Željka Požgaj[*]
Zoran Kalafatić[**]
Hrvoje Sertić[***]

UDK 004.415:33
Review
Pregledni rad

IMPROVED SOFTWARE DEVELOPMENT AS A KEY TO COMPANY BUSINESS GROWTH

ABSTRACT

The current market situation demands from companies to use complex software systems as a business foundation and connection with customers and partners. Such systems must use complex technologies and efficiently support actual business processes. Traditional development and implementation practices are no longer feasible since customer expectations can no longer be fulfilled without their participation. Development projects must continuously learn and improve their competences towards achieving a common goal: software implementation according to the actual customer expectations. This paper explores new economy software development practices and proposes methods to involve actual customers in software development to enable efficient software implementation, which will provide new business opportunities for the company. The proposed methods are applied on the actual development project to evaluate all benefits of the proposed approach.

Keywords: *new economy, software development*

JEL: O30

1. Introduction

Most of the companies competing on the global market are relying on the software systems used to provide services to customers and partners. Regardless of the position in the company or actual business process, software is nowadays the foundation of every business operation. Software systems are used to present a company's business and products to customers all over the world by providing almost real-time information about the company and its products. Customers today perceive a company and its business according to the representation they get during interaction with the company's software systems, so that the company actually depends on the software systems to provide competitive and advanced services to its customers. Company business growth directly depends on the implementation of the advanced software systems, since they are nowadays used as a business foundation.

Competition on the global market forces companies to constantly improve current and implement new software systems. Companies must continuously advance provided services to obtain an adequate position on the global market. The development and implementation of such systems is not trivial: company expectations in terms of

[*] Faculty of Economics and Business, J.F.Kennedy Square 6, 10000 Zagreb, Croatia, e-mail: zpozgaj@efzg.hr
[**] Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia, e-mail: zoran.kalafatic@fer.hr
[***] Ericsson Nikola Tesla, Krapinska 45, 10000 Zagreb, Croatia, e-mail: hrvoje.sertic@ericsson.com
Članak primljen u uredništvo: 30.09.2007.

functionality and performance increase constantly, the company wants to provide new services before the competition to obtain a constant business growth, so that the development team must implement advanced software systems in a very short time frames.

During the last several years, a company would typically order software implementation from the dedicated development team, define its own expectations, and wait for the software to be implemented. Only after the implementation, the company's representatives would test the software system to determine if it is correctly implemented. In the rare circumstances, the implemented software would comply with the company's expectations and the company would be able to base its business growth on the services of the new software system. In most of the other cases, the company representatives would recognize that the implemented software system is not what they had expected, and that it cannot be used as planned, therefore resulting in missed business opportunities. By performing acceptance tests at the end of implementation to evaluate if the software system is actually usable, the company is faced with a great risk: if the implementation effort (expressed in time, finances and company resources) does not result with a usable and efficient software system, the company will lose its market position since it will not be able to provide required services in the planned schedule.

Changes of the market conditions are forcing companies to constantly adjust and improve their own business practices. Companies must constantly develop their offerings and create new services on a daily basis to sustain business growth. A company faces changes in all aspects of its business dealings, directly impacting software development: the company will be forced to modify its own expectations during development due to the changes it constantly faces. Changes of the business conditions may become so significant that the planned software solution may become unusable. Changes of the company's expectations during implementation are difficult to handle with current development practices, which do not provide usable methodology for quick change implementation.

Together with accepting implementation changes, software developers must be able to understand actual company's needs to implement usable software system. Today's business practices and work processes which include cooperation with several business partners, outsourced parts of production and different levels of coordination are usually too complex to be simply analyzed and understood by the software development team. Significant effort is required to understand how the company performs business activities and to identify valuable details of the actual business processes. It is a common situation that the currently used business processes are not suited to the usage of the new software system; most of the development projects must include improvements of the company's business processes aimed to efficiently utilize services of the new software system. Business process complexity is the main source of the misunderstanding between the company's expectations and the developer's understanding of the new software system.

