ISSN:0975-9646

Design of an Automated System to Represent Web Requirements Engineering

Harjot Singh Virdi, Balraj Singh

Department of Computer Science
Lovely Professional University
Punjab, India, 144806

Abstract- **In the SDLC the software requirement is the foremost step. A clear requirement specification results a reliable software development. But any change in requirement will disturb the whole development process and increase the software cost. But when the requirement is clearly defined we can save such cost, effort and time of re-development. In this proposed work we are presenting the agile based requirement gathering automated system for web applications development. This system is basically designed for the web applications where we will merge the user web page requirements, respective validation and the development phase. This whole process will be done in the form of a navigational tool which will accept the user requirements in terms of design and control specification and generates the client side and server side code with validations.**

Keywords- **Agile, Requirement, Requirement Engineering.**

I. INTRODUCTION

A requirement gathering is the major field of software development that provide the basic blue print for development process. Software Engineering gives a scientific approach to the development process. Software Engineering not only defines the development approach it also includes all preprocessing and the post processing of software development. The complete development approach is represented by SDLC (Software Development Life Cycle).

As the idea of developing a software comes in mind, the work of the software development begins with understanding the work feasibility. In case of web based application such as development of websites and the web portals, the preprocessing phase is very crucial. Here the preprocessing means the requirement Engineering. The requirement engineering term itself do not define only the requirement rather it includes three main aspects of the preprocessing In figure 1, The main aspects of Web Requirement Engineering are defined.

A Website basically represents a business or the organization for a user. Because of this a Web Application owner wants his Website user-friendly and that will fulfil the need of his all stakeholders. The Web application owner finds the regular change requirement in his web application. These requirements are based on the appearance of the website or web portal. These requirements include some updation respective to the information that owner want to provide to the stakeholder. It also includes the requirements of the stakeholder that they want in that web application. The inclusion of new attribute in any form filling application is also the major requirement of Web application client. It means in a web application the Web Requirements are very clear at the initial phase but even then these requirements change very frequently.

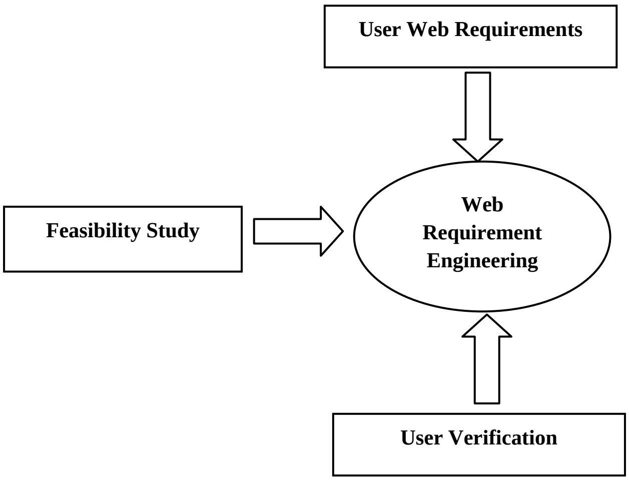

Fig. 1 Web Requirement Engineering.

In Figure 2, the factors that influence the web application requirements are defined. As the requirements are specified the next work is to check for feasibility. Feasibility is checked with respective to the system support, web support, environment support etc. But generally this process is done for one time. Generally once the feasibility is defined; there is no need to change the feasibility after every subsequent change in web requirement.

After the feasibility analysis the next step is to perform the verification and validation. The verification is done by the end user, but in case of web application the end user cannot understand the work done till he himself uses it as an application. It means in case of web application the verification process cannot be done before the code implementation. There are also the chances that during the verification phase the requirements are again modified by the user and after each modification it will be verified again till the user requirement is not fulfilled.

The validation is the process done by the developer as well as the tester to restrict the user to perform any wrong task. In web application the validation is generally performed on form fields to verify the user input. Generally such kinds of validations are common and clearly defined. For this kind of work no preprocessing is required and this whole work can be done during the coding phase.

