A

Dissertation

On

Web Application Testing

Submitted in partial fulfilment of the requirement

for the award of

MASTER OF TECHNOLOGY

(SOFTWARE ENGINEERING)

Submitted by:

SANSKRITI MADAN

Roll No.:- 2K12/SWE/23

Under the guidance of:

Ms. ABHILASHA SHARMA

Delhi Technological University

Department Of Computer Engineering

Delhi Technological University

2012-2014

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY
(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI-110042

Date: ____________________

This is to certify that the thesis entitled ***Web Application Testing*** submitted by Sanskriti Madan *(2K12/SWE/23*), in partial fulfilment of the requirements for the award of degree of Master of Technology in Software Engineering, is a work carried out by her under my guidance.

Mrs. Abhilasha Sharma
Assistant Professor
Department of Computer Engineering
Delhi Technological University
Delhi

Signature……………………………….

ACKNOWLEDGEMENT

I take this opportunity to express my deepest gratitude and appreciation to all those who have helped me directly or indirectly towards the successful completion of this thesis.

Foremost, I would like to express my sincere gratitude to my guide ***Mrs. Abhilasha Sharma***, *Assistant Professor, Department of Computer Engineering, Delhi Technological University, Delhi* whose benevolent guidance, constant support, encouragement and valuable suggestions throughout the course of my work helped me successfully complete this thesis. Without her continuous support and interest, this thesis would not have been the same as presented here.

Besides my guide, I would like to thank the entire teaching and non-teaching staff in the Department of Computer Engineering, DTU for all their help during my course of work.

SANSKRITI MADAN

Master of Technology

(Software Engineering)

College Roll No. 2K12/SWE/23

Department of Computer Engineering

Delhi Technological University

Bawana Road, Delhi-110042

ABSTRACT

The rapid diffusion of Internet and open standard technologies is producing a significant growth of the demand of web sites and web applications with more and more strict requirements of usability, reliability, interoperability and security.

While several methodological and technological proposals for developing web applications are coming both from industry and academia, there is a general lack of methods and tools to carry out the key processes that significantly impact the quality of a web application (WA), such as the Validation &Verification (V& V), and quality assurance. In this thesis, an approach WSDM(Web Site design Method) being used for constructing web applications and web sites is modified by integrating Agile methodology to it. The basis for this integration is to enhance testing of web applications so that more reliable, maintainable, flexible and effective framework shall be proposed.

The new proposed model i.e, combination of WSDM and AGILE TESTING shall perform testing with simultaneous execution of development phases. This agility will make the WSDM approach flexible enough to incorporate changes at any phase as methodology possesses the ability to successfully deliver result quickly and inexpensively in short intervals of time.

Furthermore, in this thesis a comparative analysis is drawn between the Proposed model and a model proposed by some other author using a case study. A case study, carried out with the aim of assessing the effectiveness of the proposed method and tools, produced interesting and encouraging results.

List of Figure(s)

Fig. Number	Description	Page number
Fig.1.1	Generic view of Web Application	2
Fig. 2.1	WSDM Process	9
Fig. 3.1	V- Model	15
Fig. 3.2	W- Model	17
Fig. 3.3	Fish Model	18
Fig. 3.4	Manifesto for Agile Software Development	21
Fig. 4.1	Integrating V- Model into WSDM Approach	28
Fig. 5.1	V- Model Vs Agile Methodology	33
Fig. 5.2	Integration of Agile with WSDM	35
Fig. 6.1	Job Portal using WSDM- V model	42
Fig. 6.2	Job Portal using WSDM Agile	43

List of Table(s)

Table Number	Description	Page number
Table3.1	Choice of Testing Models Based on some Parameters	25
Table 6.1	Summarised Estimation Analysis of the Job Portal system	44
Table 6.2	Comparative Analysis	44

Table of Contents

Certificate…………………………………………………………………….…[i]

Acknowledgement……………………………………………………………....[ii}

Abstract…………………………………………………………………….….[iii}

List of Figure(s)……………………………………………………….…...…[iv]

List of Table(s)……………………………………………………………….…[v]

1. **INTRODUCTION**……………………………………………………….. 1

1.1 Web Application……………………………………………………. 2

1.2 Web Application Development Process………………………………….. 3

1.3 Significance of Testing in WEB………………………………………. 4

1.4 Motivation…………………………………………………………... 5

1.5 Organization of the Thesis…………………………………………….. 5

2. **WEB SITE DESING METHOD**………………………………………….. 6

2.1 Background………………………………………………………… 7

2.1.1 WSDM (WEB SITE DESIGN METHOD)…………………. 7

2.2 Advantages and Pitfalls in WSDM…………………………………….12

3. **BACKGROUND ON TESTING MODELS**…………………………...13

3.1 Web Application Testing..…………………………………………..... 14

3.2 V-Model………………………………………………………… 14

3.3 W-Model………………………………………………………... 16

3.4 Fish Model……………………………………………………….18

3.5 Agile Methodology…………………………………………………19

3.5.1 Manifesto for Agile Software Development………………..21

3.5.2 Why Agile? ....……………23

3.5.3 Agile Methodologies……………………………………….23

3.5.4 Agile for Web……………………………………………25

3.6 Choice of Testing Models Based on some Parameters………………….25

4. **LITERATURE REVIEW**……………………………………………..26

4.1 Introduction……………………………………………………. 27

4.2 Review of Related Work………………………………………………..27

5. **RESEARCH METHODOLOGY**……………………………………...30

5.1 Introduction …………………………………………………………31

5.2 Agile Testing over other Approaches………………………………….. 31

5.2.1 Agile Vs V-Model………………………………………32

5.2.2 Agile Vs W-Model………………………………………33

5.2.3 Agile Vs Fish Model…………………………………….33

5.3 Proposed Model: “AGILE –WSDM “Approach………………………..34

6. **DISCUSSION AND RESULTS**…………………………………….37

6.1 Introduction……………………………………………………….. 38

6.2 Effort Estimation…………………………………………………...39

6.2.1 COCOMO Estimation for WSDM - V-Model ………….39

6.2.2 SCRUM Estimation for WSDM – Agile ……………..…40

6.3 Deployment of both the Models…………………………………42

6.4 Comparative Analysis……………………………………………44

CONCLUSION...47

References...49

CHAPTER 1

INTRODUCTION

1.1 Web applications

The rapid growth of Internet and open standard technologies has created a significant growth of the demand of web sites and web applications with strict emphasis on usability, reliability, interoperability and security. They have become the crucial vehicles for commerce, information exchange, and a host of social and educational activities. This increase in demand has imposed high pressure for developing and incorporating new technologies. As a result, web sites and applications were developed "naively". This rapid evolution actuated the need for efficient methodologies, tools and techniques for developing applications that fulfils the desired quality requirements.

Compared with the traditional software, the Web applications have many distinct properties[1]: firstly, because of the easy accessibility to information, the Web applications(WA) have a huge and varied user population, thus loading the server's performance and the ability of dealing with concurrent transactions, secondly, the architecture necessitate the Web applications to fit for the heterogeneous and autonomous environments, thirdly, WA mainly focus on the information search and index, so they have weaker functions but quicker updating rates in their contents and techniques, comparing with the traditional ones. Thus, extra efforts are needed in for its testing.

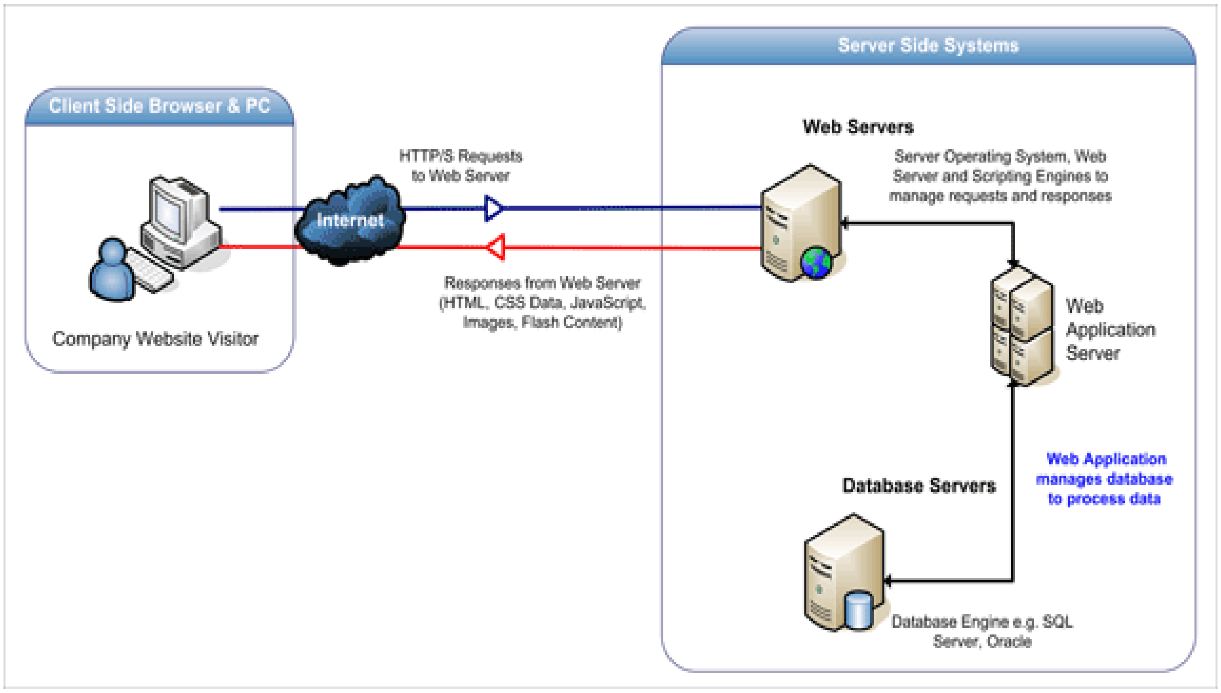

Client Side Browser & PC
Company Website Visitor
Internet
HTTP/S Requests to Web Server
Responses from Web Server (HTML, CSS Data, JavaScript, Images, Flash Content)
Server Side Systems
Web Servers
Server Operating System, Web Server and Scripting Engines to manage requests and responses
Web Application Server
Web Application manages database to process data
Database Servers
Database Engine e.g. SQL Server, Oracle

Figure 1.1 Generic view of Web Application

Web application[19],[20] uses a web browser as a client which request/post some data from/on server which again may be centralized or distributed in order to improve server response time. Client is a computer software application installed at users' side which runs web application, developed in a browser supporting language (such as javascript combined with HTML). Figure 1 represents a very generic view of web application.

Here are some types of web applications:

- *Analysis Campaign* - Create advertisements and track them in order to produce the results for analysis of the product. Analysis report may exist in form of various types of graphs and charts.
- *Order Goods Online* - There are many websites available for online purchasing like flipkart, snapdeal, myntra, amazon etc.
- *Estimate* - Analysis of user behavior is performed on various sites in order to get current user trends and taking business decisions accordingly. Example - Rossiter and Co – video estimate.
- *Educational Sites* - There are many sites running successfully providing educational material for students of various subjects. Example - w3schools, APSA etc.
- *NEWS Sites* - This is very popular and useful kind of websites providing information of happenings across the whole world. Each news channel has its own website telecasting their news bulletin over internet. Example - BBC news, NDTV etc.
- *Searching Sites* - Google, yahoo, bing etc. are some well-known name when one desires to search on internet regarding some specific website.
- *Travelling Sites* - These sites provide information about various convince like trains, buses, flights etc., also gives facility to book ticket online. Example - makemytrip, IRCTC etc.
- *Social Sites* - These sites provide a platform for internet users to connect with each other sharing their information relating to the purpose of the website. For example, *Facebook* provides users a way to share their thoughts with each other.

1.2 **Web Application Development Process**

According to Nancy L. Russo et al. *Application development methodologies are promoted as a means of improving the management and control of the application development process, structuring and simplifying the process, and standardizing the development process and product by specifying activities to be done and techniques to be used* [28].

Geoffrey Elliott defined development methodology as noun *An application development methodology is a framework that is used to structure, plan, and control the process of developing an information system- this includes the pre-definition of specific deliverables and artefacts that are created and completed by a project team to develop or maintain an application* [2].

