Strategies for Web Application Development Methodologies

Ravi Sharma
IT Department
Amity University
Greater Noida, India
Ravisrma1988@gmail.com

Shipra Ravi Kumar
CSE Department
Amity University
Greater Noida, India
Shipra.chaudhary85@gmail.com

Keshav Gupta
School of computing science and engineering
Galgotias University, Greater Noida, India
keshavgupta101@gmail.com

Abstract—Web applications have grown very unexpectedly in these recent years but due to difference between web applications and software applications, their development methodologies differ in many ways. The distinction between these two different types of applications acquire for introduction of development methodologies dedicated to web applications. The thesis work represents various development approaches and methodologies proposed specially for web applications. Agile, object oriented, UML based etc. are some example of approaches to be used for inventing techniques for web application development. Appendixes contribute to the thesis work representing implementation of Agile based and UML based web application development techniques.

Keywords—XP (Extreme Programming), WCML (Web Mark Up Language), CORBA (Common Object Request Broker Architecture), AWDWF (Agile Web Development with Web Framework)

I. INTRODUCTION

An application running on a single machine and benefiting its user only i.e., being used for solo purpose, constitute software application and an application running on a network, providing services to its users on the network itself, is called web application. Popularity of web application has risen exponentially in last decade and every day numbers of users of web applications are increasing very rapidly. Software application is all the computer software that causes a computer to perform useful tasks benefiting its users beyond the running of the computer itself, thus, making it distinct from system software which manages computer capabilities but does not perform users' tasks. Thus software application exploits the benefits of some particular computing platform and works for some particular purpose. Some applications have several versions in order to run on different computing platforms. Sometimes, a new application, made for working on a single platform, becomes so popular that it increases desirability of that particular computing platform. Such applications are known as *killer applications*. Web application uses a web browser as a client which request/post some data from/on server which again may be centralized or distributed in order to improve server response time. Client is a computer software application installed at users' side which runs web application, developed in a browser supporting language (such as javascript combined with HTML).

II. RELATED WORK

Since introduction of web into public as *Web 1.0* in 1993, web development techniques have been evolved and raised in exponentially increasing count. The work in this thesis specifies 10 web development methodologies based on different approaches. The very first in this series is *Web Application Development using CORBA* (Common Object Request Broker Architecture) presented by Eun Sook Cho et al. in 1997 at IEEE. This is an object oriented web application development approach that brought phenomenally rapid expansion in World Wide Web

Second in the series is again object oriented technique for developing web applications presented by Martin Gaedke et al. in IEEE Internet Computing in 1999 named *Web Composition*. Web Composition technique uses two modules: Component Model and Markup language. Main concept of the technique is modelling web entities like web pages or some specific web link at arbitrary level of abstraction and presenting specifications in WCML (Web Composition Markup Language) which is based on XML.

Extreme Programming (XP) is next approach based on agile development principles to develop web applications. Frank Maurer et al. presented a paper in IEEE Computing in

ISBN: 978-1-5090-1666-2/16/$31.00 ©2016 IEEE

2002 named *Extreme Programming: Rapid Development for Web based Applications* specifying 12 agile practices to be followed while developing web applications. The list of web application development methodologies continues with UML based approach *Web Application Development by Supporting Process Execution and Extended UML Model* proposed by Wookjin Lee et al. presented at Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSCE'05) organised IEEE in 2005. The development methodology is based on agile development approach consisting of navigation model and component communication model, extended from state machine package and interaction package of UML 2.0 respectively.

Next development methodologies described in the thesis is *Agile Web Development with Web Framework* proposed by Hu Ran et al. in 2008. AWDWF emphasize on communication between people for example, the communication among the development team members, and between the customer and development teams. In order to handle rapidly changing customer requirements, prototypes are released frequently and this helps in quick and timely customer feedback. The main idea behind AWDWF is keeping development team focused on the core business logic and thus avoiding programming duplication and wastage of resources, consecutively, improving system's efficiency and stability, and maintaining quality of the web application.