This paper analyzes typical causes of poor implementation of the new economy software systems which usually result in missed business opportunities. Typical problems are analyzed to define software development improvements based on the involvement of the actual company representatives and users of the future software system into all aspects of the software implementation, from requirement definition, through participation in software design and test activities to software verification and introduction into the company. The basic idea behind all proposed improvements is to adjust software development practices towards fulfilling the actual company's needs, therefore enabling the company to improve its business practices and gain competitive advantage.

This paper is organized as follows: after the introduction, the current situation and typical problems in the software development are explored. The third section proposes a method for improvement of the software development projects based on the involvement of the actual customers, which is aimed to provide means for the company's business growth by implementing efficient software systems according to the company's expectations. The fourth section analyzes the application of the proposed method in the actual software development project. Conclusions are given at the end.

2. Problems related to the current development practices

Today's software development practices have significant impact on the company's chances to implement advanced software system and base further business operations on its services. Company must understand the limits which result from today's development practices to set realistic expectations from the development team, but development practices must be adjusted and improved in order to satisfy composite company expectations (Fingar, 2001). The following areas are explored to identify major problems which impact today's development of new economy software systems:

a. Selection of the software vendor

When considering software implementation according to the company's particular requirements, the first step is the analysis of available software vendors in terms of their competences and past experiences. Selection of the appropriate software vendor is often disregarded as an important step in the software development (Allen, 2001), but proper selection of the software vendor actually determines chances to implement usable software system. The usual practice is to select the software vendor according to the recommendations from other companies, vendor's references, and experience in used technologies. The described practice fails in the most important area: past experience is not sufficient to determine if the planned software vendor can understand the company's business needs and actual expectations to implement software system which will enable the company to improve its business operations. Typical problems occur when the company realizes that the selected software vendor with valuable references can not understand their expectations and that the implemented software system represents incorrect vendor's view of the company expectations (Harmon, 2002).

b. Software development practices

Software development practices are constantly improved, meaning that most of today's development activities are aimed towards the usage of new and advanced technologies (Fioravanti, 2005). Most of the development activities are still sequentially performed, so they are not suited to accept constant changes (Berkun, 2005). Analysis of the company expectations and definition of the implementation requirements is usually the first task of every development project. After the requirements are determined, development continues with software design and implementation, and concludes with the verification phase, which must determine if the company expectations are satisfied. Typical development projects often disregard the importance of proper introduction of the new software system into the company and education of actual users to efficiently use its services (Coyle, 2002) to perform improved business operations. In that case problems arise, since then all sequential development phases must be repeated to improve an already implemented system, which can not be used as a capable business foundation.

Software development has mostly transited from traditional waterfall approach, where all development phases were performed only once in the project, towards iterative practices where software is implemented in several incremental iterations, where each iteration contains all development phases aimed to implement one small part of system functionality (Berkun, 2005). There are several drawbacks in applying iterative approach to the new economy software development: (1) Although iterative practices are aimed to implement a working prototype of the software solution as soon as possible, that is often difficult due to the complexity of such systems. The usability of such prototype directly depends on the amount of introduced changes. (2) Software vendors in general cannot produce sufficient frequent software releases to provide the company with valuable insight into the way the software system is implemented. The difference in functionality between two subsequent incremental releases can become so significant that the company can not rely on the particular release to verify that the final solution will satisfy their expectations. (3) Iterative development projects are usually based on the particular technologies, meaning that the base technologies can not be changed during software implementation. The software vendor on average selects technologies according to its own preference, without considering actual company needs (Hoffman, 2003). Improper technology selection usually results with inefficient software systems and increased development cost.

c. Requirement definition

Common software development practices are based on the concept of software requirements (Wiegers, 2003), short and precise description of the system features. The relationship between actual company expectations and requirements aimed to describe these expectations is not straightforward; requirement authors must have sufficient knowledge about the actual business domain to properly represent the company's expectations with respective requirements. Poor requirements cause significant development problems, since they are used as a main description of the future software system. Improperly defined requirements, lack of usable information in the requirements, and misunderstanding of requirement purpose cause typical implementation problems and result with unusable software systems.

