An agile requirements elicitation approach based on NFRs and business process models for micro-businesses

Article · June 2011

DOI: 10.1145/2181101.2181114

Rj Macasaet
8 PUBLICATIONS **24** CITATIONS
SEE PROFILE

L. Chung
University of Texas at Dallas
149 PUBLICATIONS **4,806** CITATIONS
SEE PROFILE

José Luis Garrido
University of Granada
134 PUBLICATIONS **814** CITATIONS
SEE PROFILE

Manuel Noguera
University of Granada
111 PUBLICATIONS **688** CITATIONS
SEE PROFILE

An Agile Requirements Elicitation Approach based on NFRs and Business Process Models for Micro-Businesses

RJ Macasaet[1,2]
[1]Pentathlon Systems Resources Incorporated, Department of Software Research and Development
Cityland Pioneer, Pioneer Street, Mandaluyong City, 1550 Metro Manila, Philippines
[2]University of Granada, Departamento de Lenguajes y Sistemas Informáticos
Calle Periodista Daniel Saucedo Aranda S/N, 18071 Granada, España
rjmacasaet@pentathlonsystems.com, rjmacasaet@ugr.es

Lawrence Chung
University of Texas at Dallas,
Department of Computer Science,
Richardson, Texas 75083, U.S.A.
chung@utdallas.edu

Manuel Noguera
University of Granada, Dpt. de Lenguajes y
Sistemas Informáticos, C/ Pdta. Saucedo Aranda
S/N, 18071 Granada, España
mnoguera@ugr.es

José Luis Garrido
University of Granada, Dpt. de Lenguajes y
Sistemas Informáticos, C/ Pdta. Saucedo Aranda
S/N, 18071 Granada, España
jgarrido@ugr.es

María Luisa Rodríguez
University of Granada, Dpt. de Lenguajes y
Sistemas Informáticos, C/ Pdta. Saucedo Aranda
S/N, 18071 Granada, España
mlra@ugr.es

ABSTRACT

Software empowers micro-businesses to be more competitive - be it accounting software, point-of-sales software, e-commerce software, and the like. Before having such software, micro-businesses must express their requirements properly to the software developers first. Since micro-businesses have restrictions with their budget, manpower, and technical exposure to software, some trade-offs must be addressed. In such case, a practical requirements approach must be used. This approach must be inexpensive, non-technical, and must not involve too much manpower. This paper will show how a software development company, Pentathlon Systems Resources Incorporated, has applied an agile requirements elicitation approach to a micro-business, an amusement arcade. This paper will demonstrate how several models and techniques such as goals, business process models, patterns, and non-functional requirements, have helped in defining the software requirements of the micro-business.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements Specifications - Methodologies

General Terms

Documentation

Keywords

Goals, Business Process Modeling, Non-Functional Requirements, Patterns, Requirements Engineering

1. INTRODUCTION

Micro-businesses will find it essential to have some sort of software – be it accounting software, point-of-sales software, e-commerce software, and the like. Software simply makes them more competitive. Hence, they must take advantage of these software solutions.

For micro-businesses - the smallest of businesses that operate – the software that they would require need not be complex as compared to those of mature, multi-national corporations. There could be hundreds of users or just one user but the kind of software that is used for their business could be made with just a few software modules or lines of code. Moreover, these micro-businesses usually have very small software budgets in comparison to large firms (See Figure 1). Hence, they cannot afford to employ a large team of software programmers, requirements engineers, and software architects to design and implement the perfect software for their business.

Most of the requirements modeling techniques available nowadays are made for larger, more mature firms. "Many people feel that the approach that is used on medium and large projects is an inappropriate guide for small projects – that the practices, policies, mechanisms, methods, techniques, and tools can't be applied" [1]. For instance, KAOS [2], one of the most popular requirements modeling approaches nowadays, has been used in large industrial projects like hospital emergency service management systems, large-scale drug delivery management systems, and large-scale information systems for daily newspaper firms [3]. Albeit useful for large projects, “a KAOS-like requirements study is worth the effort as soon as the project man power is more than 100 man days” [2]. This means that for a four-man software development team, this project would last for more than five weeks. A significant amount of micro-businesses do not require this much man days to implement software for their business.