III. DEVELOPMENT METHODOLOGY

Application development methodologies are promoted as a means of improving the management and control of the application development process, structuring and simplifying the process, and standardizing the development process and product by specifying activities to be done and techniques to be used. Using application development methodologies beneficiate the application to be developed in following ways:

1. *Conviction of why to do what to do* - First step in any development technique prescribed by any methodology to use advocates for feasible study of the system being planned to develop. Analysis of feasible study report may lead to deviation in objectives of the system being developed for better good.

2. *Complete and clear set of requirements* - Many techniques for requirement gathering and analysis have been described in various development techniques prescribed by chosen development methodology, using those, helps in getting clear idea of what to do and what is needed for developing the application. If one starts coding the application without having the clear objectives, development of application may lead to unwanted and unreliable product and thus leading to failure of organization.

3. *A proper guideline of how to do* - Now we have well known documented requirements set for starting development of application. But next thing is that how to formalise development in proper steps and manner so that minimal risk would be associated with optimal benefit. Methodologies provide a way for this according to what we want to develop and what will be its running environment.
4. *Optimal formation of teams and distribution of work* - Randomly assigning the work to random people cannot lead to optimal use of workers according to their strengths. It is must to have maximum use of man power in order to get efficient and rapid product output.
5. *Minimal risk with optimal benefit* - Methodologies do not only provide ways for efficient development of project but also guides after the development in terms of deployment and maintenance plans.

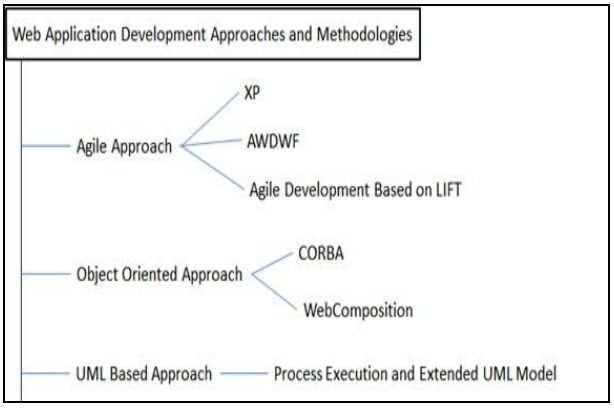

Figure 1: Web Application Development Approaches and Methodologies

IV. AGILE WEB DEVELOPMENT APPROACH

Agile software development is not a single approach but a framework which contains a family of processes which is based on some core principles, also called as *Agile Manifesto* drafted by well known developers in 2001 defining the agile development.

Agile methodologies embrace small team formation, making face-to-face communication possible, working together to define quick prototype to visually present the problem. As shown in below Figure, each iteration has its requirements followed by development, defining and running integrated test scripts, and finally getting results verified by

the user. Verification occurs much earlier in the development process than it would with waterfall, allowing stakeholders to fine-tune requirements while they're still relatively easy to change.

A. Extreme Programming

Extreme Programming (XP) is one of the most popular agile approaches to develop the web applications. Below figure shows over all process of development of web application for SMEs (small and medium enterprises) integrating requirement analysis and designing phase using XP practices.

