



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Comprehensive Model Driven Approach for Requirement Management System

Pooja Batra Nagpal

Assistant Professor, Dept. of CSE, Amity University, Gurgaon, Haryana, India

ABSTRACT: Many researchers proved that requirement management is essential part of any software system. Ignorance in implementation of requirement management system may conclude failure of process. In previous models of requirement management system negligence of requirements occurred resulting faults and delay in software delivery. In my paper a comprehensive model driven approach is discussed which focus on requirement elicitation by introducing customer involvement concept. Through which mapping of requirements is done in traceability matrix. Consequently give assurance of success of software in every manner.

KEYWORDS: Requirement Engineering, requirement management, traceability, requirement elicitation

I. INTRODUCTION

Requirements management is a process of eliciting, documenting, organizing, and controlling changes to the requirements. Generally, the process of requirements management depends upon the availability of software requirement specification document, but planning for managing the changing requirements should start during the requirements elicitation process. Many requirement elicitation techniques are used for this purpose. For example interviews, questionnaires, user observation, workshops, brainstorming, use cases, role playing and prototyping etc. Requirement management involves activities such as to recognizing the need for change in the requirements, encourage the relationship amongst stakeholders and involving them in the requirements engineering process and identifying and tracking requirements attribute.

Requirements management enables the development team to identify, control, and track requirements and changes that occur as the software development process progresses. There are some other advantages i.e. better control of complex projects, improved software quality, reduced project costs and delay, improved team communication, reduced project costs and delays associated with the requirements.

Requirements management starts with planning, which establishes the level of requirements management needed. After planning, each requirement is assigned a unique 'identifier' so that it can be crosschecked by other requirements. Once requirements are identified, requirements tracing is performed.

Requirements tracing is a medium to trace requirements from the start of development process till the software is delivered to the user. The objective of requirements tracing is to ensure that all the requirements are well understood and included in test plans and test cases. Various advantages of requirements tracing are listed below.

- It verifies whether user requirements are implemented and adequately tested.
- It enables user understanding of impact of changing requirements.

Trace ability techniques facilitate the impact of analysis on changes of the project, which is under development. Traceability matrix is used to store information stored, which relates requirements to design module. The traceability matrix refers to a mapping table that correlates high-level requirements with the detailed requirements of the product. Mainly, five types of traceability tables are maintained.

What the customer wants is the question which seems simple but very difficult to answer [1].



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirce.com

Vol. 5, Issue 5, May 2017

II. RELATED WORK

Many researchers working on software engineering and requirement engineering is always an integral part of it. In [2] authors discuss the concept of requirement management. Requirements traceability facilitates software developers to trace a requirement from its inception to its completeness [2]

In [3] authors proposed a model for software development and requirement management, it discusses that the Requirement engineering is the most essential stage of software development life cycle. It is iterative process which is for good for requirement engineering. The results implied that the successful implementation of proposed model for requirement engineering and management has a good impact on the production of quality software product.

In [4] authors tells in their research article a Requirements Management System based on an Optimization Model of the Design Process describes an engineering design process which is modeled as an improvement issue over the system design space, where the goals, results, and exchange first choices of the users are defined as measurable objective functions. In this kind of formulation, user requirements become constraints that define the viable region of all possible system designs

III. PROPOSED COMPREHENSIVE MODEL

The proposed model is the extension of the model proposed by Mogk et al, for the optimization purposes. In this research article the proposed model does not mention any process which can be used to resolve conflicts b/w requirements in case of any change arrives, called them unresolved ones. They just proposed that requirement traceability will be done by using the traceability matrix, which might not be a good option in every situation and should not report with the body of current constraints and remove unsolved constraints as irrelevant. We proposed a model which is an extension of the above mentioned model focusing on the limitation of the model proposed by Mogk et al. As traceability the issue, my focus is to improve the process of traceability in the above mentioned model. I have added a phase of traceability of unresolved requirement with a focused negotiation stage which will require user involvement with the aim to resolve the conflicts. It will definitely results in lesser conflicts plus the user will have some idea of what its product will look alike now having some of his/her changes incorporated and some of them discarded. One can't just simply ignore the conflicts at a development end, considering unresolved conflicts as irrelevant without user involvement because it might the case that the important ones left behind without the customer knowledge ultimately results in failure of the software. We proposed that traceability matrix should not be fixed there should be traceability techniques that should be used which best suits the situation .To demonstrate working of my model I am taking example of a website which computerize the loan system of a bank. To create its traceability matrix we require business requirement document and function specification document.

