Designing an Interface for Improving Reuse in Web Engineering

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Kshitij Goyal
(801031016)

Under the supervision of:
Ms. Shivani Goel
(Assistant Professor)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004
June 2012

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Designing an Interface for Improving Reuse in Web Engineering*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Ms. Shivani Goel* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Kshitij Goyal)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Ms. Shivani Goel)
Assistant Professor,
Computer Science and Engineering Department
Thapar University
Patiala.

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

I would like to thank and acknowledge each and every one who supported me during my thesis work and the whole experience of Thapar University. Many people shared their time and expertise to help me accomplish my goal.

I would like to thank God for always showing me the right direction and the desired strength which helped me stay calm and moving in oddest of times.

No volume of words is enough to express my gratitude towards my guide, Ms. Shivani Goel, Assistant Professor, CSED, Thapar University, Patiala for her constant support, time, patience and guidance. She has helped in exploring this vast topic in an organized manner and her responses to my queries, feedback and editorial comments were invaluable and motivational for writing this thesis. It was a great opportunity to work under her supervision.

I am also thankful to Dr. Maninder Singh, Head of the Department, CSED, Thapar University, Patiala for providing all the facilities and environment. I would like to thank all my professors for their time, discussion and invaluable suggestions during my work.

I would like to thank my parents, my family members for always supporting me in the smooth and tough moments of my life. With, their blessings, inspiration, motivation and love every dream seems to be achievable.

I would like to thank all of my friends for helping me throughout the thesis work, spending sleepless nights pondering on the problems and possible results.

Finally, I would like to thank all the authors whose work was available for references.

Kshitij Goyal
(801031016)

ABSTRACT

Web engineering (WebE) is a discipline which deals with the web based systems and applications, their analysis, designing, cost effective development, operations, testing of web applications according to their operations needed to be performed and maintenance of the high quality applications. WebE evolved a lot in past few years. The web applications are built using W3C standards. In developing Web-Apps quality is important aspect. But usability, performance, or security aspects need special attention. These applications must be built keeping in mind the wide range of users. There are many things which are common in many websites in a particular domain. By reusing the web based application's in a successful way, another one can be developed. The development and evolution of Web applications by reusing existing software artifacts is an important goal to reduce costs and to increase quality. But reuse in this case is not easy because of the legacy of the Web. Components are being used to promote reuse. As, these components are built with reuse point of view, they provide the general functionalities along with something unique. New component-based technologies and support for service-oriented architectures are being developed and a continuous research is going on to address the related problems with an increasing set of dedicated solutions. The selection of components is a complex mechanism.

Many challenges for reuse in web engineering have been identified from literature review and a survey has been conducted to find critical issues during life cycle phases in web engineering and solutions are provided to handle these. In this thesis work, a tool is designed for suggesting the best reusable component which will be the result of search done by the user on the priority basis in all phases of web engineering. The component will be searched by the priority assigned by the user according to his requirements.

Table of Contents

SECTIONS **Page No.**

List of Figures

List of Tables

Chapter 1
Introduction

Software Engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software [1]. It is also defined as a systematic approach to the analysis, design, assessment, implementation, test, and maintenance and reengineering of software, that is, the application of engineering to software [2].

1.1 Web Engineering

As the time goes by, the need of web based application is becoming more and more prominent. The existing software engineering models are being adapted to make them compatible for the development and maintenance of the web based applications using the web engineering model. A new approach of WebE deals with the use of sound scientific, engineering and systematic approach to the development, deployment and maintenance of high quality web based systems and applications (Web Apps). But, it can be seen that the WebE model can be seen as the adapted version of software engineering model. Analysis is done to make the model useful for the development of Web Apps [3].

Web engineering is everywhere and is becoming more and more integrated part of small and large company's business strategies. And it is spreading at a very fast pace. Almost everybody now-a-days wants to host their own websites and web based applications on the internet. As a result, million's of websites get hosted daily and become a part of the vast network.

It's also encouraging to see that as companies are hosting their websites, they have recorded a tremendous increase in their business. Earlier the websites used to be static but now these are changing to dynamic. The logic behind the design of dynamic websites is web engineering. As a traditional software process development, web engineering also has many phases of development. There are some portions of a

websites which are common to all websites. Thus, those parts can be designed from a reuse point of view to save time and efforts.

Studies indicate that these limitations in users' web development activities are not due to lack of interest but rather to the difficulties inherent in interactive web development [4]. Many end users can envision simple interactive applications that they might try to build if the right tools and techniques were available [4].

When developing a small application consisting of only a few web pages, it may be sufficient to follow a reasonably linear development sequence starting with analysis followed by design, implementation and maintenance [5].

Poorly developed web-based applications that continue to expand have a high probability of low performance and/or failure. Recently, large web-based systems have had an increasing number of failures [6].

1.2 Traditional Software Engineering Phases

Software development is an activity of developing a software product using different methods and processes. This activity can be writing codes, re-engineering, maintenance, *etc*. More often we perform all the steps involved between the conceptions to the final product.

1.2.1 Phases of Traditional Software Development

Generally, the following phases are used in software development. But these may vary according to the developer also. The phases are as follows:

a) **Information Gathering:** It is the first and foremost step in software development as it deals with collecting the requirements to get an idea why the software needs to be developed.

In this phase, the developer deals with customers and ask for requirements specific to their needs, or gather information through survey if it is not being developed for a particular customer.

The person who is interacting with the customer must ask a lot of questions to get the proper understanding of the requirements. Some of these questions can be:

what is the purpose of the software which is to be build, who are the target audience, specified requirements which have to be incorporated in the software, *etc*.

b) **Planning**: After getting the requirements, the next step is planning. In this step, usually the strategies about how to proceed with the requirements are gathered and some meaningful things are developed out of it.

During planning, we do consider things like estimation of cost, what technologies should be chosen to work upon, approximate time span needed to complete the project work, taking care of business goals and interests of the company and understanding how to achieve those goals, risk related to the project, testing and quality assurance activities, post release activities, *etc*.

c) **Design:** Designing is the process which gives us the blue print or a model using which the software is developed.

During designing and analysis phase the developer develops a model on the requirements gathered from the customer before starting the actual development work. The models are developed so that the requirements can be understood more clearly and it allows the developer to check the model from multiple ways and uncover the hidden errors. This acts as a validation of the software requirements. The analysis model contains all the data flow diagrams, class diagrams, use case diagrams, *etc*.

The design process gives the levels of abstraction which reduces the problems in the smaller and manageable modules. It also gives the architectures, interfaces, patterns, etc needed to implement the system.

The design however must contain enough information that shows how the developer can translate the requirements into the final product. The designer must adhere to the requirements and make designs accordingly. If the designs are made for unknown users, the developer must check the likes and dislikes of the target audience. The software design must be done in such a way that it fulfills all the needs of the software regardless of its domain, size and complexity. The design attributes are: simplicity, consistency, robustness, visual appeal (look and feel) and compatibility.

d) **Development**: In this phase, the actual functional software is developed. The software engineer starts the development. All the blue prints developed in the earlier phases are implemented in this phase. The coding is done in different languages

which are considered according to the user requirements or the programmers experience, *etc*.

Generally, we deal with two terms like front-end and back-end in software development. The front-end deals with the Integrated Development Environments and tools like Net-beans, Eclipse, Visual Studio, *etc* which helps in the efficient building of the software. The back-end deals with the databases which are used to store the information.

Different methodologies are used during this phase, for example sometimes we use pair programming in which we use one coder and one tester working together.

Time taken can be reduced by working on the modules in parallel and this can be achieved by dividing work to different teams according to their field, experience and other factors. But, building a team is not an easy task and making a team who jells together, is certainly also not an easy task. Certain key points must be remembered while building a new team such as establishing set of team guidelines, need of strong leadership, commitment from each member, *etc*.

e) **Testing and Delivery**: This is an important step in the life cycle where the testing of the software is done. Testing begins with the completeness of functionality, user interface, design architecture, navigation, component, integration, configurational testing, performance testing, *etc*.

So the developer must ensure that whatever he codes must be in accordance with the current technology and standards. Once the testing gets over and the developer gets an approval from one and all, the software is delivered to the customer.

f) **Maintenance:** The development is not yet over even after the delivery, as the developers have to deal with the recent technological changes, updating new content or offering new products.

The customer feedbacks are also important to keep updating the product. All these updates are considered to make changes according to customer’s interests and the new software increments are released as the new versions sometimes.

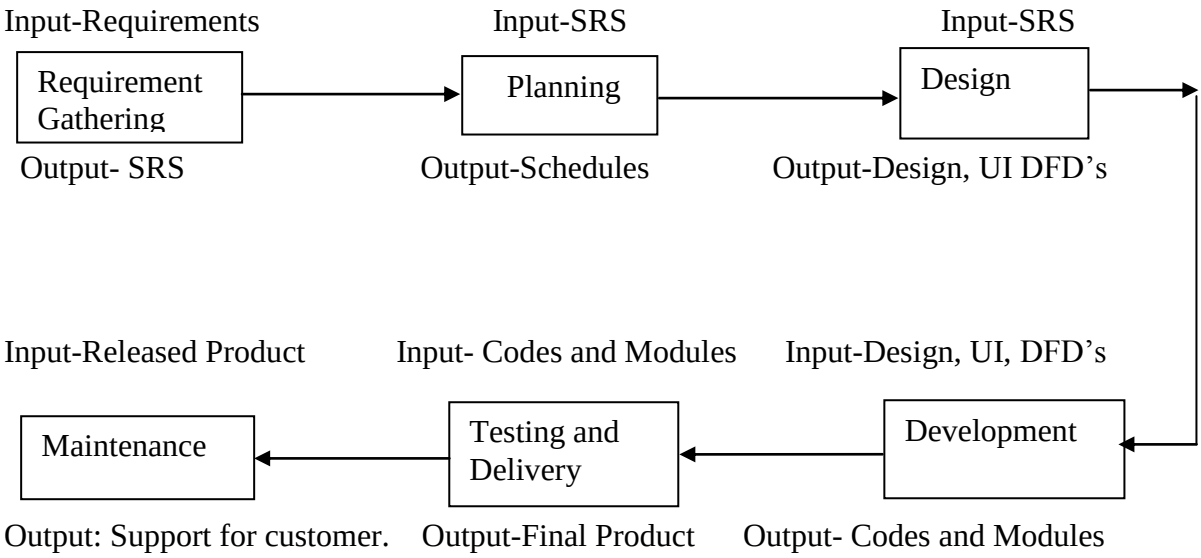

Figure 1: Phases of Software Engineering

1.3 Web Engineering Phases

Web Engineering deals with disciplined and systematic approaches to development, deployment, and maintenance of web based systems and applications. The development of Web-based systems and applications incorporates process models adapted from software engineering methods to the characteristics of WebApp development and a set of important enabling technologies [7].

1.3.1 Phases of Web Engineering

Generally, this is the web engineering process modes used which are influenced by the attributes of web based applications and systems. The web engineer chooses the best appropriate model according to his/her needs. The phases are as follows:

a) **Customer Communication:** It is the first and foremost step in website development as it deals with collecting the requirements to get an idea why the website needs to be developed. In this phase, there are two major tasks to be done, first the business analysis and secondly the formulation.

In business analysis we find why we are developing the web based applications, what help it can do or advantages it will provide to us, and will the new application be integrated with other applications and database.

Formulation is the activity which deals with customers and asks for requirements specifically to their needs, or gathers information through survey if it is

not being developed for a particular customer. The web designer must ask a lot of questions to get the proper understanding of the requirements and what problems it can solve.

b) **Planning**: After getting the requirements, the next step is planning. In this step, usually the strategies about how to proceed with the requirements gathered and some meaningful things are developed out of it.

During planning, we do consider things like what technologies should be chosen to work upon, approximate time span needed to complete the project work, taking care of business goals and interests of the company and understanding how to achieve those goals, risk related to the project, testing and quality assurances activities, even the post release activities, *etc*.

c) **Modeling:** It is the process which gives us the blue print or a model using which we can develop the web applications. During designing of a site map, the developer should keep in mind the generic views as well as some specific views, as these views are the guidelines for development. The design however must contain enough information that shows how the developer can translates the requirements into the final product.

The designer must adhere to the requirements and make designs accordingly. If the designs are made for unknown users, the developer must check the likes and dislikes of the target audience. The design goals that must be followed while developing a web application regardless of its domain, size or complexity are: simplicity, consistency, robustness, navigation, visual appeal (look and feel) and compatibility [7].

d) **Construction**: In this phase, the actual functional web based applications is developed. The aim is to develop a good interface which is easy to understand and navigate. The navigation is an important part while dealing with web applications because the visitors must be able to navigate the web application easily. There should be no dead ends in the web sites, the user must at least be able to reach home page from the current page. The navigation helps a user feel comfortable with the interface. But the question arises how to manage the links between the functions provided and the objects providing them. For this we have to apply the relationship-navigation analysis. This provides a series of steps which help us to identify the relationships.

Some basic rules that must be followed while developing are: not too much text on a single page which makes it monotonous and boring to read, use of minimum animation according to need. The development needs a wide range of skills, so teams are made and roles are assigned to people accordingly. Some of the roles assigned are content developer, web publisher, business domain experts, administrator, *etc*.

Certain key points must be remembered while building a new team such as establishing set of team guidelines, need of strong leadership, commitment from each member, *etc*.

Testing begins with the completeness of functionality, user interface, design architecture, navigation, component, integration, configurational testing, performance testing and the cross browser compatibility. Different type of testing is done to cover up multi dimensional aspects which can uncover maximum hidden errors or problems.

e) **Delivery and Feedback**: Once the testing gets over and the team gets an approval from one and all, the Web App is delivered to the customer. The development is not over yet, it is an incremental process after delivery, as the developer has to deal with the recent technological changes, updating new content or offering new products.

The customer evaluates and gives his feedback to the web engineering team, who can start working on the modifications and as a result gives the new required software increment. After every iteration, a software increment is released. Software Increment is the product which the web engineering team releases after this incremental process flow.

