advantage.

Source

Software products are usually delivered to customers in several releases through their lifetime and developers have to select which requirements to implement in each release [29]. In this context, requirements prioritization assumes an important role in the release planning [30]. According to Buyukekici [6], the priority of a requirement “reflects relevance of the requirement for market success, its urgency for implementation, its impact for future development”. The development of software products requires the allocation of resources to activities related to requirements prioritization, selection and release planning to ensure continuous delivery of competitive software releases [13].

Implications

The right selection of requirements during the release planning of a product involves the analysis of requirements importance, cost estimation and requirements interdependencies [9]. Suppliers have to understand market segments and growing opportunities to perform tradeoffs between requirements priorities [30]. Moreover, the selection of right requirements allows suppliers to meet customer expectations at controlled budget.

Hypothesis 6

Relationship between suppliers and customers is generally long but with limited proximity.

Source

Software products are developed and distributed to market where the physical distance between suppliers and customers is usually large [23]. When using products available in the market, customers perform the activities of system planning, evaluation, selection, installation and maintenance **[36]**. Compared to traditional software development (in which customers order software solutions tailored to their needs) the interaction between suppliers and customers of software products is less imperative [33].

Implications

According to Sawyer [33], customers and suppliers use a variety of intermediary strategies to interact. In particular, requirements for a particular software product are gathered from a variety of sources, such as market research and product reviews rather than from eliciting customer needs [18]. An important result from this observation is that many of the assumptions made for traditional software development may be inapplicable for packaged software development.

Hypothesis 7

The failure of product launches is largely due to the product not meeting customer needs.

Source

According to an industry survey conducted by AMR research, 46% of respondents believe that the failure of product launches are due to the product not meeting customer needs [35].

Implications

Different sources have to be examined to understand the needs of stakeholders [10]. The primary goal of market-driven software development is to create solutions that will satisfy the needs and expectations of customers within very different contexts of use [16]. For an innovative software product to be appealing to customers, it needs to bring benefits over existing technology, either as a new functionality, better quality or reduction of cost [2]. COTS products usually offer more functionalities than customers actually need [36]. Suppliers implement these extra features in an attempt to differentiate their products from competitors and hence, make their products more attractive to consumers. However, some of these extra features may cause unwanted mismatches between what the consumers need and what the product offers (i.e. between consumer requirements and product features) [3].

Hypothesis 8

Requirements are only validated after the product is released in the market.

Source

The absence of customers before the product's first release affects the requirements validation phase [9]. Final validation of packaged software can only be done post mortem by examining product's market acceptance [10].

Implications

According to Carmel [7], the acceptance of packaged software can be measured in terms of sales revenue, market share, and good product review. This means that before the first release of the product in the market, suppliers have to deal with uncertainty and risk of product commercial success/failure. This situation is particularly hard when a software product brings radically innovative features. Users may not be able to perceive the benefits brought by the product and effective customer awareness has to be developed [2].

Hypothesis 9

Packaged software suppliers generally have an ad-hoc requirements engineering process.

Source

A number of packaged software suppliers have started as small start-up business. These companies operate under quite strict budget and resources [16]. The rigid constraint of time-to-market forces them to follow a lean software development process. Given that products are developed to the market, there is no need of a formal contractual requirements document between the parties. For all these reasons, software package suppliers usually have an ad-hoc requirements engineering process [32].

Implications

It is largely agreed that developing companies following a well-defined requirements engineering process may obtain considerable benefits in the medium to long term [31]. However, some practices may be quite expensive and complex to be adopted by companies with limited resources. As a result, more lightweight requirements engineering approaches may be needed.

Hypothesis 10

Market-driven development presents fundamental differences to the RE process such that traditional RE practices cannot be used as-is.

Source

Several authors have claimed that market-driven RE differs from customer specific RE. This observation is a result of literature surveys [10][28][32][33] as well as industrial studies [8][12][14][16][25].

Implications

The development of packaged software brings new challenges and opportunities to developers that did not occur in customer specific software development. Thus, it is necessary to carry out studies to examine whether traditional RE practices can be used by packaged software suppliers and how these approaches must be adapted to fit the particular needs of market-driven software development.

