IJIRR

International Journal of Information Research and Review
Vol. 02, Issue, 11, pp. 1327-1357 November, 2015

Research Article

HYBRID MEHODLODGY (HM) FOR DEVELOPING WEB-BASED SYSTEMS

[1,*]Ahmed Hassan Mohamed Ali and [2]Mohammad Nazir Ahmad

[1]Department of Computer Science and Information Technology, University of Medical Sciences and Technology, Khartoum, Sudan
[2]Department of Computing, Universiti Teknologi Malaysia, Malaysia

ARTICLE INFO

Article History:

Received 22th, August 2015
Received in revised form
16th, September 2015
Accepted 25th, October 2015
Published online 30th, November 2015

Keywords:

web-based Systems,
Software Engineering Models,
and Web Engineering Models

ABSTRACT

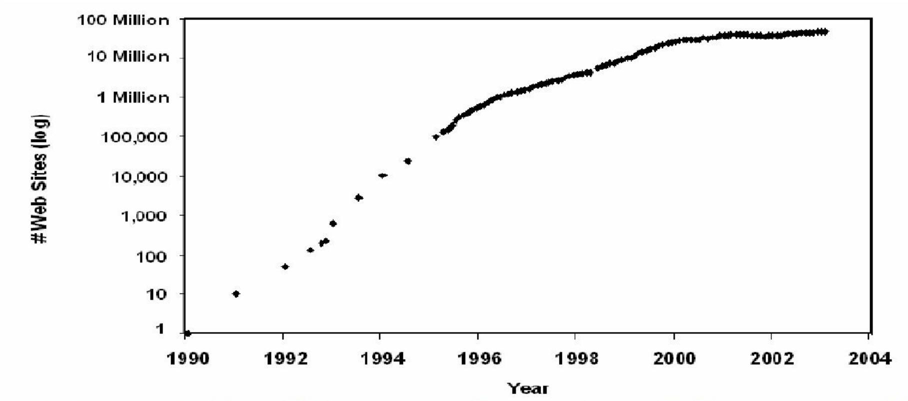
With the increasing expansion of web-based systems in their use, as well as the increase in the complexity of their development, it has been found that the conventional software engineering models such as for example Waterfall model, Prototyping, Incremental, Spiral, Rational Unified process (RUP) and Extreme programming (XP) cannot be used directly or not applicable for the development of web-based systems. As a result of failure of software engineering models in developing web-based systems several web engineering models and methodologies have been developed and tried during the first years of the 21st century to help developers master the complexity of Web application design and development, but they also failed due to nature of web-based systems. In this paper, a hybrid methodology for developing web-based system will be proposed. The methodology will help to ensure successful development of web-based systems and be able to overcome the common problems found in web development methods.

INTRODUCTION

In the beginning, the internet and the WWW were seen as means of communication and sharing information across a widely dispersed audience. As the use of the web moved from the static display of information (such as on-line brochures) to dynamic information provision and to real-time interactive applications, the design of web-based applications presented challenges and opportunities that were not normally encountered in conventional systems (Mohamad Noorman Masrek, Norhayati Hussin et al. 2008). In the past 10 years, the number of Web sites has grown dramatically from 100 to over 45 million (see Figure 1 Growth of websites) (Brandon and Dan 2008). The Web has become closely ingrained with our life and work in just a few years. From its initial objective of facilitating easy creation and sharing of information among a few scientists using simple Web sites that consisted primarily of hyperlinked text documents, the Web has grown very rapidly in its scope and extent of use, supported by constant advances in Internet and Web technologies and standards.

Web-based systems currently is essential for business operation, marketing, and strategy (Lam 2011). Enterprises, travel and hospitality industries, banks, educational and training institutions, entertainment business and governments use large-scale web-based systems and applications to improve, enhance and/or extend their operations. E-commerce has become global and widespread. Traditional legacy information and database systems are being progressively migrated to the Web. Modern web-based systems run on distributed hardware and heterogeneous computer systems. Furthermore, fuelled by recent advances in wireless technologies and portable computing and communication devices, a new wave of mobile Web applications are rapidly emerging. The Web has changed our lives and work at every level, and this trend will continue for the foreseeable future (Brandon and Dan 2008). Due to the quick evolvement of Web-based systems and applications in their scope and extent of use, their performance, reliability and quality have become very important and essential. At the same time, the developmental processes of web-based system did not progress at a sufficient rate to meet the challenges and demands of this evolution (Brandon and Dan 2008).

Corresponding author: Ahmed Hassan Mohamed Ali,
Department of Computer Science and Information Technology, University of Medical Sciences and Technology, Khartoum, Sudan.



Note: Web Sites = Number of Web servers; one host may have multiple sites by using different domains or port numbers.

Source: Hobbes' Internet Timeline, 2004, www.zakon.org/robert/internet/timeline/

Figure 1. Growth of websites

Many Web developers think that web-based system development is just a process of creation of Web page that contains few images and hyperlinks to other Web pages or documents such as personal Web pages, seminar announcements and simple online company brochures (Murugesan and Ginige 2005). There is more to Web-Based systems development than visual design and user interface; it involves planning, Web architecture and system design, testing, and quality assurance and performance evaluation. It also evolves continual update and maintenance of the systems as the requirements and usage grow and develop (Brandon and Dan 2008).

Web developers failed to address users' needs and issues such as content management, maintenance, performance, security, and scalability of Web applications (Suh 2005). Developing web-based systems in ad hoc manner causes a lot of problems such as poor quality web-based systems and applications, outdated or irrelevant information, difficulties in using the Web site and finding relevant information of interest, slow response, Web site crashes, and security breaches are common (Suh 2005). There is a survey conducted which has identified the following problems in developing web-based systems (Brandon and Dan 2008):

- 63% of projects exceeded the budget.
- 79% of projects were behind schedule.
- 53% of delivered systems did not have required functionality.
- 84% of delivered systems did not meet business needs.
- 52% of deliverables were of poor quality.

However, solutions and processes have been proposed but they have not managed to change the web development approach. The reasons are many but the most important ones are:

- Most of the proposed methodologies internally are very complex, need specialized training and to be presented in relatively short academic papers, which in general most developers or software houses do not read or follow (Bolmeson 2005).
- Methods of Web-based systems development are not universally applicable and have not been satisfactorily tested in live situations (Sherry Jeary and Keith Phalp 2009; Selmi, Semia Sonia et al. 2011).
- There are cruel problems in adapting traditional software approaches to web development (Sherry Jeary and Keith Phalp 2009).
- Most of the methodologies lack well-defined process or quality control (Brandon and Dan 2008; Selmi, Semia Sonia et al. 2011).
- Flawed design and development process (Al-allaf 2008; Selmi, Semia Sonia et al. 2011).
- Poor understanding of methodology to develop web-based systems (Al-allaf 2008).
- Most of methodologies do not support all of the Web engineering process model phases (Domingues, André L.S. et al. 2008).
- Inadequate testing and focus on implementation with amalgamation of requirements capturing and designing (Kushwaha, Dharmender Singh et al. 2006).

Therefore, the need for a sound, structured and repeatable process is unquestionable – high performance and business critical web-based systems are very hard to develop without proper processes and activities – not if a certain degree of quality is to be sustained and the systems delivered on time and satisfying the requirements.

Background and literature review

Web-based systems are those systems that deliver to users through Internet, intranet and extranet. The Internet is a worldwide collection of interconnected networks. An intranet is a private network inside a company using web-based applications, but for use only within an organization. An extranet is a private network that allows external access to customers and suppliers using web-based applications (Aranda 2007; Ingle and Meshram 2012).

Web-based systems can be categorized in many ways - there is no unique or widely accepted way. Categorization of web-based systems based on functionality (Table 1) is useful in understanding their requirements and for developing and deploying web-based systems and applications (Suh 2005).

Table 1. Categories of web-based systems based on functionality. Adapted from (Suh 2005)

Functionality/Category	Examples
Informational	Online newspapers, product catalogues, newsletters, manuals, reports, online classifieds, online books.
Interactive	Registration forms, customized information Presentation, online games.
Transactional	Online shopping (ordering goods and services), online banking, online airline reservation, online payment of bills.
Workflow oriented	Online planning and scheduling, inventory Management, status monitoring, supply chain management.
Collaborative work environments	Distributed authoring systems, collaborative design tools.
Online communities, marketplaces	Discussion groups, recommender systems, online marketplaces, e-malls (electronic shopping malls), online auctions, intermediaries.

There is no hesitation that the majority of information systems to be developed in the future will be web-based even for internal purposes and that is coming from two reasons (Ziemer, 2007; 2009):

- **Web-based systems are more accessible:** The HTTP used in web-based systems is a standard protocol that can travel across corporate firewalls. The only client software a user need is a web browser. Also, web-based systems are available on many platforms. Web browsers are packaged with most operating systems these days.
- **Web-based systems have a lower maintenance and deployment costs:** Since Web-based systems are running in web browser, they do not depend on installing client software on each user's computer. Web-based systems can be maintained by modifying code that resides on a server. This reduces the time and the cost of upgrade and deployment of web-based systems compared to traditional client/server applications.

Developing web-based systems is significantly different from traditional software development and poses many additional challenges (Holck 2003). There are obvious differences in the nature and life cycle of Web-based and software systems and the way in which they're developed and maintained. Most researchers agree that web-based systems are different from most other types of software systems, and there is an urgent need for the disciplined and systematic approaches to the successful development, deployment and maintenance of high quality Web-based systems (Zhou and Stålhane 2004) and It has been found that the conventional software engineering models such as for example Waterfall model, Prototyping, Incremental, Spiral, Rational Unified process (RUP) and Extreme programming (XP) cannot be used directly or not applicable for the development of web-based systems (Kumar and Sangwan 2011) . This is because Web applications have unique characteristics which are: (Pressman and Lowe 2009)

- **Network intensiveness:** It resides on a network and must serve the needs of a diverse community of clients.
- **Concurrency:** Large number of users can access the Web application at the same time.
- **Unpredictable load:** The number of users of the Web application may vary by orders of size from day to day.
- **Performance:** Users of Web application should not wait long time for Web application response.
- **Availability:** Users of Web application often demand access on "24/7/365" basis.
- **Data driven:** In many cases, the primary function of a Web application is to use hypermedia to present text, graphics, audio, and video content to the end user.
- **Content sensitive:** The quality and aesthetic nature of content remains an important determinant of the quality of Web application.
- **Continuous evolution:** It is not unusual for some Web application (specifically, their content) to be updated on an hourly schedule.
- **Immediacy:** The time to market for a complete Web site can be a matter of a few days or weeks.
- **Security:** In order to protect sensitive content and provide secure modes of data transmission, strong security actions must be implemented throughout the infrastructure that supports a Web application and within the application itself.
- **Aesthetics:** (Lavie and Tractinsky 2004) called the sophisticated use of web technology "web-based aesthetics". For example, researchers have identified web- based aesthetics as often including advanced graphics and multimedia features such as sound, animation and video streaming (Bansler 2000; Kautz 2007).