Be it a large industrial software project or a software project for a micro-business, “a requirements process is required for a project of any size" [1]. Given the budget and manpower constraints of the micro-business owner, a certain kind of approach must be used in order to describe the requirements of the micro-business to the software developer with the least resources expended possible. With constraints on the technical exposure of micro-business owners, the approach must not use too much technical jargon as well. Stakeholder requirements are best described in a non-technical and realistic manner [4]. The micro-business owners are those who know their businesses best and if they are empowered in expressing their requirements in their own language then the requirements document should turn out to be better. “The requirements process need not be complicated or expensive” [1].

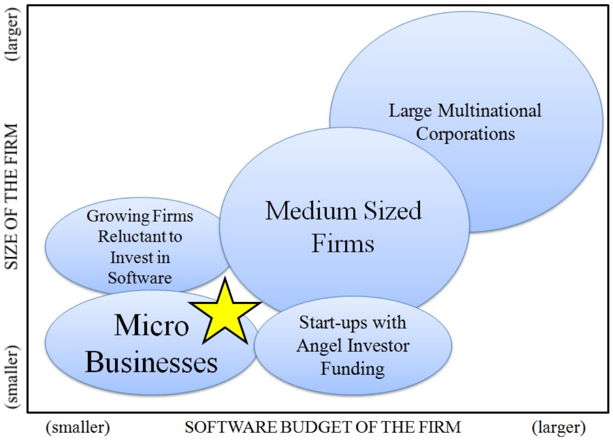
(larger)
SIZE OF THE FIRM
(smaller)
Large Multinational Corporations
Medium Sized Firms
Growing Firms Reluctant to Invest in Software
Micro Businesses
Start-ups with Angel Investor Funding
(smaller)
SOFTWARE BUDGET OF THE FIRM
(larger)

Figure 1. Firm size and their respective software budgets

This paper will show how an agile requirements elicitation approach based on non-functional requirements “NFRs” and business process models has been used in a micro-business. It is an efficient, practical, and intuitive way of acquiring software requirements from micro-businesses. Pentathlon Systems Resources Incorporated “PSRI” has continuously been using and developing this approach

2. MOTIVATION

The agile requirements elicitation approach aims to address the need of micro-businesses in defining their software requirements. Given their budget, manpower, and technical constraints, the approach uses practical techniques and models (See Figure 2) that do not demand a huge amount of resources both from the micro-business and the software developer.

Initially, the approach focuses on the problem domain. It is the domain in which a system is going to be used. It is important to focus on the problem domain so that the micro-business owner does not stray and attempt to come up with premature software solutions on his own. If the business owner defines his problem properly, without any premature implementation or solution biases, the software developer will have more liberty in coming up with software solutions later on. With an open solution space for systems engineers and architects, their ingenuity and creativity are fully harnessed, resulting in better project outcomes [4].

During the first part of the approach, the micro-business owner enumerates his goals [3]. These are what the micro-business wants to achieve in a business sense. Next, the approach uses the business process modeling notation “BPMN” [5]. BPMN aims “to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes” [5]. BPMN becomes the common illustration medium between the software developer and the micro-business owner. To speed up the definition of the micro-business processes even more, patterns are used. They “provide a common form for the transport of experiences” [6]. Finally, the concepts of functional and non-functional requirements [1] [7] are used to help both the software developer and micro-business owner agree on a requirements document.