3. Discussion

In the previous section we have presented ten hypotheses we have collected from relevant literature on market-driven software development. We have examined other surveys and studies focusing on current research as well as industrial practice in the field. This research has improved our understanding of market-driven software development from a theoretical perspective. We can draw some preliminary conclusions concerning the characteristics and challenges involved in the development of COTS products. A key observation is the general consensus among researchers that market-driven RE differs from customer specific RE. They claim that the most RE processes, methods, techniques and tools were developed to support the development of bespoke software and these approaches fit poorly the needs of market-driven RE. We summarise the main differences between market-driven and customer specific RE in Table 1.

In this paper we have focused our discussion on COTS software development from the perspective of the suppliers of these packages. Another relevant stream of research in this field is the requirements engineering for COTS-based software systems. This development paradigm involves the evaluation, selection and integration of COTS products available in the market by consumer organizations [1]. We can say that COTS-based development and market-driven development look at opposing perspectives of the same problem. In COTS-based development, the viewpoint is from consumers acquiring software products, while market-driven development takes the perspective of COTS suppliers.

Similarly to what occur to suppliers of packaged software, organizations who build software systems based on COTS products face new challenges during the requirements engineering process compared to traditional (i.e. customer specific) RE.

In traditional software development, the requirements engineering activity basically consists of eliciting stakeholder needs, refining the acquired high-level goals into non-conflicting requirements, and finally validating these requirements with stakeholders [26][31]. The main objective of the requirements engineer is to ensure that the requirements specification meets stakeholder needs and represents a concise and clear description of the system to be developed. Broadly speaking, the specified requirements will be translated into software architecture, and ultimately implemented.

Table 1. Comparison between customer specific and market-driven requirements engineering (based on [15][23][26][28][31])

Characteristics	**Customer specific RE**	**Market-Driven RE**
Key objective	Compliance to requirements specification	Time-to-market, competitive advantage
Success criteria	Customer satisfaction, acceptance	Sales, market share, profit, product reviews
Customers	Few customers, close contact between supplier and customer	Several customers, difficult to establish contact and they are initially unknown
Elicitation phase	Acquired from stakeholders using traditional elicitation techniques	Invented by developing team for the first release of the product
Specification phase	Requirements document acts as contract between customer and supplier	Less formal specification, requirements are verbally communicated
Prioritization and negotiation phases	Engage stakeholders to prioritise requirements and negotiate conflicting requirements	Support requirements selection for release planning
Validation phase	Performed together with customer before product is delivered	Product acceptance occurs after product is released in the market

Therefore, it is reasonable to assert that stakeholder requirements play a controlling role in system development [36]. In contrast to the traditional development paradigm, requirements engineering for COTS-based development (CBD) must consider new sources of influence created by the COTS marketplace.

Given that COTS products are developed to satisfy the requirements of an entire market instead of the specific requirements of the buyer organization, it is possible that mismatches may occur between what is desired by the stakeholders of the organization and what it is possible to achieve with the COTS product [3]. As a result, consumer organizations have to specify requirements as flexible as possible in order to increase the number of COTS candidates that might satisfy these requirements [24]. In CBD, the analysis of requirements is a highly interactive and incremental process where the refinement of requirements is driven by the availability of COTS products. When selecting COTS products, organizations should be prepared to make simultaneous tradeoffs among requirements the organization wishes to satisfy, availability of COTS product capabilities, risks imposed by COTS alternatives and costs to develop the system (i.e. acquisition, adaptation and integration costs).

In previous research, we have performed extensive work in the area of requirements engineering for COTS-based development [1][3]. We believe that knowledge gained in this area can benefit our current research. The rationale for that is because having a deep understanding on the processes and challenges faced by COTS consumers, we may be able to find recommendations and good practice guidelines informing COTS suppliers how to better satisfy customer needs, and consequently, better position their products in the marketplace.