The software vendor must be able to properly prioritize requirements for incremental development, so that the most important requirements are implemented first (Fioravanti, 2005). The company must be able to verify basic concepts of the software system in the early project phase; therefore requirements must be correctly prioritized. Changes of the company expectations usually result with the need to re-prioritize requirements and change the way the software system is implemented.

d. Change handling

Most of today's development practices are based on the stable set of requirements which must be stable before implementation. Waterfall approach demands a stable set of requirements to be defined at the project start, while iterative approach demands a stable set of requirements for particular development iteration (Allen, 2001). Both approaches are not suitable for new economy software development, since the company demands to execute changes through the complete implementation. Today's development practices can successfully accommodate changes at the project (or iteration) start; changes that occur later are much more difficult to handle, since there are no development practices which are prepared for the impacts of the late changes.

Depending on the project size and scope, today's software development projects use two different requirement management practices: (1) formal approach, where requirements are properly defined and each change request must be formally approved and introduced, and (2) informal approach, without detailed requirements, where changes are usually handled by the agreement between actual software developers. Both approaches have significant drawbacks in terms of change handling: the formal approach demands significant effort to introduce change, while the informal approach usually results with misunderstanding between the company and the software vendor. The company demands change handling practice that will enable a quick response to the change request, but that will also provide a way to control the overall impact of the changes.

Table 1.

Method for selection of the software vendor

Activity	Responsible	Description
Initial team setup	Company's management	Company's team for software implementation should include representatives from all impacted business processes
Market analysis	Company's team	Define list of potential software vendors
Definition of the company's expectations	Company's team and management	Both company's team and management must together define high-level expectations from the future software system
Select target set of potential vendors	Company's team	List of potential software vendors should be narrowed to two or three most capable vendors by performing initial discussions
Define requirements for future software system	Company's team and software vendor representatives	Software vendor's representatives should together with the company's team define initial requirements for the future software system. By evaluating software vendor participation in requirement definition, company can get insight into vendor's understanding of its business needs.
Define concept of the software solution	Software vendor	Software vendor should define and present to company's team actual concepts of the future software system.
Evaluate presented concepts	Company's team and management	Initial concept of the software solution is the foundation for selection of the software vendor. Presented concepts should be evaluated in terms of support for company's business processes and actual usability.
Select vendor for implementation	Company's management	Make final decision according to the results of the requirement and concept evaluation.

e. Software verification

Today's development projects commonly underestimate the importance of software verification (Dustin, 2004). The complexity of design and implementation activities usually leave a small amount of time for software verification. The importance of the software verification is very high: software solutions must be implemented with required quality the first time they are introduced into the company, since the company must be able to use the software system as a business foundation as soon as possible. There are significant challenges in software verification of the new economy development projects: (1) constant changes demand different approach to test preparation and execution, (2) it is becoming impossible to verify the complete set of functionality in the available time frame and (3) current implementation practices demand software verification to be performed together with all development activities to identify software errors as soon as possible.

3. Methods for improvement of the software development projects

To avoid problems described in the previous section, several methods for implementation of the new software systems according to the company expectations are proposed. The described methods define activities that should be performed on the development project to introduce the company representatives into significant aspects of software implementation.

a. Selection of the software vendor

Proper selection of the software vendor is a precondition for successful software implementation. Activities described in Table 1 should enable the company to select the most suitable software vendor in terms of business understanding and implementation of usable software systems.

b. Software implementation

Activities presented in Table 2 are aimed to improve software development practices and to enable efficient introduction of company representatives into development activities. The proposed activities focus on creating plans for subsequent development iterations according to the feedback from the company representatives.

Table 2.

