

# **The Importance of Effective Requirements Management in Offshore Software Development Projects**

DCS860A: Emerging Information Technology

Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Professional Studies in Computing

at

Pace University

School of Computer Science and Information Systems

by

James F. Kile  
Donald Little  
Samir Shah

November 19, 2005

## Table of Contents

1.0	Abstract.....	1
2.0	Introduction .....	3
2.1	State of Software Development .....	3
2.2	Software Continues to Fail.....	4
3.0	Requirements Management .....	6
3.1	Types of Software Requirements.....	7
3.2	Requirements and Requirements Management Challenges.....	7
3.3	Offshore Development Challenges .....	8
3.3.1	Culture.....	9
3.3.2	Stakeholder Ambiguity .....	9
3.3.3	Communication Methodology .....	10
3.3.4	Level of Sophistication .....	10
3.3.5	Distance.....	10
4.0	Proposal .....	11
5.0	Conclusion.....	13
6.0	References .....	14

## 1.0 Abstract

The number of organizations outsourcing their software development processes to other countries are increasing everyday. There are thousands of software development teams worldwide who are engaged in developing all types of software applications for various industries. The goal of any organization is to develop high-quality software that satisfies customers' needs and deliver them on time and on budget. The design and development of software is a complex and difficult process and, unfortunately, quality continues to be a persistent problem.

Many studies suggest that software projects appear to meet business objectives, but fail to deliver the functionality customers really want [8]. One of the leading root causes for failure is the inability to manage requirements effectively throughout the software development process [14]. In fact, an average of 70% of the total requirements for a project is difficult to identify and 54% of the total requirements specified are unclear [11].

Software requirements represent the interests of users. They are the heart of any project [11]. However, as organizations deal with the pressure everyday challenges such as shorter development timeframes, resource constraints, customer demands, global competition, and the need to achieve increased profitability, they must also manage requirements that are oftentimes uncertain, inconsistent, ambiguous, informal, brief, and tend to change continuously.

In order to leverage the perceived benefits of offshore outsourcing (such as lower-cost), a large number of organizations are now taking advantage of technological and infrastructure improvements available to them in other countries. This is increasingly able to support moving processes such as application development, maintenance, testing, support, implementation and new product reengineering services to the offshore locations [7].

As this trend continues to grow, it adds another dimension to the already existing requirements management challenges. For example, if the initial requirements are location sensitive – they cannot be addressed outside a country due to perceived or real sensitivity – cannot be fully disclosed to the offshore software developers, the software development outcomes may not be successful.

The purpose of this proposal is to present a model of effective requirements management in the offshore software development projects. The proposal will define the requirements management process and discuss some of the challenges experienced in context of the offshore software development. The goal is to develop a framework that contributes to improving the methods, techniques, and processes of requirements management in an offshore environment that maximizes the chances for success. This framework will focus on any methods and techniques needs to be expanded to encompass the ever-changing role of the offshore software development environment.

## **2.0 Introduction**

This paper introduces the concept of requirements management in an offshore environment. This introduction focuses on the state of current software development and some statistics about software projects and a discussion about why projects seem to fail.

### **2.1 State of Software Development**

Many organizations are engaged in Information Technology projects to meet heightened customer demands and expectations; improve business results; respond to changing security requirements; and support a growing emphasis on good governance. Information Technology is often a company's response when faced with a need to alter or improve their business processes. The efforts to meet the increased demand for IT services are hampered by budget limitations and lack of resources.

Forrester research predicts that worldwide IT spending will continue to grow, but at a rate slower than the growth in demand. For this year (2005), spending is expected to grow by 5.7% and reach \$795 billion [13]. On average, best-of-breed organizations spend 4-5% of revenue on information technology. IT operating expense currently represents one of the largest corporate expenses [2]. While some of this spending goes to hardware and software upgrades, software license fees, and other related services, the majority is spent for new software development projects. New software projects are constantly being proposed in organizations to automate and streamline their business

processes in response to stakeholder's demands. The U. S. government planned to spend \$16 billion on military software projects alone in 2004 [2].

To put the above spending in context, software is also becoming increasingly more complex. A typical cell phone, which contains 2 million lines of software code now, is estimated to contain 20 million lines of code by 2010 [2]. The General Motors Corporation estimates that its cars will run software having almost 100 million lines of software code by 2015 [2].

## **2.2 Software Continues to Fail**

Despite improvement efforts by organizations, challenges and complexities continue to cause software projects to fail. Studies suggest that more than 60% of U. S. software projects fail [9] in some manner either through outright failure to deliver or abandonment. Unsuccessful or abandoned software development projects cost the world economy billions of dollars each year. It is estimated that these failures have likely cost the U. S. economy at least \$25 billion and perhaps as much as \$75 billion.

This track record of failures is not unique to the United States. The Sydney Water Company, the largest water provider in Australia, launched a software application project to automate its customer information and billing system in 2002. The project was aborted midway after spending \$33.3 million (U. S.) due to inadequate planning and specifications [2]. Further, software development failures appear to not be restricted to any industry type, nation, or organizational culture and size [3].