Table 2 highlights some weaknesses of familiar software engineering models in developing web-based systems.

Table 2. Software Engineering models

Model	Weaknesses
Waterfall	• Cannot respond to rapid changing in requirements (Howcroft and Carroll 2007). • Inflexible (Munassar and Govardhan 2010). • System performance cannot be tested until the system is nearly fully coded (Munassar and Govardhan 2010). • Problems are often not discovered until system testing (Munassar and Govardhan 2010). • Difficult and expensive to make changes (Silberberg 2006).
Prototyping	• Requirements may frequently change significantly (Centers for Medicare & Medicaid Services 2008). • Can lead to poorly designed systems (Howcroft and Carroll 2007). • Identification of non-functional requirement is difficult to document (Centers for Medicare & Medicaid Services 2008).
Incremental (Centers for Medicare & Medicaid Services 2008)	• There is usually lack of overall consideration of the business problem and technical requirements for the overall system. • Cannot respond to rapid changing in requirements. • Did not support all web engineering frame work processes.
Spiral	• Need skilled and experienced project manager (Centers for Medicare & Medicaid Services 2008). • Can be a costly model to use (Silberberg 2006). • Doesn't work well for smaller projects(Munassar and Govardhan 2010).
XP	• XP concepts need some adaption to support web engineering frame work processes (Ali 2008). • Needs experience and skill if not to degenerate into code-and-fix (Munassar and Govardhan 2010). • Difficult to scale up to large projects (Munassar and Govardhan, 2010).
RUP (Kappel 2006)	• Did not support short development cycles. • Cannot respond to rapid changing in requirements.

As a result of failure of software engineering models in developing web-based systems and at the same time with the increasing expansion of Web applications in their use, as well as the increase in the complexity of their development, several WebE models and methodologies have been developed and tried during the first years of 21st century to help developers master the complexity of Web application design and development. Examples of them are (Moreno and Vallecillo 2008):

- Hypertext Design Model (HDM) (Garzotto 1993).
- The Relationship Management Methodology (RMM) (Isakowitz 1995).
- Object Oriented Hypermedia Design Model (OOHDM) (Rossi 1996).
- Web-Composition Process Model (WCPM) (Ali 2008).
- Hypermedia Model Based on Statecharts (HMBS) (Carvalho 1999).
- Object-Oriented Hypermedia (OO-H) (G´omez 2001).
- Web Application Extension (WAE) (Conallen 2003).
- Simple Web Method (SWM) (Griffiths 2002).
- Web-Oriented Conceptual Models (OOWS) (Fons 2003).

WebE modeling methods are categorized into four groups (Kappel 2006)

- **Data Oriented Methods:** These are originated from the field of database systems, and mainly based on ER model. The primary focus of these methods is the modeling of database-driven web applications, for Examples RMM, HERA, WebMl.
- **Hypertext-Oriented Methods:** HOM focuses on hypertext character of web applications. These methods include HDM, later HDM extended to W2000 and WSDM.
- **Object Oriented Methods:** These methods are based on either OMT or UML. This category includes OOHDM, OOWS and OO-H.
- **Software Oriented Methods:** It uses techniques that strongly follow classical software engineering. An example of this category is WAE.

Table 3 below summarizes some of the most accepted and common WebE models' and methodologies' main technique, notation, tool and some of it weaknesses.

From the above literature we can conclude that

- There are severe problems in adapting traditional software approaches to web development.

- Most of web methods are not supporting all WebE model processes and activities in specific formulation activities, validation and verification activities.
- Most of web methods are not easy to use and need specialized training.
- Most of web methods not giving concerns to the web-based system quality issues that are defined by ISO/IEC 9126 (ISO/IEC9126 2001) (Lang and Fitzgerald 2005). This fact coming from the definition of WebE as "the application of systematic, disciplined and quantifiable approaches to the cost-effective development and evolution of high-quality applications in the World Wide Web" (Heuser 2004). One possible reason for this situation is that, being the final objective of any quality evaluation process the quality in use (meeting user needs) (ISO/IEC9126 2001), and given the fact that assessing quality in use means tracking the use that real users make of the application under real exploitation conditions. The WebE field has traditionally considered such concerns out of its scope.
- Most of web methods focus on the design workflow in the life cycle, while other tasks like requirements engineering, tests and quality management are handled with less relevance or not included at all (Vidgen 2002; ESCALONA 2004).
- Our main aim in this research was to formulate a suitable, HM for developing web-based systems that is easy to learn, to use and to implement.

Table 3.A Comparison of Common WebE Models

Methodology	Technique	Notation	Tool	Weaknesses
HDM	Entity relationship	Entity relationship	N/A	Not supporting all WebE model processes (Ali 2009).
RMM	Entity relationship	Entity relationship and own	N/A	Not supporting all WebE model processes (Domingues, André L.S. et al. 2008). Need specialized training (Ali 2008).
OOHDM	Object oriented	Unified modeling language (UML) and own	OOHDM-web	Not supporting all WebE model processes (Ali 2009). Not easy to use (Ali 2008)
HMBS	Object oriented and states	Statecharts and own	HySCharts and Web-Scharts	Not supporting all WebE model processes (Ali 2009)
OO-H	Object oriented	UML and own	CASE tool	Not supporting all WebE model processes (Domingues, André L.S. et al. 2008)
WAE	Object oriented	UML and own	N/A	Not supporting all WebE model processes (Domingues, André L.S. et al. 2008)
SWM	Structured Analysis (SA).	DFD	ASCENT	Not supporting all WebE model processes (Domingues, André L.S. et al. 2008)
OOWS	Object oriented	UML and own	OlivaNOVA	Not supporting all WebE model processes (Ali 2009)

This aim was achieved by addressing the following specific objectives

- Identify and conduct a comparative study on four of the most commonly web-based development methodologies used in industry.
- Determine and categorize the problems faced by developers when developing web- based systems.
- Propose HM for developing web-based systems that can handle the special nature of web-based systems and can address the critical problems faced by developers when developing web-based systems.
- Experiment and evaluate the proposed development methodology in at least one company.

MATERIALS AND METHODS

HM is developed based on Peffer's DSRM and following design research guidelines. DSRM is providing a commonly accepted framework for successfully carrying out DS research and a mental model for its presentation. It should help researchers to present research with reference to a commonly understood framework, rather than justifying the research paradigm on an ad hoc basis with each new paper.

The DSRM includes the followings activities (Ken et al., 2007)

- **Activity 1:** Problem identification and motivation. Define the specific research problem and justify the value of a solution. Explaining the value of the solution lead to two things: it encourage the researcher and the audience of the research to accept the results and it useful to understand the reasoning associated with the researcher's understanding of the problem.
- **Activity 2:** Define the objectives for a solution. Drive the objectives of a solution from the problem definition. The objectives can be quantitative, or qualitative. The objectives should be concluded and linked rationally from the problem identification.

- **Activity 3:** Design and development. Create the artifact. Artifacts can be constructs, models, methods, or instantiations. Research artifact can be any object that contains the research contribution. This activity includes determining the artifact functionality and then creating the actual artifact.
- **Activity 4:** Demonstration. Implement the artifact to solve one or more instances of the problem. Demonstration can be in form of experimentation, simulation, case study, proof, or other appropriate activity.
- **Activity 5:** Evaluation. Measure how well the artifact supports a solution to the problem. This activity includes comparing the objectives of a solution to actual results from use of the artifact in the demonstration.
- **Activity 6:** Communication. This activity includes publishing papers demonstrating the importance of the solved problem and communicating the artifact, its utility and novelty to the research community.

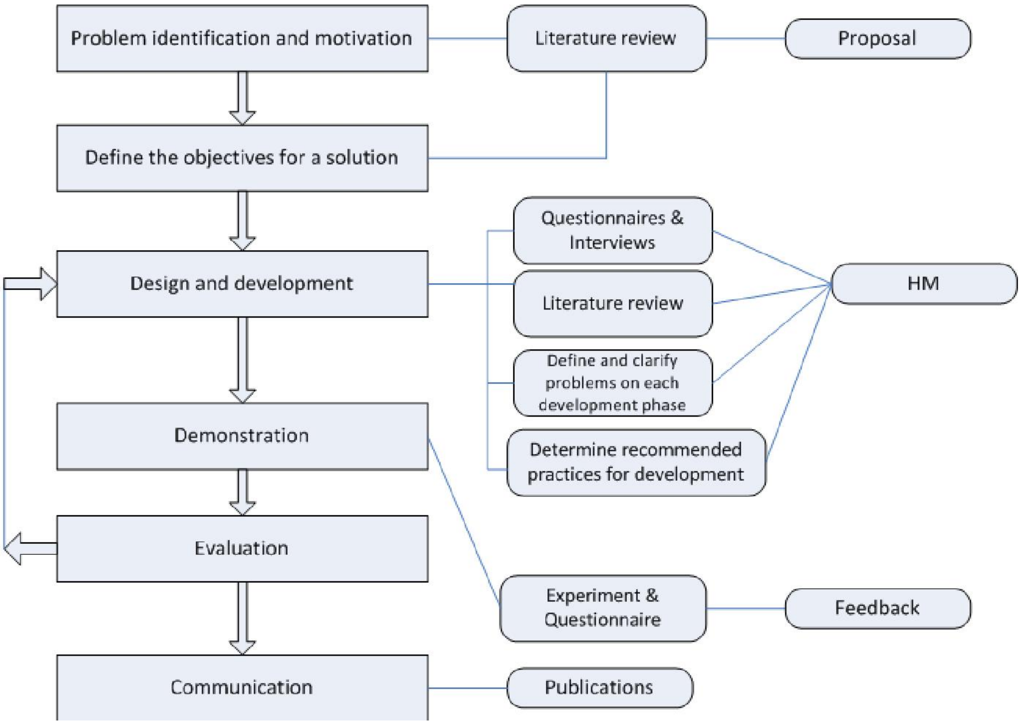

Problem identification and motivation
Literature review
Proposal
Define the objectives for a solution
Questionnaires & Interviews
Design and development
Literature review
HM
Define and clarify problems on each development phase
Determine recommended practices for development
Demonstration
Evaluation
Experiment & Questionnaire
Feedback
Communication
Publications

Figure 2. Research design

Problem Identification and Motivation

Web-based systems currently provide large number of diverse groups very complex functionalities and dependence on the web-based systems has increased. Hence, their performance, reliability and quality have become very critical and the development of web- based systems has become more complex and challenging.