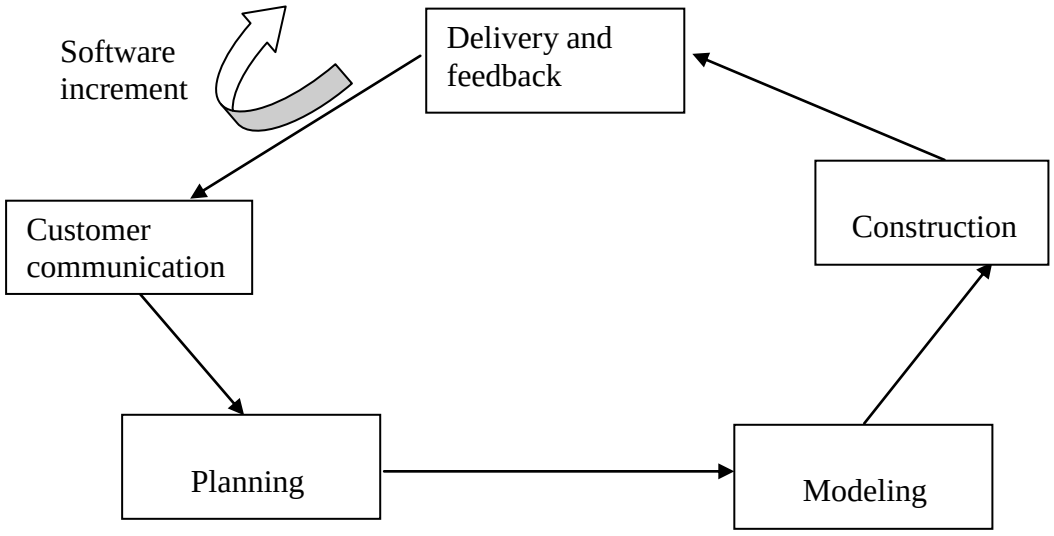

Figure 2: Phases of Web Engineering

1.4 Reuse

Reuse aims at increasing the productivity and quality in large-scale software development [8]. Krueger lists five key processes that should happen for a software artifact to be reused: classification, abstraction, selection, specialization and integration [10]. Classification organizes the stored information to help future queries and updates, both by those who build for reuse and by those who build with reuse. Abstraction helps understandability by hiding low-level details and implementation. Selection is the process where the actor building with reuse chooses what to reuse from the available reusable artifacts. Specialization is necessary in white box reuse, where an artifact needs to be changed to become reusable. To contrast, in black box reuse, an artifact is used as it is. Finally, integration is necessary to make the artifact being reused fit into the context where it is going to operate

1.4.1 Software Reuse

Some may say that software reuse is common, while others complain about the lack of reuse. Software reuse is the use of existing software components to construct new components rather than building them from scratch [9]. The software artifacts can be code, documents, architecture, designs, *etc*. Reuse involves the selection, specialization, and integration of artifacts, although different reuse techniques are used [10]. The purpose of reuse is to improve the quality and productivity of the product. Reuse based software development emphasizes strategies, techniques, and principles that enable developers to create new systems effectively using previously developed architectures and components [11].

1.4.2 History of Reuse

In 1968, NATO Software Engineering Conference was held and it was focused on software crisis; it is the problem of building large, reliable software systems in a controlled cost effective way [10].

Software reuse is the process of creating software systems from existing software rather than building them from scratch. This simple yet powerful vision was

introduced in 1968 [10]. From the initial days of software reuse, it was aspired as a means to overcome the software crisis.

1.4.3 Perspectives of Reuse

There are different perspectives from which to view software reuse. These are briefly discussed as [9]:

1) By substance: It defines the essence of the items to be reused *i.e.* ideas, artifacts.
2) By scope: It defines the form and extent to which reuse can be done *i.e.* horizontal or vertical.
3) By mode: It defines how the reuse is conducted *i.e.* planned, ad-hoc.
4) By technique: It defines what approach is used to implement reuse *i.e.* conceptual or generative.
5) By intention: It defines how elements will be reused *i.e.* black box or white box.
6) By product: It defines what work products are reused *i.e.* source code, design, *etc*.

Table 1 shows the different perspectives of reuse

Table 1: Perspectives of reuse [9]

By substance	By scope	By mode	By technique	By intention	By product
Ideas; Concepts	Vertical	Planned; systematic	Compositional	Black box; as it is.	source code
Artifacts; components	Horizontal	Ad-hoc; opportunistic	Generative	White box; modified	Design

1.4.4 Benefits of Software Reuse

These are the main benefits of software reuse.

1). Gains in productivity: It is discussed that productivity increases if the more things are produced in the allotted time. It can be achieved by avoiding re-development [11]. Reuse will reduce the effort, labor and cost on the whole project.

2). Gains in quality: When a component is reused it means it had been tested again and again with the current requirements. The component gets thoroughly tested under

different conditions which increase its quality. It is achieved by incorporating components whose reliability has already been established [11].

3). Gains in development schedule: A lot of development time can be saved by using reusable assets. Reusing gives an opportunity to save effort, development time. This result in reduced time to market and a quick release in the market gains more profit.

Other benefits are:

- Increased reliability: It can be seen by components already in working systems.
- Reduced process risk: Uncertainty reduces with the reused components.
- Effective use of specialists: By reusing components, the expert people can work on different projects.
- Standards compliance: Embed standards in reusable components.
- Accelerated development: It speeds up development and delivery process.

1.4.5 Components

Brown and Wallnau, suggested it as a non trivial, nearly independent, and replaceable part of a system that fulfills a clear function in the context of a well defined architecture [7]. Components are defined by its state and behavior. The state is defined by the properties. The properties can be represented as name, value pair.

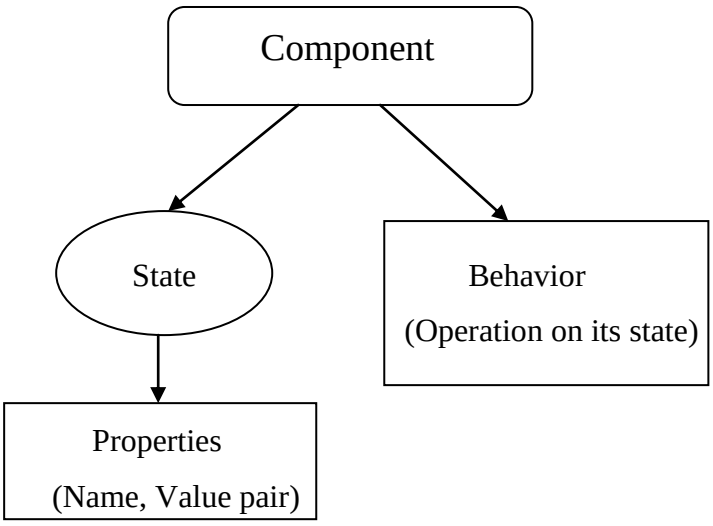

Figure 3: Component in detail.

1.4.5.1 Types of Components

Different types of components can be characterized based on their use in CBSE. These are:

1) Qualified components.
2) Adapted components.
3) Assembled components.
4) Updated components.

1.4.6 CBSE and CBD

Component based software engineering (CBSE) is a process which is similar to other software engineering processes. CBSE emphasizes on the use of reusable components in the development of new systems. The commonalities are in abundance in large systems which can be exploited and gives an opportunity to reuse and fulfill the user requirements.

Component based development (CBD) is a CBSE activity that occurs in parallel with domain engineering. After the establishment of architecture components must be used, either available from library or customized according to the user needs.

These components must qualify and adapted in conformance of the requirements. If new components are needed, they must be built for reuse. Then, the components must be integrated and tested. The analysis of components is done to find the requirements or specification in a component to fulfill the user needs of the current application. The analysis extracts the requirement or specification which is used to search for components in the library. The components found can be reengineered and customized according to the customers need. When new components must be designed use of standard data, templates, protocols must be used.

1.4.7 COTS

COTS stand for Commercial off the shelf components. It is defined as; a third party component that is not developed in-house, available for reuse at an reasonable price without the access to the source code and the user depends on the vendor to provide the support.

1.4.7.1 Categorization of COTS

There are a number of classifications and categorizations given by various researchers. These are discussed here. According to [12], the three COTS classification categories are:

1. Architectural level: It considers the architectural pattern chosen for the system like pipe and filter, 2-tier or 3-tier, *etc* and for COTS.
2. Product kind: It refers to classifying whether COTS is an executable, standard or a specific service.
3. Life cycle phase: It classifies the COTS according to the phase of life cycle in which it is used which could be during development or during execution.

According to [13], four categories of COTS classification are defined:

1. Source: It considers the origin of the COTS component, *i.e.* is it bought from the market or it was acquired from a vendor by a contract.
2. Role: The role of the COTS component also helps in deciding the importance of COTS components. The COTS are categorized as whether the COTS are general purpose or specific to a particular domain. The functionality providing will limit the scope of its reuse. The COTS are categorized as the one providing the core functionality for the application at architectural level or it provides only the support function for the specific application where it is reused.
3. Customization: The COTS may require some modifications before being reused. There could be various modes of modifications according to which the COTS may be categorized. The modification could be by plugging in executable form, just by setting certain configuration parameters, by glueware, addware or change of source code. This also indicates the estimate of cost of modification and hence the difficulty of COTS reuse.
4. Bundle: This categorizes the COTS based on whether it is delivered as a package and what will be the size. This can also help in deciding whether the COTS is suitable for current requirement or not.

1.4.7.2 Challenges of COTS evaluation and selection

The selection of best component from the all candidates is a difficult task. Various challenges are faced during the evaluation and selection of components. These challenges summarized as:

1) Lack of defined process: There is no specific process defined for the selection of COTS. Most of the times, the COTS are selected by an ad-hoc manner. This makes phases like planning, development, *etc*, more difficult as appropriate tools and methods can't be used.
2) Evaluation criteria: There are no proper criteria defined and used for the evaluation of the COTS. This leads to further problems like compatibility issues. They only focus on technical issues, not on non technical ones.
3) Black box nature of COTS: The COTS components are black box in nature. As, they are developed by third party people, the internal working structure of the COTS remain hidden from the user. The documentation of such system is also incomplete or unmaintained. Whenever any changes are done they are not documented properly.
4) Rapid changes in market: New component released, may not have the functionality needed in the current project. It is said that the life of the component becomes half within 6 months of release.

1.4.7.3 Current approaches of COTS evaluation

There are three approaches currently in the market:

1) Progressive filtering represents a strategy whereby a component is selected from a larger set of potential components. Starting with a candidate set of components, progressively more discriminating evaluation mechanisms are applied in order to eliminate less "fit" components [14].
2) In keystone selection strategy, a keystone characteristic such as vendor or type of technology is selected first before selecting the COTS products. Often, interoperability with the keystone will become an overriding concern, effectively eliminating a large number of other products from consideration [14].
3) The puzzle assembly model begins with the premise that a valid COTS solution will require fitting the various components of the system together as a puzzle. The

puzzle assembly approach applies an evolutionary prototyping technique to build versions that are progressively closer to the final system [14].

1.4.7.4 Challenges in Reusing Components

There are many challenges in reusing the components:

1) There may not be enough “good” components around to make reuse worthwhile [15].
2) The recall and precision of the retrieval technologies used to find suitable components may not be sufficient [15].
3) The overall risk and effort involved in finding and evaluating components for reuse may be too high compared to the risk and effort involved in building them from scratch [15].

Chapter 2
Literature Survey

2.1 Developing Web based Applications

As web based applications are becoming increasingly important to all aspects of life, there is a need to encourage practitioners to adopt best practices so as to improve the quality of the processes in use, and therefore achieve targets related to time, budget and quality. The web development industry worldwide is dominated by a myriad (great number) of small firms. This presents a challenge in terms of determining the current practices of industry participants, and in devising improvement initiatives which are feasible for small firms. There were several development methodologies for web-based application. If the proposed software process framework is to be successful then it must be laid on web engineering practices as it is becoming popular day by day [16].

This describes a flexible, model-driven approach for engineering web applications that succeeds through a focus on user interface design and usability. It's simple, model-driven techniques work well for novel applications and integrate readily with lightweight or agile development processes under compressed development schedules [17].

Developing Web Apps is significantly different from traditional software development and poses many additional challenges. Web design frameworks combine generic conceptual, navigational, and context schemas. Such frameworks offer developers a conceptual approach to maximize design reuse rather than code reuse [18].

By nature, Web Apps are designed for a number of unknown users, residing at remote locations, whose requirements are not precisely known. Web Apps are always under the continuous evolution due to technology upgradation and business dynamics. The frequency of changes and modifications are very high. Many web applications are designed for high performance, scalability, and highly reliable utilization and are capable of extending modes of usage and growing throughput. Successful application development in these environments involves bridging the gap between exercising

available programming specifications and the proper design, coding, and life-cycle management of the application [19].

The pressure to be the first to market with a new product has accelerated the development process. Today's process models require agility, defined as the ability to operate in real time and to adapt quickly to changing requirements and conditions [20].

2.1.1 Web Engineering Practices

It is not possible for every web application developer to use the same WebE framework every time they want to develop an application. Modifications are made to suit and fulfill the desired requirements. The fundamental practices should be applied to develop quality Web Apps. The practices are:

- The requirements must be understood properly as it must clearly describe the project objectives and other details.
- They should be clear about the idea how a user will interact with the Web App.
- A project plan describing schedules must be made before starting the project.
- Building a prototype before moving to the actual work is a good practice.
- Construct the application with as many reusable components as possible.

2.1.2 Web Engineering Activities

The activities during web engineering are:

- Web-based system analysis and design
- Web development methodologies and techniques
- Migration of legacy system to web environments
- Web-based real-time applications development
- Web-based multimedia application development
- Testing, verification and validation techniques and tools
- "Web metrics" - metrics for estimation of development efforts
- Performance specification and evaluation
- Update and maintenance
- User-centric development

2.2 Characteristics of Web Based Application Development

The characteristics of web based applications are necessary to be identified because the extent to which the current solutions can be reused is limited. Further characterization of web based application development must be done to ensure effective understanding and development of web based applications.

2.2.1 Application Related Characteristics

Web application development is not only about the functionality, it's also about the content, hypertext, aesthetics, *etc.*

2.2.1.1 Content: The content consists of data written as text, in tables, structured in databases. The content is an important part of the whole web application. It is a textual representation giving details about small units of the application. The content must be of high quality which means it should be accurate, dynamic and updated continuously. Content is only useful when it is of high quality [21].