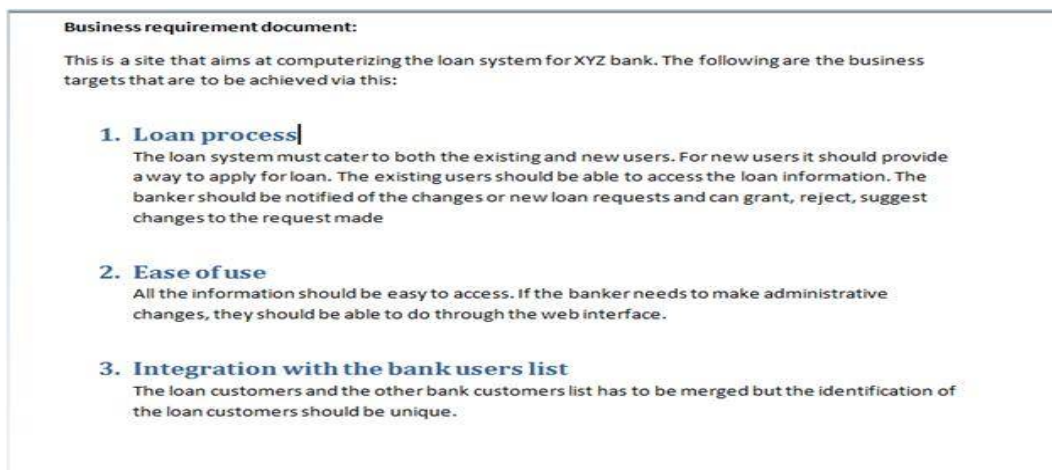


Fig1. Business requirement document of bank loan website

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirccce.com

Vol. 5, Issue 5, May 2017

Fig. 1 shows the BRD (business requirement document) of bank loan system. BRD may be of many pages but here I am including just few points to ease analysis. These are initial important artifacts of our system.

Fig. 2 shows FSD (Functional specification document) of bank loan website. Same as above in the BRD I am just giving here snapshot of few points of FSD. It may also be of many pages.

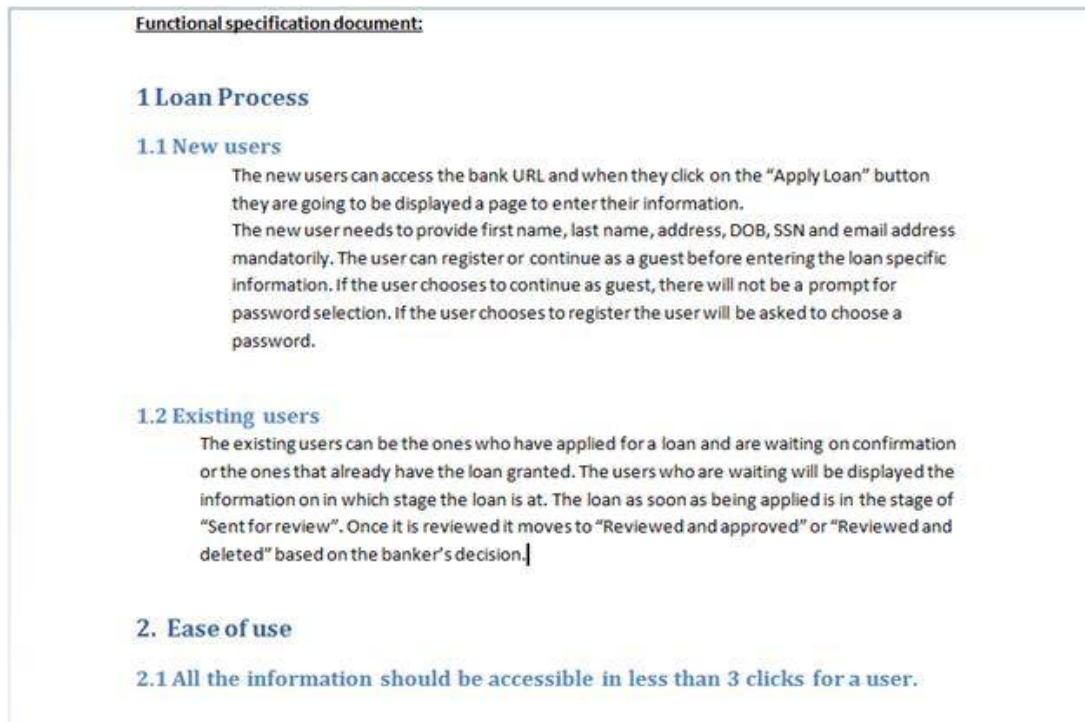


Fig2. Functional specification document of Bank Loan Website

Based on the above two documents, test case scenario file is prepared through any software testing tool. I used here Selenium tool. Following figure 3 presents test case scenario file of my dataset.