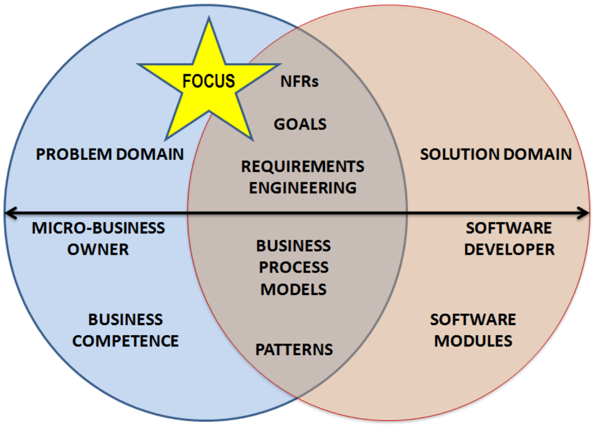

Figure 2. Overview of the Agile Requirements Elicitation Approach

3. THE APPROACH

This section will illustrate, step by step, how the agile requirements elicitation approach has been applied in a micro-business. The approach can be seen in Figure 3.

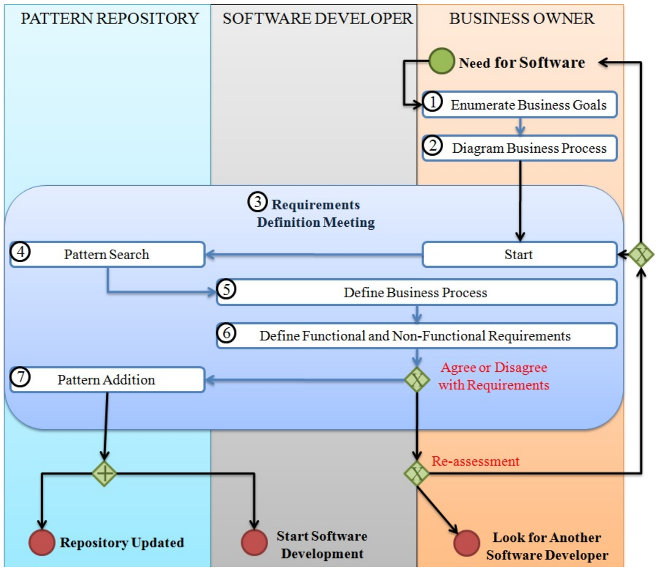
PATTERN REPOSITORY
SOFTWARE DEVELOPER
BUSINESS OWNER
Need for Software
1 Enumerate Business Goals
2 Diagram Business Process
3 Requirements Definition Meeting
4 Pattern Search
Start
5 Define Business Process
6 Define Functional and Non-Functional Requirements
7 Pattern Addition
Agree or Disagree with Requirements
Re-assessment
Repository Updated
Start Software Development
Look for Another Software Developer

Figure 3. The Agile Requirements Elicitation Approach

3.1 Preparation

The starting point of the agile requirements elicitation approach is the need for software. In this case, the micro-business sampled is an amusement arcade. It sells tokens to its customers for use in its amusements. The arcade is having difficulty managing the queues of customers and the micro-business owner believes that a software system would be able to aid in managing these queues.

The micro-business owner begins the approach by enumerating his business goals. (Refer to Step 1 in Figure 3). Goals capture the various objectives the system under consideration should achieve. They may be formulated at different levels of abstraction, ranging from high-level, strategic concerns (such as "serve more passengers" for a train transportation system or "provide ubiquitous cash service" for an ATM network system) to low-level, technical concerns (such as "acceleration command delivered on time" for a train transportation system or "card kept after 3 wrong password entries" for an ATM system) [3]. In the case of micro-business application systems, goals may be used to relate the software-to-be to organizational and business contexts [8]. A transcription of business goals made by the micro-business owner is as follows.

List of Business Goals (Step 1)

1. Increase revenue of the arcade through token sales

2. Improve customer satisfaction

3. Reduce the build-up of queues

4. Speed up transactions at the sales counter

5. Train sales personnel

6. Allot monthly training sessions for sales personnel

7. Purchase new computer hardware

8. Keep track of sales – daily, weekly, monthly, etc.

9. Manage the availability of tokens in the inventory

10. Manage the forms of payment at the sales counter – cash or credit

The next step of the approach is the diagramming of the business process by the micro-business owner (Refer to Step 2 in Figure 3). Every micro-business could have their own unique way of diagramming their processes. There are an infinite number of ways. Despite this ambiguity, these diagrams will aid in business process pattern recognition and business process definition in the succeeding stages of the approach. Therefore, these diagrams are essential to the approach. If a micro-business does not already have a business diagram of their own, they ought to express their business process in a diagram using their own means of illustration. An internal company diagram of the micro-business process is shown in Figure 4.