4. Conclusions and Future Work

This paper presents the initial results of our research in market-driven software development. Our contribution here was to provide an explorative and critical analysis of the relevant literature in this field. Important insights could be obtained from this study. In particular, we have observed that market-driven requirements engineering presents new challenges (here described in the top 10 list) to software developing companies that did not occur in customer specific requirements engineering. As a result, it is necessary to further investigate whether traditional RE models, methods and techniques fit the needs of packaged software development. Another interesting observation is that many of the challenges and opportunities presented in this paper are particularly relevant for SMEs.

The second stage of our research involves an empirical study with software product companies in order to gather deeper knowledge in this area. Similar experiments have been performed and published in the literature and data was collected regarding the software product industry from several countries [12][16][30]. An interesting exercise would be to replicate these studies in the context of Brazilian software industry.

We aim to empirically examine the validity of the hypotheses presented in this paper. Our objective is to conduct studies in market-driven SMEs based in the state of Pernambuco (the Brazilian region where this research is being conducted). These studies will rely on quantitative and qualitative research methods. Firstly, we plan to conduct questionnaires and interviews with studied supplier companies. We want to gather information concerning company profile, types of products being developed, software development processes, requirements engineering practices, among others. In addition, we want to perform more in-depth analysis to obtain qualitative insights and feedback from suppliers. Focus group seems to be a promising method to achieve that purpose [19].

In the following stage, we aim to analyse the data gathered in the empirical studies. Based on these results, we will propose a requirements engineering framework with a set of good practice guidelines, appropriate techniques and methods to address problems faced by studied companies. Finally, we aim to transfer the results of this research to the studied companies. We believe that by empirically investigating current practices adopted by packaged software suppliers, we are able to gain realistic understanding on the problems faced by these companies, and hence, provide a set of solutions to successfully address these problems.

5. Acknowledgements

This work was supported by the following research grants: CNPq 551824/2005-0(Carina Alves), CNPq 304982/2002-4 and CAPES BEX 1775/2005-7 (Jaelson Castro).

6. References

[1] Alves, C. and Castro, J. CRE: A systematic method for COTS selection. In Proceedings of the 15th Brazilian Symposium on Software Engineering, 2001.

[2] Alves, C. Castro, J. The Evolution of Emerging Technologies in Market-Driven Software Product Development. International Workshop on Technology Transfer in Software Engineering, 2006.

[3] Alves, C. Managing Mismatches is COTS-based Development. PhD Thesis. University College London, 2005.

[4] Basili, V. and Boehm, B. COTS-based Systems Top 10 List. IEEE Computer, 34(5):91–93, 2001.

[5] Boehm, B. Basili, V. Software Defect Reduction Top 10 List. IEEE Computer, 34(1), 2001.

[6] Büyükekici, B., Deifel, B., Jacobi, C. and Sandner, R. Prioritization of Complex COTS, Fifth International Workshop on Requirements Engineering for Software Quality, 1999.

[7] Carmel, E. and Sawyer, S. Packaged Software Teams: What makes them Different? Information Technology & People, 11 (1), 6-17, 1998.

[8] Dahlstedt, A. Requirements Management in a Life-Cycle Perspective. Seventh International Workshop on Requirements Engineering: Foundation for Software Quality, 2001.

[9] Dahlstedt, A. Study of Current Practices in Market-Driven Requirements Engineering. Third Conference for the Promotion of Research in IT at New Universities and University Colleges in Sweden. 2003.

[10] Deifel, B. Requirements Engineering for Complex COTS. Fourth International Workshop on Requirements Engineering for Software Quality, 1998.

[11] Fellows, L. and Hooks, I. A Case for Priority Classifying Requirements. Eighth Annual International Symposium on Systems Engineering, 1998.

[12] Hietala, J. Kontio, J. Jokinen, J. Pyysiainen, J. Challenges of Software Product Companies: Results of a National Survey in Finland. 10th International Software Metrics Symposium, 2004.

[13] Höst, M. Regnell, B. Lindesvärd, J. Nedstam, J. and Nyberg, C. Exploring Bottlenecks in Market-Driven Requirements Management Process with Discrete Event Simulations Journal of Systems and Software, pp. 323-332, 2001.