Method for the software implementation

Activity	Responsible	Description
Define requirement priorities	Company's team	Defined requirements should be prioritized in terms of business value, importance and stability, so that the most important requirements with relatively small probability of change will be implemented first.
Define development plan	Software vendor and company's team	High-level development plan that will include iteration schedule, definition of planned releases and development activities should be defined. Project plan should define about 10% of effort through the complete project to be performed by the company's team.
Plan next iteration	Software vendor and company's team	Development iteration must be time-limited to two weeks to create frequent releases. Company's team and software vendor should define set of most important requirements for next iteration. Plans for next iteration should, beside development activities, focus to the activities important for the company: (1) Test execution, (2) Usability improvements, (3) Preparations for software introduction into the company and (4) Creation of the user documentation.
Involve company's team into development activities	Software vendor	Company's team should participate in high-level design and test activities. By utilizing company's team knowledge, software vendor can improve software design and implementation. By including company's team into test activities, software vendor can get instant feedback about implemented functionality.
Perform increment verification	Company's team	At the end of development iteration, company's team must evaluate implemented software system increment by performing evaluations on two different areas: (1) Software system functionality, achieved performance, together with previously implemented functionality, and (2) usability from company's business perspective. Feedback from the increment verification must be used for planning of the subsequent iterations and adjusting development activities.
Adjust next iteration	Software vendor and company’s team	Preparation of the next iteration should include the following: (1) Re-prioritization of requirements (changes in priorities set by company representatives according to the current business environment), (2) Improvements defined according to the verification of the previous increment, (3) Changes proposed by the company's team, (4) Test improvements aimed to better evaluate further increments. Each subsequent iteration must be focused to implement features defined by company's team

c. Requirement definition and change handling

Activities presented in Table 3 are aimed to provide foundation for efficient requirement handling practices and to enable the software development team to quickly respond to changes proposed by the company representatives.

Table 3.

Method for the requirement definition and change handling

Activity	Responsible	Description
Describe high-level features	Company representatives	Company representatives should create simple presentation of the expected features on the high-level, so that development team gathers basic understanding of the system that should be implemented
Describe relationship between proposed features and business practices	Company representatives and software vendor	First step in creating detailed requirements is to describe actual business value of the expected software system features. By creating such descriptions, company is assured that the proposed features will be actually usable in everyday business processes. Development team should work together with company representatives to obtain information about the expected functionality, but also to propose improvements of the current company's business processes which should improve efficiency of the future software system
Model requirements, create in-depth description of features and system behaviour	Software vendor	Requirement descriptions are foundation for the actual requirement models that represent functionality of the future system in interaction with its users. Together with requirement models, development team must create detailed descriptions that will be used to actually implement software system.
Add detailed information	Company representatives	Company representatives are responsible for adding detailed information to the requirement models and in-depth descriptions.
Verify and explain requirements	Company representatives	To present actual expectations to the development team, company representatives should create presentations of the defined requirements to the complete development project. Requirement explanation should include description of the expected system functionality and overview of the global business environment.
Select requirement for implementation	Company representatives and software vendor	Both company representatives and software vendor must together select requirements that should be implemented in further iteration.
Create requirements change request	Company representatives or software vendor	Company representatives or software vendor can create request to change requirements. Company representatives can require changes in the system implementation during iteration execution, while software vendor can propose change of requirements according to the implementation experience. Change request should contain clear description of the expected change and priority for implementation.
Create impact analysis	Software vendor	Software vendor should create description of the changes in project execution introduced with requirement change request. Impact analysis should contain detailed description of change impacts to the software system and user documentation, schedule for the change implementation, and cost of the change implementation.
Decide about change implementation	Company representatives and software vendor	Both company representatives and software vendor should gain agreement on the change implementation according to the impact analysis. It is mandatory that both company and software vendor have the same opinion on the change implementation.
Implement change	Software vendor	If the change request gets approval from the company representatives and software vendor, it should be executed as soon as possible, but without compromising existing iteration plan.

d. Software verification

The proposed activities for software verification shown in Table 4 are aimed to enable the company representatives to gain an actual view on the software verification activities performed in the development activities.

Table 4.