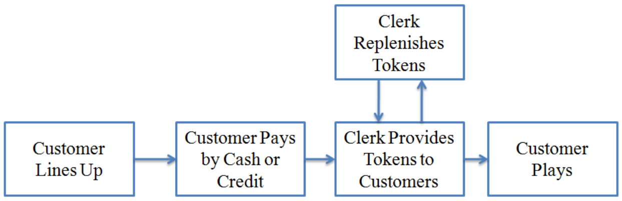
Clerk Replenishes Tokens
Customer Lines Up
Customer Pays by Cash or Credit
Clerk Provides Tokens to Customers
Customer Plays

Figure 4. Business Process Diagram of the Amusement Arcade Token Purchase (Step 2)

3.2 Requirements Definition Meeting

Once the micro-business owners have enumerated their business goals and diagrammed their business processes, they may proceed with a requirements definition meeting (Refer to Step 3 in Figure 3) with the software developer.

The next step of the approach is to compare the business process diagrams of the micro-business and find similar existing business process patterns in the pattern repository (Refer to Step 4 in Figure 3) or find existing business process patterns that can be of use. An example of a similar business process pattern is shown in figure 5 and an example of a useful business process pattern is shown in Figure 6.

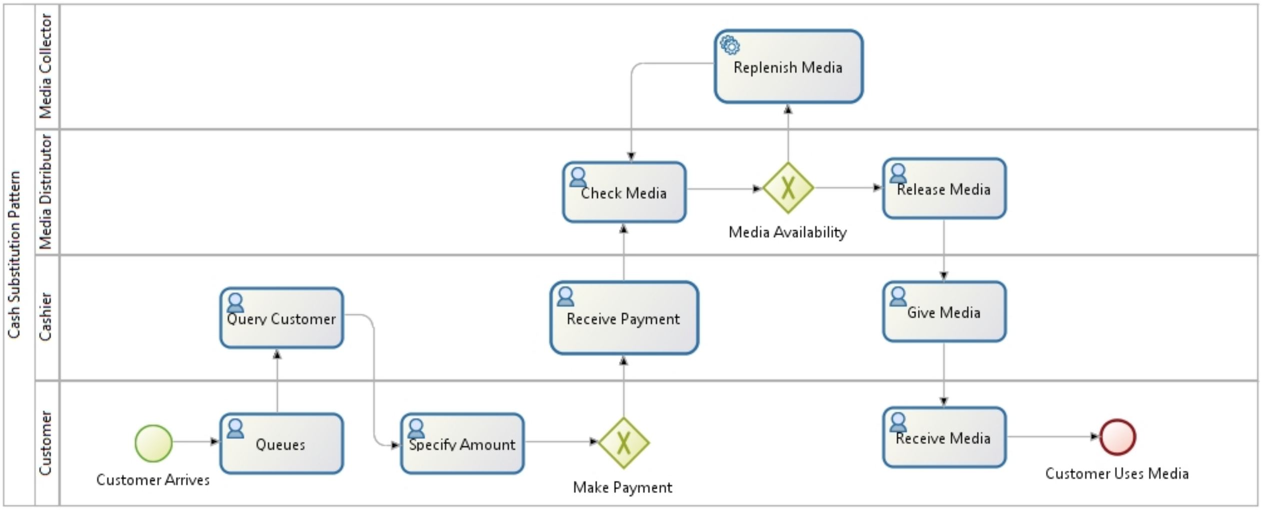

Figure 5. Cash Substitution Pattern – A Business Process Pattern in the Pattern Repository that is similar to the Micro-Business Process (Step 4)