2.2.1.2 Hypertext: It is a basic part of any web application structure. The web application will contain hyperlinks, tags, *etc,* as an integrated part of it. They are included in navigation design, browsing, *etc* [21].

2.2.1.3 Aesthetics: The first thing which is noticed in any web application is the look and feel of the application. The look and feel should accompany the purpose of the web application. The appearance may depend on the type of users, current technologies, trends, *etc* [21].

2.2.2 Usage Related Characteristics

The users of a web application vary not only in number, but also in background, sophistication level, cultures, *etc.* It depends on the wide range of user, unpredictable infrastructure, *etc.*

2.2.2.1 Natural context: The user's sometimes demand 24*7 availability of services. These requirements need special attention to be fulfilled [21].

2.2.2.2 Unpredictable infrastructure: Plenty of devices are available for user's which vary a lot depending on the capabilities. The quality of service is also affected

because of absence of standards. The developer's can only control the execution at their end. At users end, the complexity varies a lot because the representation of the web applications depends on the client device [21].

2.2.2.3 Wide user base: With increase of internet, web applications are becoming important day by day. The no. of people accessing web has increased tremendously. This increased the user's expectation from the applications. User's with different culture and background leads to variation and conflicts in requirements [21].

2.2.3 Development Related Characteristics

During the development stage, the developers need to focus on various aspects such as team, environment, process, *etc.*

2.2.3.1 Development team: Web application development is done with a team of members which deals with different technology, requirements to develop a product. The team may use Commercial of the shelf (COTS) components during development [21].

2.2.3.2 Developing environment: The environments in which web applications are developed are unstable. Many COTS components are available for integration. Factors like time to market, changing requirements, *etc* make components unreliable and unstable [21].

2.2.3.3 Legacy integration: The legacy systems are integrated to the web applications to make them stable. But, sometimes when changes are made to these legacy systems they are hardly documented and this affects the quality of the developed product [21].

2.2.3.4 Process models: Often, the web applications are developed with frequently changing requirements, compressed schedules, competition driven environment. This increased the need of flexible and incremental process models [21].

2.2.4 Evolution Related Characteristics

The software process models were well planned and they developed software with a method which evolves in years over a series of planned releases. On the other hand, the web development is continuously evolving, without any planned increments. The

web applications are also developed in urgency due to the short time to market, competitive needs. This needs a highly flexible and incremental approach to handle development which helps in the evolution of web based applications.

2.3 Attributes of Web Based Applications

There are many attributes which differentiate Web Apps from traditionally developed software. These are:

1) **Network intensiveness**: It deals with the overhead or the number of clients in the network. Example: number of clients will differ on WAN, LAN, MAN.
2) **Concurrency**: It is tested when large number of users is accessing the application at once.
3) **Unpredictable load**: It represents the number of users accessing concurrently which keeps on changing. Example: rush during peak hours will also vary day to day.
4) **Performance**: It is defined as the time taken for producing the result of the query after submitting the query.
5) **Availability**: The system is expected to always keep on functioning and work without any failure. The scheduled down time does not affect the availability of the system.
6) **Data driven**: The web applications are also used to fetch information from the database which is not an integral part of the web based engineering.
7) **Content sensitive**: The quality of the content displayed in every web page is an important characteristic.
8) **Continuous evolution**: Web applications do not undergo a planned evolution approach for releases of the next increment or added functionality.
9) **Immediacy**: Time to market is an important characteristic in success of web apps. So, the process is adapted to reduce the schedules needed for Web App development.
10) **Security**: As, the webapp are accessed through network, strong security measures must be taken to protect the sensitive information from unauthorized people.
11) **Aesthetics**: It is the overall look and feel of the application which also contributes a lot in making a web page attractive to user.

12) **Usability**: It is the ease in use while using the web application.
13) **Functionality**: The number of different work done by a single application.
14) **Efficient and Reliable**: The time consumed should be less and result must be correct.
15) **Compatibility**: When a new application can be easily integrated with the older ones.
16) **Interoperability**: Deals about working with a number of different other applications.

2.4 Reuse in Web Engineering

2.4.1 Need of Reuse

The reasons why we need reuse in web engineering are because a survey on Web-based project development by the Cutter Consortium highlighted problems plaguing large Web-based projects [16]:

- Delivered systems didn't meet business needs 84 percent of the time.
- Schedule delays plagued the projects 79 percent of the time.
- Projects exceeded the budget 63 percent of the time.
- Delivered systems didn't have the required functionality 53 percent of the time.
- Deliverables were of poor quality 52 percent of time.

2.4.2 Reusing the Web Application UI Controls:

Web application user-interface (UI) is often composed of distinctive UI elements, the so called UI controls. Similar controls are often used in different web applications and facilitating their reuse could lead to faster development [22].

The web applications are developed under tight schedules, rapidly changing requirements, low budget, *etc*. This applies for the process of reuse. Rather than writing new codes, making new design, they look for already available components. This saves a lot of time and effort. The developers now have the luxury to extract UI controls of client. Using available tools the developer extracts the UI controls by selecting and interacting with the desired web page. All the essential code and

resources are extracted, all necessary adjustments are made and the control is packed as a reuse friendly web page [22]. It depends on developer where he wants to use these extracted UI controls on a web page.

2.4.2.1 Extracting and Reusing UI Controls: Before reusing the extracted UI controls, all the details necessary for the control to be functionally working and behavior must be extracted. The extracted details can be the code, CSS, scripts, *etc*.

The process can be separated into three phases as:

1) Interaction recording,

2) Resource extraction, and

3) UI control reuses (figure 4)

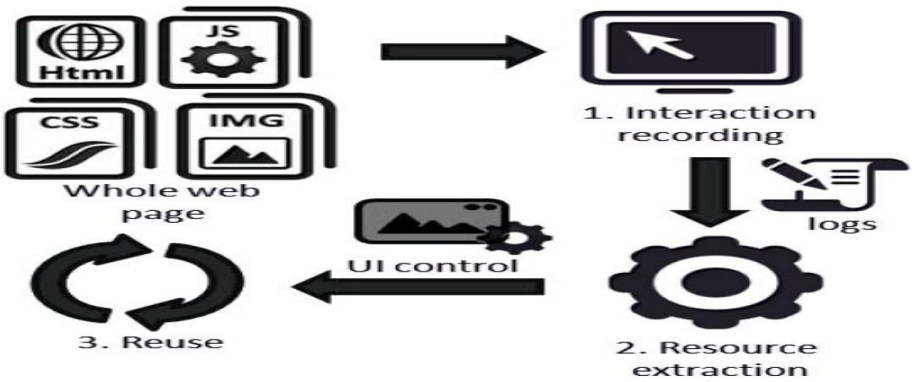

Figure 4: Extracting and Reusing UI Controls in Web Engineering [22]

Chapter 3

Problem Statement

Previous chapters have discussed about web engineering phases, components and reuse. This chapter focuses on problem statement taken up in the thesis.

Over the past few decades, the web application development has grown from simple implementation to complex. Earlier the website used to be static but now these are changing to dynamic. The logic behind the design of dynamic websites is web engineering. Web engineering is everywhere and it is encouraging because it is becoming an integrated part of small and large company's business strategies. As a traditional software process development, web engineering also has many phases of development.

There are some portions of a websites which are common to all and can be reused. Thus, those parts can be designed from a reuse point of view to save time and efforts. This is needed to cater the ever rising demands and solving other problems as well. For example, the number of users has increased tremendously, the expected time for development has reduced but change in requirements has increased a lot [23]. Due to this, there are a number of challenges faced in each life cycle phase of web based application development.

The developers, managers and other stakeholders are in favor of reusing. But, how to do it is a critical issue? To develop a web application, study must be done to find the issues for reuse in all phases of web engineering. A survey was conducted to extract the information about the major issues in all the life cycle phases of web engineering related to reuse. The survey questionnaire is provided in Appendix A.

COTS components are the basic ingredients for a reusable application. Integration of COTS components is carried out to make the application reusable and providing the functionality which is needed to fulfill the user needs. So, it's a complex process of choosing the best suitable COTS component. There are a plenty of candidate components available, but do they really fulfill the criteria that's a question to be

asked. For picking out the best component there is a need of an efficient way of searching the components.

Mostly the main focus by the researchers is the need of an efficient searching technique which can improve the list of candidates which are fit for the purpose. The amount of unwanted results retrieved increases the time, effort, cost, *etc* in managing these results and finding the one which is most appropriate to fit the purpose. So, the searching needs to be more efficient for better reuse.

Here, the aim is to identify various COTS components which can be used in various phase of web engineering.

The objectives that are taken in this thesis report are:

1) To identify and find the solutions of challenges for reuse in web engineering.
2) To find the major issues in all the phases of web engineering related to reuse.
3) To improve reuse in web engineering by suggesting components (including COTS) which are best fit to the purpose.

Chapter 4

Reuse Challenges

Over the period of time web engineering has made strides forward to achieve what it is today. And to keep growing with this pace and prosper it needs to add/ update itself with time. The current requirement is to increase reusability in web engineering to support the development of web applications because of the constraints under which these applications are developed. Whether it is rapidly changing requirements, tight schedules, low budget, *etc*, reuse seems to be the solution for these problems. To integrate or enhance reusability in web engineering, challenges must be identified and encountered successfully.

Here, some challenges for reusability in web engineering are being discussed with proposed solution which can be helpful in encountering these challenges.

4.1 Challenges, Causes and Solutions for Reuse in Different Phases of Web Engineering

The challenges in various phases are:

Table 2: Reuse Challenges in Web Engineering

Web Engineering Phase	**Reuse Challenges**
Customer Communication	Complex and rapidly changing requirements, wider scope , multiple users with different cultures leading to more variations and conflicts in requirements [3]
Planning	Time constraints; cost effectiveness [24]
Modeling	Object orientation to promote reuse of objects across webapp [25], data structuring capabilities to reuse the data structures effectively [26], genericity, flexibility and adaptability, navigability and usability, use of UML [27]
Construction	Complexity, changeability, invisibility and unrealistic schedule [5]
Testing	Vast users with different demands, so extensive navigation testing and usability testing; security testing for web application

	transmitting secure data like online shopping and banking etc; stress and load testing to see the effects on performance with increase in the number of users accessing the web sites; rigorous regression testing.
Delivery and Feedback	Availability; immediacy; concurrency; network intensiveness [3]

4.1.1 Customer Communication phase:

4.1.1.1 Complex and rapidly changing requirements [3]:

Cause: Mostly the customer who gives the requirements in starting doesn't exactly know about them and then (s)he asks the developer to incorporate new requirements which results in the increase in complexity of the project.

Solution: Perform interviews with the clients, timely scheduled meetings with the client to avoid last minute changes, try to improve the clarity of thoughts about the final product, prioritize the requirements to be incorporated.

4.1.1.2 Wider scope [3]:

Cause: As number of people using the internet is increasing day by day. Hence, the functionalities, services expected from an application have increased.

Solution: Find the target audience, find the feasibility of the application and find out the main objective of the web application which it should fulfill and collect requirements for providing different functionality to the user.

4.1.1.3 Multiple users with different cultures [3]:

Cause: The way of using an application is never the same. This leads to more variations and conflict in requirements. Example: In India, we have more than 20 regional languages. So, an application must support all of them.

Solution: Find out the users sophistication level and background and collect the requirements at various levels *i.e.* manager, team leader, developers, user, etc. This will help us in understanding the users and their needs.

4.1.2 Planning Phase

4.1.2.1 Time constraints [24]:

Cause: Mostly, the teams have to work with unrealistic deadlines. So, this phase is either eradicated/not performed or not given enough time.

Solution: Making a brief project plan will help in scheduling and tracking of the project.

4.1.2.2 Cost effectiveness [24]:

Cause: To save money, either it is performed or minimum cost is put on planning phase. But due to lack of planning the projects get over budget.

Solution: Web Applications creates opportunities for cutting development cost and easing evolution via reuse [24]. Example: At architectural and code levels for simplifying design and to meet new challenges reuse must be done.

4.1.3 Modeling

4.1.3.1 Object orientation to promote reuse of objects across websites [25]:

Cause: The web was not designed to handle the complexities and efforts often run into the limitations of web infrastructure.

Solution: To support complex application it must provide general capabilities similar to Object Modeling Architecture (OMA). A web object model must be used which develops and caters the requirements of the web applications [26].

4.1.3.2 Data structuring capabilities to reuse the data structures effectively [26].

Cause: The basic infrastructure of web consists of hyperlinked HTML document which is too simple a structure to support the complex web applications

Solution: Integrating aspects of object technology with the web infrastructure can increase its structuring power. Object technology enhances the functionality of web because it adds the behavior provided by the objects to the static content of html [26].

4.1.3.3 Genericity: It is the ease with which an asset can be instantiated and specialized [28].

Cause: Most of the applications developed are specific to requirements or made to fulfill a particular objective.

Solution: To enhance the reusability aspect of an asset they should have general characteristics. The assets must be built with reuse point of view not for single purpose\use.

4.1.3.4 Navigability and usability [25]: Trying to make the Web Apps as user friendly as possible.

Cause: The links are not appropriate sometimes leading to dead end, errors while linking. The ease to use the application is not present, inconsistent, etc.

Solution: Check for the links; are they working properly or not. The user must have an option to return to homepage from the current web page. The choice of colors, fonts must be made keeping in mind the target users and their ease for accessing the web pages.

4.1.3.5 Use of UML [27]: It is done to make UML diagrams which help in understanding.

Cause: The UML's further improvement is refrained because of two major strains viz. make it precise, make it executable [29].

Solution: UML is not a programming language. The diagrammatic models have the unique feature which puts it into an entirely different category and makes it stand out of crowd.

4.1.4 Development

4.1.4.1 Complexity [25]: Deals with difficulty in the use of Web Apps.

Cause: It increases day by day with Web Apps and becomes unmanageable.

Solution: Choose an appropriate model, infrastructure from the beginning. The secret's hidden in itself, we have to find a path about how we manage changes to the information.