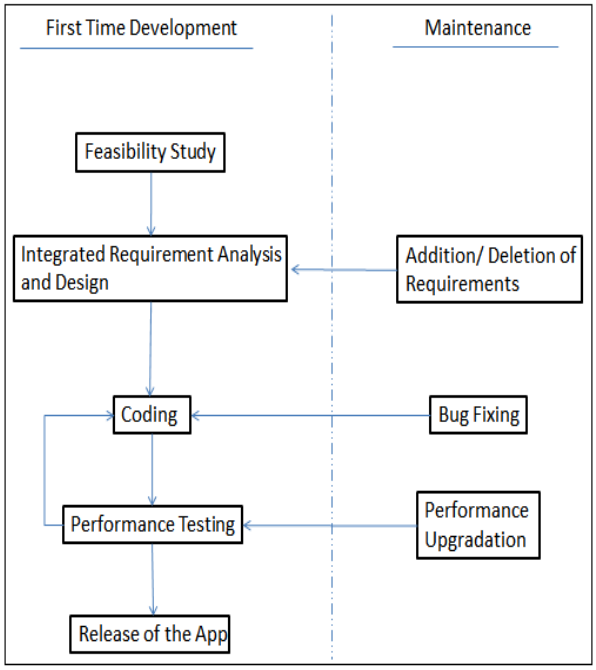

Figure 2: Development of Web Application for SMEs Integrating Requirement Analysis and Designing Phase using XP Practices

Extreme Programming (XP) approach focuses on building small teams of 5 to 15 members and follows 12 related practices. Also, XP concentrates on delivering executable code and automated test drivers rather than spending effort on paper based requirement and design documentation part. Building small teams for development helps in avoiding documentation part as face to face communication can take place effectively. But, as the development organization grows time spent exchanging product knowledge and training new people increases and often renders XP unsuitable. XP proposes a set of software development practices to increase productivity while maintaining quality

- *Development of website red drop using XP practices*

Red Drop is a social website motivating people for donating blood to needy ones. This website provides information about blood donors available in required location. Also, people, who are willing to donate their blood, can register with the website and update their contact information on the server so that needy persons could contact them and get the blood. For requirement elicitation, *Use Case* diagrams have been used. First of all, home page designing takes place for which use case diagram is drawn in order to capture main requirements set. Below Figure shows use case diagram for *Home Page*.

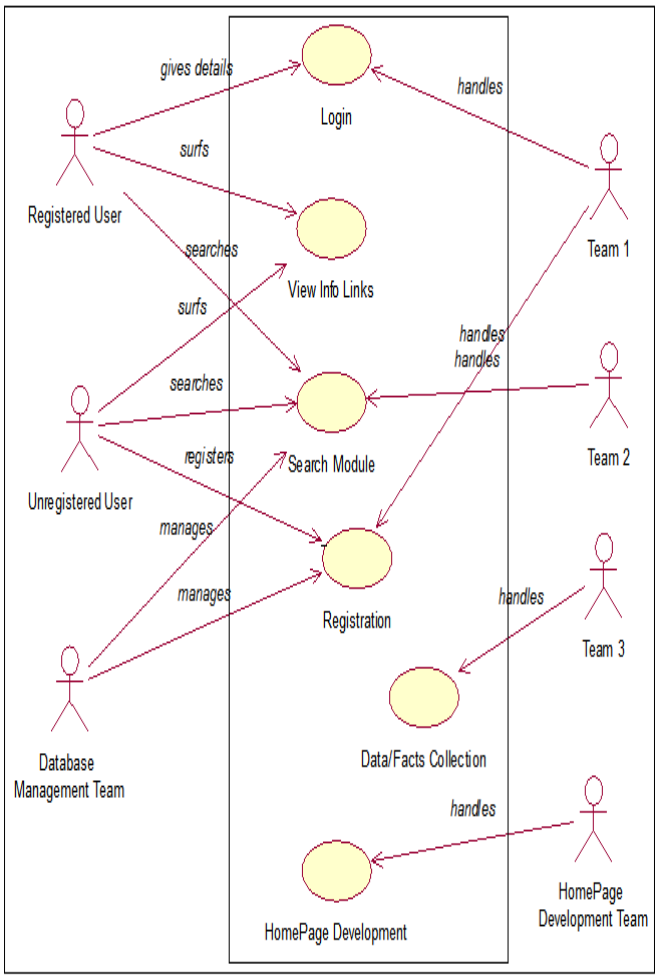

Figure 3: use case diagram (homepage)

Here we present user interface view of all the web pages of the website *Red Drop* from the perspective of anonymous visitor, registered user, and admin.