Using application development methodologies beneficiate the application to be developed in following ways:

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

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

3. *A proper guideline of how to do* - Now we have well known documented requirements set for starting development of application. But next thing is that how to formalise development in proper steps and manner so that minimal risk would be associated with optimal benefit. Methodologies provide a way for this according to what we want to develop and what will be its running environment.

4. *Optimal formation of teams and distribution of work* - Randomly assigning the work to random people cannot lead to optimal use of workers according to their strengths. It is must to have maximum use of man power in order to get efficient and rapid product output.

5. *Minimal risk with optimal benefit* - Methodologies do not only provide ways for efficient development of project but also guides after the development in terms of deployment and maintenance plans.

1.3 **Significance of Testing in WEB**

Development of Web Applications and web sites have become the mainstream business for several companies in major market areas. Consequently, the quality of WA is getting crucial.

WA and web sites have heterogeneous and autonomous environment. It is multi-platform, interactive and highly dynamic, thus making its development and maintenance highly strenuous unlike the traditional ones making it more complex.

The goal of testing is systematical detection of different classes of errors in a minimum amount of time and with a minimum amount of effort thus providing a more reliable and a better quality product. Complete web testing of a system before going live is the primary step to get assured of an entire web application’s ability to work properly. After performing web tests you’ll be able to find bottlenecks in your systems before they happen in a production environment. Testing should not be a one time job. Web system’s testing demands a continuous approach. A key to testing Web based systems is ensuring their functional quality, providing high security, seeing to the fact that the intended purpose for which it was developed has been met with user satisfaction.

Mostly ad hoc methods are used to achieve the quality of Web based applications. These methods and techniques are important to understand and improve the quality but due to the growing complex nature of Web-based applications more systematic methods and tools are required. Testing is the most applicable method to verify the performance and functional requirements of the applications. It is one of the most important processes (phase) in the software development life cycle because it ensures the quality, stability, adoptability, sustainability and acceptability of Web-based applications. In other words we can say that it include all verification and validation activities of the Web applications.

1.4 **Motivation**

As explained earlier, Web Applications are dynamic web sites combined with server side programming which provide functionalities such as interacting with users, connecting to back-end databases, and generating results to browsers. Despite the fact that Web Engineering has made remarkable advancement but still there occurs several flaws. As we deal with more of the online data, security, reliability and maintainability are some of the major issues that demand attention. Thus testing seems a necessary and crucial prerequisite for web based systems.

A very famous approach "WSDM or Website Design Method" was being used for web application development process but it concentrated only on development criteria. Testing phase was lacking in WSDM. Due to this limitation, maintainability and flexibility issues arises.

Thus there arise the need of integrating Testing into Web Development process. An approach was presented by Tolba, R., Mushtaha, A, in which they integrated V-model with WSDM. V-Model is a Testing model used in SDLC. In this approach, testing was undertaken after the coding phase and no immediate deliverables were produced. In this era of changing user needs and requirements, deployment of this model was quite expensive and ineffective.

1.5 **Organisation of the Thesis**

This thesis consists of 6 chapters. The chapter deals with.

- **Chapter 1** introduces the theme of research work and introduction about Web Applications and its framework
- **Chapter 2** provides description about WSDM (Web Site Degign Method).
- **Chapter 3** provides Background on Testing Models.
- **Chapter 4** provides review of literature in order to create an adequate framework for conducting this research work.
- **Chapter 5** deals with the research methodology and explains the proposed model.
- **Chapter 6** includes discussions and results of the thesis work.

CHAPTER 2

WEB SITE DESIGN METHOD

2.1 Background

With the advent of web as *Web 1.0* in 1993, web development techniques have evolved tremendously over time and have grown exponentially. Because of the strong user base, and because of an inherent ease of publishing on the web, the number of web applications has grown at a significant pace. However, most of the web sites and web applications have been created opportunistically without prior planning or analysis. Due to which their development and maintenance has been an ad hoc process, leading to poor-quality of various applications. To an extent the reason behind this can be attributed to Web-based application development lacking a systematic engineering approach.

Web engineering is a systematic and quantifiable approach for development of web based systems and applications. Web site development discipline requires a sound web engineering methodology which includes various sub processes such as- modeling, design, implementation and maintenance.

A number of researchers have already recognised the lack of design methods for web sites, or in general for web-based information systems. They have proposed the following methods:

- HDM (Garzotto, Paolini and Schwabe, 1993) and its successors Garzotto, Paolini and Mainetti, 1993) and OOHDM (Schwabe and Rossi, 1995), (Schwabe et al., 1996; Schwabe and Rossi, 1998);
- RMM (Isakowitz et al, 1995);
- W3DT (Bichler and Nusser, 1996);
- The method for analysis and design of web sites in (Takahashi and Liang, 1997);
- SOHDM (Lee et al, 1998).

Older methods (HDM, RMM) were originally designed for hypertext or hypermedia applications and do not deal comfortably with web-specific issues. In addition, these methods are data-driven or implementation oriented. Some of these have their origin in database design methods like the E-R method (Chen, 1976) or object oriented (OO) methods such as OMT (Rumbaugh et al., 1991). These methods may solve maintenance problems to an extent but they do not succeed much in addressing the typical usability problems.

2.1.1 WSDM (WEB SITE DESIGN METHOD)

WSDM is developed by the Web & Information
Systems Engineering research group, WISE, at the Vrije Universiteit Brussel.
WSDM was introduced by De Troyer and Leune, 1998; (De Troyer, Goedefroy et al., 1998) for developing web sites. Back then, WSDM was the acronym that stood for Web Site Design Method and targeted only information-giving web sites. With the growth of the industry, methodology too has evolved to a great extent and even allows traditional web applications as well as semantic web applications to be designed. Therefore the acronym was renamed to **Web**

Semantics Design Method. Improved version of WSDM was mentioned by De Troyer (De Troyer,2001).

It enables development of web site and web applications in a systematic way. It also provides modeling primitives that allow a web developer to construct models that describe the web site/application from different perspectives and at different levels of abstraction.

WSDM[18] understands that web sites have different types of visitors who in turn have different needs. And fulfilling these needs leads to customized web sites, better usability and increased satisfaction. WSDM differs from traditional approaches as the older versions gave more importance to the organization’s database thus being termed as data-centric methodologies whereas WSDM is audience- driven. WSDM takes the needs and requirements of the intended user of the web site as the starting point. By explicitly starting from the requirements of the users or visitors, WSDM solves Web site issues that are primarily caused by that fact that a site has no underlying design at all.

WSDM[15] starts the design with the identification of the different audience classes, their needs and requirements. These different audiences and their requirements will drive the entire development process.

Authors De Troyer and Goedefroy gave the framework of WSDM which comprised of five phases. Figure below shows the five design phases.

Phase 1: Mission Statement Specification

The first step is to define the *mission statement* of the web site or web application. This phase determines the purpose and subject of the project followed by specification of the intended users, and declaring the target audience. It answers the following questions:

- *What is the **purpose** of the web site*?
- *What is the **subject?***
- *Who are the **target audiences?***

Without giving due consideration to the purpose of the web site, there is no proper basis for making decisions, or for evaluating the effectiveness of the web site.
To be able to address the target audience in an effective way, one needs to have a clear understanding of one’s target audience.

Phase 2: Audience Modeling Phase

In general, the mission statement only gives a general indication of the target audience of the site. E.g. for a commercial site, the target audience may be customers and potential customers. However this specification is too broad to decide whether this target audience should be segmented as one audience class or as two or more audience classes.

The audience classes are built as hierarchy. It consists out of two sub-phases, Audience Classification and Audience Class Characterization.

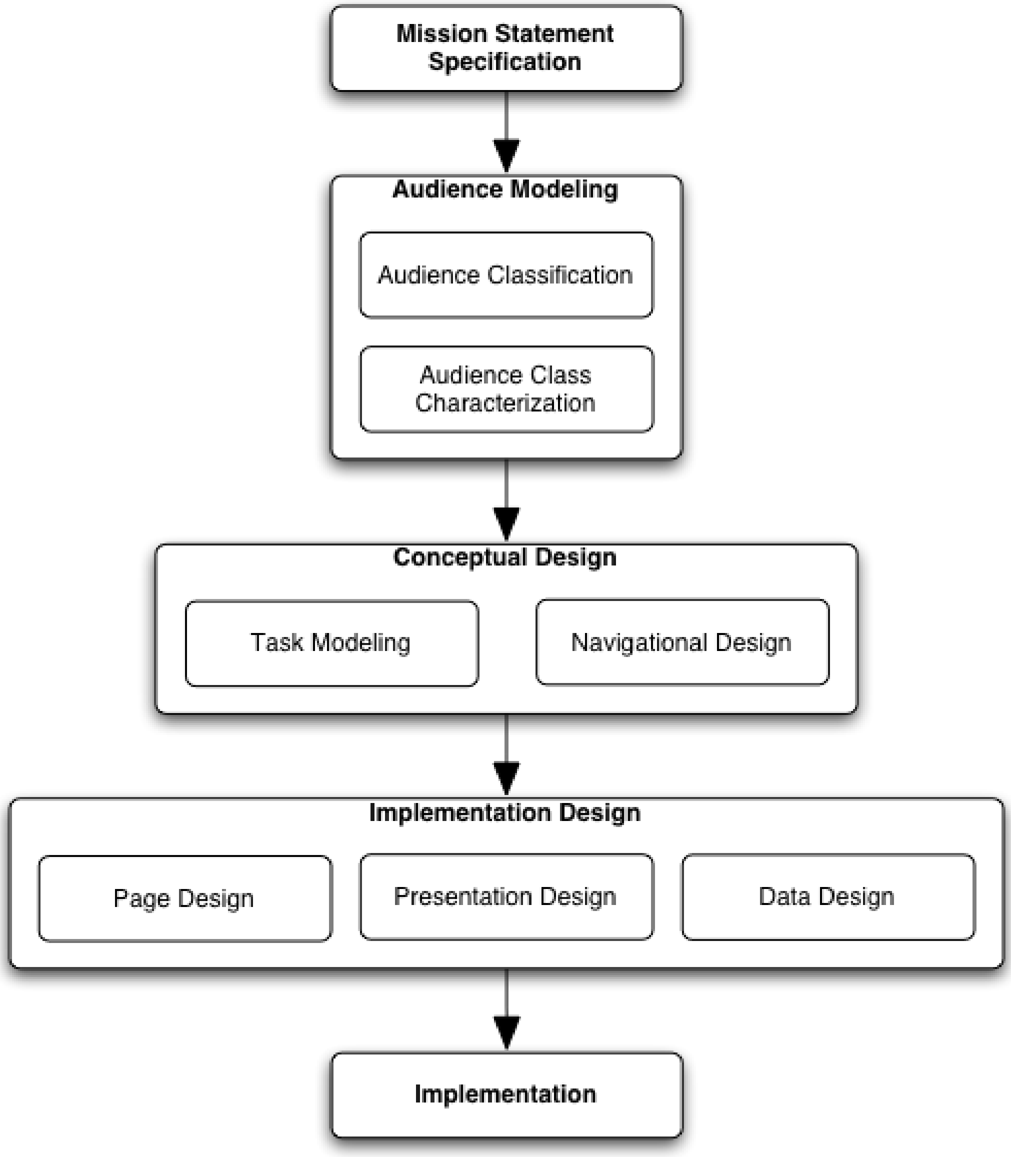


Figure 2.1 : WSDM Process

- Audience Classification

It basically deals with identification of Audience classes by following the steps :

Step 1 we are going to consider the activities of the organization related to the purpose of the web site.

Step 2 for each activity we have to:
1. Identify the people who are involved in the activity.

2. Restrict them to the target audience. So only consider those people which belong to the target audience formulated in the mission statement.
3. Divide them into audience classes based on different information or functional requirements.
4. Decompose the activity if possible.
5. Repeat Step 2 until no new audience classes are found.
6. *If decomposition of activities doesn't result into any new audience classes it should be stopped, because it is no longer useful.*

For each audience one must be aware of the functional requirements and information requirements.

- Audience Class Characterization.

As we know, all members of an audience class might have the same information and functional requirements but even then the members of one audience class may differ on how the information should be presented to them i.e their usability requirements may vary. If we can distinguish, within one Audience Class, groups of members with different characteristics, we can introduce audience class variants.