[14] Kamsties, E. Hormann, K. Schlich, M. Requirements Engineering in Small and Medium Enterprises: State-of-the-Practice, Problems, Solutions and Technology Transfer. Conference on European Industrial Requirements Engineering, 1998.

[15] Karlsson, J. and Gurd, A. Introduction to Market-Driven Product Management. Telelogic White Paper. 2006.

[16] Karlsson, L. Dahlstedt, A. Natt och Dag, J. Regnell, B. Persson, A. Challenges in Market-Driven Requirements Engineering – an Industrial Interview Study. Eighth International Workshop on Requirements Engineering: Foundation for Software Quality, 2002.

[17] Karlsson, L. Regnell, B. Aligning the Requirements Engineering Process with the Maturity of Markets and Products. Tenth International Workshop on Requirements Engineering: Foundation for Software Quality, 2004.

[18] Keil, M. Carmel, E. Customer-developer Links in Software Development. Communications of ACM. 38(5) 1995.

[19] Kontio, J. Lehtola, L. Bragge, J. Using the Focus Group Method in Software Engineering: Obtaining Pratitioner and User Experiences. International Symposium on Empirical Software Engineering, 2004.

[20] Lehtola, L. Kauppinen, M. Kujala, S. Linking the Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping. 13th International Conference on Requirements Engineering, 2005.

[21] Maiden N., Robertson S. & Gizikis A., 2004, Provoking Creativity: Imagine What Your Requirements Could be Like. IEEE Software, September/October 2004 21(5), 68-75.

[22] Morisio, M., Seaman, C. B., Basili, V. R., Parra, A. T., Kraft, S. E. and Condon, S. E. COTS-based software development: Processes and open issues. The Journal of Systems and Software, 61(3):189–199, 2002.

[23] Natt och Dag, J. Elicitation and Management of User Requirements in Market-Driven Software Development, Department of Communication Systems Lund Institute of Technology, Licentiate Thesis, 2002.

[24] Ncube, C. and Maiden, N. PORE: Procurement Oriented Requirements Engineering Method for the Component Based Systems Engineering Development Paradigm. International Workshop on Component Based Software Engineering, 1999.

[25] Nikula, U. Sajaniemi, J. Kalviainem, H. A State-of-the-Practice Survey on Requirements Engineering in Small and Medium Sized Enterprises. Telecom Business Research Center Lappenranta, 2000.

[26] Nuseibeh, B. and Easterbrook, S. Requirements engineering: a roadmap. In The Future of Software Engineering, pages 35–46. ACM Press, 2000.

[27] Oudshoorn, I. Development of Packaged Software. Vrije Universiteit Amsterdam, 2004.

[28] Potts, C. Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software. Second International Symposium on Requirements Engineering, York, 1995.

[29] Regnell, B. Karlsson, Host, M. An Analytical Model for Requirements Selection Quality Evaluation in Product Software Development. 11th International Requirements Engineering Conference, 2003.

[30] Regnell, B., Host, M., Natt och Dag, J., Beremark, P., and Hjelm, T. An Industrial Case Study on Distributed Prioritization in Market-Driven Requirements Engineering for Packaged Software. Requirements Engineering Journal, pp. 51-62, 2001.

[31] Robertson, S. and Robertson, J. Mastering the Requirements Process. ACM Press/Addison-Wesley, 1999.

[32] Sawyer, P., Sommerville, I. and Kotonya, G. Improving Market-Driven RE Processes. International Conference on Product Focused Software Process Improvement, 1999.

[33] Sawyer, S. A Market-Based Perspective on Information Systems Development. Communications of the ACM, vol. 44, issue 11, pp 97-102, 2001.

[34] Soffer, P. Goldin, L. Kuflik, T. A Unified RE Approach for Software Product Evolution: Challenges and Research Agenda. International Workshop on Situational Requirements Engineering Processes, 2005.

[35] The CIO's Guide to the Perfect Product Launch. AMR Research, June 2005.

[36] Wallnau, K., Hissam, S., and Seacord, R. Building Systems from Commercial Components. Addison-Wesley, 2002.