On the other hand, conventional software engineering methodologies failed to develop web-based systems because the nature of web-based system is different from traditional software. Several Web Engineering models and methodologies have been developed and tried during the first years of 21st century as a result of failure of software engineering models in developing web-based systems, but they also failed to change development approach of web-based systems due to many reasons.

Define the Objectives for a Solution

The research general objective is to formulate a suitable, HM for developing web-based systems that is easy to learn, to use and to implement.

The research specific objectives were as follows:

- To identify and conduct a comparative study on four of the most commonly web- based development methodologies used in industry.
- To determine and categorize the problems faced by developers when developing web-based systems.
- To propose a hybrid methodology (HM) for developing web-based systems that can handle the special nature of web-based systems and can address the critical problems faced by developers when developing web-based systems.
- To test and evaluate the proposed development methodology in at least one company.

Design and Development

It was the intention of this research to involve companies that develop web-based systems and to focus on industrial development practices. This was to ensure that the research dealt with real world situations and that the research results would be relevant for industrial web- based systems developers. Contact was made with several companies to learn how they develop web-based systems and to find problems and challenges these companies are struggling with. Web-based systems are developed in many areas and for different purposes. The interest of this research was in large companies that develop advanced web-based systems. Hence, companies developing this type of web-based systems were the preferred choice for the research. To achieve research objective one, we conducted an intensive literature review. From the information gathered we concluded on what are the four most commonly used web development methodologies in the industry. Then to investigate the problems faced by developers when developing web-based systems in objective two, two questionnaires was used and our population was companies that undertaken development of web-based systems. The first questionnaire (see Appendix 1 Companies Characteristics Questionnaire Results) was distributed to thirty-nine companies that are developing web-based systems in order to know their main characteristics (see Appendix 1: Companies Characteristics Questionnaire Results).

The characteristics of large companies that are determined from questionnaire one are

- Company age more than 4 years in developing web-based systems and the development held inside the company.
- Has more than 50 employees and 5 or less developer involve in development.
- Develop all types of web-based systems (static, internet, intranet and extranet).
- Develop more than 200 web page or web form.
- Measure the size of web-based system by number of requirements, time needed for development and number of developers involved.

The second questionnaire (see Appendix 2: Web Development Questionnaire Results) was used to determine and identify the development methodologies that are used by these companies and the problems that developers are facing during development of web-based systems (see Appendix 2: Web Development Questionnaire Results).

The analysis of the second questionnaire shows the following results:

- Most of web engineering methodologies is not used in developing web-based systems.

The reasons are many but the most important reasons are

  - Not known.
  - Methods are not easy to understand and learn due to its internal complexity.
  - Methods are not easy to implement.
  - Methods are not support all web engineering process model activities.

- Critical problems on each phase based on second questionnaire results analysis

- **Analysis phase**

  - Lack of good planning
  - Lack of user involvement
  - Requirement gathering techniques

- **Design phase**

  - Lack of user involvement
  - Difficulties with frequent changes in requirements

- **Implementation phase**

  - Developers with various background
  - Low time-to-market

- **Integration phase**

  - Difficulties with frequent changes in requirements

- **Evaluation phase**

- Lack of good planning for maintenance and evaluation
- Low usability

For Objectives three and four we developed a HM that can address web-based development problems. The proposed methodology was tested and evaluated by selecting three companies to experiment on it, and its feedback was used to conclude the research project.

Demonstration

After HM developed, it was tested by selecting three companies to experiment on it, and its feedback was used to evaluate the research project.

Evaluation

The purpose of this phase was to evaluate the proposed development methodology. User's feedback that was collected on the experiments was compared to our research general objective which was to develop hybrid methodology that is:

- Easy to learn, to use and to implement.
- Can handle the special nature of web-based systems.
- Contains the recommended practices for developing web-based systems.
- Support all web engineering model processes and activities.

Communication

Publishing papers and demonstrating the importance of the solved problem is included in this activity.

Statistical Methods used in Data Analysis

The descriptive statistical techniques were used in the data analysis. Descriptive statistics such as frequencies, percentage were used in the first questionnaire to know the major characteristics of the companies that are developing web-based systems and used in the second questionnaire to determine and identify the common methodologies that were using in development of web-based systems, the main problems and its significance on each phase of development and what are the problems that are expecting the proposed methodology (HM) to solve based on the problems significance.

Hybrid Methodology

HM is a methodology that built in the intention of to be suited to develop all types of web- based systems. Our main intention during designing the methodology was to include all the recommended practices and activities of developing different types of web-based:

- Parallel development to meet the short Time-to-Market requirement, developers have to try to shorten the time needed to develop a new release. This can only be done by parallel development, meaning that the development team is working on two different releases simultaneously.
- Release orientation in the early phases of a Web Application, releases are made in short cycles. The release cycle can be between 2 and 15 days.
- Tool dependence many Web Application development organizations make heavy use of development tools to speed up the design and coding process.
- Customer involvement since the requirements are evolving during the development, customers are involved intimately in the development effort. Customers are often co-located with the development team, and participate closely in all phases of development.
- Prototyping are used to deal with the unstable and evolving requirements. They are used to agree on requirements, and to receive feedback both from the customer and from the end users. This is possible since the prototypes are published. Prototyping is used as a way to communicate with the customers to validate and refine requirements.
- Increment and iterative concepts should be used in development because of rapid requirements evolving.
- Support all web engineering model processes and activities.

HM is designed to support main web engineering processes and activities, to be flexible enough in order to be able to respond to rapid changes in requirements, to be easy to learn and adapt. It consists of phases to make it manageable and fundamental software engineering activities such as (specification, design, implementation, verification and validation). HM is hybrid because it borrows ideas from other models such as incremental, iterative, prototype, relationship management model (RMM) and object oriented analysis and design due to the nature of web-based systems.

How it Works

HM is divided into three main phases (pre-development, analysis and release phase), each phase contains a set of activities. The activities in each phase are determined according to the web-based system type. In the pre-development phase web-based system aim and objectives are determined as well as development priority for each objective. Then analysis phase start with the most significant objective. Analysis and release phase is repetitive and incremental process, on each iteration or increment one or more than one objective based on its importance or development priority is take to analysis phase to determine its functional and non-functional requirements, database and classes design, user interfaces. Each objective is tied up to it set of requirements.

The idea behind that is to make process of measuring and evaluating web-based system is easy, at the same time it useful in check whether web-based system aim is achieved or not.Relationship management model (RMM) concept is used in analysis phase because it's useful in designing navigational structure of web-based system objective. Object oriented analysis and design concept also is used during the analysis phase in figuring out users requirements and documenting them. After finish from objective analysis phase, release phase is start for that objective. Each objective can be represented by one or more than one release and that is depends on if it contains large number or complex requirements. In the release phase concept of prototype model is used. Prototype is developed based on the currently known requirements. By using this prototype, the client/customer can get an "actual feel" of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired objective.

After finish from the release phase, another objective is take again based on its business priority and start analysis and release phase for it again. Figure 1 shows the HM phases.

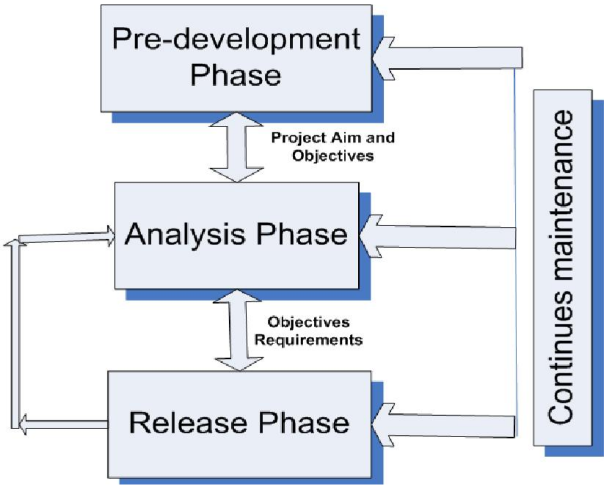


Figure 3 HM phases

Based on the recommended practices of developing web-based system, methodologies of web-based development should support parallel development. Figure 4 shows how HM support parallel development throughout it different phases.