Phase 3: Conceptual Design phase

In Conceptual Design we describe the structure of the web site and model how the members from different audience classes will be able to navigate through the site. The goal of this phase is to concentrate on the conceptual "what and how" rather than on the visual "what and how". The conceptual "what" is covered by the Task & Information Modeling sub phase and deals with modeling of the content and functionality of the web system; the conceptual "how" is covered by the Navigational Design sub phase. This means in database design we describe not only what kind of information will be presented (objects, object types and relationships) (the conceptual "what"), but also we describe how the information will be structured and how it will be possible to navigate through the information (the conceptual "how"). This is required because navigating through the information space is an essential characteristic of web sites. If the navigation is not designed as per the target audience, serious usability problems be arise. The conceptual "what" is mainly covered by the information modeling step, the conceptual "how" by the navigation design.
Two sub phases are:

- *Task and Information Modeling*

The information modeling step is intended for the data intensive web sites. The purpose of the information modeling is to model the structured data offered by the web site. The reason to do this is the same as in database design. It offers structure and enhances the maintenance.
The purpose of the task modeling is detailed modeling so that the members of each audience class can perform different tasks and to formally describe the data and functionality that is required for those tasks. WSDM uses the *Concurrent Task Tree* technique for its task modeling

because it allows not only to sequentially decompose a task in subtasks but also to specify temporal relationships between different subtasks.

1. Define a task for this requirement.
2. Elaborate the task into more detail.
3. Decompose the task into elementary tasks (using Concurrent Task Tree).
4. *For each elementary task make an Object Chunk that models the information and/or functionality required by this task.*

- *Navigational design*

During this stage, we describe the structure of the web site and model how the members from different audience classes will be able to navigate through the site. For each audience class (variant) a different *navigation track* is created. All navigation tracks together form the *navigation model*. A navigation track is described in terms of *components* and *links*. Components represent units of information. Such a unit can contain text, pictures and some other kind of multimedia information as well as structured information modeled by means of the audience object models. Components are connected by means of links. We use links to model the structure of the web site but also to indicate the need for navigation links.

Phase 4: Implementation Design phase

The aim of this phase is to provide the conceptual design, with necessary implementation aspects. It consists of three sub phases.

- Page design
  In this sub phase the designer describes where the information on a page should be positioned and how it should be laid out. It is also decided how and where the links should be presented.

- Presentation Design
  This sub phase gives the 'look and feel' of the web site and the layout of the individual pages.

WSDM provides three different sets of modeling concepts to describe the layout of a page. Each set provides a different level of abstraction.

- *Primitive presentation concepts*: There are two kinds of such primitives, positioning elements and multimedia elements (Audio, Email, Image, Integer, String and Video).

- *High-level presentation concepts*: These express more high-level presentation concepts like lists, menus and sections, logos, banners, breadcrumbs, etc. These concepts are defined on top of the primitive presentation concepts.
- *Template concepts*: are defined on top of the primitive presentation concepts and meant to define templates that will be used for different types of pages.

- Logical data Design

The Data Design can be used to specify a logical data schema for a database using the Object Chunks specified in the Conceptual Design phase. If there is no need for a full-fledged database, logical schemas for XML DTD, RDF definitions, etc. can be produced here.

Phase 5: Implementation

This last phase of the methodology, Implementation, consists of realizing the physical Web Site or web application. Firstly, an implementation environment (HTML, XML) is chosen and then the result of the Implementation Design can be converted to the chosen environment. This can be automated depending on the complexity of the model and the availability of usable tools.

2.2 Advantages and Pitfalls in WSDM

WSDM is a web site design method based on a new approach, called audience-driven. This means that not the data available in the organisation or the available technology but the requirements of the target audience is the starting point of the modelling process. This approach prevent so-called “jumble sites” and “ego sites”. Jumble sites are sites that contain all information the developer could collect and that are organised in a way only obvious for the developer. Ego sites (Powell, 1998) are those sites that are built to satisfy the ego of the developer, not to fulfill user needs.

The method is based on the principle that the web site should be designed for and adapted to its target audience. We have also explained the need for a conceptual design phase in web site design similar to the conceptual design phase in database systems. Therefore, the emphasis in this method is on audience modelling and conceptual modelling rather than on layout and presentation aspects. In principle, an implementation can be generated once the conceptual design is completed. As a consequence of our audience-oriented approach, the conceptual information schema of a web site cannot be seen as a single schema but as a collection of schemes; each audience class has its own conceptual information schema.

With all these advantages, there exists a major limitation. It does not give consideration to testing phase. WSDM approach can be improved greatly by adding a ‘Testing Phase’ to it. With the absence of Testing phase maintainability and the effectiveness of the website shall be endangered.

Thus for the successful web applications in general and e-business in particular this issue should not be neglected.

CHAPTER 3

BACKGROUND ON TESTING MODELS

3.1 Web Application Testing

The heterogeneous and highly dynamic nature of web applications gave rise to need of web application. Testing aims at uncovering errors in the tested object and executing the tested object with selected input values. Web applications frequently undergo maintenance that to at a faster rate than other software systems and this maintenance often demands small incremental changes. Web application is strictly interwoven to its running environment, it is not possible to test it separately to find out exactly what component is responsible for each exhibited failure. Therefore, different types of testing have to be executed to uncover these diverse types of failures. A highly adaptable test suites and automatable web testing approaches are essential to fit in with such changes. However, WA raises important and challenging test issues that cannot be solved by traditional strategies and existing tools and frameworks.
Therefore, all the entities of a Web application must be tested thoroughly to ensure that the application is reliable and meets its original design specifications.

3.2 V-Model

Initially defined by the late Paul Rook in the late 1980s, the V was included in the U.K.'s National Computing Centre publications in the 1990s with the aim of improving the efficiency and effectiveness of software development. It's accepted in Europe and the U.K

The V-model represents a software development process. It is an extension to the waterfall model. Execution of processes happens in a sequential manner in V-shape. The V-model is also called as **Verification and Validation model**. The testing activity is performed in each phase of Software Testing Life Cycle phase. In the first half of the model validations testing activity is integrated in each phase like review user requirements, System Design document & in the next half the Verification testing activity is come in picture.

Typical V-model shows Software Development activities on the Left hand side of model and the Right hand side of the model actual Testing Phases can be performed.

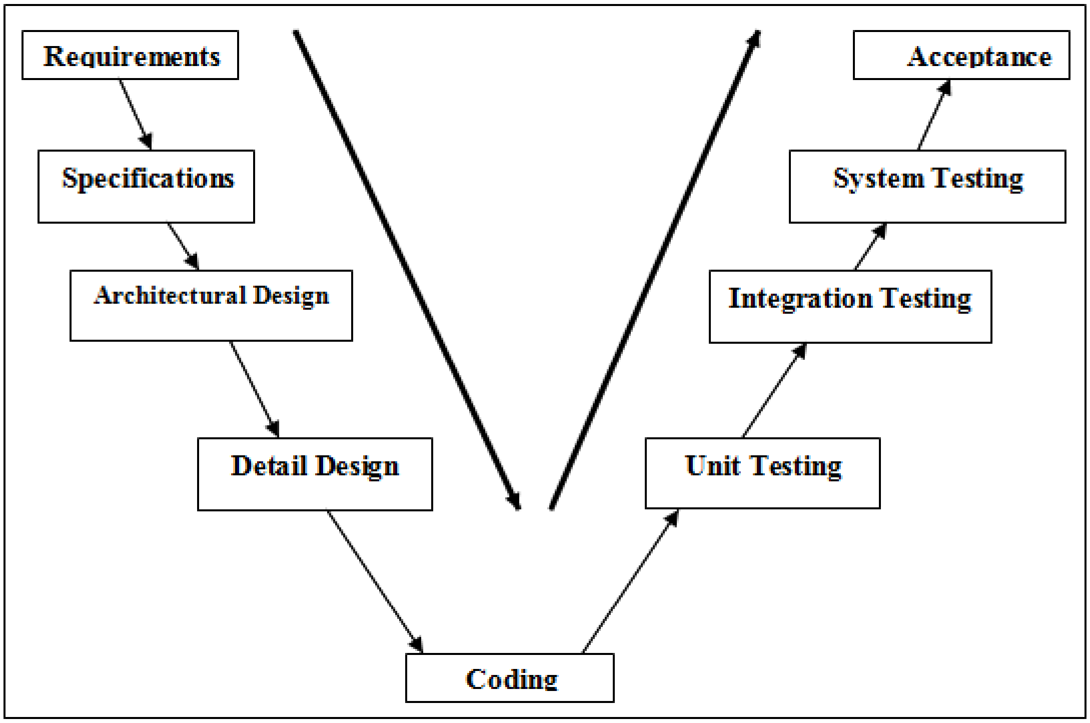


Figure 3.1 : V- Model

Verification Phases

Following are the Verification phases in V-Model:

- Requirement Analysis**:**

This is the first phase in the development cycle where the product requirements are understood from the customer perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement. The acceptance test design planning is done at this stage as requirements can be used as an input for acceptance testing.

- System Design**:**

Once clear and detailed product requirements have been listed, it's time to design the complete system. System design would comprise of understanding and detailing the complete hardware and communication setup for the product under development. System test plan is developed based on the system design.

- Architectural Design**:**

Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. System design is broken down further into modules taking up different functionality. This is also referred to as High Level Design (HLD).

- Detailed Design**:**

In this phase the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems. Unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. Unit tests can be designed at this stage based on the internal module designs.

Coding Phase

The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements. The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

Validation Phases

Following are the Validation phases in V-Model:

- **Unit Testing:**

Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

- **Integration Testing:**

Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

- **System Testing:**

System testing is directly associated with the System design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during system test execution.

- **Acceptance Testing:**

Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non functional issues such as load and performance defects in the actual user environment.

3.3 W – Model

Most of the models have some deficiencies with regard to test activities. Testing starts after the development phase at an instance when the implementation approaches completion. The link between testing, debugging, change and re-test is not usually the focus in such development models and the handling of this process is unclear. The W-model explained here aims to overcome such hurdles and improve the effectiveness of the testing process in the development life cycle.

Paul Herzlich introduced the W-Model approach in 1993. . Rather than focusing on specific dynamic test stages, as the V-Model does, the W-Model focuses on the development products themselves. Essentially, every development activity that produces a work product is “shadowed” by a test activity. In its most generic form, the W-Model presents a standard development lifecycle with every development stage followeds by a test activity. On the left hand side, typically, the deliverables of a development activity (for example, write requirements) is accompanied by a test activity “test the requirements” and so on. Testing usually receives relatively less attention in the traditional models and usually appears as a trivial and unimportant task to be carried out after coding. In order to place testing on an equal footing, a second “V” dedicated to testing is integrated into the V-Model. Both “V”s together give the “W” of the W-Model.[7]. W-model is represented in the figure below[8]

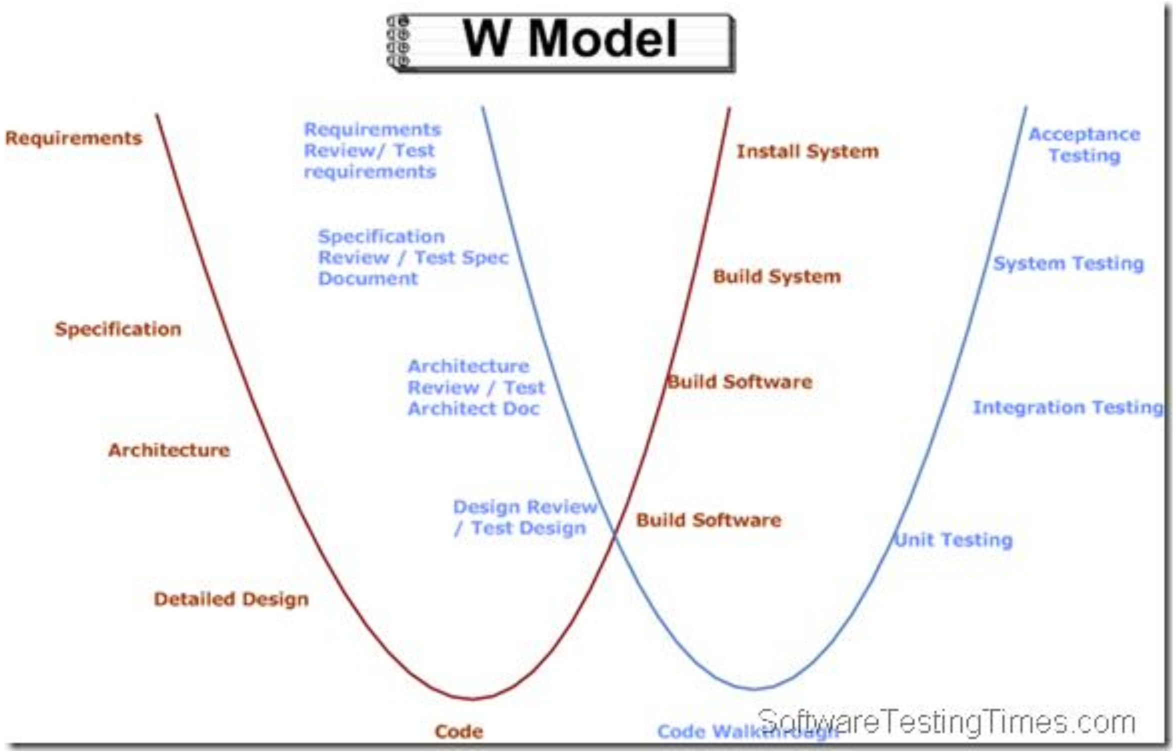


Figure 3.2 : W- Model

It is already clear in the early development phases, testing activities must be carried out. In this model, it is not only the planning and management tasks that have to be carried out but also, for example, as the system is divided into components the corresponding test cases for checking the component interfaces can also be developed in parallel. When the tasks that a particular component should involve are clear, the test cases to check these tasks individually lie relatively clear for creation. Should these considerations be first placed at the integration test phase (as the V-model suggests) then a considerable increase in cost is necessary - the detailed knowledge of the tasks of the component must first be relearned.