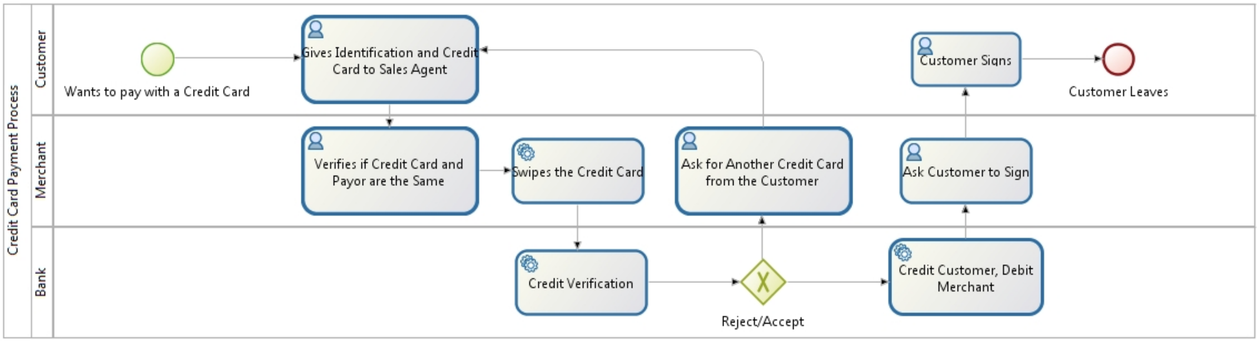
Credit Card Payment Process
Customer
Merchant
Bank
Wants to pay with a Credit Card
Gives Identification and Credit Card to Sales Agent
Customer Signs
Customer Leaves
Verifies if Credit Card and Payor are the Same
Swipes the Credit Card
Ask for Another Credit Card from the Customer
Ask Customer to Sign
Credit Verification
Reject/Accept
Credit Customer, Debit Merchant

Figure 6. A Credit Card Payment Process – A Useful Business Process Pattern in the Pattern Repository (Step 4)

After finding similar business process patterns and useful patterns in the pattern repository, the software developer and the micro-business owner proceed to defining the business process (Refer to Step 5 in Figure 3).

Using BPMN, the software developer and the micro-business owner collaborate on defining the business process. Tools such as the Bonita Studio [9] or the eBPMN Designer [10] can be used. In this case, the software developer and the micro-business owner used the Bonita Studio to illustrate the business process. A unison of Figure 5 and a "+" symbol below the "Credit Payment" Activity representing Figure 6 can be seen in the business process definition in Figure 7.

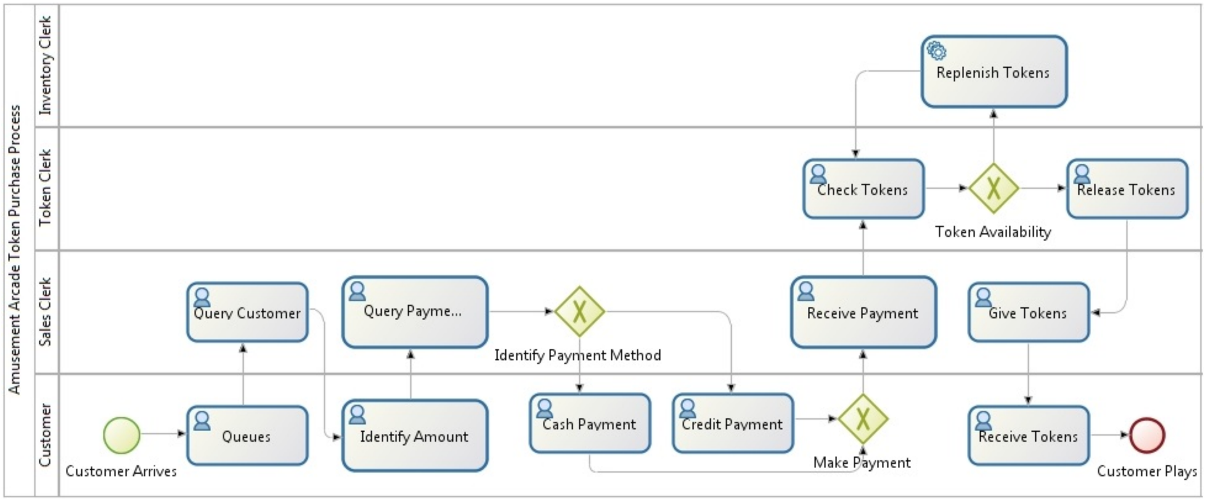

Figure 7. Business Process Definition of the Amusement Arcade Token Purchase Process as Agreed Upon by the Micro-business Owner and Software Developer (Step 5)