After completing all these specifications, the work of web requirement engineering is fulfilled, but without the code implementation no phase is yet completed. It means the coding can affect each step and after each step requirements and the code will be changed. Because of this there is the requirement of such a model that will automate the complete system in such way that the requirement, validation, verification and implementation can be done on initial phase. In this proposed work such a model is created with the agile concept.

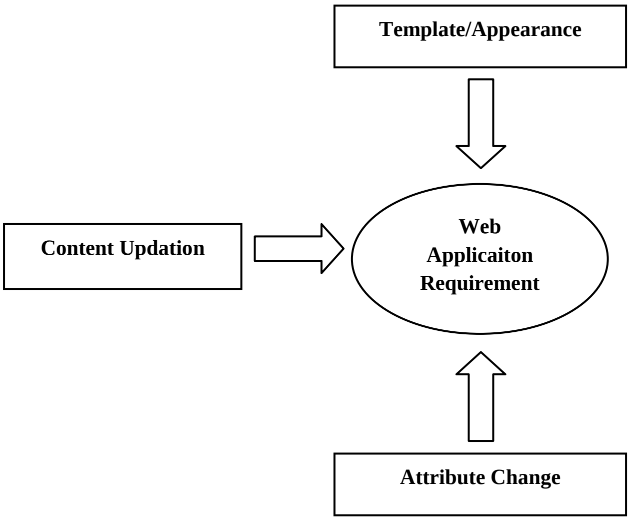

Fig.2 Web Application Requirement

In comparison to the traditional software processes, agile development is less document-centric and more code oriented. Agile methods are adaptive rather than predictive. With traditional methods, most of the software process is planned in detail for a large time frame. This works well if not much is changing (i.e. low requirements change) and the application domain and the software technologies are well understood by the development team. Agile methods were developed to adapt and thrive on frequent changes. Agile methods are people-oriented rather than process oriented. They rely on people's expertise, competency and direct collaboration rather than on rigorous, document centric processes to produce high-quality software [1].

In this proposed application we are using the model based approach of agile engineering. Agile Modeling (AM**.** The basic idea of AM [2] is to give developers a guideline of how to build models that resolve design problems but not 'over-build' these models. Like XP (Extreme Programming), AM (Agile Modeling) point outs that changes are normal in software development. AM does not explicitly refer to any Requirement Engineering techniques but some of the practices support several Requirement techniques (e.g. tests and brainstorming). AM highlights the difference between informal models whose sole purpose is to support face-to-face communication and models that are preserved and maintained as part of the system documentation. The later are what is often found in RE approaches.

Model-driven engineering (MDE) is a software development methodology which focuses on creating models, or abstractions of something more tangible, that describe the elements of a system. It is meant to increase productivity by maximizing compatibility between systems, simplifying the process of design, and promoting communication between individuals and teams working on the system [3].

II. Related Work

The agile learning environment is explained by the Peter M. Jansson[3], In this paper a research is performed on clinical innovations by performing the project based learning analysis. Another article presents the agile based Product model. It is the integrated model to gain the attention in research and controlling the planned reuse [4]. This kind of process model is called Software Product line engineering.

In year 2010 a work is presented to describe the agile efficiency- and quality-focused change management mainly based on scenario-driven regression simulation. The main contribution is the demonstration of the modern software engineering techniques' applicability to develop distributed embedded systems [5].

In year 2010, Zornitza Racheva presented a paper to investigate the fit between real option thinking and agile requirements engineering. The paper presented the published experiences in the agile software engineering literature to identify (i) 'experience clusters' suggesting the ways in which real option concepts fit into the agile requirements process and (ii) 'experience gaps' and under-researched agile requirements decision-making topics which require further empirical studies[6].