The W-Model removes the rather artificial constraint of having the same number of dynamic test stages as development stages. If there are five development stages concerned with the definition, design and construction of code in your project, it might be sensible to have only three stages of dynamic testing only.

W-Model in test strategy as follows:
Having identified the specific risks of concern,

- Specify the products that need to be tested
- Select test techniques (static reviews or dynamic test stages) to be used on those products to address the risks
- Schedule test activities as close as practicable to the development activity that generated the products to be tested.

3.4 Fish Model

This model of software development resembles the famous & ancient Waterfall model. It involves continuous / simultaneous deployment of verification & validation processes which depicts the effective testing phase. But, this model is costly to follow and it happens to be a very time consuming model.

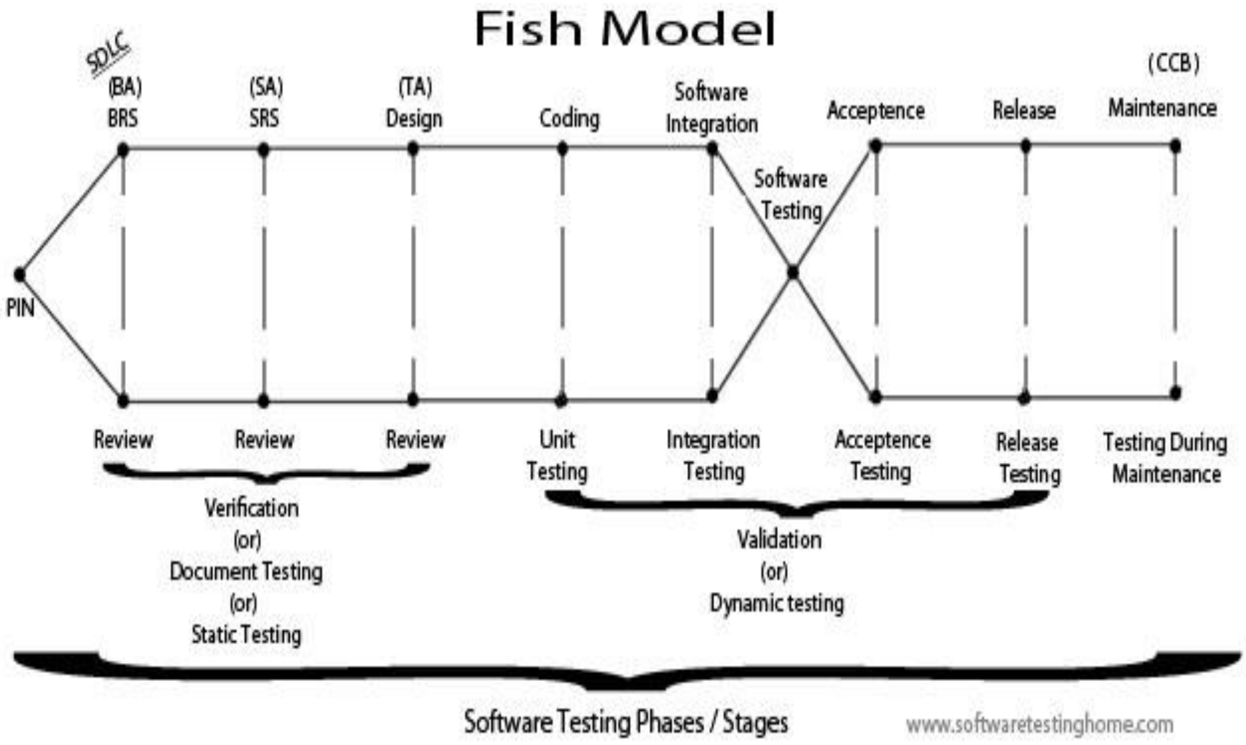


Figure 3.3 : Fish Model

Explanation of the above diagram is given below.

1. At first project initiation note (PIN) will be given to the CEO, from there the work is going to begin.
2. The business analyst(BA) will gather all the requirements from the customer and for those gathered requirements testing purpose a 'separate testing team' will be put together..
3. Then system analyst (SA) will do the analysis of those requirements and for the review of those analyzed requirements a separate testing team will be assigned.
4. Technical architect will do the design of those analyzed requirements in the form of HLD and LLD's after that a separate testing team will be assigned for the review of the design.
5. All the above conducted testing or reviewing is called as 'documents testing' or 'verification' or 'static testing'.
6. Unit testing: Unit testing involves testing on a single program in java, .NET, oracle, SAP,C,C++ etc.
7. Integration testing: Integration testing means conducting testing on the interconnection of two or more programs.
8. Software testing: Software testing means conducting the testing on a complete software w.r.t. to the customer requirements and expectations.
9. Acceptance testing: Acceptance testing means that collecting the feedback on the software from the real customers(Clients) or model customers(Public).
10. Release testing: Release testing means that the whether software was completely and correctly ported into customer site or not. It is also called as onsite testing/ delivery testing/port testing.
11. Maintenance testing: During the maintenance testing, CCB people can modify and test software w.r.t to customer change request.

3.5 Agile Methodology

The concept of agile software development consists of a group of software development methods in which requirements and solutions evolve through the collaboration between self-organizing, cross-functional teams. In software application development process, the methodology for the creative process which anticipates the need for flexibility and applying a level of pragmatism into the delivery of the finished product is known as Agile Software Development (ASD). Agile Software Development focuses on keeping code simplicity, testing often, and delivering functional bits of the application as soon as they're ready. To build upon small client-approved parts as the project progresses, as opposed to delivering one large application at the end of the project is the main goal of ASD.

The meaning Agile programming is different things to different people, but there are few principles which reside at the core of all agile development methodologies: Business stakeholders are collocated with small, autonomous development teams; the reliance of teams is less on up-front requirements and documentation than on face-to-face conversations; a continuous dialogue for software design, testing and refocusing are provided by those conversations. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages rapid and flexible response to change. It is a conceptual framework that focuses on delivering working software with the minimum amount of work.

Thus, the ascendancy of Agile Development Methodology is in direct response to IT organizations' dolorous history of software project failure, cost overruns and the concomitant business dissatisfaction with traditional IT design and development.

Agile practices allow teams to release software features at frequent intervals, as it emphasize iterative software design and development,. The Agile Methodology thus differs from the more traditional waterfall model, which involves gathering all requirements up front and delivering one finished software application at the end. In these changing business times, conversion from waterfall to Agile occurred in some companies because of its flexible processes and ability to continuously improve products during the development lifecycle.

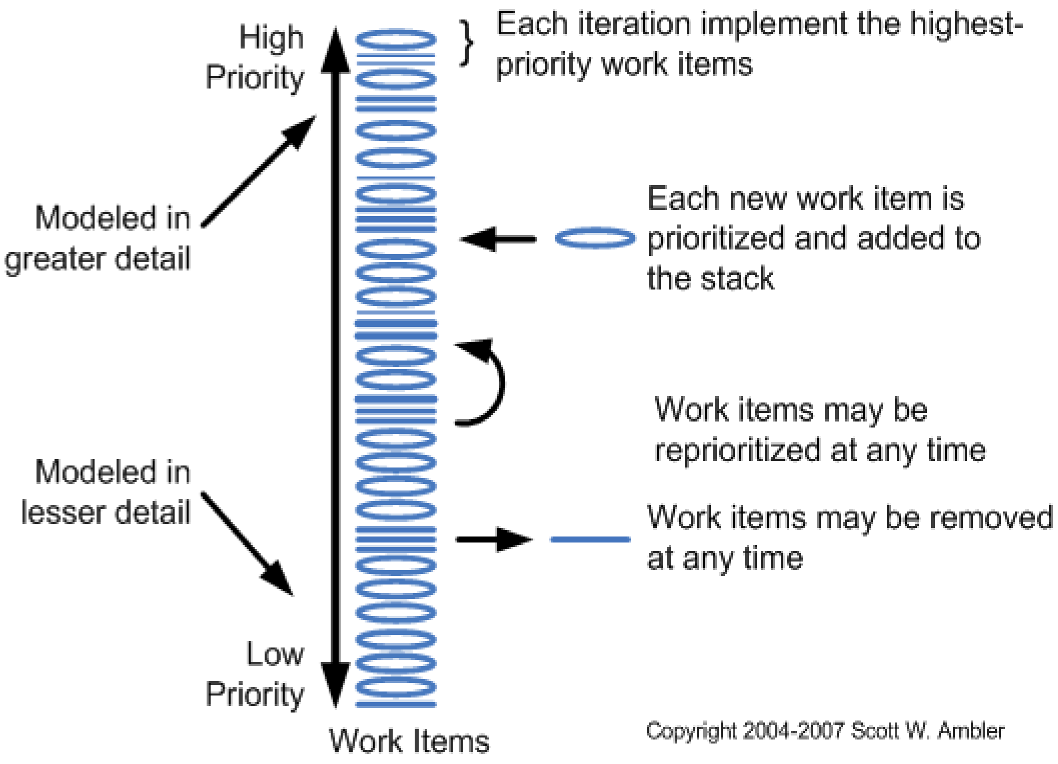

High Priority
Each iteration implement the highest-priority work items
Modeled in greater detail
Each new work item is prioritized and added to the stack
Work items may be reprioritized at any time
Modeled in lesser detail
Work items may be removed at any time
Low Priority
Work Items
Copyright 2004-2007 Scott W. Ambler

Figure 3.4 : Working of Agile

3.5.1 Manifesto for Agile Software Development

In February 2001, 17 software developers met at the Snowbird, Utahresort and lightweight development methods were discussed. They published the Manifesto for Agile Software Development to define the approach now known as agile software development. Some of the manifesto's authors formed the Agile Alliance, a non-profit organization that promotes software development according to the manifesto's principles.

Agile Manifesto reads, in its entirety, as follows:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

> Customer collaboration over contract negotiation
>
> Responding to change over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more."

The meanings of the Manifesto items on the left within the agile software development context are described below.

- Individuals and Interactions – self-organization and motivation are important in agile development, as are interactions like co-location and pair programming.
- Working software – working software will be more useful and welcome than just presenting documents to clients in meetings.
- Customer collaboration – requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important.
- Responding to change – agile development is focused on quick responses to change and continuous development.

Twelve principles underlie the Agile Manifesto, including:

- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development
- Working software is delivered frequently (weeks rather than months)
- Working software is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Close, daily co-operation between business people and developers
- Face-to-face conversation is the best form of communication (co-location)
- Projects are built around motivated individuals, who should be trusted
- Continuous attention to technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular adaptation to changing circumstances

3.5.2 Why Agile?

There are three major drivers for agile adoption in the enterprise.

1. The first is the frequency of release. Using agile practices, deliveries can be made faster and more frequent releases of their products by the companies.

2. Second is the desire to make midcourse corrections. Historically, most of the companies plan long-term programs that can take 18 months to two years to complete. The concept of Agile gives the ability to continuously monitor where the project is going and make necessary changes to the plan during the process instead at the end. Thus it makes easier for the companies to correction.