ANONYMOUS VISITOR (HOME PAGE)

Figure 4: Screen shot of home page of RED BLOOD website

ADMIN (Admin Panel)

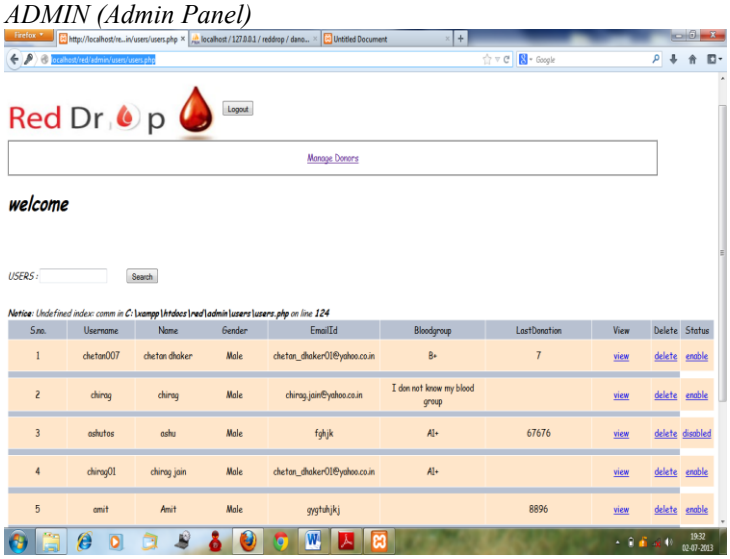

Figure 5: Screenshot of Admin Panel

B. *Agile Web Development with Web Framework (AWDWF)*

AWDWF exploits advantages of agile development using web framework. It is an excellent Web development technique for the Web application system which fulfils the requirement of quick service, quick response, and adoption of rapid changes. AWDWF emphasize on communication between people for example, the communication among the development team members, and between the customer and development teams. The success of project depends on regular and continuous customer feedback. In order to handle rapidly changing customer requirements, prototypes are released frequently and this helps in quick and timely customer feedback. This process of receiving feedback, testing the product and then evaluation take place throughout the entire life cycle. The main idea behind AWDWF is keeping development team focused on the core business logic and thus avoiding programming duplication and wastage of resources, consecutively, improving system's efficiency and stability, and maintaining quality of the web application. AWDWF is based on the web framework and therefore, the most of the web application system is based on the MVC multi-level structure to develop.

C. *Agile Web Application Development Based on Lift Framework*

Lift is framework for development of web application which is based on Scala programming language combining advantages of popular frameworks like rails and django.

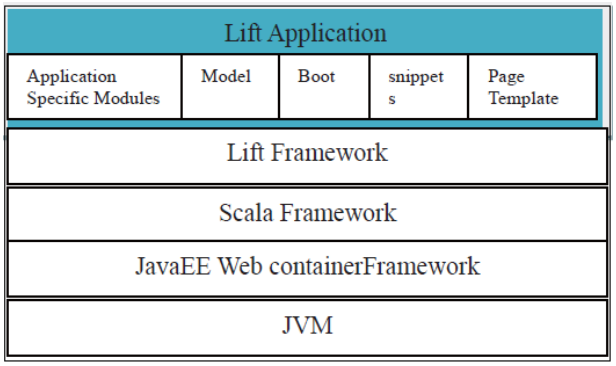

Figure 6: LIFT application Framework's Position

The very first layer represents the agile application framework which is here point of interest and HU Dong, and Yu Xue have described this framework in their research paper "Designing and implantation of agile framework based on Lift" published in 2010 at IEEE.

- *Designing of AGILE application framework*

Below figure shows a pipe filter architecture depicting main components and work process of agile framework. The XML file is input to analysis engine which parse the file and generate the corresponding components. These components

along with the database make input for lift framework which then outputs the information general parts.