Another paper focuses on identifying the tensions between usability and agile methods. The research aims is to identify the common approach of the agile methods and usability engineering by surveying extensive related work on integration of usability and agile methods [7]. The paper presented by David Oxenham identifies four challenges facing researchers and practitioners of Systems Engineering and Systems Thinking in defense capability: Language, Complexity, Longevity and Agility. Its purpose is to stimulate a debate on how best to meet these through systems research and systems engineering practices for all parts of lifecycle and particularly, more agile approaches in R&D [8].

In year 2010, Shinpei Ogata, has defined Evaluation of a Use-Case-Driven Requirements Analysis Tool Employing Web UI Prototype Generation. They compare the proposed method with traditional use case modeling to evaluate the effectiveness of the proposed method [9]. A work is proposed on metamodel for WebML which is based on the Meta Object Facility (MOF). WebML accomplishes the following aims: First, it represents an initial step towards a transition to employing MDE techniques (e.g., model transformations or language extensions through profiles) within the WebML design methodology. Second, it represents an important step towards a common metamodel for Web modeling[10]. María José Escalona has illustrate how graph transformations can be used as a suitable technology and associated formalism to automate the transformations from Web requirement models to Web design models[11].

III. Approach

Today the web applications are being more and more complex. Due to this, to understand the problem and to identify the solution easily, it is required to define the web requirement at the early stage of web development. It is necessary to know the requirements needed as soon as possible or to at least control their growth to guarantee the quality of the system. The software development begins with the requirement phase. A clear and precise SRS makes the developer work very simple. In case of Web Application the user requirements with respective to webpage components and the design is clearly defined but along with these changing requirements by the user it is most important to understand that these changing requirements are the most crucial factor for development of web applications. In such case each user change will rollback the whole process. We are providing the solution for the above defined problem by using the concept of Web Requirement Engineering. In this work we are using the concept of agile to automate the concept of Web Requirement, Design and Development as a single unit.

We are presenting the concept of Extreme Programming Methodology to capture the knowledge requirement of the user and then this knowledge acquisition will be automated to design and develop the web page. We are developing a tool to combine all required phases in a single unit called Navigational Development Approach. According to this approach during the development, we select the field and it will itself design fully tested the web pages. The model is presented in figure 3. The model is the user interactive product oriented model that will do complete Web Development process in a single shot. This whole process is divided in three modules.

1) Navigational Specification

In this module we create user Interface for Navigational Development technique. (i.e.) the User Interface which contain File, Edit, View etc. This has to specify the page name, action page name and the form requirement on the page, here the action page means the server side page name. User can also specify the design for the webpage. For this some predefined templates would be attached with the page specification.

2) Code Generation

In this module the code will be generated respective to the specified design. The client side code is done using HTML scripting. The server side code will be done in JSP (Java Server Pages). It also includes the client side and server side validations.

3) Webpage Generation

As the final stage a complete web application will be generated and published. If the web server is installed and is in the running position the webpage will work for both the client side and the server side.