3. The third and final driver for adoption is improved relations with the business. By the use of agile practices, the IT organization can work in better synchronization with the business. Agile allows IT to still plan for long-term projects but deliver short-term results to the business.

3.5.3 Agile Methodologies

The Different methodologies of Agile Framework are as follows:

- DSDM

The Agile framework for software projects implements the Dynamic System Development Methodology to synchronize with the traditional approaches. The most recent version of DSDM is called DSDM Atern. The name Atern is a shortening of Arctic Tern, which resembles to a collaborative bird that can travel vast distances and epitomizes many facets of the method which are natural ways of working most importantly prioritization and collaboration. The DSDM addresses the most common failures of information systems projects, which include exceeding budgets, missing deadlines, and lack of user involvement and top-management commitment.

- SCRUM

The most popular agile framework in the world is scrum, which uses iterative and incremental development model. The concentration of Scrum is particularly on how to manage tasks within a team-based development environment. It provides the simple framework of basic tenets to solve problems and deliver good results - more valuable software faster. The actual implementation can involve keeping a track of the development and the blockages in the process on a daily basis. A scrum master is assigned for such a purpose who keeps a track of all the team activities and aids in faster development process

- XP

The type of agile software development, which advocates frequent "releases" in short development cycles, is Extreme Programming. It is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. The name for this methodology is derived from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels. It is a software-development discipline aiming at organizing people to produce higher-quality software more productively. XP addresses the analysis, development and test phases with novel approaches in order to make a substantial difference to the quality of the end product.

- TDD

Test-driven development (TDD) a type of software development process relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case for defining a desired improvement or new function, then it produces the least amount of code to pass that test, and finally refracting the new code to acceptable standards.

- LEAN

The production practice that considers the expenditure of resources for any goal other than the creation of value for the end customer to be wasteful, and thus a target for elimination is termed as Lean. Working from the perspective of the customer who consumes a product or service, "value" is defined as any action or process that a customer would be willing to pay for. Lean is centered on preserving value with less work.

- KANBAN

Kanban is a system to control the logistical chain from a production point of view, and is not an inventory control system. It was developed by Taiichi Ohno, at Toyota, to find a system to improve and keep up a high level of production. Kanban is one method of achieving JIT . Kanban became an effective tool in support of running a production system as a whole, and it proved to be an excellent way for promoting improvement.

3.5.4 Agile for Web

Agile welcomes changing requirements, even late in development. The Agile processes harness change for the customer’s competitive advantage.

The Agile Manifesto

The flexibility feature that is offered by agile methodology Manifesto makes agile incredibly apt for web development as, perhaps, no technology changes faster than the Internet. New technologies, products, techniques, styles and businesses pop up on an almost daily basis and working with agile gives both developers and product owners the best possible chance to adapt. What if a new, better, cheaper payment system arises half way through a project? Would it make any sense to stick to what was agreed several months ago just because it was written on a piece of paper (or go through numerous change orders and respecting and recostings)? Or what if a complete new payment style, such as freemium, becomes the popular – wouldn’t the entire project be better off pivoting the business model to adapt to it instead?

Ultimately, agile lets you focus on delivering a product rapidly, within set boundaries of time and budget, ensuring that every iteration sees tangible and valuable features being developed and released. The flexibility such as an approach provides, especially in fast-paced web world, is incredibly valuable.

3.5 Choice of Testing Models Based on some Parameters.

S. No.	Parameters	V-Model	W-Model	Fish Model	Agile
1	Do Requirements change often?	×	×	×	✓
2	Requirements indicate that system is complex?	×	✓	×	✓
3	Programmers have experience on similar projects	×	✓	✓	✓
4	Experience of tools to be used	×	✓	×	✓
5	Tight project schedule	×	×	×	✓
6	Limited user participation	✓	×	✓	×
7	Customer feedback is required	×	✓	×	✓
8	Working deliverable is required in early stages	×	×	×	✓

Table 3.1 : Choice of Testing Models Based on some Parameters

CHAPTER 4

LITERATURE REVIEW

4.1 Introduction

A Literature Review is a body of text that aims to identify, evaluate, and synthesize the critical points of current knowledge and methodological contributions to a particular topic by the author. Literature reviews are secondary sources, and as such, do not report any new or original experimental work. It should give a theoretical basis for the research and helps to determine the nature of research. It identifies what is already known about an area of research.
For the present study, literature has been collected from different sources such as journal articles, Internet and conference proceeding paper.

4.2 Review of Related Work

Olga De Troyer, (2001) [13], addressed the issues that arise in constructing a web site. It also provides the detailed explanation of working of WSDM.

Tolba, R., Mushtaha, A(2006), proposed an approach to integrate V-Model into the web site development process. This approach is given as an extension of the Web Site Design Method (WSDM). WSDM was a new methodology for web development. Despite being a user-centric approach, it seems incomplete and lacks "Testing Phase" after implementation phase. Testing is important phase in order to validate and verify user's requirements because WDSM is user-centered approach.

Integration Of WSDM and V-Model

By integrating V-model with WSDM, author's vision was to make WDSM more flexible and satisfy more of user requirements by delivering a better quality product. In this proposed model, the development process of web is done using WSDM and the lack of testing process is overcome using V-model.

Each phase in the left side of V-model is compatible with WSDM's phases, and the integration is done from the right side of V-model as shown in the figure. The integration process is discussed in the following 4 points;

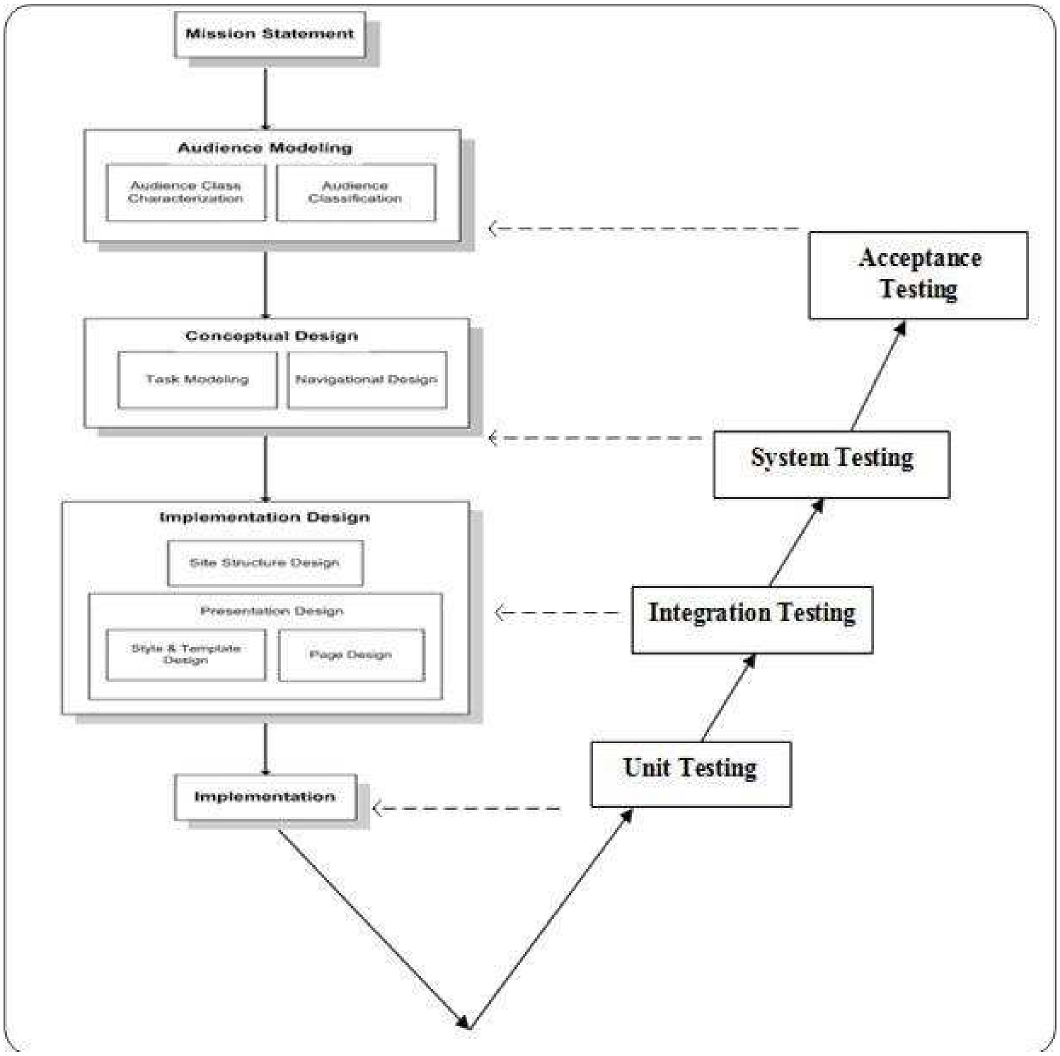


Figure 4.1 : Integrating V- Model into WSDM Approach

.

1. ***Unit Testing:*** The unit testing is performed on the units while the programmer is coding them. Units were considered as methods, classes that it is the bias of building system, testing them by tracing the code and discover faults then correct it. Testing also included the reused methods and classes with its methods, or the Scripting codes that can be used to enhance the capability of the system.

2. Integration Testing: In integration testing the separate reused methods, classes with its methods and scripting codes were tested together to expose faults in the interfaces and in the interaction between integrated components. Tracing here is done in a logical way. Integration testing also includes the integration of pages and tracing the links between web pages and check that navigation go correctly.

3. System Testing: System testing compares the system specifications against the actual system. This means that system results need to be compared with the mission statement goals that were put it initially. Mission Statement aims that the testing process must guarantee that the website satisfy the audience requirements and reach to the goal that we put in mission statement phase. System Testing receives feedback from audience by using the website in order to enhance performance and identify faults.

***4. User Acceptance Testing*:** This is the final phase of testing, we carry the website to the real user's environment by uploading and hosting and testing it using real data .The aim is to decide whether a system satisfies its acceptance criteria or not.

Steven H. VanderLeest, Calvin College and Dorner Work (2009) [14], presented successful integration of test-driven development, pair programming, refactoring, an iterative approach, and other Agile methods into a traditional DO-178B software development process. They observed that the transition to Agile development does not require sudden, sweeping change, but instead can be accomplished through incorporating Agile methods into an existing process.

Martin McHugh1, Oisín Cawley2, Fergal McCaffery1, Ita Richardson2, and Xiaofeng Wang3,(2013),[11] proposes an AGILE-VModel for Medical Device Software Development to Overcome the Challenges with Plan- Driven Software Development Lifecycles.

Anil Agarwal, NKGarg, Avirag Jain, (2014), presented this paper to highlight broadly the role of QA within Agile development model. It explained the the importance of agile methodology. It also focused on fresh thoughts and approaches to improve the overall quality of product developed using Agile methodology.

CHAPTER 5

RESEARCH METHODOLOGY

5.1 Introduction To Agile Testing

Agile software development is iterative & incremental. An important thing about testing is that we test to validate that were building the right thing.

The approach of Agile Testing is different in many aspects:

- Reduce social risk: Previously people worked in silos, being told what to be done for years.
- Schedule & cost risk: Price & delivery date are often already fixed, this contradicts to asking people to take responsibility, since the decisions have already been taken
- Business risk: Caused by different understandings of the requirements, this can be reduced by face-to-face communications and by showing your work to the users as this is the only possibility to verify.
- Technical risk: Are you using the right technology/ framework? Do we have the right skills? Consider that you never do the same project twice!