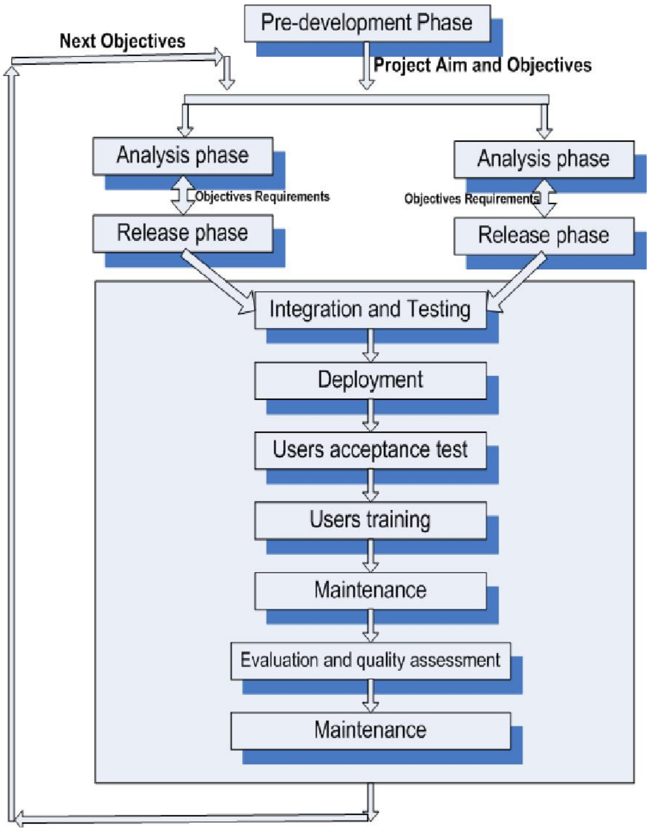


Figure 4. HM process model for parallel development

Hybrid Methodology Phases

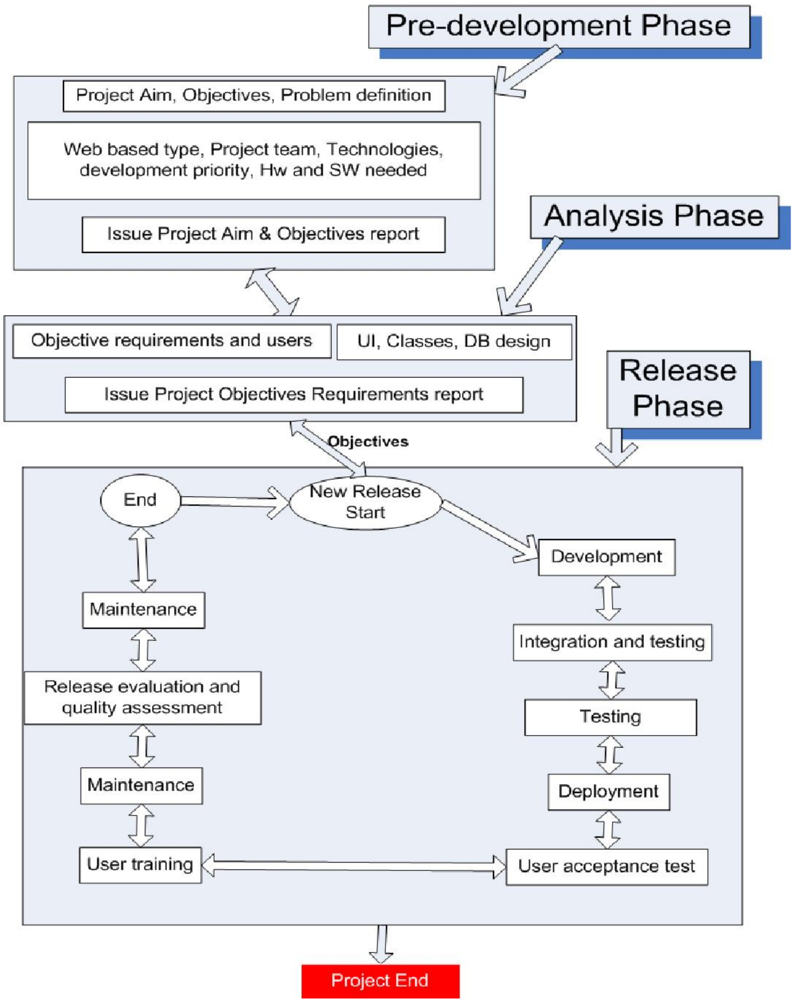

Pre-development Phase
Project Aim, Objectives, Problem definition
Web based type, Project team, Technologies, development priority, Hw and SW needed
Issue Project Aim & Objectives report
Analysis Phase
Objective requirements and users
UI, Classes, DB design
Issue Project Objectives Requirements report
Release Phase
Objectives
End
New Release Start
Development
Maintenance
Integration and testing
Release evaluation and quality assessment
Testing
Maintenance
Deployment
User training
User acceptance test
Project End

Figure 5. HM phases and activities

The methodology is composed from three main phases:

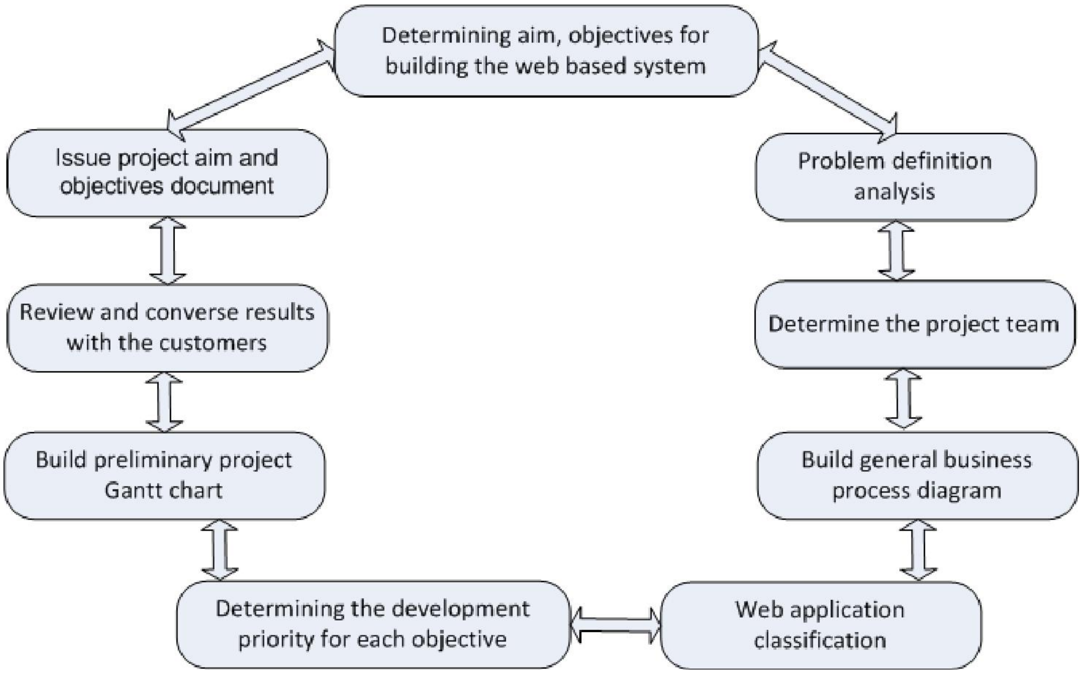

Determining aim, objectives for building the web based system
Issue project aim and objectives document
Problem definition analysis
Review and converse results with the customers
Determine the project team
Build preliminary project Gantt chart
Build general business process diagram
Determining the development priority for each objective
Web application classification

Figure 6. Pre-development phase activities

Pre-development phase

The main aim of the pre-development phase is to build a foundation for the web development project. Web-based system objectives, motivation, purpose, problem statement, web-based system type and development priorities are determined in this phase. The major output from the pre-development phase is the web-based system or project aim and objectives report (see appendix 3: Project aim and objectives report), which considered as input to the analysis phase.

Project aim and objectives report contains the following major elements

- Project aim
- Problem statement
- Project team
- General business process diagram
- Project objectives, each objective has description and development priority
- Project Gantt chart
- Project technical details such as:

  - Web-based system type
  - Technologies will use o Programing languages o Database engine
  - Tools
  - Hardware and software requirements

Pre-development phase includes the following activities for all types of web-based systems

- Determining aim and objectives for building the web-based system
- Problem definition analysis
- Determine the project team
- Build General business process diagram
- Web-based system classification
- Determining the development priority for each objective
- Build preliminary project Gantt chart
- Review and converse results with the customer
- Issue project aim and objectives report

Phase tools

- HM tool or software to save project aim, objectives, project team, business process diagram, web-based system classification, preliminary project Gantt chart and to issue project aim and objectives document
- HM tool or software to track progress of phase different activities
- Microsoft Visio or Open Office draw or Dia or LucidChart or Diagramly or Pencil Project software to draw general business process diagram and preliminary project Gantt chart

Analysis phase

The main aim of the analysis phase is to determine and identify each objective requirements, users, conceptual design of user interfaces, classes and database. In the pre-development phase, objectives development priority is determined. Analysis phase start with the most significant or high development priority objective.

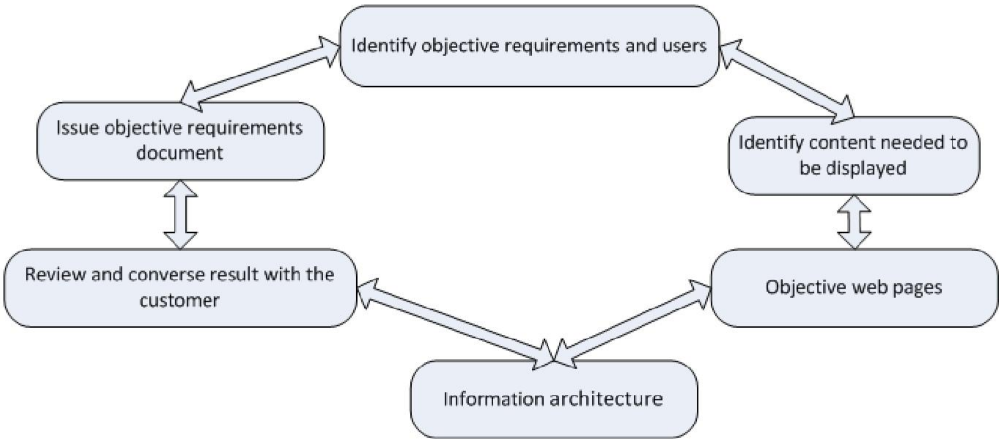

Identify objective requirements and users
Identify content needed to be displayed
Objective web pages
Information architecture
Review and converse result with the customer
Issue objective requirements document

Figure 7. Analysis phase activities for static web-based system

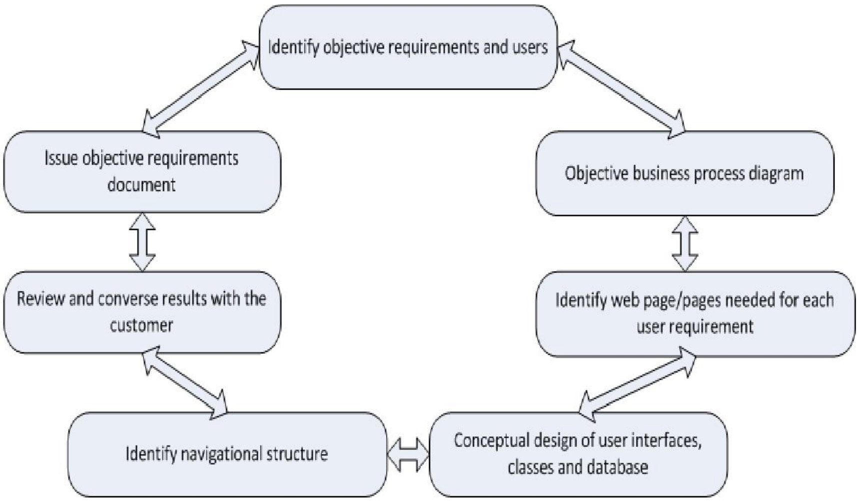

Identify objective requirements and users
Issue objective requirements document
Objective business process diagram
Review and converse results with the customer
Identify web page/pages needed for each user requirement
Identify navigational structure
Conceptual design of user interfaces, classes and database