Some of the reasons why software projects fail are: resource constraints; priority conflicts; inability to work in dynamic development environment; poor requirements management; lack of a body of knowledge in software project management; and poor communication between project teams. These failures can be classified into two broad categories, the first is the inability of implemented software to perform to the expectations of the user resulting from a failure of requirements management and the second is the inability of software developers to produce working or functioning software [3].

### 3.0 Requirements Management

A requirement is a capability that the resulting software must deliver [14]. They are represented as a formal listing of the users needs. Studies suggest that one of the top five reasons why software projects continue to fail is poor requirements and scope creep [9]. Scope creep is the insidious increase of requirements without formal acknowledgment. Scope creep is responsible for 80% of project schedule overruns [9].

Requirements Management is a process that captures, traces, and manages stakeholder needs and changes that occur throughout a project's lifecycle [8]. Requirements Management is one of the major areas and activities under Requirements Engineering. The purpose of Requirements Management is to manage the scope of the project, manage specification changes, reduce project risk and ensure that the project meets the real business objectives. According to the research, 40% of the requirements generate rework during the project life cycle [10].

Many software tools, methods, techniques and standardized processes exist that can help manage requirements more effectively. Some of the available tools and methods are – IBM's RequisitePro, UML diagrams and Telelogic DOORS. The standardized processes such as Software Engineering Institute's Capability Maturity Model (CMM), British Standards Institute's Guide to Specifying User Requirements, Canadian Standards Association's Basic Guidelines for the Structure of Documentation, Department of



Defense's documentation standard MIL-std 498, and the ISO 9000 quality management standards are also available.

### **3.1 Types of Software Requirements**

Specifications for software include both functional and non-functional requirements. Functional requirements are those that directly relate to the features and functions that will be produced as part of the final product. Examples include screen layouts, specific user action definitions, and external system interface specifications.

Non-functional requirements are those requirements that do not necessarily address a specific feature of a product, but addresses an overall desire. Examples of non-functional requirements include usability, implementation of a specific vendor package, performance specifications, and overall reliability.

### **3.2 Requirements and Requirements Management Challenges**

One traditional method used to facilitate the discovery of requirements is to define them through a system's anticipated inputs, outputs, functions, and attributes [14].

Regardless of the method chosen, many medium to large software products have thousands of requirements. Managing them effectively can be a major challenge. Since the goal of requirements specifications is not only to achieve agreement, but to form the basis for software product design, understanding between the user and the development organization is critical.

Requirements Management continues to be a major challenge in all sizes of software development projects. For smaller projects, requirements may be informal and brief, thus

making them more difficult to manage effectively. In larger projects – especially those with aggressive development schedules, the requirements tend to change in a continuous stream, and again, managing them effectively continues to be a major issue. If requirements are not managed effectively, the problems of organizing, prioritizing, controlling, managing, and providing resources can become major issues during development.

### **3.3 Offshore Development Challenges**

Many organizations are moving toward offshore software development in order to realize lower cost solutions. Outsourcing is the delegation of administrative, engineering, research, development, or technical support processes to a third party vendor in a lower cost location [7]. For this proposal, the focus is on offshore software development outsourcing. This includes software development processes such as application development, maintenance, testing, support, implementation and new product reengineering services [7].

Although outsourcing and a global work environment are not new phenomenon, distributed software development work in offshore countries is a growing relatively new area. It is anticipated that U. S. offshore spending will quadruple to \$46 billion by 2007 [4]. The estimated export of software and technology services from India alone was about \$ 12 billion during 2004 [12]. NASSCOM estimates that more than 185 of Fortune 500 firms engage in offshore outsourcing to India. It was also found that 44% of U. S. firms with over \$1 billion in revenue had offshore IT activities in 2001 [1]. This grew to 67% by 2003 [1].

Although the software tools, methods, techniques and processes are improving, the requirements management challenges in the offshore software development environment creates situations in which these tools may not be adequate. Therefore, methods and techniques for requirements management may need to be expanded or modified to meet the challenges in the offshore developments environment.

Though there is a high level of investment (and a presumably returns on those investments), there are unique challenges to offshore software development including culture, stakeholder ambiguity, communications, level of sophistication, and distance.

### *3.3.1 Culture*

Communicating with offshore providers can be hampered by cultural differences. For example, a common observation is that Americans and Indians tend to communicate differently. Offshore experts believe that, on the whole, Americans tend to be informal and expansive, whereas Indians tend to focus on topics and discuss them point by point. "[Indians] will give you exactly what you ask for, so it's a literal kind of thing," ICICI Infotech's Kunkalienkar says. Although that can often be advantageous for the outsourcing client, it can sometimes lead to misunderstandings.

### *3.3.2 Stakeholder Ambiguity*

With time zone differences and physical distances, it may be difficult for the developers to determine who the key stakeholders are as they do not have any local or informal knowledge. Additionally, the vendor may not have appropriate access to these stakeholders. Hence the requirements gathered may be tailored to those users who were available at the time.