5.2 Agile Testing over other approaches

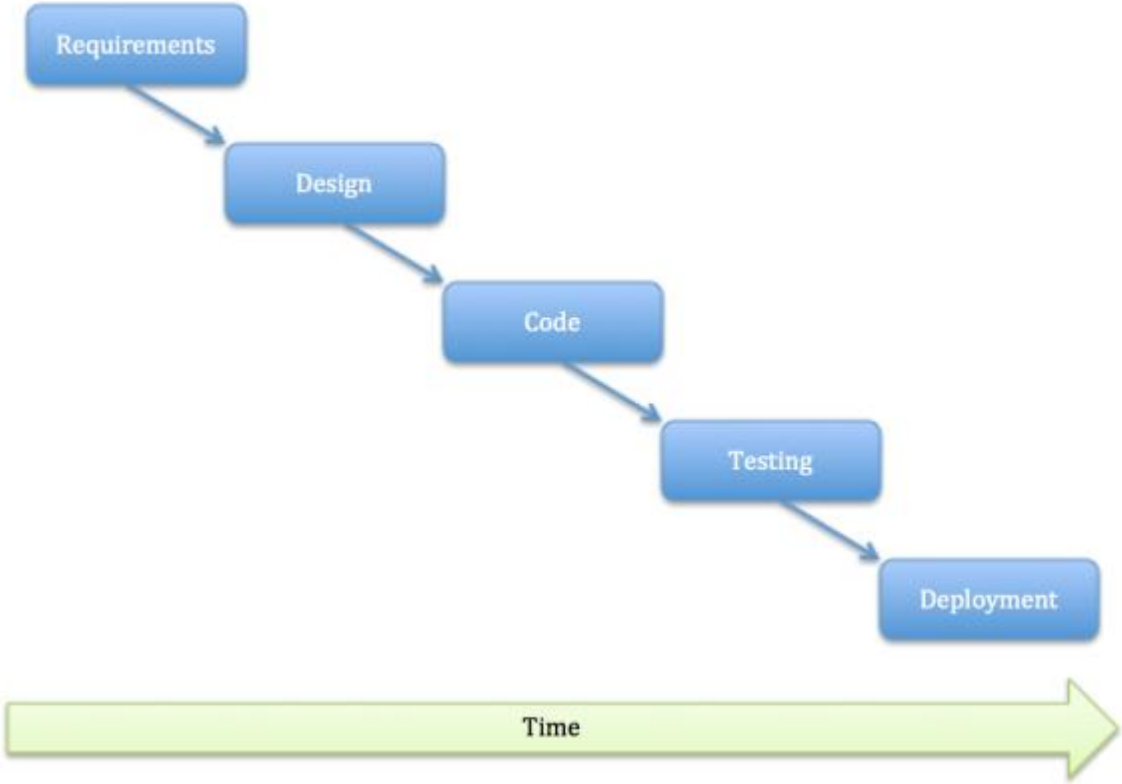
Over time agile methodology has been recognized as one of the most effective testing methodologies. The reason for the wide acceptance of this method lies in the practicality of the approach. When it comes to actual software development the theoretical and the actual scenarios are completely different. The development cycle if at all planned in advance is subject to a lot of changes in its cycle. Real time problems may arise which may result in a change in the original plan. For example a module in a website may take longer than planned, or let’s say in case a resource decides to leave in the middle of the development cycle, it would take time to find a new resource and train him or her for the project. It could also be the case where the number of resources may be lesser than anticipated during planning or change of requirements or addition in requirements, the list is endless.

Looking at all these examples it will suffice to say that unprecedented incidents can take place which make long term planning in software development obsolete. Agile facilitates the breakdown of the software development into small realistic chunks which can be tested and monitored easily. It involves testing from the customer perspective as early as possible; testing early and often as code becomes available and stable enough from module/unit level testing It does not involve a lot of documentation and the development can take place swiftly. Another feature is that a working version of the product can be made available to the client in the initial phases of development. This means that the customer actually gets a visual of the product very soon and suggestions and improvements can be made in very early stages. This saves a lot of rework which would take place if the revisions are made at a later stage.

5.2.1 Agile Vs V model

V-model is a software development model that involves building a logical V shape sequence where the testing techniques associated with the design are reflected as descending and are applied for the “verification” and connected to the requirements or specifications parts are reflected as ascending and are applied for “validation”. Equal weight to coding and testing in the V-model gives software development process.

As compared to this model Agile methodology gives a faster output and is more suited to web applications owing to the fast trend changes occurring in web development.
As more features are delivered incrementally, customer can realize some of the benefits early on. Testing cycle time of Agile is relatively short compared to V-Model, because testing is done parallel to development. Agile is a proactive model (due to its very short cycles) compared to the much more reactive V-Model. V-Model is very rigid and relatively less flexible than Agile model.



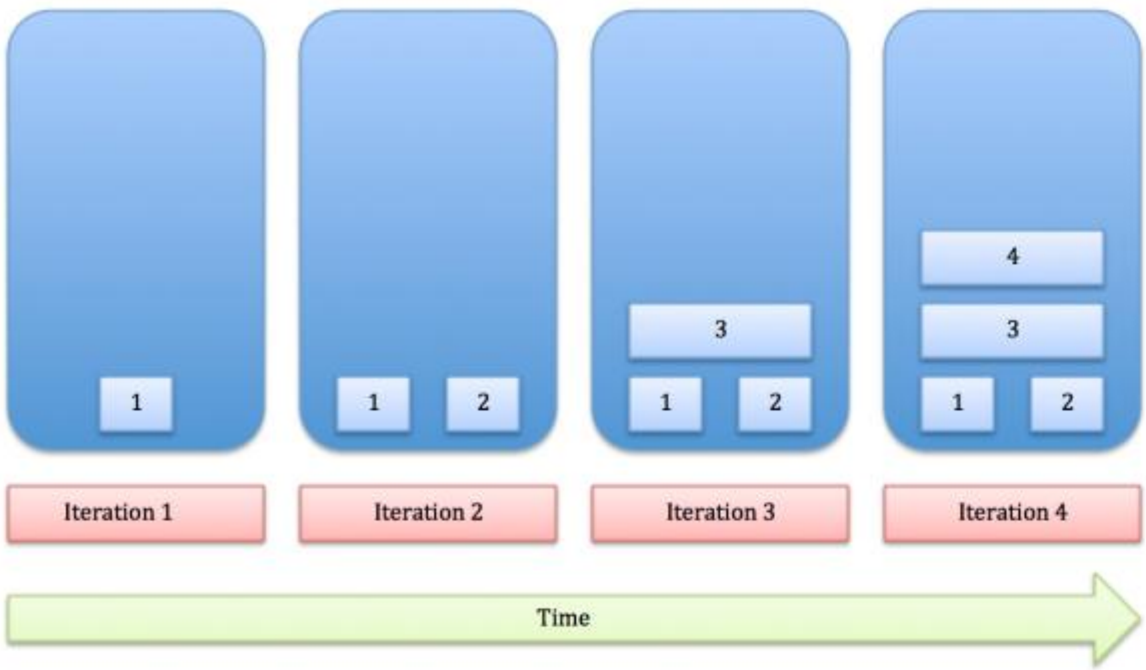


Figure 5.1 : V vs Agile methodology

5.2.2 Agile Vs W model

The models of software development presented do not clarify the expenditure needed for resources that need to be assigned to the individual activities. Also in the W-model it appears that the different activities have an equal requirement for resources (time, personnel, etc.) In practice this is certainly not the case. In each project the most important aspects may vary and so therefore the resource allocation is unlikely to be equal across activities.

Unlike the W model, all activities do not have equal weightage. Since the development cycle is broken down in small realistic sprints the expenditure and resources are assigned accordingly and if at all there is an imbalance, it is short lived because the sprints are small in size and correctional measures can be taken to improve effectiveness.

5.2.3 Agile Vs Fish

The fish model is an expensive and time consuming model. At the end of every phase two reports are created; one for verification and one for validation. This can be very time consuming and expensive as such extensive reports require time and manpower.

Unlike that agile encourages swift development. There is no extensive report generation hence a lot of time and manpower is saved.

5.3 Proposed Model : "AGILE – WSDM" Approach

In this model, WSDM approach is enhanced by integrating Agile Testing Methodology into WSDM. The AGILE-WSDM duo would help to create a more flexible, reliable, usable, maintainable, customizable and effective Web System.

The development process shall be done using WSDM approach, furthermore continuous testing will be performed by agile testing methodology.

The integration process will be discussed in the following points.

Phase 1 & Phase 2 : Mission statement specification & audience modeling
This phase will be performing the same task as discussed in Section 2.1.1. It will be discussed in detail with an example in the next section.

Phase 3 : Conceptual design phase
Originally in WSDM process, the output of this phase is Navigation models and Task Models. But in this proposed model there is a slight change. As agile is an incremental planning approach, the output will be conceptual design model for functionality 1. Navigational and task models would be constructed for particular 1st iteration only. It depicts the ability of agile modeling to flexibly schedule the implementation of functionality, responding to changing business needs. Hence every iteration will be tested in the testing phase and integrated with the forthcoming functionalities.

Phase 4 : Implementation Design Phase

In the Implementation Design step the Page Design, the Presentation Design, and the Logical Data Design is done. But for this proposed model the respective designing will be done only for certain functionality in 1st iteration rather than designing for the whole system at the beginning itself. With the subsequent iterations, the functionalities shall be increased which will ultimately create the desired system.

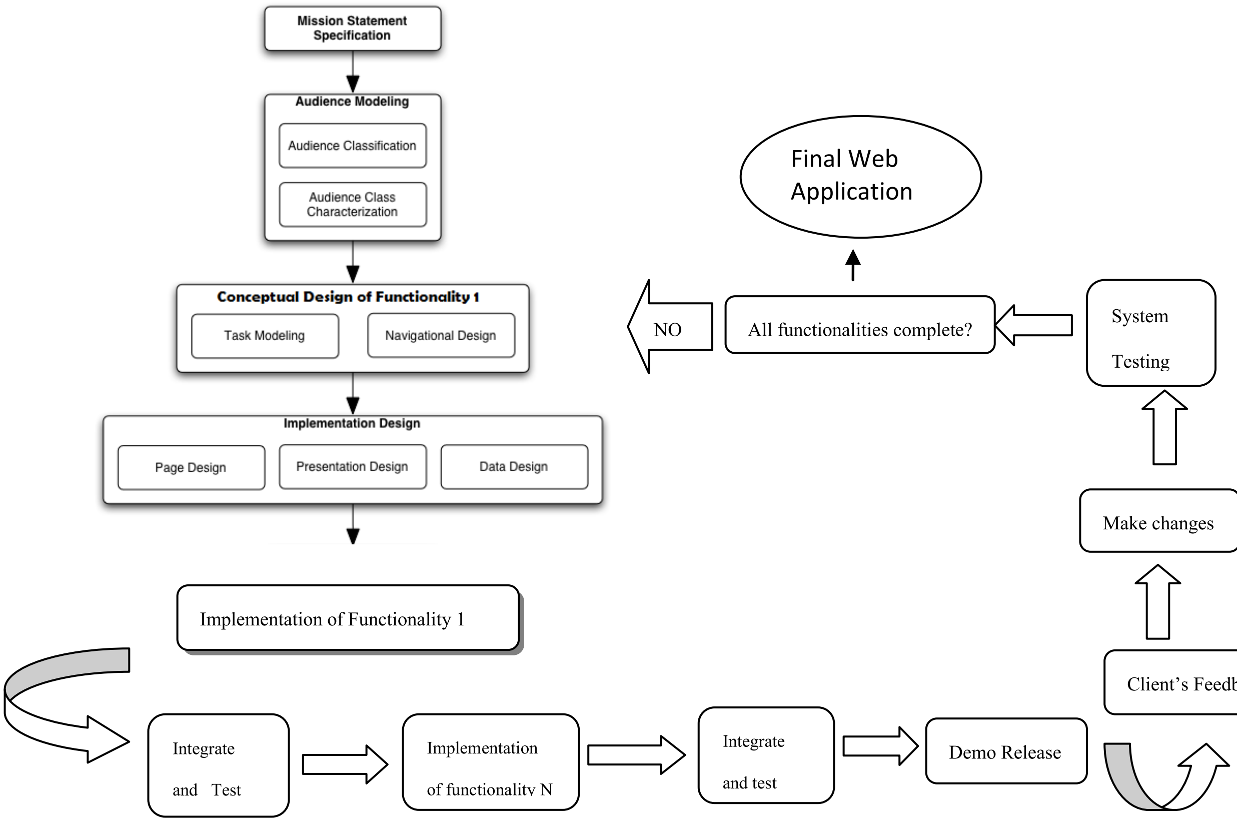


Figure 5.2 : Integration of Agile with WSDM

Phase 5 : Implementation phase for 'Functionality 1'

It is the actual implementation using automated and existing tools and technologies. All the Information collected in the previous phases is taken as input and Functionality 1 (in the form of iteration 1) in the chosen implementation environment is automatically generated. This phase is about realizing the physical web site. Whereas in WSDM-Vmodel approach, complete physical web site is released in this phase which is then tested.