Test scenario ID	Test Objective/Test scenarios
TS_Loan_001	Validate the "Apply Loan" feature as a new user
TS_Loan_002	validate the "Apply Loan" feature as a already existing user
TS_Loan_003	For a new user in the "Apply loan", check the guest customer option and apply loan
TS_Loan_004	For a new user in the "Apply loan", check the Register option and apply loan
TS_Loan_005	Login to the loan portal as an already a customer with a loan and check the information displayed
TS_Loan_006	Check the Loan whose status is "Sent for review"
TS_Loan_007	Check the Loan whose status is "Reviewed and approved"
TS_Loan_008	Check the Loan whose status is "Reviewed and deleted"
TS_Loan_009	Check for a visitor if the information on the site is accessible in less than 3 clicks or not
TS_Loan_010	Check for a registered user if the information on the site is accessible in less than 3 clicks or not
TS_Loan_011	Check for a banker if the information on the site is accessible in less than 3 clicks or not

Fig.3 Test case scenarios file of bank loan website

IV. ANALYSIS & RESULTS

In my model user involvement is introduced and requirements are mapped through traceability matrix. Here objectives are given priority and are binded with constraints. Binding constraints and non-binding constraints are the



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

stages which iteratively refine themselves and by each other. Now the role of user involvement is to resolve those unresolved conflicts that came into the phase while mapping. To deploy the right product it is necessary to diminish all conflicts that will ensure the successful requirement engineering. In earlier model user involvement was missing resulting into possibility of system failure. Our aim was to focus those unresolved conflicts and finding out the importance of these requirements causing the conflicts in user view, and resolving them.

Following figure 4 shows a traceability matrix that is formed through BRD, FSD and test case scenario files.

BRD- Sectio	FSD- Section	Test scenario ID
1	1.1	TS_Loan_001
		TS_Loan_002
		TS_Loan_003
		TS_Loan_004
	1.2	TS_Loan_005
		TS_Loan_006
		TS_Loan_007
		TS_Loan_008
2	2.1	TS_Loan_009
		TS_Loan_010
		TS_Loan_011

Fig4. Traceability Matrix of Bank Loan Website

The above document establishes a trace between, the BRD to FSD and eventually to the test scenarios. By creating a document like this, we can make sure every aspect of the initial requirements have been taken into consideration by the testing team for creating their test suites.

V. CONCLUSION AND FUTURE WORK

My proposed comprehensive model driven approach reduce the gap between user or stakeholders' requirement and actual product's functionality resulting into full satisfaction of customer. Requirement can be mapped now through traceability matrix which was the highest drawback of previous models.

In future work we can take big data [6] as dataset and can apply clustering techniques [7][8] on data set. After that we can create its traceability matrix to expand our model. Clustering techniques used for large spatial data such as DBSCAN will be aptly applied on our big dataset. Further dataset can be analyse by decision tree algorithm.[9]. Also we can take dataset as various images [10].

REFERENCES

1. Zielczynski P (2008) Requirements Management Using IBM® Rational® RequisitePro®. Pearson, Inform IT
2. Torkar R, Gorschek T, Feldt R, Raja UA, Kamran K (2012) Requirements Traceability State-of-the-Art: A Systematic Review and Industry Case Study. International Journal of Software Engineering and Knowledge Engineering 22: 1-49.
3. Pandey, D., Suman, U., & Ramani, A. (2011). A Framework for Modelling Software Requirements. International Journal of Computer Science, 8.
4. Mogka NW (2014) A Requirements Management System Based on an Optimization Model of the Design Process. Procedia Computer Science 28: 221-227.
5. Chauhan, Rashi, Pooja Batra, and Sarika Chaudhary. "An Efficient Approach for Test Suite Reduction using Density based Clustering Technique." International Journal of Computer Applications 97.11 (2014).
6. Patnaik, Preetishree, Pooja Batra Nagpal, and Ulya Sabeel. "An Emerging 3-Tier Architecture model and frameworks for Big Data Analytics." (2014).
7. Nagpal, P. Batra, and P. Ahlawat Mann. "Survey of Density Based Clustering Algorithms." International journal of Computer Science and its Applications 1.1 (2011): 313-317.
8. Nagpal, Pooja Batra, and Priyanka Ahlawat Mann. "Comparative study of density based clustering algorithms." International Journal of Computer Applications 27.11 (2011): 421-435.
9. Malik, Sanjay, and Sarika Chaudhary. "Comparative Study Of Decision Tree Algorithms For Data Analysis." International Journal of Research in Computer Engineering & Electronics 2.3 (2013).
10. Mandiratta, Sonam, Pooja Batra Nagpal, and Sarika Chaudhary. "A Perlustration of Various Image Segmentation Techniques." International Journal of Computer Science and its Applications, Volume 139,issue 12(2016)