The next step of the approach is to define the functional and non-functional requirements of the micro-business. It is important to keep in mind that there is no clear distinction between functional and non-functional requirements. The distinction depends on the level of detail of the requirements document or the degree of trust between the micro-business owner and the software developer. Moreover, stakeholders may have vague goals that cannot be expressed precisely. These vague and imprecise requirements included in a requirements document can cause problems both for the micro-business owner and the software developer [11].

“Experience shows that user non-acceptance originates more often from inadequate non-functional requirements more than from problems with functional requirements” [12]. Therefore, in order to significantly increase the chances of user-acceptance, it is imperative to define both functional and non-functional requirements, with an emphasis on the latter. In order to address the problems of ambiguity and vagueness of non-functional requirements, non-functional requirements must satisfy two attributes. First, they must be objective. This means that the non-functional requirement does not express a wish, a goal, or a personal opinion. Second, they must be testable. There must be some process by which the requirement can be tested [13] [14]. If a software project’s functional and non-functional requirements are defined clearly, the software developer merely has to “satisfice” them, i.e., satisfying at some level a variety of needs, without necessarily optimizing results [7]. With the open solution space [4] of the software developer, chances of “satisficing” these requirements is higher, which eventually would result in higher user acceptance rates upon project completion.

Table 1 shows a list of functional and non-functional requirements that the software developer and the micro-business owner have agreed to.

Table 1. A List of Functional and Non-Functional Requirements that the Software Developer and the Micro-busniess Owner have agreed on (Step 6)

Functional Requirements ***"what the system can do"***	**Non-Functional Requirements** ***"ability of the system to do it"***
Record both cash and credit sales at the counter	cash transactions should take less than 10 seconds credit transactions should take less than 30 seconds
Notify clerks when token inventory is low	clerk is notified immediately through PA system when token inventory reaches 15%
Display sales data	information must be shown on the screen in less than three seconds
Has its own Graphical User Interface	"GUI" must have the company's colors and must display company logo
Has a back-up database	daily data can be backed-up on the server in less than one hour
Must be cross-platform	system can be installed on Windows and Linux OS

All the non-functional requirements are both objective and testable. With the collaborated documents on hand, the software developer and the micro-business owner can either agree to proceed with software development or not. This agreed requirements document provides the basis for planning the development of a system and accepting it on completion. The ability to arrive at a requirements document where all stakeholders of the software system are in agreement is a critical point. Without reaching this point, a project would never be able to commence [4]. If the parties agree to develop the software, the business process diagrams may be added to the repository (Refer to Step 7 in Figure 3) for future reference. If the parties do not agree to develop the software, both parties may re-evaluate.

4. EVALUATION

In this example, the software developer and the micro-business owner have agreed to proceed with the development of the software. As a quick and simple evaluation of the approach, five software developers from PSRI were asked to classify the resulting requirements documents. All five software developers have at least three years of experience in developing software systems for micro-businesses. Their opinion about the resulting requirements documents is solely based on how they will be of use when actual software development takes place. From their foresight, the resulting requirements documents have been classified under these three categories:

1) SUFFICIENT – The documents are sufficient enough to produce the software required
2) INTUITIVE – There is lacking information in the documents but some degree of intuition and creativity from the developers could produce the software required
3) INSUFFICIENT – The documents are insufficient and some other requirements gathering method should have been used. It is highly unlikely that the developers can produce the software required

Three out of the five software developers said that the requirements documents produced were sufficient enough to produce the software required. Two out of the five software developers said that the requirements documents produced were intuitive enough to produce the software required.