Phase 6 : Integrate and Testing Phase

This is a repetitive phase. It occurs after release of every functionality or set of functionalitiues i.e after every iteration. For very 1st iteration only testing is performed and then hereafter after every subsequent releases, new functionality is integrated to the previous one followed by testing. This repeats for ‘n’ iteration. Thus customer feedback after every iteration helps to conduct testing in a more effective manner.

Phase 7 : Implementation phase for ‘Functionality 2…N’

With every iteration, a deliverable is produced for the client. N no of functionalities are produced and resetive integration and testing is performed. This works as a circular loop. It is depicted in the figure.

Phase 8 : Demo Release

This is known as iteration 1. It can have N no. of functionalities depending upon user needs and size of project. This demo release gives a visual idea to the users about the look and feel of the web site or web application. Sometimes user doesn’t have a clear picture in the beginning of the project, but with the help of this demo release, there comes a better understanding of the objective. By having frequent demos of the software, you give your clients opportunities to react to where you are on the development trail. Sometimes you'll get a simple, "Great! That's what we want." Other times you'll get a correction of your misunderstandings.

Phase 9 : Clients Feedback

Users review is very important. Agile methodology seek customer feedback throughout the point. There is no point performing all the testing if the purpose is not being met. The purpose of the web application should be served. Then the effort put for testing is worthwhile. Hence uses feedback to make constant adjustments in a highly collaborative environment. Thus with demo release users get to have a clear picture and can provide a valuable feedback so that necessary action be taken.

Phase 10 : Make changes

This phase incorporates the necessary changes in the system, that might come because of user feedback or as a result of integration and testing. The uncovered bugs are removed in this phase.

Phase 11 : System Testing

Finally system testing is performed. This testing typically starts after the completion of development and functional verification testing.

CHAPTER 6

DISCUSSIONS AND RESULTS

6.1 Introduction

In the previous section, a new model was proposed which I believe is an extended version of existing WSDM approach. My aim was to blend testing phases with development phases.
To understand the workflow of our proposed WSDM-Agile model and its fundamental differences with the previously proposed WSDM-V model, both the models are required to be deployed in practice so that clear picture can be foreseen.

Thus both the models undertake the example an online job vacancy site.
One group, named as group A was told to follow the proposed WSDM-Agile model for development of the website while the other group, group V was told to follow the WSDM-V model for the same. The requirements of the system were specified to both the groups. Typical users of such a Website are employers, visitors, and the owners. These different users have different needs and requirements:

- Employers should be able to offer new job opportunities. To do so, they will need a login and password that is provided by the owners of the online job vacancy site. They can also provide a company presentation which will be placed with each job vacancy. Once a company is registered they can start offering jobs. Therefore, they have to provide a job description, a description of the profile they expect of the candidates, a description of the job vacancy, the type of contract, a contact person, etc. Note that not all of these are mandatory. There should also be the possibility of deleting a job offering. Employers should also have the opportunity to search in a CV database for suitable candidates they want for a particular job.
- Visitors should be able to browse through the online job vacancies. They should have the opportunity to search for jobs by function, area (country, state, province, city, ...), statute (laborer, clerk, interim, ...) and language. They should also be able to search for jobs from a specific company. Next thing they may want to do is to apply for a job vacancy. When they do so, they have two possibilities. On one hand they can choose to stay anonymous for the online job vacancy company and do their application by standard email. On the other hand there should be a possibility to register and to create and modify an online CV which is automatically available in the database where employers can search for suitable candidates. The CV is also sent by email to the company of the job vacancy. Job-seekers may also want to create and modify a job profile (what they are searching for), so that when a vacancy is entered by a company which is suitable for that job-seeker, he immediately receives an email.
- Owners of the company are responsible for the management of the job vacancy Website. When a new job has been added, the managers are notified and can inspect the information about the job. A manager can change the category or subcategory of the job function if necessary. Next to that the manager can decide to notify all people who have the category or subcategory in their job profile.

This example was taken from [33] and will follow the different phases of the Design Method(WSDM).

The JobSite mission is to assist employers in meeting their workforce and assist local job seekers to enter, remain, and progress in the workforce using job offers and job profiles.

6.2 Effort Estimation

After the analysis of the problem statement, Effort Estimation for both the model was undertaken.

- For WSDM-Vmodel ‘Basic Cocomo Model’ was used for Effort estimation.
- For WSDM-Agile ‘Scrum Estimation Technique’ was used for Effort Estimation.

6.2.1 COCOMO Estimation for WSDM-Vmodel

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics. The COCOMO cost estimation model is used by thousands of software project managers, and is based on a study of hundreds of software projects.

Basic COCOMO computes development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (KLOC).

COCOMO applies to three classes of software projects:

- Organic projects - "small" teams with "good" experience working with "less than rigid" requirements
- Semi-detached projects - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects - developed within a set of "tight" constraints. It is also combination of organic and semi-detached projects.(hardware, software, operational, ...)

The basic COCOMO equations take the form

Effort Applied (E) = $a_b(\text{KLOC})^{b}{}_{b}$ (1)
Development Time (D) = $c_b(\text{Effort Applied})^{d}{}_{b}$ (2)
People required (P) = Effort Applied / Development Time (3)

KLOC is the estimated number of delivered lines (expressed in thousands) of code for project.

The coefficients a_b, b_b, c_b and d_b are given in the following table 1:

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Basic COCOMO is good for quick estimate of software costs. However it does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and so on.

Estimated LOC for the online job portal is 10 KLOC, and this makes our case study of the online job portal come under the case of an ‘organic’ project.

So, for the estimation using the Basic COCOMO, using equation (1) and (2) and the coefficients of organic model form Table 1,

Effort= $2.4(10)^{1.05}$ = 26.92 Person Months

Development Time = $2.5(26.92)^{0.38}$= 8.73 months

Manpower required = 3 persons

6.2.2 SCRUM Estimation for WSDM - Agile

Scrum is a form of agile that optimizes the estimation and execution of work. Scrum teams work in fixed intervals, known as sprints, which means they iterate on the product frequently. This allows the team to get feedback about each release, and to ensure maximum value back to the customer in the next sprint. First of all, Scrum assigns work to an entire team, not an individual. Secondly, instead of a manager estimating time on behalf of other individuals and assigning tasks based on conjecture, team members in Scrum use effort and degree of difficulty to estimate their own work.

Scrum does not prescribe a single way for teams to estimate their work. However, it does ask that teams not estimate in terms of time, but, instead, use a more abstracted metric to quantify effort. Common estimating methods include numeric sizing (1 through 10), t-shirt sizes (XS, S, M, L, XL, XXL, XXXL), the Fibonacci sequence (1, 2, 3, 5, 8, 13, 21, 34, etc.) In the Sprint Planning Meeting, the team sits down to estimate its effort for the stories in the backlog. TheProduct Owner needs these estimates, so that he or she is empowered to effectively prioritize items in thebacklog.

Following scenario was used:

1. Put all the stories (objects) in front of developers.
2. Let them to choose one story (object) that will get the number **Effort=1**. Typically it is small story like **„I as a user want to sign out my acount“**.
3. Then compare all remaining stories **relatively** to referenced story and other, already estimated stories.

For example, **Story 2 –** „I as a user want to search for a job “ is harder story than Story 1 mention above. Developer thinks that this story is 3 times harder than Story 1. Thus I will put Effort = 3 story points.

Similarly for the project "Job Portal Website" the above explained methodology was employed. Fibonacci series was used for calculating story points. For the Job Portal website 46 stories were listed by the developers and testers.

No. of days required to complete the task = total No. of story points/ no. of persons

Thus, through this effort can be estimated.

In this particular example, Total story points for 46 stories : 603

If manpower is taken 10, then No. of days Required = 60.3

6.3 Deployment of both the Models

Given below is the flowchart that illustrates working process for both the models. Figure[] shows the diagrammatic representation for WSDM – VMODEL and Figure [] for WSDM – AGILE.

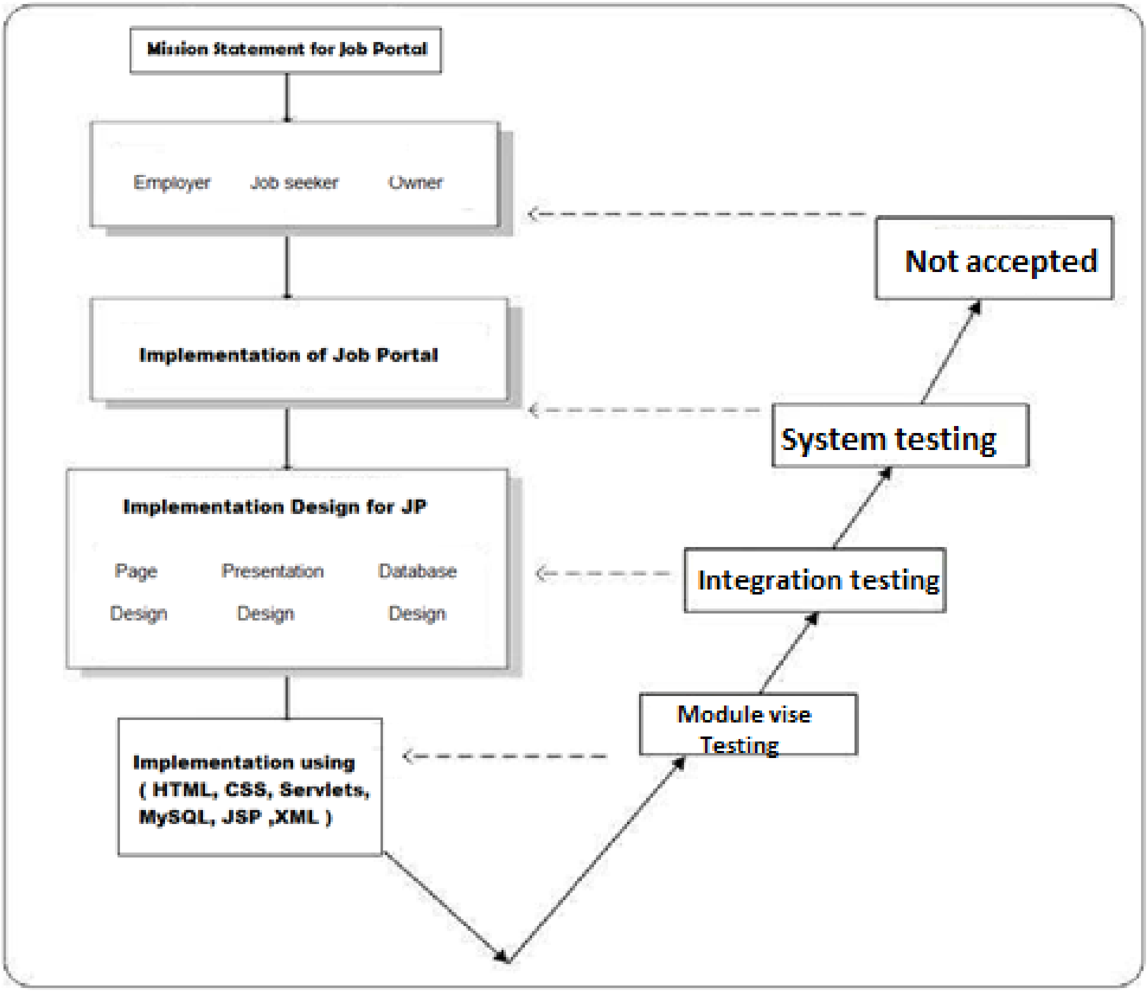

Mission Statement for Job Portal
Employer
Job seeker
Owner
Implementation of Job Portal
Implementation Design for JP
Page Design
Presentation Design
Database Design
Implementation using (HTML, CSS, Servlets, MySQL, JSP ,XML)
Module vise Testing
Integration testing
System testing
Not accepted