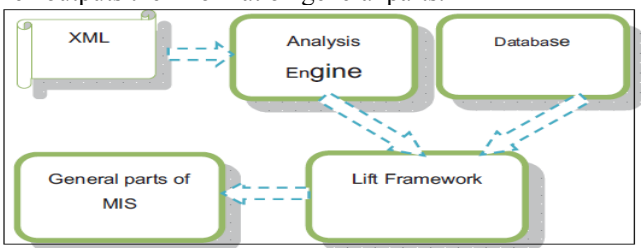

Figure 7: Pipe Filter Architecture

- *Implementation of Framework*

(i) *Analysis Engine Subsystem* - XML analysis engine is based on javax.xml which is a standard java library and its main function is transforming the XML information into Document Object Model (DOM) tree structure and then analyzing them using Xpath mechanism.

(ii) *Menu Generation Subsystem* - Page information is converted into the main menu of the system. Also, menu structure, page view style, and access control strategies are defined by menu generation subsystem.

(iii) *Template Generation Subsystem* - This is one of the kernel function of the framework responsible for generating different components like add, delete, update, query etc. Figure 19 shows the process of template generation.

V. OBJECT ORIENTED WEB APPLICATION DEVELOPMENT METHODOLOGIES

A. Web Application Development based on CORBA

Common Object Request Broker Architecture (CORBA) is an architecture that is proposed by OMG for object oriented distributed computing environment. CORBA provides service suit which includes services like naming, events, trading, security, externalization, licensing, concurrency, transactions, persistence, time, and so on. It allows different implementation languages in server and client. In CORBA, objects are accessed using Object Request Broker (ORB) which maintains two databases: Interface Repository, and Implementation Repository. This model is effective for establishing peer-to-peer distributed systems. CORBA architecture provides fast web application through load balancing. Client can access all of the objects without physical location because both process and data distribution are supported by CORBA. Following points explains CORBA in brief -

1. *Object partitioning* - There may be three types of object partitioning - client objects, server objects, and client/server objects. Main advantage here is that almost all objects may play role of client and server because each object can access the method of other object without physical location.
2. *Process/Data distribution* - It means that local machine at client side should process the data thus reducing the network traffic and enhancing response time.
3. *Multiple services* - CORBA provides various services as named earlier which supports the integration and interoperation of the distributed objects. For example, Object Life Cycle service defines how CORBA objects are created, deleted, moved, and copied.
4. *Synchronization* - For efficient client/server system, blocking and non-blocking, two model are considered. In non-blocking model, clients can process other tasks after issuing request.

B. Web-Composition (Object Oriented Web Application Development)

The approach is based on a Web component model that abstracts from low-level Web implementation technologies to support seamless, reversible development of Web applications. Figure shows the main idea of Web Composition which is composed of mainly two parts - component model, and Mark-up Language. A resource generator maps the component model to a standard web implementation. The overall process is deriving Web view from developer's view of an application which is maintained by component model. Web Composition Mark-up Languauge is based on XML language which is a Meta language facilitating user defined tag based textual format. The figure – 21 shows an example of structure of a WCML document. The XML document type definition of WCML describes a Markup notation for WebComposition concepts i.e., for component descriptions, properties, and relationships.

```
<wcml>
<component id='CHeader'>
   <property name='text' value=''/>
   <property name='level' value=''/>
   <property name='content'>
      <H<<refprop name='level'/>>
      <refprop name='text'>
      </H<<refprop name='level'/>>
   </property>
</component>
<component id='CFooHeader'>
   <prototype is='CHeader'/>
   <property name='text' value='This is a level 2 header'/>
   <property name='level' value='2'/>
</component>
</wcml>
```

Figure 8: Code describing the structure of a WCML document

VI. UML BASED WEB APPLICATION DEVELOPMENT METHODOLOGIES

A. Web Application Development by Supporting Process Execution and Extended UML Model

Although UML is one of the commonly used modelling language for modelling software projects but when it comes to web application development, it is not sufficient. This technique uses two more features of UML i.e., conceptual model, and architecture, drawn using class diagram and component diagram without extension, respectively.