4.1.4.2 Changeability [5]: Less no. of alternatives available.

Cause: Not many applications are present to choose from.

Solution: Build the applications with reuse point of view. It will add to the alternatives and lead to increase in reuse with its benefits following.

4.1.4.3 Invisibility and unrealistic schedule [5]: Trying to complete the project in time under tight schedule.

Cause: Most projects rolled out are either incomplete in functionality or over budget.

Solution: The schedules must be planned properly keeping in mind the resources, effort to be put in, budget, etc. Following the schedule strictly is equally important as making schedules.

4.1.5 Testing and Delivery

4.1.5.1 Vast users with different demands [3]:

Cause: Need for extensive functionality testing and usability testing.

Solution: Testing of all the links on web pages, database connections, forms, etc. must be done. Usability testing includes that website should be easy to use, consistent, fits to purpose.

4.1.5.2 Navigational testing: It deals with surfing of web pages [25].

Cause: Unknown way of navigation; the user can navigate in any manner (s)he wants to.

Solution: The user should not be forced to navigate in a particular direction. He shall always have the option to where (s)he wants to navigate. Different controls like buttons, boxes and links must be provided to encourage easy navigation.

4.1.5.3 Security testing: It deals with authorized access [25].

Cause: Because web applications are transmitting secure data like online shopping and banking applications.

Solution: The testing becomes critical when we deal with such applications. We should perform steps like testing the URL directly without login, test the CAPTCHA script, look for HTTPS, all transactions and errors must be logged and handled carefully.

4.1.5.4 Stress and load testing: Load testing deals with ability to sustain heavy load (mostly on peak time) [25].

Cause: To see the effects on performance with increase in the number of users accessing the web sites.

Solution: Load testing, it test's how many users are accessing or requesting the same page. System sustain in peak load times. Site should handle many

simultaneous user requests, connections to database, heavy load on specific pages, etc.

Stress testing: It means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and checked how system reacts to stress and how system recovers from crashes. Stress is generally given on input fields, login and sign up areas.

4.1.5.5 Rigorous regression testing [25]:

Cause: Changes are done rapidly. Need to check the effect of change to all existing functionality.

Solution: The automated tools are the best options for regression testing. The test cases must be used to record the scripts on.

4.1.6 Software Increment

4.1.6.1 Availability [25]:

Cause: Web Apps are expected to be available 24 X 7 X 365.

Solution: The scheduled downtime should be in effect to maintain the Web App and make it fit to solve the purpose.

4.1.6.2 Immediacy [3]:

Cause: Time to market is less, as fast an increment can be launched the better it is.

Solution: Encourage reuse as it will save the time that would have consumed in putting efforts to make the component or artifact which we are using. Hence, it will reduce time to market with other factors as well.

4.1.6.3 Network intensiveness [3]:

Cause: Different challenges are faced on different network like WAN, MAN, etc.

Solution: It can meet the need of different group of users via intranet, internet.

These are proposed solutions to the challenges for reuse in web engineering. Further study will refine these proposals.

Chapter 5

Experimental and Survey Results

5.1 Research Methodology

A questionnaire was designed to find the reuse options which the developers of web applications want during web engineering. It is provided as appendix A. The questions were divided as per the phases of web engineering i.e. requirements analysis and design, planning, construction, testing, delivery and feedback and the umbrella activities. The questionnaire was sent to 40 software engineers aware of web engineering. Only 15 complete responses were obtained. The users included software engineers, technology consultant's, area head's, project engineer, developer, Ph.D. scholars from various industries including HCL, HP, WIPRO, India NIC Infotech Limited, Provogue India Ltd.

5.2 Research Finding

The findings are discussed as per the phases of web engineering.

5.2.1 Requirements Analysis Phase: There were 10 questions in this phase. First three questions aimed to find the importance of level of clarity of requirements in web engineering, user's background and sophistication and feasibility checking of web application. The results are shown in figure 5.

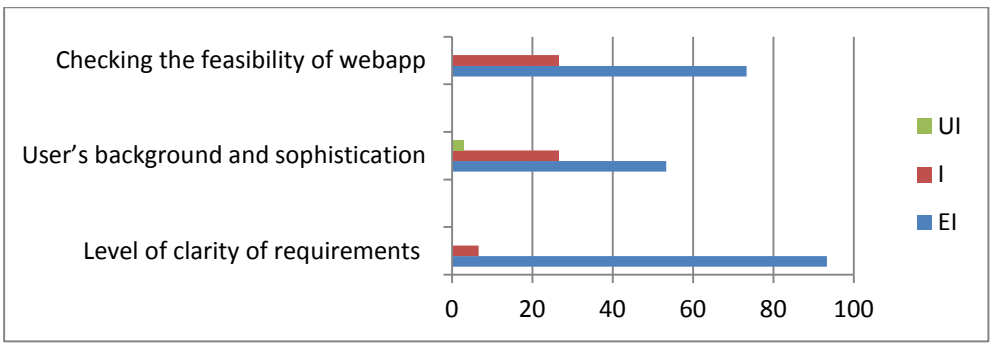

Figure 5: Requirements Analysis and Design Phase (R1-R3)

93% of users indicated that checking the level of clarity of requirements of web application is extremely important. User's background and sophistication was found to be extremely important by 53% of users while 27% found it to be important. The feasibility check of web application was found to be extremely important by 73% while 27% users found it to be important. The next question aimed at identifying which software process mode is considered useful for web application development. The results are shown in figure 6.

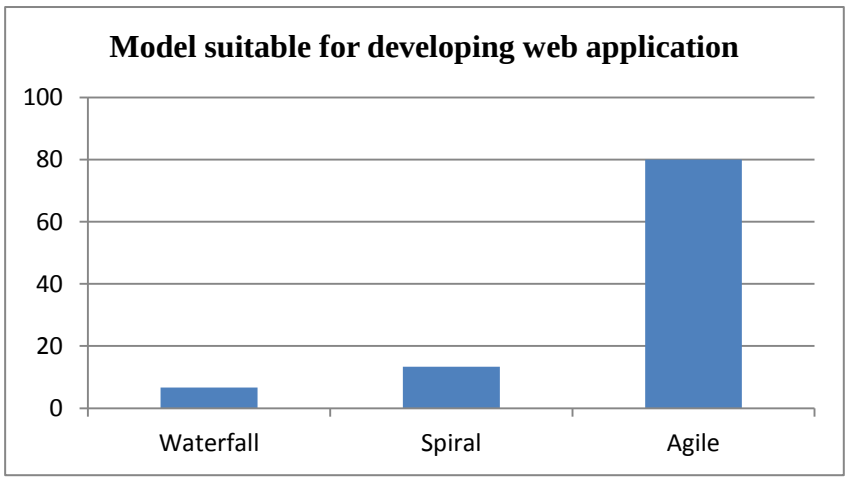

Figure 6: Requirements Analysis and Design Phase (R4)

80% of the users found agile models to be suitable for web application development. Only 13% preferred spiral model while only 7% preferred to use the waterfall model. The next question aimed at identifying the priority of people from whom the requirements gathering should be done. The results are shown in figure 7.

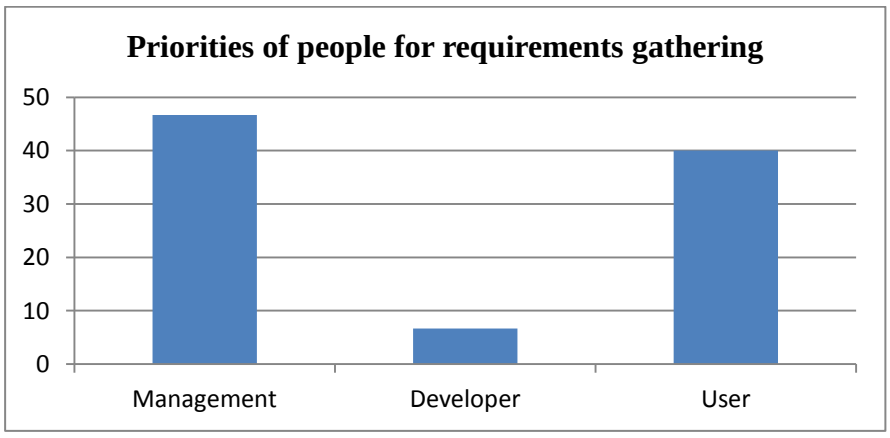

Figure 7: Requirements Analysis and Design Phase (R5)

47% users voted for management for getting the requirements while 40% were in favor of users for getting the requirements. Only 7% voted that requirements should be gathered from developers. The next question R6 aimed at identifying whether there should be a scheduled meeting with the customer for communication during web application development or on adhoc manner. All the users agreed for a schedules meeting. Question R7 was formulated to check the level of rigor required for survey of requirements and all the users agreed for a rigorous survey. The results of question R8 indicated that the web engineers should be extremely focused for handling the user's problem. Next question R9 aimed at identifying the design approach considered suitable for web application development. The design approaches can be top down, bottom up or a combination of both. 53% users confirmed that top down design is best suitable for web application development while 20% confirmed that bottom up approach is more suitable for web application development. 27% of users said that a combination of the two approaches is best option for web application development. The results are shown in figure 8.

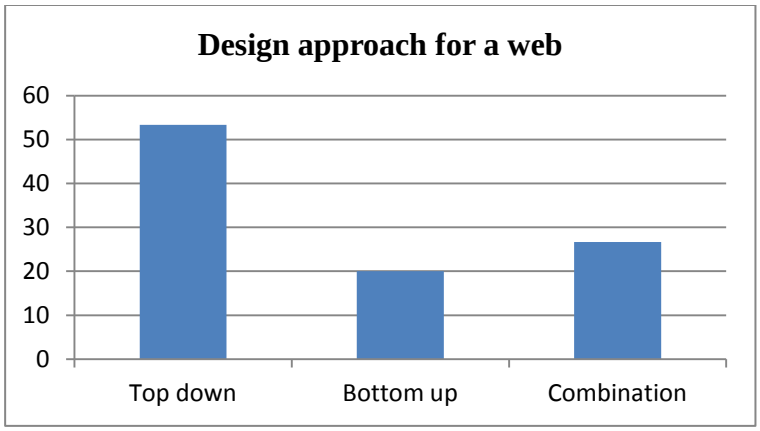

Figure 8: Requirements Analysis and Design Phase (R9)

The next question R10 aimed at identifying the priorities of web application for areas like academics, sports, health, government, business and banking sectors. Seven attributes of web applications were considered. The users were asked to prioritize these for different domain. The results of academic domain are shown in table 3.

Table 3 : Priorities for Academic domain (R10)

Academics	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Sum
User friendly	1	2	1	1	1	1	1	1	1	1	3	1	1	1	1	18

Simple design	3	3	3	3	3	2	3	2	2	3	2	2	3	4	2	40
Security	2	1	2	2	2	3	2	6	3	2	1	6	2	6	3	43
Extensibility	7	4	7	7	7	4	7	5	4	7	5	5	7	5	4	85
Customizability	4	7	4	4	4	6	4	4	5	4	6	4	4	7	5	72
Detailed information	5	6	5	5	5	5	5	3	6	5	7	3	5	2	6	73
Media content	6	5	6	6	6	7	6	7	7	6	4	7	6	3	7	89

The highest priority is given to user friendly and simple interface design. The next priority is given to security of web application. Customizability and detailed information was at next priority level. Extensibility and media content were given a low priority.

Similarly, the data was collected for other domains and the results are summarized in table 4:

Table 4: Priority of Various Domains

	Sports	**Health**	**Government**	**Business**	**Banking**
User friendly	17	21	27	44	30
Simple design	42	42	51	46	46
Security	58	55	22	22	27
Extensibility	80	80	82	78	82
Customizability	62	70	74	69	80
Detailed information	72	61	60	48	60
Media content	59	91	90	89	71

The highest priority is given to user friendly by sports and health domains. Security was given the highest priority by government, business and banking domains. Media content was given the least priority by health, government and business domains. User friendly and simple design interface was given priority by all the domains.

5.2.2 Planning Phase: The first question in planning phase is to identify the importance of planning phase during web engineering. 80% of users have reported it as extremely important while 20% have reported it as important. Next question aims at prioritizing the factors for planning. The factors considered are estimation, quality attribute and risks. The results are shown in figure 9.

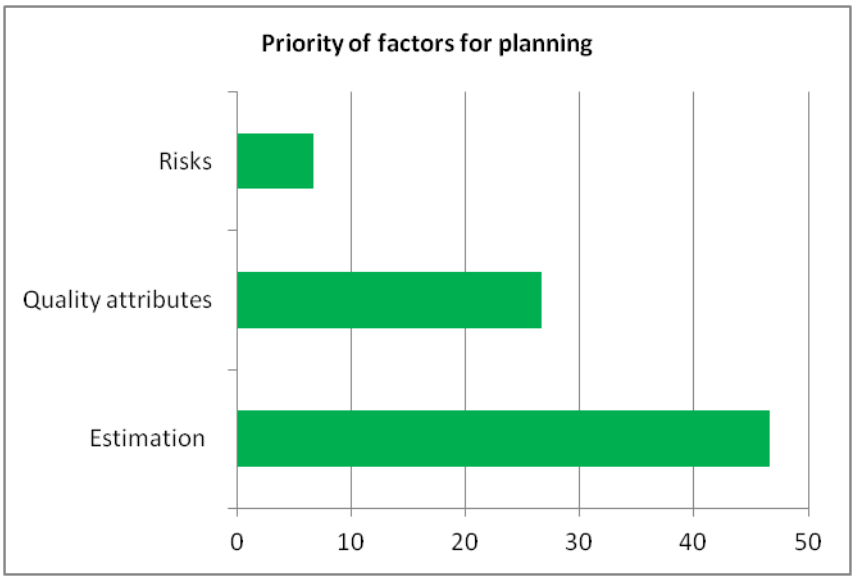

Figure 9: Planning Phase (P2)

Estimation has been given the first priority by 47% of users while 27% users have given priority to quality attributes. Only 7% have given first priority to risks in planning. The next question aimed at identifying the priority of technical specifications needed during planning phase. The results are shown in figure 10.