Figure 8. Analysis phase activities for intranet and internet web-based systems

For each objective in the analysis phase, the following is determined:

- Objective business process diagram, if required
- Objective users
- Objective requirements
- for each objective requirement, the following is determined:
  - description
  - users
  - Pre-condition: What must be true before the use case can be considered;
  - validation
  - Post-condition: What is true after the use case is done, i.e. the state of the system
  - User interface design

The major output of the analysis phase is project objectives reports (see appendix 4: Project objectives requirements report), which used as input to the release phase. Project objectives requirements report contains the following major elements for each objective:

- Business process diagram, if required
- Users of the objective
- User use case diagrams
- Class diagram
- Objective requirements
- For each requirement, the following is determined:
  - description
  - users
  - Pre-condition
  - validation
  - Post-condition
  - User interface design

The activities on the analysis phase are vary based on the type of web-based system. It includes the following activities for static web pages:

- Identify objective requirements and users
- Identify content needed to be displayed in order to achieve phase objective
- objective web pages
- information or content architecture
- Review and converse results with the customer
- Issue objective requirements report

For intranet and internet web-based systems, it includes the following activities:

- Identify objective requirements and users
- Create objective business process diagram
- Identify web page/pages needed for each user requirement

- Conceptual design of user interfaces, classes and database
- Identify navigational structure
- Review and converse results with customer
- Issue objective requirements report

Phase tools

- HM tool or software to track progress of phase different activities and to issue objective requirements report.
- Microsoft Visio or Open Office draw or Dia or LucidChart or Diagramly or Pencil

Project software to draw general business process diagram

Release phase

The release phase features a key step in the project: system construction. The previous phases lay the foundation for system development; the following phases ensure that the product functions as required. The purpose of the release Phase is to convert the system design prototyped in the analysis phase into a working information system that addresses all documented system requirements.

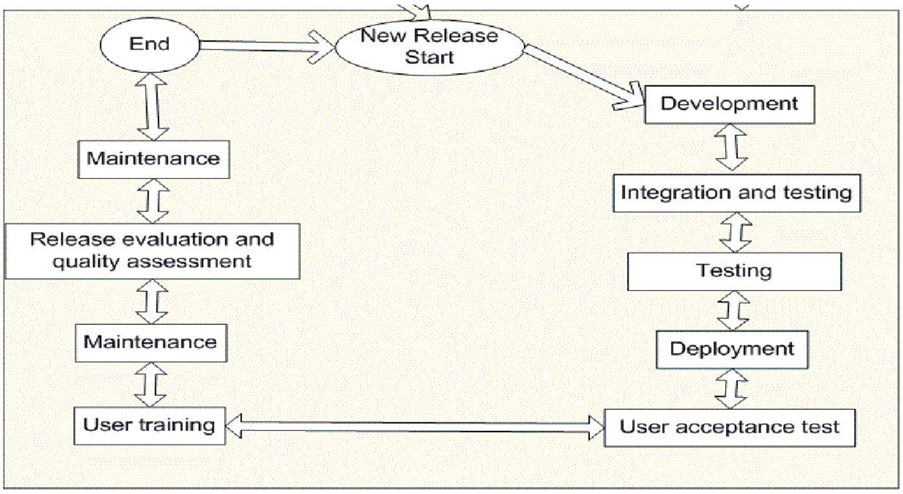


Figure 9. release phase activities for internet and intranet web-based systems

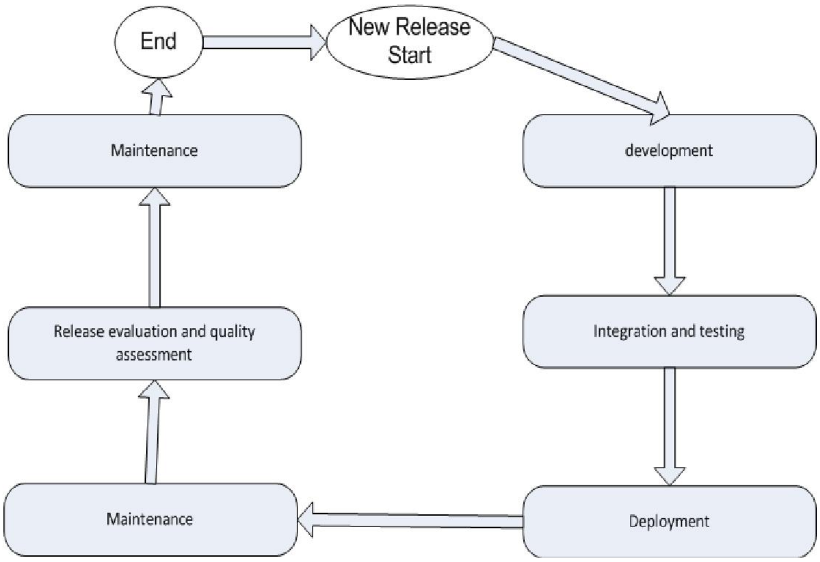


Figure 10. release phase activities for static web pages

During the release phase, the system developer takes the detailed logical information documented in the previous phase and transform it into execu form, and ensure that all of the individual components of the automated application/system function correctly and interface properly with other components within the application/system.

Successful completion of the release phase should comprise

- Building the release system
- Testing and integration the units into larger components
- Preparing the technical environment for the system
- Users testing and training
- System evaluation and quality assessment

The outcome of this phase should be finished release that was tested, reviewed with users, users trained on it, verified against the objective that it was developed for and can be used. The activities on this phase are determined based on web-based system type.

Release phase activities for static web pages

- Development
- Integration and testing
- Deployment
- Maintenance
- Release evaluation and quality assessment
- Maintenance again

Release phase activities for intranet and internet web-based systems:

- Development
- Integration and Testing
- Testing
- Deployment
- User acceptance test
- Users training
- Maintenance
- Release evaluation and quality assessment
- Maintenance again

Phase tools

This phase is tool independent.

Evaluation of HM

HM was experimented in three large companies in developing extranet web-based systems, then questionnaires were distributed to these companies to get their feedbacks and to check:

- Whether HM can handle the special nature of web-based systems.
- Easy to learn, easy to understand and easy to adapt.
- Easy to use.
- Easy to implement.
- Contains the recommended practices for developing web-based systems.
- Support main web engineering model processes and activities.
- Overcome, control, avoid, or solve partially common development problems.

The result of experiments shows that HM can be used to develop web-based systems successfully and contains all recommended practices of developing web-based systems. Table 4 shows the summary of the experiments.

Table 4. experiment questionnaire summary

HM design objective	Percentage
whether HM can handle the special nature of web-based systems	100%
easy to learn, easy to understand and easy to apt	98%
Contains the recommended practices for developing web-based systems	100%
Support all web engineering model processes and activities	100%
Avoid development problems	55%
Eliminate totally development problems	60%
Control development problems	90%
Solve development problems partially	15%

DISCUSSION

HM is divided into three major phases (pre-development, analysis and release), it composes ideas from other models such as increment, iterative, prototype and RMM models. HM was made to support all recommended practices of developing web-based systems. Table 5 shows comparison between HM and recommended practices of developing web- based systems.

The main differences between HM and other models are

- HM divides the whole web-based system project into set of business objectives and then applied its phases and activities on each business objective separately, each business objective link to it set of users requirements. The idea behind using business objectives is to make the process of evaluation is easy and measurable.
- HM focus in customer involvement due to rapid changes in requirements. Thus customers are involved on all its phases.
- HM focus on quality of web-based systems. Because it contains activity to evaluate the quality of web-based system.
- HM contains just three phases in order to meet time to market constraints and make it easy to manage.
- HM borrows ideas from other models to design its phases and activities in order to meet recommended practices of developing web-based system.

Table 5. HM and practices of developing web-based systems

Recommended practices	HM
Support all web engineering model processes and activities	HM contains all main activities of web engineering process model
Consist of a phased approach to the development life-cycle in order to make it manageable	HM contains three phases pre-development, analysis and release phase
Release orientation and support parallel development	HM support release and parallel development
Consist of the fundamental SE-activities specification, design, implementation, verification and validation	HM contains main SE-activities
Be able to encompass other software engineering processes as sub-activities	HM able to encompass other process model as sub-activities since it only is a general framework with different phases and purposed activities
Be flexible enough in order to be able to respond to rapid changes in requirements	HM is flexible, because movement from one activity to another or from one phase to other can be backward and forward
Be easy to learn and adapt	HM phases and activities is easy to learn and adapt
Be using concept of increments, iterative and prototype models due to the nature of web-based systems	HM is hybrid because it borrows ideas from other models
Customer involvement	HM is involving customers in all its phases

Conclusion

The proposed HM was made hybrid due to the unique characteristics of web-based systems. HM borrows ideas from incremental, iterative, prototype, relationship management model (RMM) and object oriented analysis and design in order to design its phases and activities. HM was made to meet and fulfill the recommended practices of developing web-based systems.

HM advantages are

- Its hybrid methodology because it uses ideas of other models such as incremental, iterative, relationship management methodology, prototype and object oriented analysis and design.
- Support parallel analysis and development.
- Does encompass the fundamental and traditional software engineering activities such as analysis, specification, design, implementation and verification and validation.
- Is flexible and responds well to rapid changes in requirement since it is based on the notion of incremental development with different software-increments and prototypes.
- Does consist of phases approach based on the ideas of the traditional evolutionary spiral model.
- Support major web engineering model processes and activities.
- HM is easy to manage because it consists of phases.

- Is able to encompass other process model as sub-activities since it only is a general framework with different phases and purposed activities.
- Is able to encompass other software engineering processes as sub-activities.
- Is decomposable into both different phases and activities. The methodology offers a recommended path to take during the development, but it does not require the user to use it exactly.

Further Work

HM was only experimented in large companies. Small and medium companies that are undertaken in development of Web-based systems have fewer resources and employees than large companies and this factor affect their development capabilities directly. Therefore, HM should be test in both medium and small companies to check any needed modification and HM should also be tested and experimented for developing intranet and internet web-bases systems.

Author contribution

A hybrid methodology for developing web-based system was proposed. The aim of the methodology is to ensure successful development of web-based systems and be able to overcome the common problems found in web development methodologies.