Figure 6.1 : Job Portal using WSDM- Vmodel

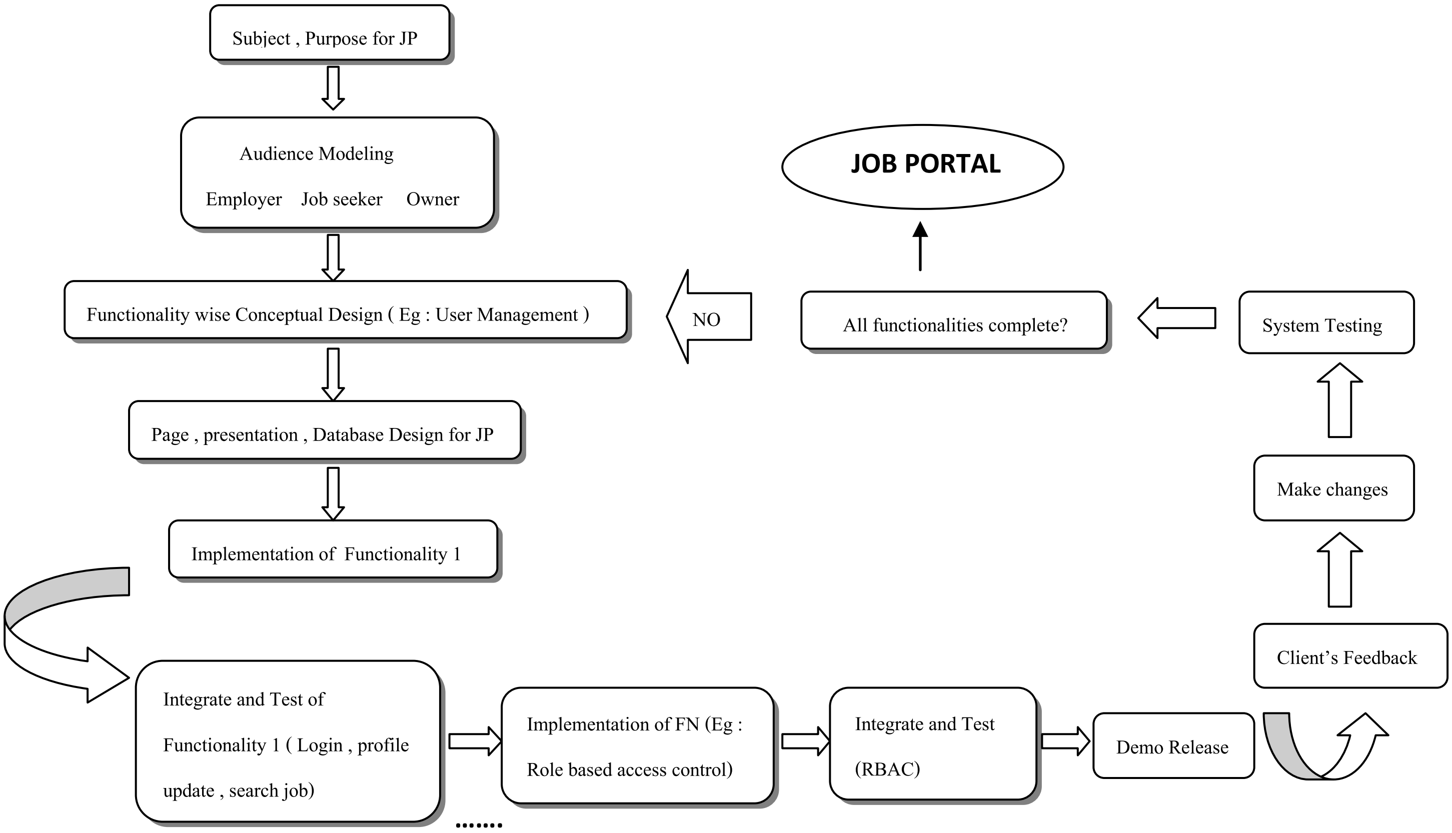

Subject , Purpose for JP
Audience Modeling
Employer Job seeker Owner
Functionality wise Conceptual Design (Eg : User Management)
Page , presentation , Database Design for JP
Implementation of Functionality 1
Integrate and Test of Functionality 1 (Login , profile update , search job)
....
Implementation of FN (Eg : Role based access control)
Integrate and Test (RBAC)
Demo Release
Client's Feedback
Make changes
System Testing
All functionalities complete?
NO
JOB PORTAL

Figure 6.2 : Job Portal using WSDM-Agile

Following the respective approaches for constructing the website, models yielded the following results:

The WSDM-V model approach was completed and the deliverable that it produced in the end failed in the acceptance testing phase because the customer interaction was minimum in this model throughout the web application development phase. This required a lot of rework and much more effort for modifying the web application to meet the demand and expectations of the customer. The guarantee of success in such a project is not high as the customer involvement is the last stage. And as the requirements of the customer keep changing, not involving the customer till the end of the project increases the risk factor in this case by a huge amount.

However, in the case of WSDM-Agile model, the end product was successful because a series of demo releases in between incorporated the customer feedback in the development cycle itself and the rework that had to be done was minimum as compared to the previous case. Though this project, may exceed its time due to the amount of daily interactions and series of communications required but the time spent this way is eventually utilized as it helps in the delivery of a better quality product and a more acceptable one for the customers. This ensures that the risk factor is minimized in this model.

Summarized Estimation Analysis of the Online Job Portal System

Criteria	WSDM + V-Model	WSDM +Agile Methodology
Size(in KLOC)	20	20
Effort (in person months)	26.92 person months	603 Story Points
Manpower	10	10

6.4 Comparative Analysis

Criteria	V-Model	Agile Methodology
Specification of All the Requirements in the beginning	Yes	Not all and Frequently Changed
Project Cost	Almost as Estimated	Almost as Estimated
Guarantee of Success	Moderate	High
Required Expertise	Moderate	Very High
Overlapping Phases	No	Yes
Process	Heavyweight Process	Lightweight Process
Framework type	Non linear	iterative and incremental
Rework cost	High	Low
Testing	After completion of Each Iteration	while coding
Customer Involvement	Low	Continuous
Basic business Knowledge Required	Not Much	Very Much
Suitable Project Size	Large Scale	Almost Small scale
Cost Control	Yes	No
Simplicity	Moderate	Simple
Risk Involvement	High	Not High
Flexibility	Little Flexible	Very Flexible
Maintenance	Little Maintainable	Easily Maintainable
Changes Incorporated	Difficult	Easily
Reusability	Little Possibility	Yes
Documentation and Training	Yes	Limited

Time Frame	Long	Short
Availability of Working Software	At the End of the Life Cycle	At the End of Every Iteration
Customized product	Least Possible	Possible
Customer Control over Administrator	Low	Very Much
Required Team Creativity	No	Yes
Knowledge Transfer	No	Yes
Team size	Small Team	Small Team
Primary Objective	High Assurance	Rapid Development
Implementation	Easy	Easy
Release Cycle	Big band (All Functionality at Once)	In Phases
Integrity & Security	Least	Robust

Thus, by this table we can summarize that in this changing era, when requirements keep on changing, we cannot restrict ourselves to wait for the completion of coding stage as for Vmodel. Instead WSDM-AGILE model provides higher flexibility and minimize risk of rejection. If the product is unaccepted, the effort applied to it becomes zero because whole of the work needs to be redone. The model built from the new proposed approach is more maintainable and provides high user satisfaction as the deliverables provided in between the development gave a better idea to users what the product will be like. This removes the risk of rejection. Through the new approach highly customized product can be achieved.

CONCLUSION

Conclusion

Popularity of web applications has been increasing very rapidly for last some years. Web applications are mostly accessed via server which has to ensure better user experience on his/her host machine along with security from online threats. Also, requirements change very rapidly in case of web applications and thus resulting in more controlled and fast change management. Thus need of systematic approach arises.

WSDM(Web Site Design Method), a Web Development approach provided modeling primitives that allow a web developer to construct models that describe the web site/application from different perspectives and at different levels of abstraction. Although it was a systematic approach, it also lacked a important primitive i.e a Testing Phase. Thus a model was proposed to overcome this drawback. Model proposed was Integration of WSDM approach with AGILE testing Methodology.

Earlier also a model was proposed which integrated WSDM with V-Model. To overcome the pitfalls of this model Agile testing Methodology was integrated with WSDM.

A case study of developing Job Portal Site was undertaken that clearly shows that WSDM-agile approach fares well in considerably large number of parameters like flexibility, maintainability, less risk involvement, less amount of documentation and so on when compared to WSDM-V model.

With the approach proposed in this paper, agility has been introduced into the process of website development using WSDM. One important benefit that needs to be highlighted between WSDM-Agile Approach and WSDM-V-Model Approach is that the former methodology possesses the ability to successfully deliver result quickly and inexpensively on complex projects with ill-defined requirements whereas in the latter it is delivered at the end. This agility will make the WSDM approach flexible enough to incorporate changes at any phase with the least cost and also integrate the website testing process into the website development process thereby reducing the effort of the testing in the end because it is difficult to go back and change a functionality if we follow WSDM-Vmodel approach. The proposed model will remove this risk.

The proposed framework was developed to provide more customer satisfaction, speedy deliverables, to shorten the development life cycle, to reduce bug rates, effective verification and validation and to accommodate changing business requirement during the development process. Deployment of both the models for the Job Portal Site shows the hindrances that can occur in the traditional model increasing the effort. This thesis also provides the Effort estimation and comparitive analysis for the development of the website using both the approaches. Thus the results depict the improved behavior of the proposed model.

REFERENCES

References

[1] L. Xu, B. W. Xu, and Z.Q. Chen, "Survey of Web Testing", Computer Science (in Chinese), 2003, 30(3): 100-104.

[2]Geoffrey Elliott (2004). Global Business Information Technology: an integrated systems approach. Pearson Education. p.87.

[3]De Troyer, O. and C. Leune (1998), "WSDM: A User-Centered Design Method for Web Sites," In Computer Networks and ISDN systems, *Proceedings of the 7th International World Wide Web Conference*, Elsevier, pp. 85 - 94.

[4]Goedefroy, W., R. Meersman and O. De Troyer (1998), "UR-WSDM: Adding User Requirement Granularity to Model Web Based Information Systems," In *Proceedings of 1st Workshop on Hypermedia Development*, Pittsburgh, USA.

[5] *De Troyer, O., Casteleyn, S.: "The Conference Review System with WSDM"*, First International Workshop on Web-Oriented Software Technology, IWWOST'01, http://www.dsic.upv.es/~west2001/iwwost01/), Eds. Oscar Pastor, Valencia (University of Technology), Spain.

[6] Tolba, R., Mushtaha, A. :,"Integrating V-model into Web Development Process" International Arab Conference on e-Technology - IACeT, Amman, Jordan (2008)

[7] Andreas SpillnerUniversity of Applied Sciences Bremen
Germany," The W-MODEL –Strengthening the Bond Between Development and Test ",2000

[8]Agile development: An overview; http://www.richappsconsulting.com/blog/blog-detail/agile-development-an-overview/; Retrieved 9th April, 2013

[9] www.softwaretestingtimes.com

[10]http://www.softwaretestinghome.com/2013/01/old-and-new-sdlc-models.html#axzz38litFRA5

[11] Martin McHugh1, Oisín Cawley2, Fergal McCaffery1, Ita Richardson2, and Xiaofeng Wang3,"An Agile V-Model for Medical Device Software Development to Overcome the Challenges with Plan- Driven Software Development Lifecycles, 2013 IEEE

[12]. Nancy L. Russo, Judy L. Wynekoop, Diane B. Walz; "THE USE AND ADAPTATION OF SYSTEM DEVELOPMENT METHODOLOGIES"; International Resources Management Association, International Conference, Atlanta, Georgia, 1995.

.

[13]Olga De Troyer: “Audience-driven Web Design”, Updated version published in: Information modelling in the new millennium, Eds. Matt Rossi & Keng Siau, IDEA GroupPublishing, ISBN 1-878289-77-2, 2001.

[14] *Steven H. VanderLeest, Calvin College and DornerWorks, Ltd., Grand Rapids, MI Andrew Buter, DornerWorks, Ltd., Grand Rapids, MI “*ESCAPE THE WATERFALL: AGILE FOR AEROSPACE”

[15] Dieter Blomme, Nico Goeminney, Frank Gielen and Filip De Turck; “A Two Layer Approach for Ubiquitous Web Application Development”; IEEE 2009: Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns.

[16] Hu Dong, Yu Xue; “Designing and implantation of agile framework based on Lift”; IEEE 2010.

[17] Hu Ran, Wang Zhuo, Hu Jun, Xu Jianfeng, Xie Jun; “Agile Web Development with Web Framework”; IEEE 2008.

[18] Paolucci. M, Kawamura.T, Payne.T.R, Sycara.K.; “Semantic Matching of Web Services Capabilities”; International Semantic Web Conference, 2002.

[19] Web application. http://en.wikipedia.org/wiki/Web_application. Retrieved 7th May, 2013.

[20] Web development that doesn’t hurt; http://rubyonrails.org/ ; Retrieved 13th May, 2013.