### *3.3.3 Communication Methodology*

The success of requirements analysis depends on how well the users and analysts communicate [5]. Email works well for sending documents. However, the explanation of requirements usually involves further conversation, responses, and a continuous validation to ensure universal understanding. The lack of a good communication methodology can hamper the understanding of the impact of a specification or change, and the effective management of it. It may be difficult to convey the need for quickly incorporating changes in business requirements into a software application in the offshore environment when the communication methodology is inadequate.

### *3.3.4 Level of Sophistication*

Software companies in developing countries may not have the appropriate requirements management tools and business practices often seen as a necessity for developing good requirements. Without this maturity, the ability to appropriately capture requirements is hampered and their translation into equivalent specifications that match the user's understanding may not be possible with any degree of accuracy.

### *3.3.5 Distance*

According to [12], the highest degree of interaction occurs between the user and the analyst during the requirement analysis stage of a project. The difference in time and distance of offshore development teams add complexity to any project. It may inhibit a clear understanding of the project's goals and objectives and can lead to requirements management conflicts [12].

## 4.0 Proposal

The extensive use of offshore development is new enough that there is no comprehensive model for successful implementation. As described in [12], we need to have effective requirements methods and tools in the global environment but more research is needed to make this happen [12]. Without a model, it is difficult to decide whether and which particular issues should be addressed as part of offshore software development requirement management processes. We are proposing development of such a model. This framework is focused on what drives good requirements management practices in the offshore software development environment to maximize the chances of successful software development.

The first step is to determine that state of the practice. This will be accomplished by a comprehensive review of the current requirements management literature and the offshore sourcing literature. Based upon this review, a survey metric will be prepared and distributed to select, already identified, individuals for comment and production of test data. This data will be processed using appropriate data mining tools to produce an initial hypothesis. The survey metric will be modified to reflect the data needed to test the hypothesis. The metric will be tested on a small subset of the population. This second test survey will be analyzed with data mining tools and appropriate statistics.

This carefully tested metric will be distributed to 5-10 software companies in India and 5-10 clients in the U. S. The effort will involve considerable follow up to obtain the

required data and to help the companies understand what is required. After the data is gathered, the results will be processed with the data mining and statistical tools. Based upon the results we will be able to accept or reject the hypotheses.

The results will be made available to the participating companies for their review and comments. Their comments will be incorporated in the offshore framework. This framework will help organizations to better understand how to effectively manage requirements in the global environment to minimize the chances of failure and improve overall software development performance in the delivery of new applications.

## **5.0 Conclusion**

As described in [6], software engineering is witnessing a transition from a traditional co-located form of development to a form in which global software teams collaborate across national borders. Requirements Management is gaining increased interest and appreciation as a difficult and extremely important part of software development. Stable and unchanging requirements, though highly desirable, are unlikely in a typical software development project. Moreover, most people believe that software development outsourcing will continue. A strategy and/or framework is needed that will allow companies to successfully manage offshore requirements definition.

## 6.0 References

- [1] E. Carmel and R. Agarwal, "The Maturation of Offshore Sourcing of Information Technology Work", [http://heim.ifi.uio.no/~vegari/hovedfag/Offshore\\_Sourcing\\_MISQE\\_Submission\\_Jan\\_02.pdf](http://heim.ifi.uio.no/~vegari/hovedfag/Offshore_Sourcing_MISQE_Submission_Jan_02.pdf), 2002, Accessed November 15, 2005.
- [2] R. N. Cherlette, "Why Software Fails", <http://www.spectrum.ieee.org/sep05/1685>, Accessed September 2005.
- [3] K. Ewusi-Mensah, *Software Development Failures*. The MIT Press, 2003.
- [4] S. Frank, "Source Out Risk In", <http://www.spectrum.ieee.org/careers/careerstemplate.jsp?ArticleId=i040405>.
- [5] K. Holtzblatt and H. R. Beyer, "Requirements Gathering: The Human Factor", *Communications of ACM*, vol. 38, no. 5, pp. 30-32, May 1995.
- [6] H. Jo and C. Brian, "Requirements Engineering During Global Software Development: Some Impediments to the Requirements Engineering Process – A Case Study".
- [7] R. Kalakota, *Agile Enterprises and Offshore Outsourcing*. The Agile Enterprise, Pal, N. and Pantaleo, D., eds.: Springer Publishing, 2005.
- [8] J. MacLeod, "Requirements Management Tools: Friend or Foe?" *International Developer*, pp. 52-53.
- [9] D. Martin, "An Innovation Approach To Managing Software Requirements", <http://www.mks.com>.
- [10] R. Priklandnicki, J. L. N. Audy and R. Evaristo, "An Empirical Study on Global Software Development: Offshore Insourcing of IT Projects".
- [11] R. Priklandnicki and G. d. Sul, "Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM Context ", *Proceedings of the Thirty-third Hawaii International Conference on Systems Sciences*, 2003.
- [12] E. S. Sakthivel, "Virtual Workgroups in offshore Systems Development", *Information and Software Technology*, vol. 47, pp. 305-318.



- [13] K. Schwalbe, *IT Project Management*. 4th edition ed.: Thompson Publishing, 2005.
- [14] E. Yourdon, *Managing Software Requirements*.