Method for the software verification

Activity	Responsible	Description
Perform iteration test analysis	Software vendor and company representatives	Software vendor together with company representatives should analyze requirements planned to be implemented in current iteration to identify most important test areas and functionality that must be verified.
Define mandatory test areas	Company representatives	Company representatives should define mandatory test areas which must be successfully verified in order to accept iteration outcome.
Define set of iteration tests	Software vendor and company representatives	Set of tests that will be executed in particular iteration must be defined according to the test analysis by software vendor. Company representatives should evaluate proposed tests and add tests required to verify software system behaviour from usability and business efficiency perspective.
Create automated tests	Software vendor	According to the defined set of tests, software vendor must create automated tests which should be used to verify behaviour of implemented functionality on a daily basis.
Execute tests	Software vendor	According to the test plan, automated iteration tests should be executed on a daily or weekly basis, depending on the test scope.
Evaluate test results and suggest improvements	Company representatives	Company representatives should evaluate test results at the iteration end to accept iteration outcome and to make sure that the implemented parts of the system functionality comply with their expectations. According to the test results and current business perspective, company representatives should propose test improvements aimed to better evaluate implemented functionality.
Implement test improvements	Software vendor	Test improvements should be implemented in each subsequent iteration to enable efficient software testing.

4. Method Application on the actual software development project

The proposed method was applied on the e-commerce development project aimed to develop a specific solution for the mobile phone operator that will enable the mobile operator to provide specific mobile payment services to its customers. Within the project, a team of 21 customer representatives (mobile operator employees) and developers was founded. Since the development project was organized with the partner mobile operator, activities related to the software vendor selection were only partially performed. The mobile operator created a usable description of its expectations in terms of features that the planned software system should provide. The development team used these descriptions as a foundation for the initial requirement definition. The proposed method, where feature descriptions were used as a foundation for requirement definition showed its advantages in discovered differences between the company's expectations and software implementation planned by the development team. Significant differences were experienced in different understanding of functionality for customer interaction and mobile payment handling. From feature descriptions and discussions with mobile operator employees, the development team could not understand all relevant details about mobile payment legal aspects and the implications on the software system functionality. By evaluating requirements created by the development team, the company was able to identify these problems and provide valuable information to software developers.

The initial project time plan defined development in eight iterations, where each iteration lasted two weeks. Detailed plans were created only for one iteration in advance according to the prioritized requirements. The mobile operator assigned a team of three employees to constantly participate in the development project. In that team, one person was constantly assigned in first four iterations to analyze requirements, support developers in requirement clarification and verify that the requirements are followed during the iteration execution. The other two mobile operator's employees were assigned to test activities, focusing on software system usability and interaction with the customers. In the first two iterations, the software development team was focused on implementing the initial usable release of the software system. In that phase, the project faced a conflict between mobile operators and the developer's view of the requirement priority. The mobile operator wanted to implement requirements important for business functionality (service charging and usage reporting) in the initial project phase, while the development team wanted to implement functionality mandatory for system operation first. This conflict was resolved by implementing business functionality first, since it was decided that there is greater risk related to the misunderstanding of the business requirements than the risk related to the used technologies for system operation. It must be noted that the first iteration was too short for the development team to implement the complete basic business functionality. To follow the iteration plan, the project decided to create a basic release after the first iteration, which was used only to verify that the planned technologies can actually be used for system implementation. The testing of the business functionality started at the end of the second iteration and lasted until the end of system implementation.

The development project manager and the mobile operator's project leader scheduled meetings every two days to discuss the iteration progress and implementation issues. At these meetings, the mobile operator's project leader analyzed current implementation progress and together with the development project manager proposed adjustments of the iteration execution. Small adjustments were performed during the iteration, while adjustments of the requirement priority and iteration scope were only performed in the next iteration.

The development team assigned two persons to work on requirement description and clarification. They have in the first four iterations created requirement models and descriptions of the system behavior, which was presented to other team members on a daily basis. The presence of the mobile operator's employees on the development site enabled them to constantly monitor the requirement definition and adjust the software implementation according to the mobile operator's actual needs. The project faced lots of conflicts in requirement understanding between the mobile operator's employees and software developers. Some of these misunderstandings could not be resolved during the iteration execution; instead, unclear requirements were resolved with the later testing activities, when mobile operator's employees got the opportunity to actually use the implemented parts of the software system. Significant changes of the requirements, resulted from the testing performed by mobile operator's employees, were handled through the change requests, which were discussed on regular meetings by the development project manager and the mobile operator's project leader. Some of these change requests were approved without actually performing impact analysis due to the short iterations and mostly known impact.