5. CONCLUSION

The agile requirements elicitation approach has demonstrated several benefits. First, it has allowed better self-expression on the part of the micro-business owner. The tools and techniques used – goals, business process models, and NFRs - are highly intuitive for the micro-business owner. Technical jargon has not impeded the micro-business owner in expressing his software requirements. Second, the focus of the approach on the problem domain – the definition of the micro-business owner's problem or software requirements – permits the software developer to use his creativity and ingenuity in software implementation later on (in the solution domain [4]). Third, requirements documents have been produced quickly without serious compromises. The use of the pattern repository has made the production of requirements documents faster. Later on, during software implementation, pre-existing modules that correspond to previous patterns can be used for faster software implementation as well.

The agile requirements elicitation approach is an attempt to advance and specialize the field of requirements engineering [15] by focusing on the needs of micro-businesses. Moving forward, the authors of the approach and software developers from PSRI are continuing to develop the pattern repository. As more and more patterns are added to the repository, a more efficient search-and-recognize technique for the patterns must be put into place. Moreover, enhanced visual representations of the relationship between functional and non-functional matters are under development as well.

6. ACKNOWLEDGEMENTS

This research has been funded by the Ministry of Science and Innovation of the Spanish Government through Projects TIN 2008-05995/TSI, TIN 2007-60199, and by Pentathlon Systems Resources Incorporated.

7. REFERENCES

[1] Young, R. 2004. Requirements Engineering Handbook. Artech House, Incorporated, Norwood, Massachusetts.

[2] Respect-IT. 2007. KAOS Tutorial Version 1.0. URL = http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf (Last accessed on March 10, 2011)

[3] Lamsweerde, A. 2001. Goal Oriented Requirements Engineering: A Guided Tour. *Invited mini-tutorial paper which appeared in Requirements Engineering, International Symposium on Toronto, August 2001, 249-263. Proceedings RE'01 5th IEEE*. Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.

[4] Dick, J., Jackson, K., and Hull, E. 2002. Requirements Engineering. Springer Verlag London Limited, Great Britain.

[5] Object Management Group, Inc. 2008. Business Process Modeling Notation Version 1.1. URL = http://www.omg.org/spec/BPMN/1.1/PDF (Last accessed on March 10, 2011)

[6] Gröne, B. 2006. Conceptual Patterns. *Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems*. (Potsdam, Germany, March 27-30, 2006).

[7] Chung, L., Mylopoulos, J., Nixon, B., Yu, E. 2000. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Boston, Dordrecht, London.

[8] Yu, E.S.K. 1993. Modeling Organizations for Information Systems Requirements Engineering. Proceedings of the *1st IEEE International Symposium on Requirements Engineering*.

[9] Bonitasoft S.A.. 2010. Bonita Studio. URL = http://www.bonitasoft.com/products/BPM_download.php (Last accessed on March 10, 2011)

[10] Soyatec Open Solution Company. 2010. eBPMN Designer. URL = http://www.soyatec.com/ebpmn/download.html (Last accessed on March 10, 2011)

[11] Kotonya, G., Sommerville, I. 2003. Requirements Engineering: Processes and Techniques. John Wiley and Sons Limited, England.

[12] Doerr, J., Marhold, C., Rohleder, C., and Salinesi, C. 2009. Clarifying Non-Functional Requirements to Improve User Acceptance – Experience at Siemens. *Proceedings of the 15th REFSQ International Working Conference on Requirements Engineering: Foundation for Software Quality* (Amsterdam, The Netherlands, June 8-9, 2009).

[13] Sommerville, I. 1996. Software Engineering. Addison-Wesley, England.

[14] Deutsch, M.S., Willis, R.R. 1998. Software Quality Engineering. Prentice-Hall, New Jersey.

[15] Nuseibeh, B., and Easterbrook, S. 2000. Requirements Engineering: A Roadmap. *Proceeding of the ICSE '00 Conference on Software Engineering.* (Limerick, Ireland, June 4-11, 2000)