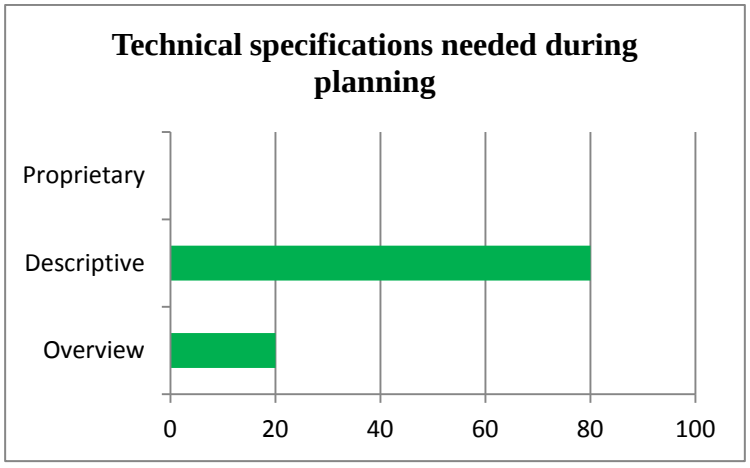

Figure 10: Planning Phase (P3)

80% of users have identified the descriptive specification to be useful while only 20% have suggested that overview of specification will be sufficient. None of the user opted for proprietary specification. The next question in planning phase was to identify the priority of objectives during planning phase. Three objectives were

considered: risk management, estimation and customer satisfaction. The results are summarized in figure 11.

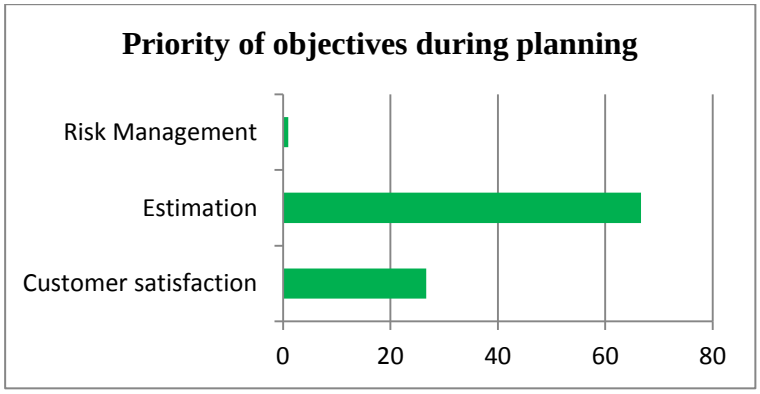

Figure 11: Planning Phase (P4)

The results show that estimation was given the highest priority by majority of the users (67%). The next priority was of customer satisfaction. Risk management was given highest priority only by 1% of users. All the users agreed for importance of planning for risks in question P5. When they were asked (P6) about the feasibility of breaking the planning tasks into smaller milestones, all the users agreed that it is feasible. In the next question, the users were asked for the ideal size of team for planning. The results are shown in figure 12.

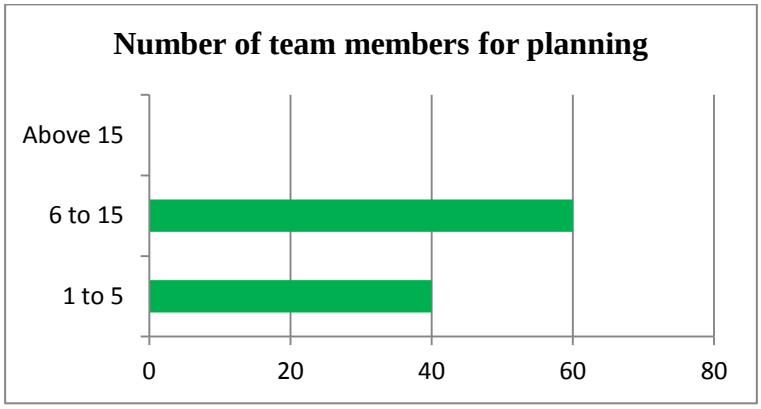

Figure 12: Planning Phase (P7)

The results indicate that the ideal team size preferred was 6 to 15 people (60%). Though 40% of users were in the favour of team size of 1 to 5 people. The next question was to ask about which release process should be used? The results are summarized in figure 13 below:

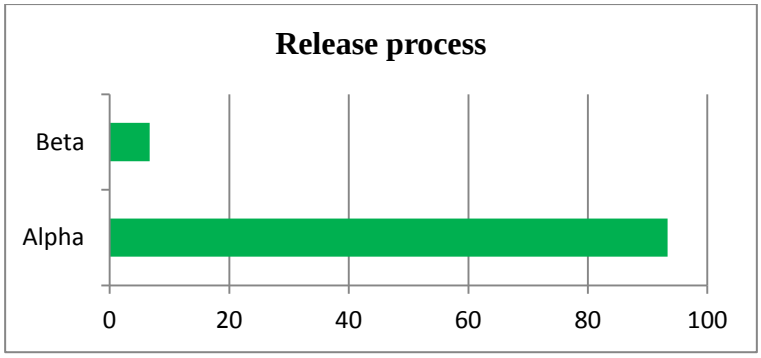

Figure 13: Planning Phase (P8)

As it is clear the preferred release is Alpha release. The next question is to check which plan for increment is best suitable. The results are shown in figure 14.

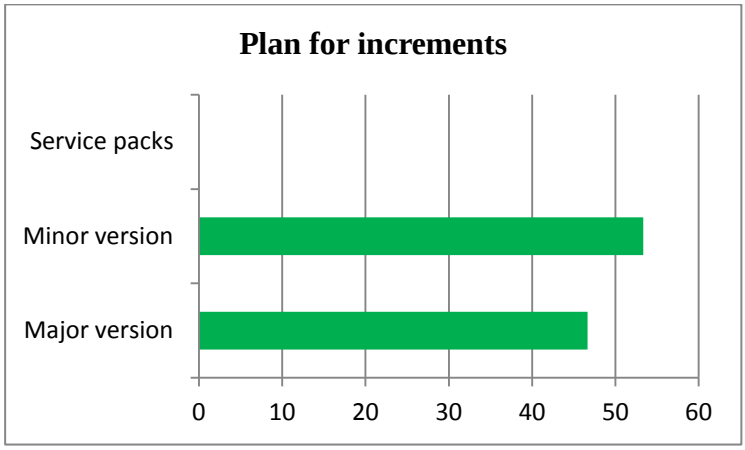

Figure 14: Planning Phase (P9)

No clear consensus was observed as 53% of users were in favor of minor revision while 47% were in the favor of major revision. The next question (P10) aimed at identifying the importance of the browser's compatibility which was found to be supported by all users.

5.2.3 Modeling/Designing Phase: First of all, all the users agreed that security is important for web application (M1). The next question aimed at identifying the priority of design in web applications. The results are shown in figure 15.

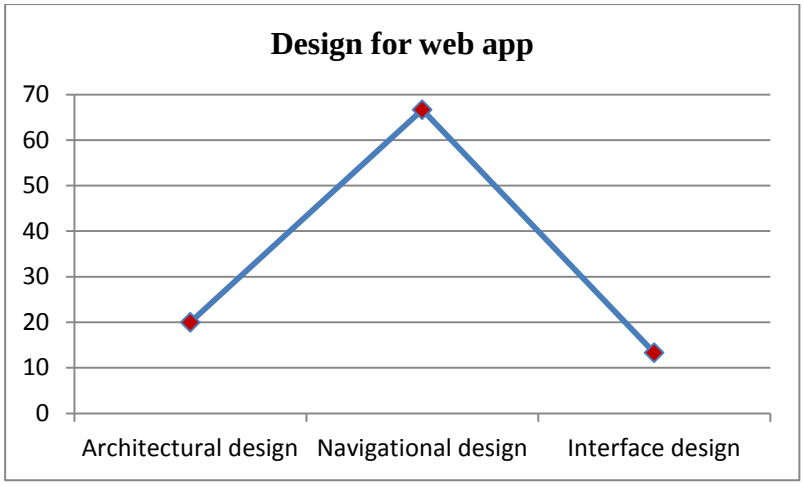

Figure 15: Modeling Phase (M2)

The highest priority was given to navigational design (67%), followed by architectural design (33%) and interface design (13%). The next question was related with navigational links. The users were asked whether they will prefer the new webpage to be opened in a new window, new tab or same tab. The results are shown in figure 16.

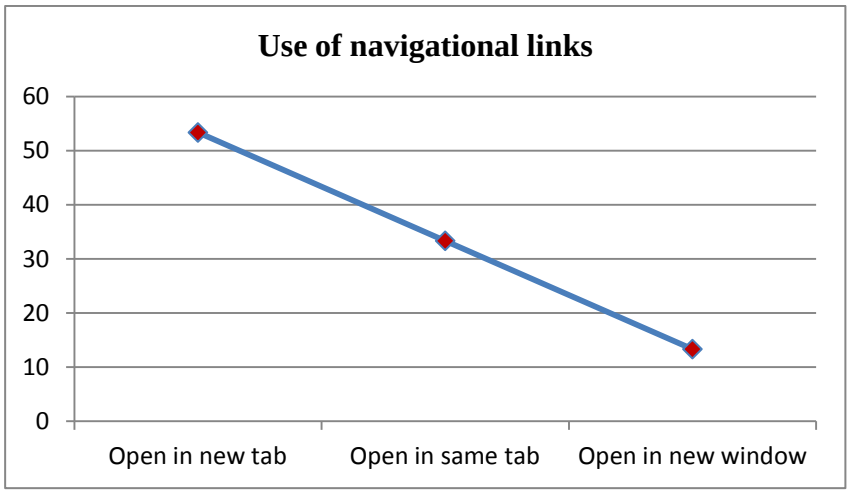

Figure 16: Modeling Phase (M3)

53% of users opted for opening the link in a new tab while 33% opted that the link should open in the same tab. Only 13% opted that the link should open in a new window. Almost 99% users agreed that the users should not be forced to navigate in a particular direction only (M4). All the users agreed that the navigation should be done according to user's expected behavior (M5). The next question aimed at identifying the priority given to various diagrams for design of web applications. The results are summarized in table 5.

Table 5 : Priority of Diagrams for Design (M6)							
	Data flow Diagram	E-R Diagram	UML Diagrams	Flow charts	Control flow Diagram	Decision tables	Decision trees
1	4	5	7	1	6	2	3
2	1	2	3	4	5	6	7
3	4	5	7	1	6	2	3
4	4	5	7	1	6	2	3
5	4	5	7	1	6	2	3
6	2	3	4	1	5	6	7
7	4	5	7	1	6	2	3
8	3	2	4	1	5	6	7
9	2	3	4	1	5	6	7
10	4	5	7	1	6	2	3
11	2	3	4	1	5	6	7
12	3	2	4	1	5	6	7
13	4	5	7	1	6	2	3
14	1	6	7	2	4	3	5
15	2	1	3	4	5	7	6
Sum	44	57	82	22	81	60	74

The results indicate that the order of priority followed by designers is flow charts, dataflow diagrams, ER diagrams, decision tables, decision trees, control flow diagram and UML diagrams.

5.2.4 Construction Phase: The first three questions during construction phase indentified whether the content provided was consistent, was the webapp extensible and whether the table organization understandable and efficient. All the users responded positively for extensibility of webapp and efficiency of table organization. They deviated in consistency of content. The results are shown in figure 17.

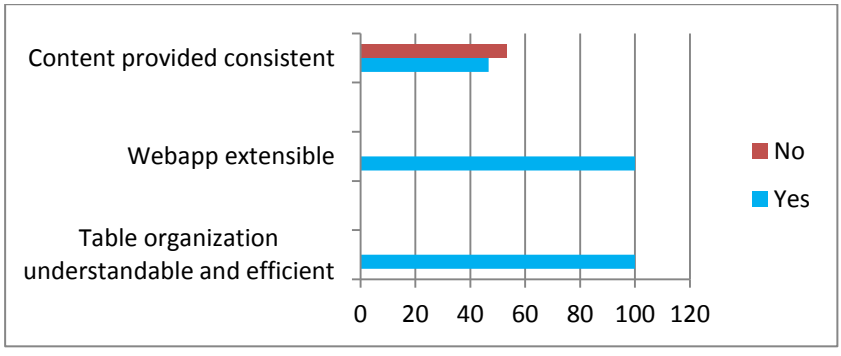

Figure 17: Construction Phase (C1-C3)

All the users agreed that rigorous testing is needed for web application development (C4). They all reported the tool used was Dreamweaver (C5). In the next question, the users were asked about the development mode was in house or it was outsourced? The results are summarized in figure 18.

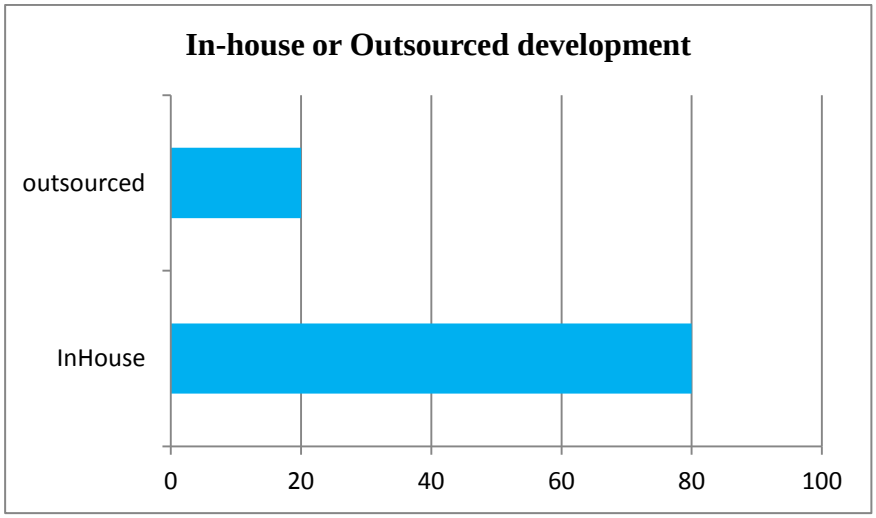

Figure 18: Construction Phase (C6)

80% users reported in house development while 20% reported outsourcing. In the next question, the users were asked about the efficient way of construction of web application. Development from scratch was reported as more efficient way (53%) as compared to development with reuse (47%) as shown in figure 19 below.