Competing interest

The authors declare that they have no competing interests.

Acknowledgment

This paper research is a part of my PhD thesis work on web engineering and support from university of medical sciences and technology is gratefully acknowledged.

REFERENCES

"This book presents current, effective software engineering methods for the design and development of modern Web-based applications"--Provided by publisher.

Ahmed Ali, and Ahmad, M. 2015. "RECOMMENDED PRACTISES FOR DEVELOPING WEB-BASED SYSTEMS." *International Journal of Development Research 1.*

Al-allaf, O. 2008. "A proposed hybrid web engineering process model for large –scale web-based application development in large web development enterprises". Faculty of Information system and Technology. *Amman-Jordan, The Arab Academy for Banking and Financial Sciences*, PHd: 208.

Ali, A. H. M. H. 2008. "A novel Methodology for Developing Web-based Systems". *International*

Aranda, P. J. V. 2007. Requirements Engineering Approach for the Development of Web Applications. Department of Information Systems and Computation, Technical University of Valencia. PhD: 360.

Bansler, J. 2000. "Corporate Intranet Implementation: Managing emergent technologies and organizational practices." AIS 1: 1-39.

Bolmeson, J. 2005. "Web Development Road Map".

Brandon and Dan 2008. "Software engineering for modern Web applications : methodologies and technologies". Hershey, PA, Information Science Reference.

Centers for Medicare and Medicaid Services, D. O. H. A. H. S. 2008. "SELECTING A DEVELOPMENT APPROACH." 10. *Conference on Software Engineering Theory and Practice,* (SETP08). Orlando, FL, USA.

Convergence, 2009. ICNM 2009. International Conference on.

Domingues, André L.S. et al. 2008. "Web application development methods: a comparison". Brazilian

Engels, G. 2004. Chapter 10. Entwiklingsprozess von Web-Anwendungen: 239–263.

Holck, J. 2003. "4 Perspectives on Web Information Systems". 36th Hawaii International Conference on

Howcroft , D. and J. Carroll 2007. "Proposed Methodology for Web Development". IS Research Centre, University of Salford.

Ingle, D. and D. B. B. Meshram 2012. "Hybrid Analysis and Design Model for Building Web Information

Kappel, G. 2006. Web engineering : the discipline of systematic development of web applications. Hoboken, NJ, John Wiley & Sons.

Kautz, K. 2007. "Persistent problems and practices in information systems development." *Information Systems*, 17: 217–239.

Kumar, S. and S. Sangwan 2011. " *Adapting the Software Engineering Process to Web Engineering* Process." International *Journal of Computing and Business Research*, 2(1 2011).

Kushwaha, Dharmender Singh, et al. 2006. "Cognitive Web-based Software Development Process: Towards a more Reliable Approach". ACM SIGSOFT Software Engineering Notes. NY, USA, ACM. 31.

Lam, M. 2011. "Methodologies, tools, and techniques in practice for web application development." *Journal of Technology Research,* 3.

Lavie, T. and N. Tractinsky 2004. "Assesing dimensions of perceived visual aesthetics of web sites." *International Journal Human-Computer Studies*, 60: 269-298.

Mohamad Noorman Masrek, Norhayati Hussin, et al. 2008. "An exploratory study on systems development methodologies for web-based applications." Information Management & Computer Security 16(2): 137-149.

Munassar, N. M. A. and Govardhan, A. 2010. "A Comparison Between Five Models Of Software Engineering." IJCSI *International Journal of Computer Science,* 7(5).

Murugesan and Ginige 2005. Web Engineering:Introduction and Perspectives, Idea Group Inc.

Pressman, R. S. and D. B. Lowe 2009. "Web engineering : a practitioner's approach". Boston, McGraw- Hill Higher Education.

Ramesh, B. 2002. "Internet software engineering: A different class of processes." Annals of Software Engineering 14(4): 169–195.

Selmi, Semia Sonia, et al. 2011. Guidance in Web Applications Design. MoDISE-EUS. Montpellier - France.

Sherry Jeary and Keith Phalp (2009). "An evaluation of the utility of web development methods." Springer, Software Quality 17(2): 125-150.

Silberberg, R. 2006. "An investigation into methods used to develop software systems." Elektron.

Suh, W. 2005. "Web engineering : principles and techniques". Hershey, PA, Idea Group Pub.

Symposium on Multimedia and the Web (Webmedia), Brazil.

System Sciences, Big Island, Hawaii, USA, IEEE Computer Society.

System." IJCSI International Journal of Computer Science Issues 9(4).

Technical session SW. 2009. Software and web engineering [breaker page]. Networking and Media

Zhou, J. and T. Stålhane, 2004. "Web-Based System Development: Status in the Norwegian IT Organizations." Norwegian IT Organizations.

Ziemer, 2007. "An Architecture for Web Applications." Essay in DIF 8914 Distributed Information Systems.

Ziemer, 2004. Web Engineering. Computer Science Graduate Student Conference 2004. Norway.

Appendixes

Appendix 1 Companies Characteristics Questionnaire Results

How old is your company?

Table 4.1 Company age

	Frequency	%
less than 3 years	7	17.9
3 - 6 years	5	12.8
6 - 10 years	6	15.4
more than 10 years	21	53.8
Total	39	100.0

Table 4.1 shows the age range of companies where more than half of the total respondents fall on the range of more than 10 years, while only nearly 18% of respondents are less than 3 years.

Is your company national or an international company?

Table 4.2. Company work scope

	Frequency	%
National	26	66.7
International	13	33.3
Total	39	100.0

Table 4.2 evidently shows that majority of the respondents are local companies and only 33% are international.

How many employees are there in your company?

Table 4.3. Number of employees

	Frequency	%
10 or less	8	20.5
11 – 25	5	12.8
26 – 49	6	15.4
50 and more	20	51.3
Total	39	100.0

Table 4.3 shows the number of range of employees in a company where 51.3% are with more than 50 employees whereas almost 21% of respondents are with 10 or less employees higher than companies with employees ranging 11-25 or 26-49.

Did your company develop web-based systems?

Table 4.4. Development of web-based systems

	Frequency	%
No	3	7.7
Yes	36	92.3
Total	39	100.0

Tables 4.4 An affirmative response of majority of respondents of 92.3% developed web-based systems, whereas only few with almost 8% respondents have not developed web-based system.

How many year/s has your company been developing web-based systems?

Table 4.5. Company age in development of web-based systems

	Frequency	%
less than 1 year	6	15.4
1 - 2 years	4	10.3
2 - 4 years	10	25.6
more than 4 years	19	48.7
Total	39	100.0

Table 4.5 shows the range of time where almost half of the total respondents have been developing web-based systems in more than 4 years time, whereas the lowest respondents of 10% have developed the system within 1-2 years in comparison with the rate 15.4% of newly opened companies within less than a year.

Is the development process of web-based system accomplished only in your company?

Table 4.6. Place of development

	Frequency	%
Inside	19	48.7
Outside	15	38.5
In and out	5	12.8
Total	39	100.0

Table 4.6 shows that almost half of the total respondents accomplished the development of web-based system inside the company, in comparison with 38.5% accomplished outside the company, whereas only 12.8% were accomplished in and out.

What is the number of web pages developed by your company?

Table 4.7. Number of web pages developed

	Frequency	%
50 or less	15	38.5
51 – 100	10	25.6
101 – 200	5	12.8
more than 200	9	23.1
Total	39	100.0

Table 4.7 shows the range number of web pages developed by the company, where only 23% have developed more than 200 web-pages lesser rate than with 15% of respondents that have developed number of web pages of 50 or less only.

How does your company measure the size of web-based system?

Table 4.8. Web-based systems size measurement technique

Factor	Always	Ofte	Rarely	Neve
Number of requirements	28	2	3	6
Number of developers	9	9	10	11
Number of programming languages used to program the web-based system	8	8	8	15
Size of Budget	13	10	6	10
Time required to develop the web-based system	20	6	5	8
Number of lines (code) in program	8	5	11	15
Number of web pages	7	13	8	11
Number of companies to develop the web-based system	4	5	12	18

Table 4.8 obviously shows that with all of the above factors the "number of requirements" with the highest number of respondents of 28 were always been used and also the least factor that were often and rarely used to measure web-based size, whereas the factor of "number of companies to develop the web-based system" were never used with 18% of respondents among the above-mentioned factors.

How many web-based system developers are there in your company?

Table 4.9 Number of developers

	Frequency	%
5 or less	25	64.1
6 – 10	8	20.5
11 – 20	6	15.4
Total	39	100.0

Table 4.9 shows the range number of web-based systems developers where the least of respondents of 15.4% having 11-20 developers, whereas respondents with 5 or less developers with high respondents of 64.1%.

What is the typical size of a web-based systems developed by your company? (in term of lines of code)

Table 4.10. Typical size of web-based systems developed (In term of lines of code)

	Frequency	%
less than 5000 lines	8	2.50
5 001 – 10 000 lines	15	38.5
10 001 – 50 000 lines	9	23.1
50 001 – 100 000 lines	2	5.1
100 001 – 250 000 lines	2	5.1
250 001 – 500 000 lines	1	2.6
500 001 – 750 000 lines	2	5.1
more than 1000000 lines	39	100.0

Table 4.10 shows the range of size of web-based systems based on line of code where almost 39% of the respondents have the typical size of 5001-10000 lines, whereas the least rate of almost 3% have only the size of 50 0001-750 000 lines.

Does the IT unit use any development models, methodologies?

Table 4.11 Use of development models, methodologies by IT unit

	Frequency	%
No	18	46
Yes	21	54
Total	39	100.0

Table 4.11 shows that more than half (54) of IT units used development methodologies in developing systems.

Does the company send its IT staff for training to update their development skills

Table 4.12 IT staff training for skills update

	Frequency	%
No	18	46
Yes	21	54
Total	39	100.0

Table 4.12 shows that 54% of IT staff send for training to update their skills.

What are programming languages and tools used by developers?

Table 4.13 Programming languages and tools used by developers

.

Programming languages	No. of users
C#	24
Java	25
Vb.Net	21
Asp.Net	26
Php	22
Joomla	7
Jsp	7
Ado.Net	14
Microsoft Sql server	27
Oracle	18
Ajax	18
Java Script	25
Dream weaver	10
Visual Studio.Net	22
Microsoft visio	11
Adobe Photoshop	20

Table 4.13 obviously shows that Microsoft Sql server with the highest respondents of 27 users was the program language popularly used, followed by the ASP.Net, Java script and C#, whereas the least used were the Joomla and Jsp with 7 users only.