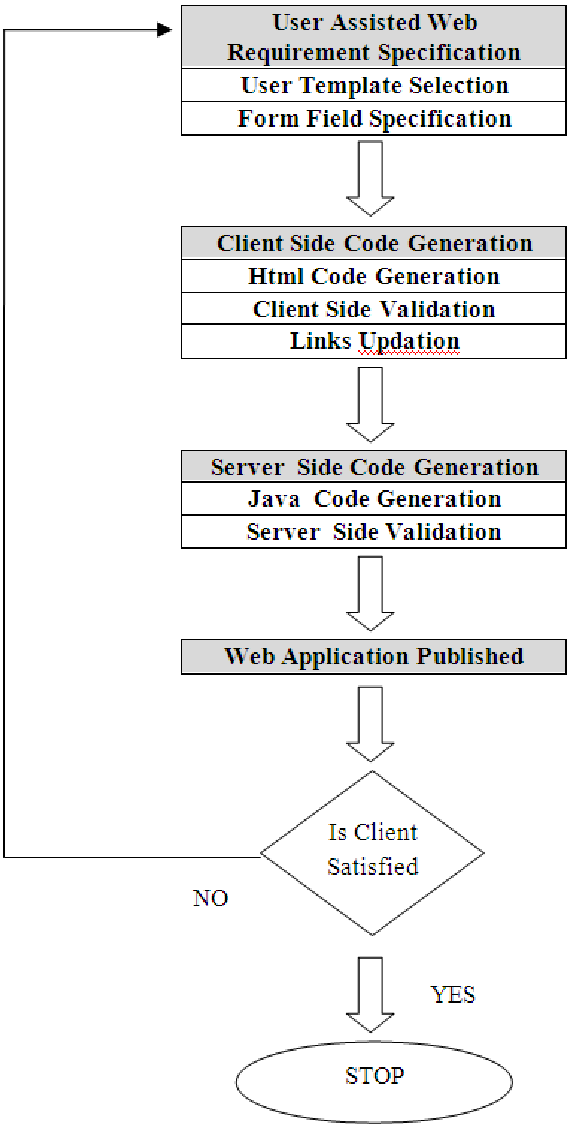

Fig. 3 Interactive product oriented model that will do complete Web Development process.

IV. Conclusion And Future Work

The model created provides the development of a navigational tool that will solve most of the problems faced by a web developer. The system will accept the user requirements and also accept the web template provided by the Navigational tool. Now on the basis of user requirements it will develop a dynamic web page automatically. The complete process is automated.

Acknowledgment

The author would like to thank the faculty of Department of Computer Science, Lovely Professional University, Punjab (India) for their able guidance.

REFERENCES

[1] Vladan Devedˇzic´ and Saˇsa R. Milenkovic, *Teaching Agile Software Development: A Case Study*. IEEE TRANSACTIONS ON EDUCATION, VOL.54, NO.2, MAY 2011 Digital Object Identifier 10.1109/TE. 2010 .2052104

[2] P. Chen, *The Entity-Relationship Approach: Towards a Unified Behavior of Data*. ACM Trans. Database Systems, vol. 1, no. 1, pp. 9-36, Jan. 2006.

[3] C. Barry and M. Lang, *A Survey of Multimedia and Web Development Techniques and Methodology Usage,* IEEE Multimedia, pp. 52-56, Apr.-June 2007.

[4] C. Cachero, *Una Extensio´n a los Ne ´todos OO para el Modelado y Generacio ´n Automa´tica de Interfaces Hipermediales*, PhD dissertation, Univ. of licante, 2006.

[5] S. Ceri, P. Fraternali, and P. Bongio, *Web Modelling Language (WebML): A Modelling Language for Designing Web Sites*, Proc. Ninth Int'l World Wide Web Conf., Computer Networks, vol. 33, nos. 1-6, pp. 137-157, May 2008.

[6] Zornitza Racheva, Maya Daneva, *How Do Real Options Concepts Fit in Agile Requirements Engineering?.*

[7] Osama Sohaib and Khalid Khan, *Integrating Usability Engineering and Agile Software Development: A Literature Review*. 2010 International Conference On Computer Design And Applications (ICCDA 2010) 978-1-4244-7164-5/ © 2010 IEEE

[8] David Oxenham, *Agile Approaches to Meet Complex System of System Engineering Challenges: A Defence Perspective".* DOI 10.1109/SOSE.2010.668

[9] Shinpei Ogata, Saeko Matsuura, *Evaluation of a Use-Case-Driven Requirements Analysis Tool Employing Web UI Prototype Generation,* WSEAS Transactions on Information Science and Application, Volume 7 Issue 2, February 2010.

[10] A. Schauerhuber, M. Wimmer, and E. Kapsammer, *Bridging Existing Web Modeling Languages to Model-Driven Engineering: A Metamodel for WebML,* Proc. Second Int'l Workshop Model- Driven Web Eng., 2006.

[11] P. Valderas, V. Pelechano, and O. Pastor, *A Transformational Approach to Produce Web Application Prototypes from a Web Requirements Model,* Int'l J. Web Eng. and Technology, 1476-1289, 2006.