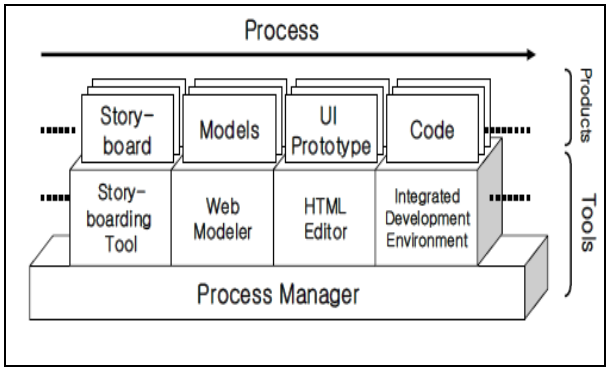

Figure 9: Overview of Methodology

First work product shown in the figure is storyboard which is representation of part of web application developed in the current active cycle along with its interaction with user. Story boarding is performed by software analyst who defines the goals of the web application.

By using the information provided by the storyboard, architecture and user interface prototype are produced. UML component diagram is used for designing architecture. For confirming the UI design, client gives his report formalized as attribute wise such as layout, images etc. The component developer implements the components, which constitute a business logic layer of the currently developed Web application. Finally, the components created by component developer, UI prototype confirmed by clients and DB system are combined into a subsystem of Web application. Each development cycle produces a subsystem and their integration after performing test activities gives final web application.

VII. CONCLUSION

Agile development approach forces on quick development concept avoiding need for large documentation. Three methodologies have been discussed based on agile development approach: Web application development using extreme programming, AWDWF, and agile development based on LIFT. XP methodology is useful when customer wants his continuous involvement with developer team. Next methodology based on agile approach is AWDWF, this approach gives some better quality result as development team is kept focused on the core business logic. One more methodology is, LIFT based development, which is more managed development method in compare to previous two, and used for somewhat complex applications provided they do not require very continuous handling of change in requirements. In Object oriented development approach CORBA is one of the methodologies for developing web applications. Web Composition, is another object oriented methodology which is again capable of developing much complex web applications in a very cost effective manner.UML based web application approach provides a very easy and effective way to manage and control development process by use of UML diagrams. This methodology can be used when there is a requirement of handling very quick and major changes in requirements very effectively in very less time.

REFERENCES

1. Agile development: An overview; http://www.richappsconsulting.com/blog/blog-detail/agile-development-an-overview/; Retrieved 9th April, 2013.
2. Campbell-Kelly, Martin; Aspray, William (1996), “Computer: A History of the Information Machine”; New York: Basic Books. ISBN 0-465-02990-6.
3. Center of online learning, research and service. http://www.uis.edu/colrs/learning/technologies/web20/ Retrieved 20th May, 2013.
4. D. Ungar and R.B. Smith; “Self: The Power of Simplicity”; Proc. OOPSLA 87, ACM Press, New York, 1987, pp. 227- 242.
5. Dieter Blomme, Nico Goeminney, Frank Gielen and Filip De Turck; “A Two Layer Approach for Ubiquitous Web Application Development”; IEEE 2009: Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns.
6. Douglas C. Schmidt, Steve Vinoski; “Object Interconnections: An Introduction to CORBA Messaging”; SIGS C++ Report magazine, November/December 1998, column 15.
7. Dr. Dobb, The world of software development; http://www.drdobbs.com/; Retrieved 9th April, 2013.
8. Eric J. Byrne; Software Reverse Engineering: A Case Study, Software—Practice and Experience; Vol. 21(12), 1349–1364 (December 1991).
9. Eun Sook Cho, So0 Dong Kim, Sung Yul Rhew, Sang Duck Lee, Chang Gap Kim; “Object-Oriented Web Application Architectures and Development Strategies”; IEEE 1997.
10. F. Ricca; “Analysis, Testing and Restructuring of Web Applications”; IEEE ICSM 2004.