Appendix 2. Web Development Questionnaire Results

Which development methodology does your company use in developing web-based system?

Table 4.14 The uses of development methodology in web-based system

	Frequency	%
Ad Hoc (No-methodology)	6	28.6
Waterfall	5	23.8
Incremental	1	4.8
Spiral	1	4.8
SCRUM	3	14.3
Others	5	23.8

Table 4.14 obviously shows that Ad Hoc was used by 29% respondents more than other methods, whereas the least used methodology were incremental and spiral with only
4.8% of respondents in developing we-based system.

In your opinion, is the methodology that you have been using suitable for developing all types of web-based systems (static, internet, intranet & extranet)?

Table 4.15 Suitability of the type of web-based system used

Type	Strongly disagree F (%)	Disagree F (%)	Fair F (%)	Agree F (%)	Strongly agree F (%)	Total F (%)
Static web pages	1 (6.7)	-	1 (6.7)	-	13 (86.7)	15 (100)
Internet web-based system	4 (26.7)	-	1 (6.7)	2 13.3)	8 (53.3)	15 (100)
Intranet web-based system	3 (20)	-	3 (20)	1 (6.7)	8 (53.3)	15 (100)
Extranet web-bases system	5 (33.3)	-	1 (6.7)	1 (6.7)	8 (53.3)	15 (100)

F=frequency

Table 4.15 shows that with all types of web-based system the Static web pages with almost 87% of respondents strongly agreed with its suitability for development of web-based system comparing with the other types that has same rate of respondents of 53.3%, whereas almost 33.3% strongly disagreed with the Extranet web-based system type, while 20% suits Intranet fairly, and only few with 13.3% suits with Internet web-based system.

What is your opinion on the current methodology that you are using?

Table 4.16. Opinion on the current methodology used

Methodology has...	Strongly disagree F (%)	Disagree F (%)	Fair F (%)	Agree F (%)	Strongly agree F (%)	Total F (%)
Isufficient well-defined process	4 (26.7)	2 (13.3)	-	1 (6.7)	8 (53.3)	15 (100)
Isufficient quality control	6 (40.0)	-	1 (6.7)	1 (6.7)	7 (46.7)	15 (100)
sufficient focus on implementation with amalgamation of requirements capturing and designing	5 (33.3)	2 (13.3)	4 (26.7)	-	4 (26.7)	15 (100)
supports all of the Web engineering process model phases	10 (66.7)	-	3 (20.0)	-	2 (13.3)	15 (100)
consists of the fundamental software engineering activities (specification, design, implementation, verification and validation)	2 (13.3)	2 (13.3)	4 (26.7)	1 (6.7)	6 (40.0)	15 (100)
consists of phases to the development life cycle	4 (26.7)	-	3 (20.0)	4 (26.7)	4 (26.7)	15 (100)
Sufficient focus on implementation with amalgamation of requirements of capturing and designing	5 (33.3)	2 (13.3)	4 (26.7)	-	4 (26.7)	15 (100)
Uses concept of increments and spiral	6 (40.0)	-	3 (20.0)	2 (13.3)	4 (26.7)	15 (100)

F=frequency

Table 4.16 shows that the highest percentage of almost 67% of respondents strongly disagreed with the use of the methodology has supports all of the web engineering process model phases, whereas with the highest rate of 53.3% of respondents agreed with use of methodology has Insufficient well-defined process, comparing with the rate of respondents of the rest of the other methodology used.

Some methodologies are easy to understand. In which development phases did you use the current methodology based on its understandability? How would you rate them?

Table 4.17 shows with the highest rate of 93.3% of respondents among the other phases found it easy to understand the use of the current methodology by pre-analysis, whereas understanding the current methodology with quality is difficult to almost 67% of the respondents.

The least respondents of almost 7% found it moderate in pre-analysis and quality in the used of the current methodology used on the basis of their understandability.

Table 4.17. Phases of rating the current methodology used on basis of understanding

Phase	Easy F (%)	Moderate F (%)	Difficult F (%)	Total F (%)
Pre-analysis	14 (93.3)	1 (6.7)	-	15 (100)
Analysis	13 (86.7)	2 (13.3)	-	15 (100)
Design	6 (40.0)	4 (26.7)	5 (33.3)	15 (100)
Implementation	6 (40.0)	3 (20.0)	6 (40.0)	15 (100)
Quality	4 (26.7)	1 (6.7)	10 (66.7)	15 (100)
Test	5 (33.3)	2 (13.3)	8 (53.3)	15 (100)
Maintenance	5 (33.3)	2 (13.3)	8 (53.3)	15 (100)
Evaluation	4 (26.7)	4 (26.7)	7 (46.7)	15 (100)

F=frequency

Table 4.17 shows with the highest rate of 93.3% of respondents among the other phases found it easy to understand the use of the current methodology by pre-analysis, whereas understanding the current methodology with quality is difficult to almost 67% of the respondents. The least respondents of almost 7% found it moderate in pre-analysis and quality in the used of the current methodology used on the basis of their understandability.

Some methodologies are able to handle internal complexity very well. In which development phases did you use the current methodology based on its ability to handle internal complexity?

Table 4.18 Phases used on basis of its ability to handle internal complexity

Phase	Easy F (%)	Moderate F (%)	Difficult F (%)	Total F (%)
Pre-analysis	14 (93.3)	1 (6.7)	-	15 (100)
Analysis	12 (80.0)	3 (20.0)	-	15 (100)
Design	6 (40.0)	4 (26.7)	5 (33.3)	15 (100)
Implementation	6 (40.0)	3 (20.0)	6 (40.0)	15 (100)
Quality	4 (26.7)	3 (20.0)	8 (53.3)	15 (100)
Test	5 (33.3)	1 (6.7)	9 (60.0)	15 (100)
Maintenance	4 (26.7)	1 (6.7)	10 (66.7)	15 (100)
Evaluation	4 (26.7)	1 (6.7)	10 (66.7)	15 (100)

F=frequency

Table 4.18 evidently shows that the highest respondents of 93.3% have used the pre- analysis rated as easy phase, whereas the least respondents of almost 7% rated as moderate in pre-analysis, test, maintenance and evaluation, whereas 80% of respondents rated as easy for analysis. With almost 67% respondents rated maintenance and evaluation as difficult with the current methodology used based on their ability to handle internal complexity.

Some methodologies are easy to adapt according to the company's requirements. In which development phases did you use the current methodology based on its adaptability? How would you rate the

Table 4.19 Phases used on basis of adaptability

Phase	Easy F (%)	Moderate F (%)	Difficult F (%)	Total F (%)
Pre-analysis	13 (86.7)	2 (13.3)	-	15 (100)
Analysis	13 (86.7)	2 (13.3)	-	15 (100)
Design	5 (33.3)	5 (33.3)	5 (33.3)	15 (100)
Implementation	5 (33.3)	3 (20.0)	7 (46.7)	15 (100)
Quality	2 (13.3)	3 (20.0)	10 (66.7)	15 (100)
Test	6 (40.0)	2 (13.3)	7 (46.7)	15 (100)
Maintenance	5 (33.3)	2 (13.3)	8 (53.3)	15 (100)
Evaluation	4 (26.7)	4 (26.7)	7 (46.7)	15 (100)

F=frequency

Table 4.19 shows that the highest respondents of almost 87% have used the pre-analysis and analysis both rated as an easy phase, whereas quality was responded by almost 67% rated as difficult, whereas the lowest respondents of 13.3% in pre-analysis, analysis, quality, test and maintenance were rated as moderate in phase of the current methodology used based on their adaptability.

What is the methodology used for requirements gathering? How often are they used?

Table 4.20 Phases used on requirements gathering

Requirements gathering methodology	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Ad Hoc (No-methodology)	2 (13.3)	1 (6.7)	3 (20.0)	9 (60.0)	15 (100)
Interview	11 (73.3)	2 (13.3)	-	2 (13.3)	15 (100)
Prototyping	4 (26.7)	3 (20.0)	6 (40.0)	2 (13.3)	15 (100)
Questionnaire	-	2 (13.3)	6 (40.0)	7 (46.7)	15 (100)
Others (Please specify)	-	-	-	-	-

F=frequency

Table 4.20 obviously shows that the highest respondents that has always been used is Interview with 73.3% respondents, whereas the least often use is Ad Hoc with only 6.7% of respondents. Ad Hoc with 60% response was never used, whereas Prototyping and Questionnaire has been used rarely on basis of requirements gathering.

Did your company involve customers or users during the development process?

Table 4.21 Customers & users involvement in development processes

	Frequency	%
No	13	86.7
Yes	2	13.3
Total	15	100.0

Table 4.21 shows that almost 87% of respondents did not involve customers or users during development process, whereas only a maximum of 2 respondents were involved in requirement specification, analysis, design, implementation, integration, testing & evaluation.

If yes, what part of the development process are the users involved?

Table4. 22 Users involvement during development process

Development phase	√
Requirement specification	2
Analysis	2
Design	1
Implementation	2
Integration	2
Testing	2
Evaluation	2

Table 4.22 shows that whereas only a maximum of 2 respondents were involved in the above development phases, except one only in design phase.

What is the test approach used to test the web-based systems in your company and how often are they used?

Table 4.23 obviously shows 100% of respondents used the Unit test methodology always, and seconding was the Integration test with 93.3% respondents, whereas scalability test approach with 73.3% respondents was never used as approach to test the web-based system.

Does your company use any web metric (measurement tools) to improve the web- based system quality?

Table 4.24 evidently shows that zero respondents used any web metric measurement tools to improve web-bases system quality.

If yes, please specify the measurement tools used:

N/A

Please rank the problems you faced during requirement specification and analysis phase.

Table 4.26 shows that the problem rarely faced was ambiguous requirements with 60% of respondents, but problem of lack of good planning and insufficient estimating size of web project with 80% of respondents was never faced during requirement specification and analysis phase.