Test activities were organized and lead by the development team, who prepared a complete test environment, test plans and defined a complete set of test cases. The mobile operator's employees selected valuable test cases which were executed during particular iterations. Due to the short time frame, the project was unable to automate the complete set of test cases; instead, only the code-level test cases were automated, while the functionality test cases were executed manually. This decision was made according to the results of test analysis, which showed that only a small part of actual test cases can be automated. The requirement set was also not stable enough in the first several iterations, so that no significant effort was invested in developing automated test cases. By performing the described testing, the project managed to deliver a usable system release in the planned schedule, but it was discovered that the development must be continued after the project's planned schedule, since the late problems occurred in several important features.

Conclusion

Companies today base their business growth on the advanced software systems. By using new and advanced software systems, companies can provide services to their customers better than the competition. The development of software systems which are used as business foundation includes significant challenges: used technologies demand significant knowledge and experience, companies demand advanced software systems to be implemented in very short time frames, and most of all, the business environment changes so significantly that the development project can not freeze requirements and implement a software solution according to the requirements defined at the project's start. This paper proposes the inclusion of the company representatives in all aspects of the software development project, from requirement definition, to design and test activities, so that the company can actually guide software development towards implementing a software system with the required functionality. The results achieved with this work show that the participation of the company representatives in the development project can improve software development, but the software development team must also adjust their development practices and change the approach to project organization and execution. By investing resources into participating in the development project, the company can significantly improve software implementation and reduce related risks, while the software development team can receive valuable support and improve development efficiency to create a software system that will provide means for the company to improve its business in the years to come.

REFERENCES

Allen, P., (2001), *Realizing e-Business with Components*, (New York: Addison Wesley)

Berkun, S., (2005), *The Art of Project Management*, (New York: O'Riley)

Coyle, D., (2002), *Getting the Measure of the New Economy*, (London: Isociety)

Dustin, E., (2004), *Effective Software Testing*, (New York: Addison Wesley)

Fioravanti, F., (2005), *Skills for Managing Rapidly Changing IT Projects*, (London: IRM Press)

Fingar, P., (2001), *The Death of "e" and the Birth of the Real New Economy: Business Models, Technologies and Strategies for the 21st Century*, (London: Meghan-Kiffer Press)

Harmon, P.,(2002), *Business Process Change: A Manager's Guide to Improving, Redesigning, and Automating Processes*, (San Francisco: Morgan Kaufmann)

Hoffman, L., (2003), *Beyond Software Architecture*, (New York: Addison Wesley)

Wiegers, K., (2003), *Software Requirements*, (Redmond, Microsoft Press)

POBOLJŠANI RAZVOJ SOFTWAREA KAO KLJUČ RASTA TVRTKE

SAŽETAK

Trenutna tržišna situacija zahtjeva od kompanija da koriste složene softverske sustave kao osnovu poslovanja i osnovnu vezu sa korisnicima i partnerima. Takvi sustavi moraju koristiti složene tehnologije i učinkovito podržavati stvarne poslovne procese. Uobičajene razvojne metode više nisu prikladne obzirom da se očekivanja korisnika ne mogu više ispuniti bez njihovog sudjelovanja. Razvojni projekti moraju konstantno učiti i podizati kompetencije ka ispunjenju zajedničkog cilja: implementaciji softvera prema stvarnim potrebama korisnika. Ovaj rad istražuje metode razvoja softvera za novu ekonomiju i predlaže metode kojima se stvarni korisnici uključuju u projekte razvoja softvera kako bi se omogućila učinkovita implementacija softvera koja će donijeti nove poslovne mogućnosti za kompaniju. Predložene metode primijenjene su na stvarnom projektu razvoja softvera kako bi se vrednovale sve prednosti predloženog pristupa.

Ključne riječi: *nova ekonomija, razvoj softvera*