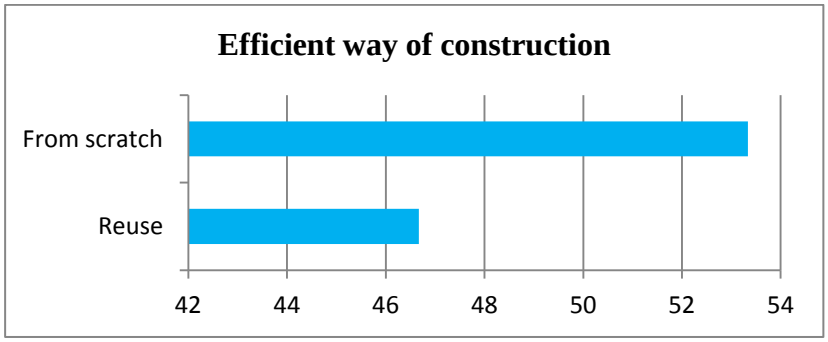

Figure 19: Construction Phase (C7)

5.2.5 Delivery, Feedback Phase and Umbrella Activities: All the users agreed to the two questions in this section namely the process should be extensible and user requirements are the factors driving the increments (F1 and F2). All the users agreed that the customer should be fully satisfied (F3). Regarding the testing of web applications, 80% of users gave priority to unit testing while

13% gave priority to integration testing. Only 7% gave priority to configuration testing. The results are shown in figure 20.

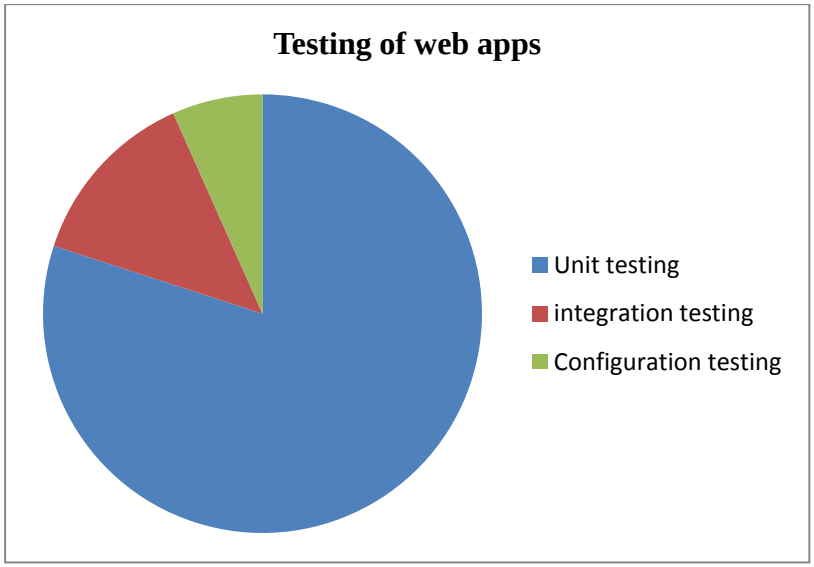

Figure 20: Delivery and Feedback Phase (U1)

The next question aimed at identifying the measures for quality. ISO standards were accepted by 93% of users while 7% accepted the quality planning and control. The results are summarized in figure 21

.

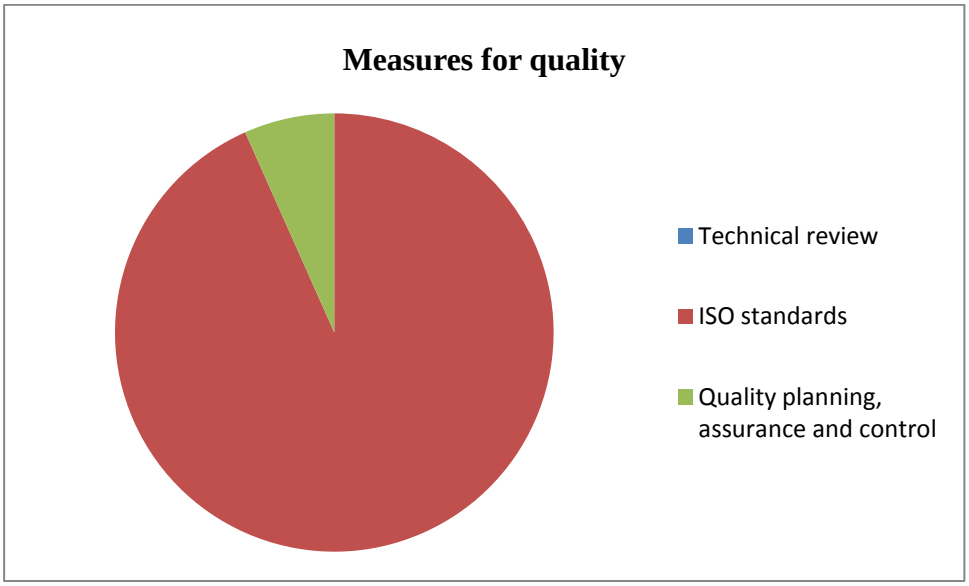

Figure 21: Delivery and Feedback Phase (U2)

Chapter 6

Prioritized Search Tool

COTS stand for Commercial off the Shelf components. Using one or more COTS products has effects on nearly all activities and products of the software process: architecture and design, effort and cost estimation, validation and testing, and reliability [30]. Growing research is boosting and improving the knowledge about COTS. These components are replaceable parts which are a complete package of interfaces, functionality, implementation details and source code. “Pick and fix” method is used in reusing the COTS components. A COTS component is chosen “pick” among the probable candidates and then integrated “fix” into the application built.

The components are built for the purpose of reuse. Reuse is done to reduce effort and cost of development. Sometimes, for the sake of reuse a component is integrated. But, reusing an un-appropriate component leads to system failures, increased no. of errors, cost and effort. To save our self from these situations, choosing an appropriate component is very much a need of the hour. A question arises how to choose the best possible component? To answer this, new priority based searching technique is proposed which will suggest the best component from the suitable candidates based on the priority assigned by the person who is searching for the components. A developer would best fit in the role of the person searching for components because (s)he is the one who will eventually reuse the component in the application (s)he is building.

6.1 Priority Searching

There are a number of searching techniques available for component selection. A tool has been designed to suggest components on the priority basis. In this tool, the person who is searching for the component needs to fill a simple form, filling the details about what type of component (s)he wants. During the form filling process he will be provided with an option of prioritizing the search for the components he needs. The

searching will be done giving preferences to the factors. The search results will be sorted according to the priority and an effective list of components will be retrieved from the database or the repository where the info about the components are stored. The use of prioritizing is discussed below.

6.2 Working of Tool

6.2.1 Need of Prioritizing

How prioritizing is concerned with the searching of the component? This may be answered as a component would be searched on the basis of criteria's filled in the form. It may happen that there is no component present in the repository or library which matches with all the criteria's filled in the form. Then the choice of priority filled against the particular field by the user plays an important role. The field with the lowest priority will be removed i.e. the constraint of searching by the specific keyword in that particular field will be removed, leaving that field open for other keywords also. The searching will again start but with one less constraint. The working of this process is like a loop. This loop is iterated until we get an appropriate component. Advantage of this loop can be seen when this loop will end and a suitable and appropriate, according to the user's need COTS component is suggested.

6.2.2 Webpages of Tool

6.2.2.1 User Login/ Welcome Page: The user has to login every time (s)he wish to access the Web App and take advantage of a prioritized component search. Figure 22, shows the login area where the user has to enter his authenticated username and password to get access to the system. The user must also select the account type either user or admin. By selecting admin account type and entering authenticated username and password, the user is directed on admin page where he can add, delete and update components. By selecting the account type user and providing valid username and password, the user will be redirected on the home page.

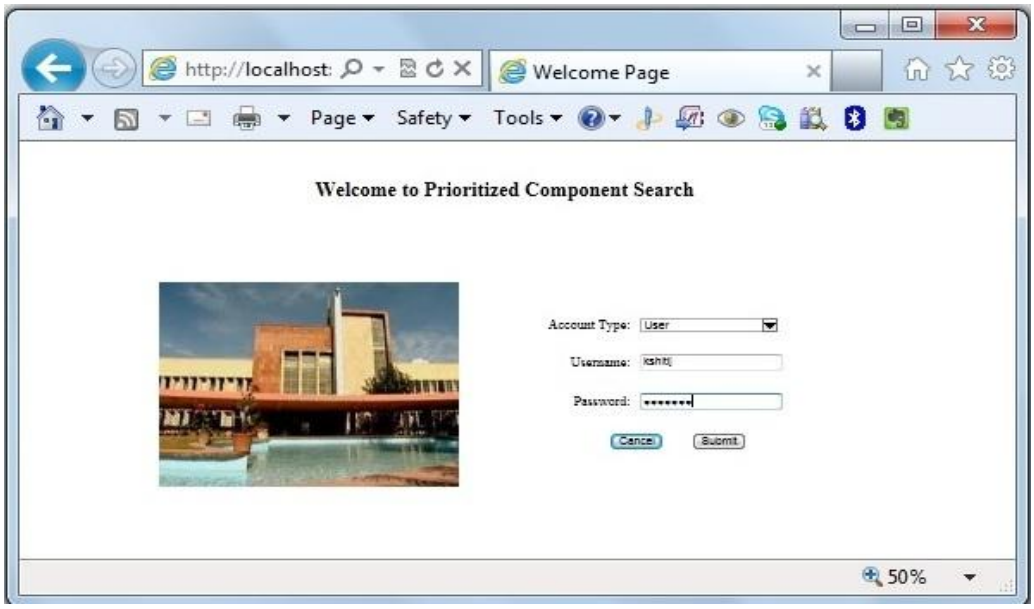

Figure 22: Snapshot of Welcome Page.

6.2.2.2 Home Page/ Index Page: This is the home page of the Web App. It is provided with information on basic layout of all the other pages contain. A user manual is also provided to help user better understand the system and can use it to his maximum advantage. The links of pages are put in a linear manner, emphasizing on each web engineering phase as shown in figure 23.

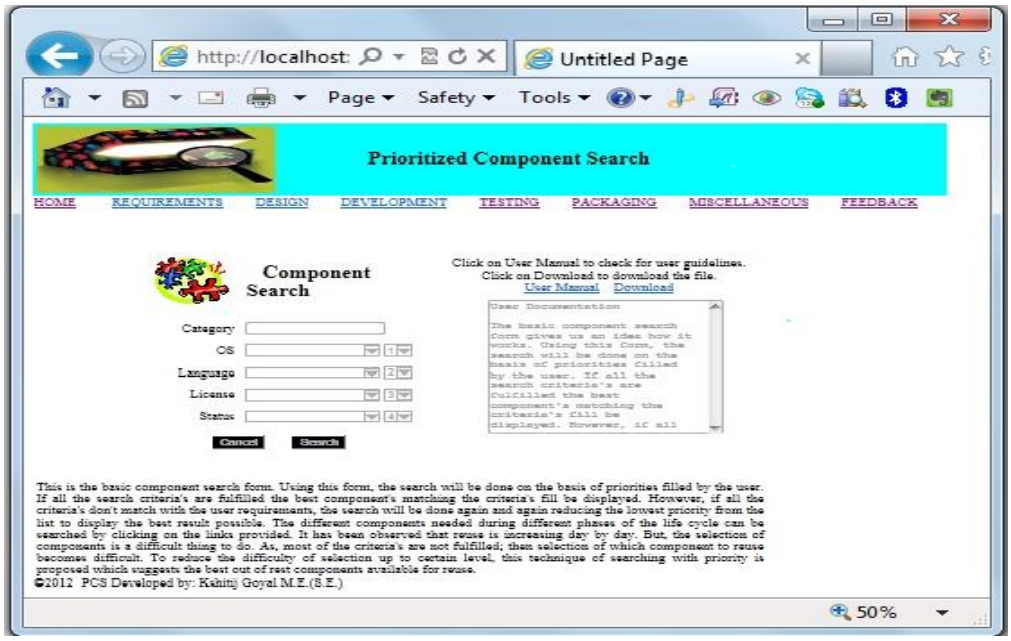

Figure 23: Snapshot of Home/Index page

6.2.2.3 Requirements Page: The requirements are changing rapidly. To inhabit reuse, certain common requirements have been gathered, collected and analyzed to make them general and widely acceptable. These requirements can be reused later according to the domain they belong to as shown in figure 24.

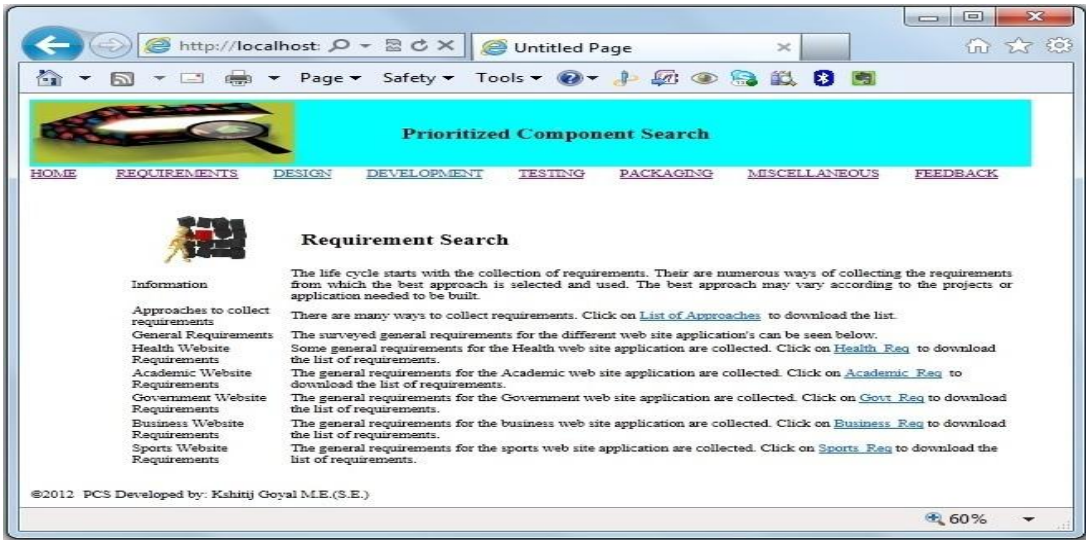

Figure 24: Snapshot of Requirements Page

6.2.2.4 Design Page: The components are searched according to the priorities user fills against the specifications given in dropdown boxes. As shown in figure 25 the specifications are chosen and the required priorities are filled against the specification. As the user clicks on submit, a priority based query is formed which generates the best possible result with accuracy.

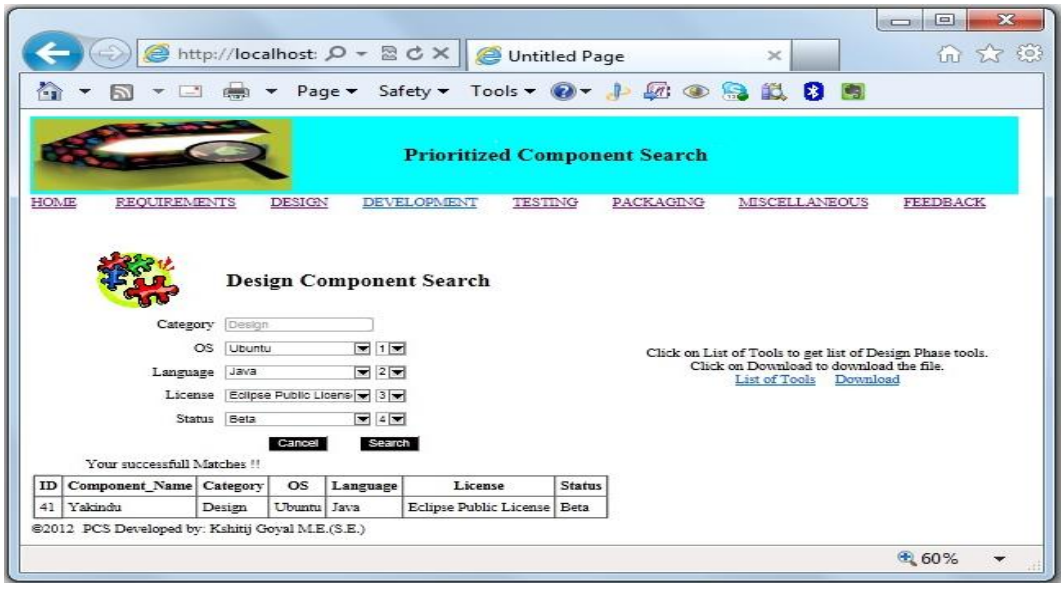

Figure 25: Snapshot of Design page.

6.2.2.5 Development Page: This page used for searching of development components. As shown in figure 26 the user fills the form and searches for development tools. Additional list of tools is also provided. The file can be previewed on the right side of the page by clicking on list of tools and can be downloaded by clicking on the download button.

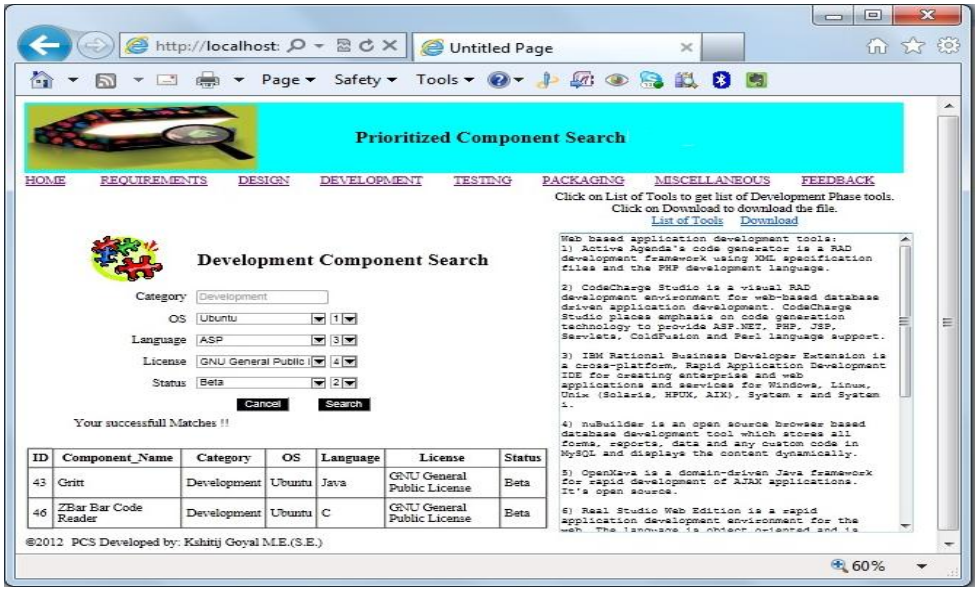

Figure 26: Snapshot of Development page

6.2.2.6 Testing Page: This page provides testing and layout checklist with searching components. As shown in figure 27 the form contains questions on the layout and can be used as testing layout questionnaire.

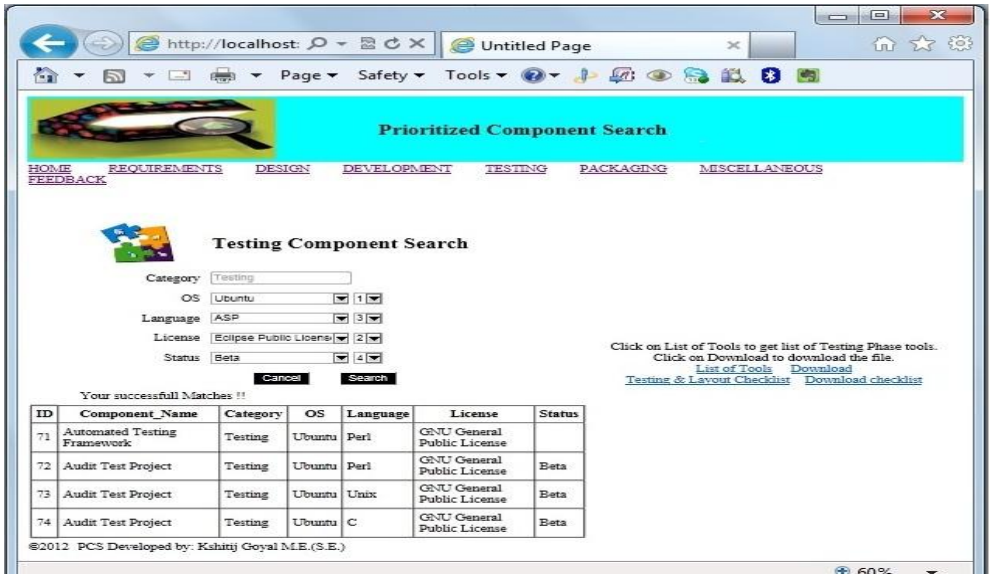

Figure 27: Snapshot of Testing page

6.2.2.7 Packaging Page: The different packaging components can searched which can help user in simplifying the packaging process. The list of tools is also given with the option of download as shown in figure 28.

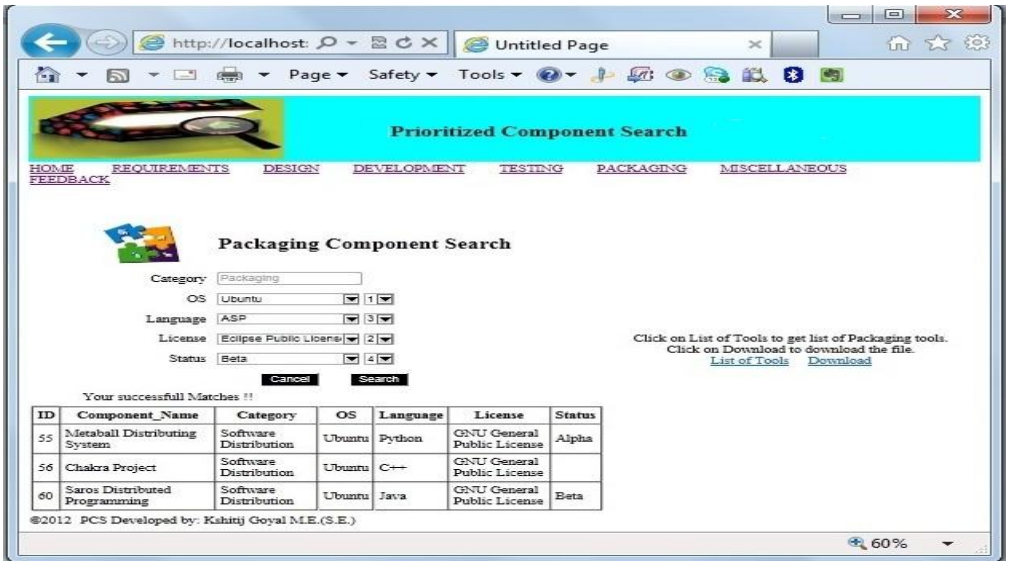

Figure 28: Snapshot of Packaging Page

6.2.2.8 Miscellaneous Page: This page is developed to facilitate the users in finding the components even when they are not clear about what type of component they need. This page is shown in figure 29.

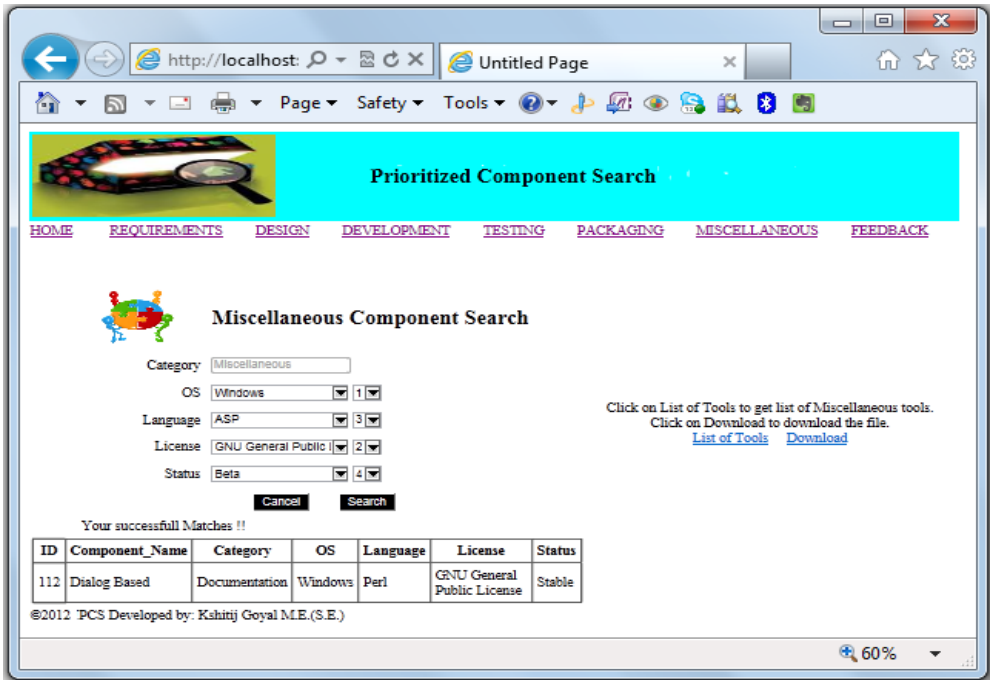

Figure 29: Snapshot of Miscellaneous Page

6.2.2.9 Feedback Page: This is a feedback page which is used to accept feedback from different users which may help the developers, analyzers, managers, *etc* to increase reusability in web engineering.

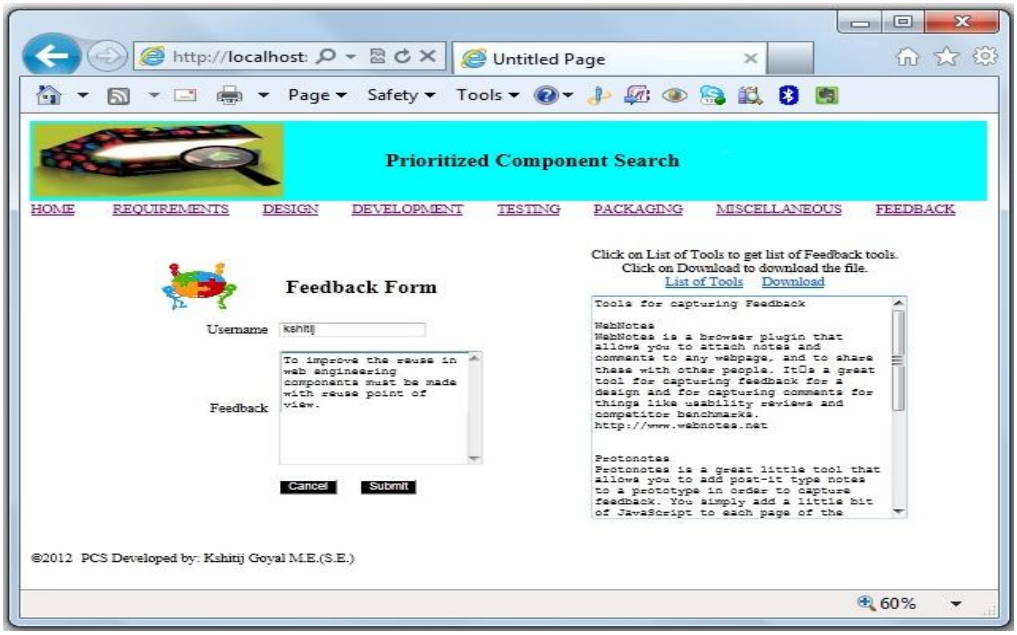

Figure 30: Snapshot of Feedback Page

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

Web engineering is important as the use of websites and Web Apps is increasing with use of Internet. In order to find the scope of reuse in web based applications, some of the challenges that must be handled effectively include rapid changing requirements, tailoring, updating, availability, security, *etc*. All should be considered critical in order to succeed in high level of reuse in development which can reduce the time, effort and cost by a large proportion and also improve the quality of the web based applications.

In reuse, the research is focus on two major areas: component retrieval systems, and enforcing reuse discipline in organizations. The .first focus identifies issues in reuse and second is efficient component retrieval system. Integrating the retrieval system with the development environment could greatly contribute to increasing developer's intention to reuse for web application development

However, developing tools to facilitate the reuse process is not enough; we also need tools to help managers enforce reuse in organizations. We are interested in developing such tools by expanding common software process models to include reuse practices. These models help developers with standards they need to follow when creating reusable components. Properly created components are, for example, easier to index and searched for in the repository.

7.2 Future scope

Web is evolving continuously. This will always cater for the need of something new to be invented. Work shall be done in future to make each and every process standardized. It shall strive to achieve the standards for identification of components, storage of components, evaluation of components, standard for documentation and knowledge support.

References

[1] Abran A., Moore J. W., Bourque P., Dupuis D., "Guide to the Software Engineering Body of Knowledge", IEEE Computer Society. pp. 1–1, 2004

[2] Laplante P., "What Every Engineer Should Know about Software Engineering", CRC Press, Inc. USA, 2007

[3] Kumar S., Sangwan S, "Adapting The Software Engineering Process To Web Engineering Process", International Journal of Computing and Business Research (IJCBR), Volume 2 Issue 1, pp. 2-3, 2011.

[4] Rode J. and Rosson M.B., "Programming at Runtime: Requirements & Paradigms for Nonprogrammer Web Application Development." In proc. IEEE HCC 2003, Auckland, New Zealand, Oct. 28-31, 2003.

[5] Ginige A., "Web Engineering: Methodologies for Developing Large and Maintainable WebBased Information Systems," in Proc. IEEE International Conference on Networking the Indiaand the World CNIW'98, Ahmedabad, India, Dec 1998.

[6] Ginige A, Murugesan S, "Web Engineering: An Introduction", J. of IEEE Multimedia, Vol. 8 Issue 1, pp. 14-18, 2001

[7] Pressman R.S., "Software engineering A Practitioner's Approach", 6th Ed. pp. 561-562, TMH.

[8] Silva M. and Werner C., "Packaging reusable components using patterns and hypermedia", In Proc. of the Fourth International Conference on Software Reuse, pp. 146–155, IEEE Computer Society Press, 1996.

[9] Prieto D. R., "Status Report: Software Reusability", J. of IEEE Software, Vol. 10 Issue 3, pp. 61-66, May 1993.

[10] Krueger C. W., "Software Reuse", J. of ACM Computing Surveys (CSUR), Vol. 24 Issue 2, pp. 131-183, June 1992.

[11] Selby R. W., "Enabling Reuse Based Software Development of Large Scale Systems", J. of IEEE Transactions on Software Engineering, Vol. 31 Issue 6, pp. 495-510, June 2005.

[12] Jaccheri L., Torchiano M., "Classifying COTS Products", In Proc. of the 7th Int. Conf. on Software Quality, pp. 246-255, London 2002.

[13] Morisio M., Torchiano M., "Definition and Classification of COTS: a Proposal", In Proc. Of ICCBSS 2002, LNCS 2255, pp. 165-175.

[14] Douglas K., Laurence B., "Applying social technical approach for COTS selection", In Proc. of 4th UKAIS Conf., University of York, April 1999.

[15] Janjic W., *et. al.,* "Lowering the Barrier to Reuse through Test-Driven Search", IEEE SUITE'09, May 16, Vancouver, Canada, 2009.

[16] Altarawneh H., El Shiekh A., "A Theoretical Agile Process Framework for Web Applications Development in Small Software Firms," in Proc. IEEE Software Engineering Research, Management and Applications (SERA'08), 2008.

[17] Constantine L.L., Lockwood L.A.D, "Usage Centered Engineering For Web Application", J. of IEEE Computer Society, Vol. 19 Issue 2, pp. 42-50, 2002.

[18] Schwabe D., Esmeraldo L., Rossi G., Lyardet F., "Engineering Web applications for reuse", J. of IEEE Multimedia, Vol. 8 Issue 1, pp. 20-31, Mar 2001.

[19] Johnson R. D., Reimer D., "Issues in the development of transactional Web applications", IBM Systems Journal, Vol. 43 Issue 2, pp. 430-440, April 2010

[20] Aoyama M., "Web Based Agile Software Development," J. of IEEE Computer Society, Vol. 15 Issue 6, pp. 56-65, Nov/Dec 1998.

[21] Kappel G., Michlmayr E., *et. al.*, "Web Engineering – Old Wine in New Bottles?", In Proc. of Int. Conf. on Web Engineering, Munich, July 28-30, 2004.

[22] Maras J., *et. al.,* "Reusing Web Application User-Interface Controls", lecture notes in Computer Science, Springer, pp. 228-242, 2011.

[23] Ahmad R., Li Z., Azam F., "Web Engineering: A New Emerging Discipline", In Proc. of Int. Conf. on Emerging Technology, IEEE 2005, Sept. 17-18, Islamabad.

[24] Jarzabek S., Pettersson U., "Cost-Effective Engineering of Web Applications," In Proc. of ICSE'06, Shanghai, China, pp. 1053-1054, May 20-28, 2006.

[25] Murugesan S., Deshpande Y., Hansen S. and Ginige A., "Web Engineering: A New Discipline for Development of Web-based Systems", pp. 1-9.

[26] Manola, F., "Technologies for a Web Object Model," J. of IEEE Internet Computing, Vol. 3 Issue 1, pp. 38-47, 1999.

[27] Hadjerrouit S., "Designing a Pedagogical Model for Web Engineering Education: An Evolutionary Perspective", J. of Info. Tech. Edu. Volume 4, pp. 115-140, 2005.

[28] Mili A., Mili H., "Reuse Based Software Engineering", Wiley, 2001.

[29] Steimann F., Vollmer H., "Exploiting Practical Limitations of UML Diagrams for Model Validation and Execution", pp. 1-2.

[30] Stroustrup B., "Language – Technical Aspects of Reuse", In Proc. of 4th Int. Conf. on Software Reuse (ICSR'96), pp. 11-16, 1996.

List of Publications

[1] Kshitij Goyal, Shivani Goel, "Attributes in Classification of Commercial Off The Shelf Components", In Proc. of IEEE-ICAER'11, UIET, Panjab University, Chandigarh, 2011.

[2] Kshitij Goyal, Shivani Goel, "Challenges for Reuse in Web Engineering", IRACST - Int. J. of Computer Science and Information Technology & Security, Vol. 1, No. 2, December 2011.

Appendix A

Questionnaire

The survey questionnaire is divided according to different phases of web engineering:

NAME:_____ORGANISATION:_____

DESIGNATON:_____CONTACT:_____

Question's on Requirement phase:

- **R1** What should be the level of clarity of requirements in the customer's mind when he comes to the developer for making the web app?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of clarity	1-Highest	2-Medium	3-Low
Ratings			

- **R2** How important is to find the user's background and sophistication?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of importance	1-Highest	2-Medium	3-Low
Ratings			

- **R3** How important is it to check the feasibility of the web application?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of feasibility	1-Highest	2-Medium	3-Low
Ratings			

- **R4** Using which of the models will make it easier for the developer to build the web app?
  Rate this as 1, 2, 3.

Methods/models	Rating
1-Waterfall model(Fixed requirements)	
2-Spiral model (New increments required)	
3-Agile method(Requirements changes frequently)	

- **R5** What are the priorities of the people/persons from whom we should collect the requirements for the web application?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level	1- Stakeholder/management	2- Developers	3- User's
Ratings			

- **R6** How important is to communicate with the customers with a scheduled chart?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of communication	1-Highest	2-Medium	3-Low
Ratings			

- **R7** What level of survey must be needed to cross check the priority of requirements and the need of those requirements?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of survey	1- Rigorous survey (New application to be developed)	2- Moderate Survey (New increment of web app)	3- No survey (Already collected requirements are sufficient)
Ratings			

- **R8** What should be the priority of the focus to handle the problems faced by the web users?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of focus Ratings	1-Extremely focused (Research will be done to find the solution.)	2-Focused (Negotiable)	3-Not at all focused (To deliver something new)

- **R9** What approach should be used in designing for a web app: top-down or bottom up or a combination of the two?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of survey	1-Top down	2-Bottom up	3-Combination of the two
Ratings			

- **R10** What is the main objective that the web app must full fill?
  Rate this as 1,2,3,4,5,6,7 **(**1 has the highest priority and 7 has the lowest priority). According to the importance required in the field of web application.

Academics

Objectives	Ratings
User friendly	
Simple design	
Security	
Extensibility	
Customizability	
Detailed information	
Media content	

Sports

Objectives	Ratings
User friendly	
Simple design	
Security	
Extensibility	
Customizability	
Detailed information	
Media content	

Health

Objectives	Ratings
User friendly	
Simple design	
Security	
Extensibility	
Customizability	
Detailed information	
Media content	

Government

Objectives	Ratings
User friendly	
Simple design	
Security	
Extensibility	
Customizability	
Detailed information	
Media content	

Business

Objectives	Ratings
User friendly	
Simple design	
Security	
Extensibility	
Customizability	
Detailed information	
Media content	

Banking

Objectives	Ratings
User friendly	
Simple design	
Security	
Extensibility	
Customizability	
Detailed information	
Media content	

Question's on planning:

- **P1** In your opinion, what is the importance of the planning phase in web engineering?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of importance	1-Extremely important	2-Important	3-Un-important
Ratings			

- **P2** What should be the priority of the factors which must be considered during the planning phase?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Factors	1-Estimation (cost, time, effort)	2-Quality attributes (usability, efficiency, maintainability, security)	3-Risks
Ratings			

- **P3** In your opinion, how should the technical specifications be needed?
  A. Overview. B. Descriptive. C. Proprietary
  Overview: basic info about tools and tech. to be used, how many members will use.
  Descriptive: defines details about tools and tech, who will use it, how these will be used.
  Proprietary: states the info. about the actual model used, no. of product used, etc.
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Technical specification	1-Overview	2-Descriptive	3-Proprietary	Any Other, please specify _____
Ratings				

- **P4** What is the priority of different ways or objectives which can help during planning phase.
  Objectives: customer satisfaction, risk mitigation, time management, etc.
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Objectives	1-Customer satisfaction	2-Estimation	3-Risk management
Ratings			

- **P5** Is it important to plan for the risks and their mitigation?
  Advantage: already identified, knows severity, what actions to take, able to avoid fire-fighting approach
  Disadvantage: may waste time, wrong identification or severity leads to problems, extra resources will be needed, etc.
  a. Yes b. No. Ans._____
- **P6** How much is it feasible to break the task into small milestones??
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of feasibility	Extremely feasible	Feasible	Not feasible
Ratings			

- **P7** What should be the number of team members needed during planning?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

No. of team members needed	1-5	6-15	Above 15
Ratings			

- **P8** What should be the release process?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Release process	Alpha	Beta	Relase to market
Ratings			

- **P9** What should be the plan for increments?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Incremental plan	Major version	Minor version	Service packs
Ratings			

- **P10** How much the browser's compatibility is important?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of importance	Extremely important	Important	Unimportant
Ratings			

Question's on modeling/ designing:

- **M1** What is the importance of security needed according to the web app?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of importance	Extremely important	Important	Unimportant
Ratings			

- **M2** Arrange with priority, the architectural, interface & navigation design for web app?
  Architectural design: overall hypermedia structure for web app.
  Navigation design: navigational flow between objects and all webapp functions.
  Interface design: screen layout, modes of interaction.
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Design	Architectural design	Navigational design	Interface design
Ratings			

- **M3** How do you prefer to use the navigational links?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Links	Open in new window	Open in new tab	Open in same tab
Ratings			

- **M4** Should we emphasize to force users to navigate in a particular direction (ordering of web pages)?
  a. Yes b. No Ans._____
- **M5** Should navigation be designed for the most commonly expected user behaviors? (Example: Forward navigation and backtracking, clicking on button to see drop down menu, etc.)
  a. Yes b. No Ans._____
- **M6** Prioritize these in terms of their importance to use : Data flow diagrams, E-R diagram, Decision table and decision trees, UML diagram, ?
  Rate this as 1, 2, 3, 4, 5, 6, 7. (Priority: 1. Highest and 7. Lowest)

Diagram	Data flow diagram	E-R diagram	UML Diagrams	Flow charts	Control flow diagram	Decision tables	Decision trees
Ratings							

Question's on Construction phase:

- **C1** Are the tables organized and sized in a manner that makes them understandable and displayed efficiently?
  a. Yes b. No Ans._____
- **C2** What is the most efficient way to get the desired results in a web application?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Factors	Reuse	Start from scratch	Reuse without analyzing
Ranking			

- **C3** Is the webapp extensible?
  a. Yes b. No Ans._____
- **C4** Is the content provided is consistent?
  a. Yes b. No Ans._____
- **C5** How much testing is needed?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of testing	Rigorous	Testing needed	Not needed
Ratings			

- **C6** What tools and technologies you prefer?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Tools and technologies	Dreamweaver (CSS)	Joomla(CMS)	FrontPage(WYSIWYG)
Ratings			

- **C7** What do we prefer In-house or outsourced development?
  In-house: less costly, possess skills, have time, resources
  Outsourced: no time to train people, don't have the skills required, resources, etc.
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Development	In-house	Outsourced	Sometimes both
Ratings			

Question's on Delivery and feedback phase:

- **F1** Is the process extensible?
  a. Yes b. No Ans._____
- **F2** Which of the following factors would drive the increments?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)
  User requirements: solving the problems of the customers.
  New research: giving something new, never seen before.
  Competitive: adding functionalities to remain in the market.

Factors	User Requirements	New research	competitive
Ratings			

- **F3** How much the customer is satisfied??
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Level of satisfaction	1-Fully satisfied	2-Satisfied	3-Unsatisfied
Ratings			

Question's related to the Umbrella activities:

- **U1**How should the testing of web apps must be done?
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Testing	Unit testing	Integration testing	Configuration testing
Ratings			

- **U2** How the quality shall be maintained throughout all the phases??
  Rate this as 1, 2, 3. (Priority: 1. Highest, 2. Medium, 3. Low)

Technique	Use technical review methods	Follow ISO standards	Quality assurance, planning, control
Ratings			