Table 4.23 Approach used to test the web-based system

Test methodology	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Unit test	15 (100)	-	-	-	15 (100)
Integration test	14 (93.3)	1(6.7)	-	-	15 (100)
Acceptance test	11 (73.3)	4 (26.7)	-	-	15 (100)
Performance test	3 (20.0)	5 (33.3)	2 (13.3)	5 (33.3)	15 (100)
Functionality test	4 (26.7)	3 (20.0)	2 (13.3)	6 (40.0)	15 (100)
Database test	-	4 (26.7)	2 (13.3)	9 (60.0)	15 (100)
Usability test	-	3 (20.0)	3 (20.0)	9 (60.0)	15 (100)
Security test	2 (13.3)	3 (20.0)	2(13.3)	8 (53.3)	15 (100)
Scalability test	-	2 (13.3)	2 (13.3)	11 (73.3)	15 (100)

F=frequency

Table 4. 24 Measurements tools used to improve web-based systems quality

	Frequency	%
Use	0	0.0
Not Use	15	100.0

Table 4.26 Problem faced during requirement specification and analysis phase

Problem	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Lack of good planning	-	-	3 (30.0)	12 (80.0)	15 (100)
Insufficient estimating size of web project	-	-	3 (20.0)	12 (80.0)	15 (100)
Customers without clear goal specifications	-	-	5 (33.3)	10(66.7)	15 (100)
Lack of involvement of non-engineering personnel	-	1(6.7)	5(33.3)	9(60.0)	15 (100)
Lack of extensive stake-holder analysis	-	1(6.7)	5(33.3)	9(60.0)	15 (100)
Lack of user involvement	1(6.7)	1(6.7)	2(13.3)	11(73.3)	15 (100)
Difficulty to understand customers'		1(6.7)	6(40.0)	8(53.3)	15 (100)
Requirements conflict	1(6.7)	2(13.3)	7(46.7)	5(33.3)	15 (100)
Incomplete requirements	1(6.7)	2(13.3)	7(46.7)	5(33.3)	15 (100)
Ambiguous requirements	1(6.7)		9(60.0)	5(33.3)	15 (100)
Lack of requirement documentation	1(6.7)		6(40.0)	8(53.3)	15 (100)

What problems can the methodology you use solve (completely, partially, unable to solve) during requirement specification and analysis phase?

Table4. 27 shows that the problems of lack of involvement of non-engineering personnel and lack of extensive stake-holder analysis showed the highest respondents of almost 67% has solved completely, whereas the least respondents of 6.7 respondents were unable to solve the problems of requirements conflict, incomplete requirements, ambiguous requirements and lack of requirement documentation during requirement specification and analysis phase.

Table 4.27 Problems on methodology used

Problem	Completely F (%)	Partially F (%)	Unable tosolve F (%)	Total F (%)
Lack of good planning	7(46.7)	8(53.3)	-	15 (100)
Insufficient estimating size of web project	8(53.3)	7(46.7)	-	15 (100)
Customers without clear goal specifications	7(46.7)	8(53.3)	-	15 (100)
Lack of involvement of non-engineering personnel	10(66.7)	5(33.3)	-	15 (100)
Lack of extensive stake-holder analysis	10(66.7)	4(26.7)	1(6.7)	15 (100)
Lack of user involvement	8(53.3)	5(33.3)	2(13.3)	15 (100)
Difficulty in understand customers needs	5(33.3)	8(53.3)	2(13.3)	15 (100)
Requirements conflict	5(33.3)	9(60.0)	1(6.7)	15 (100)
Incomplete requirements	5(33.3)	9(60.0)	1(6.7)	15 (100)
Ambiguous requirements	6(40.0)	8(53.3)	1(6.7)	15 (100)
Lack of requirement documentation	6(40.0)	8(53.3)	1(6.7)	15 (100)

F=frequency

Which problems do you think are most critical and you want the new methodology to solve during requirement specification and analysis phase? (Please rank 1, 2, 3…….12)

Table 4.28 Most critical problem during requirement Specification and analysis phase

Problem	Frequency	%
Lack of good planning	4	26.7
Insufficient estimating size of web project	1	6.7
Lack of involvement of non-engineering personnel	1	6.7
Lack of extensive stake-holder analysis	1	6.7
Lack of user involvement	4	26.7
Difficult to understand customers needs	1	6.7
Requirements conflict	1	6.7
Ambiguous requirements	1	6.7
Lack of requirement documentation	1	6.7
Total	15	100.00

Table 4.28 shows that lack of good planning and lack of user involvement with almost 27% respondents are the most critical and need to be solved during requirement specification and analysis phase.

16. What problems do you face during the design phase and how often do they occur?

Problem	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Lack of user involvement	1(6.7)	1(6.7)	-	13(86.7)	15 (100)
Difficulties with frequent changes in requirements			2(13.3)	13(86.7)	15 (100)
No standard procedure to control changes in requirements	1(6.7)		2(13.3)	12(80.0)	15 (100)
Lack of documentation	-	-	1(6.7)	14(93.3)	15 (100)

F=frequency

Table 4. 29 Problems faced during design phase and occurrences

Problem	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Lack of user involvement	1(6.7)	1(6.7)	-	13(86.7)	15 (100)
Difficulties with frequent changes in requirements			2(13.3)	13(86.7)	15 (100)
No standard procedure to control changes in requirements	1(6.7)		2(13.3)	12(80.0)	15 (100)
Lack of documentation	-	-	1(6.7)	14(93.3)	15 (100)

F=frequency

Table 4.29 shows that the majority of the respondents has never faced any problems, whereas a few like 6.7% only always face problem of lack of user involvement and no standard procedure to control changes in requirements, whereas few respondents rarely face problems of lack of documentation and difficulties with frequent changes in requirements.

What problems can the methodology you use solve (completely, partially, unable to solve) during the design phase?

Table 4.30 Problems solve during design phase of methodology

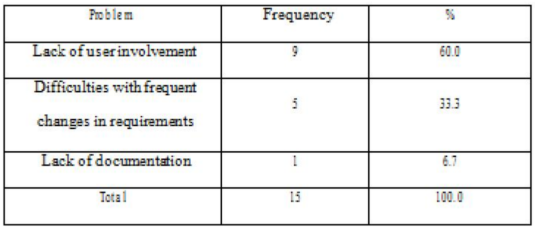
Problem	Completely F (%)	Partially F (%)	Unable to solve F (%)	Total F (%)
Lack of user involvement	10(66.7)	4(26.7)	1(6.7)	15 (100)
Difficulties with frequent changes in requirements	11(73.3)	4(26.7)	-	15 (100)
No standard procedure to control changes in requirements	10(66.7)	4(26.7)	1(6.7)	15 (100)
Lack of documentation	9(60.0)	6(40.0)	-	15 (100)

F=frequency

Table 4.30 shows that the problem that was solved with the highest respondents was the difficulties with frequent changes in requirement, whereas problem with lack of documentation was partially solved with 40% of respondents while the problems that was unable to solved were the lack of user involvement and no standard procedure to control changes in requirements having 6.7% of respondents.

Which problems do you think are most critical and you want the new methodology to solve during design phase?

Table 4.31 Most critical problems during design phase



Problem	Frequency	%
Lack of user involvement	9	60.0
Difficulties with frequent changes in requirements	5	33.3
Lack of documentation	1	6.7
Total	15	100.0

Table 4.31 shows that the most critical problem was the lack of user involvement with 60% of respondents and would want the new methodology to solve during design phase.

What are the problems you faced during implementation phase and how often do they occur?

Table 32. Problems faced and its occurrence during implementation phase

Problem	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Developers with various background	-	-	2(13.3)	13(86.7)	15 (100)
Difficulties with frequent changes in requirements	-		2(13.3)	13(86.7)	15 (100)
Low time-to-market frames	-	-	2(13.3)	13(86.7)	15 (100)
Lack of documentation	-	-	4(26.7)	11(73.3)	15 (100)
Lack of unit test	2(13.3)	1(6.7)	6(40.0)	6(40.0)	15 (100)
Lack of module test	2(13.3)	1(6.7)	6(40.0)	6(40.0)	15 (100)

F=frequency

Table 4.32 shows that the least respondents of 13.3% always faced problem of lack of unit test and lack of module test, whereas high respondents of 86.7% never faced problems of developers with various background, difficulties with frequent changes in requirements and low time-to-time market frames occurred during implementation phase.

What problems can the methodology you use solve (completely, partially, unable to solve) during implementation phase?

Table 4.33: Problems solved by methodology you used during implementation phase

Problem	Frequency	%
Developers with various background	8	53 3
Difficulties with frequent changes in requirements	3	20 .0
Low time-to-market frames	3	20 .0
Lack of module test	1	6 .
Total	1 5	100.0

Table 4.33 shows that more than half of the total respondents of 53.3% solved problems of developers with various backgrounds used during implementation phase.

Which problems do you think are most critical and you want the new methodology to solve during the implementation phase?

Table 4.34: Most critical problems during implementation phase

Problem	Frequency
Developers with various background	3
Difficulties with frequent changes in requirements	8
No environment, programming or coding standards	2
Low time-to-market frames	7
Lack of documentation	4
Lack of unit test	6
Lack of module test	5
Others (please specify)	1

Table 4.34 shows that frequent changes in requirements, low time to market and testing are most critical problems that the new methodology should solve.

What are the problems you faced during the integration phase and how often do they occur?

Table 4.35 Problems faced during the integration phase

Problem	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Lack of product test	2(13.3)	2(13.3)	7(46.7)	4(26.7)	15 (100)
Lack of user acceptance test	2(13.3)	4(26.7)	6(40.0)	3(20.0)	15 (100)
Lack of documentation	2(13.3)	2(13.3)	7(46.7)	4(26.7)	15 (100)
Difficulties with frequent changes in requirements	-	1(6.7)	4(26.7)	10(66.7)	15 (100)

F=frequency

Table 4.35 shows that the problems faced always with only 13.3% respondents were lack of product test, lack of use acceptance test and lack of documentation, whereas the highest respondents of 66.7% that never faced problem of difficulties with frequent changes in requirement during the integration phase.

What problems can the methodology you use solve (completely, partially, unable to solve) during the integration phase?

Table 4.36 Problems solved during integration phase

Problem	Completely F (%)	Partially F (%)	Unable to solve F (%)	Total F (%)
Lack of product test	3(20.0)	7(46.7)	5(33.3)	15 (100)
Lack of user acceptance test	3(20.0)	7(46.7)	5(33.3)	15(100)
Lack of documentation	3(20.0)	9(60.0)	3(20.0)	15 (100)
Difficulties with frequent changes in requirements	11(73.3)	4(26.7)	-	15 (100)

F=frequency

Table 4.36 shows that problem of difficulties with frequent changes in requirements were completely solved with highest respondents of 73.3%, whereas the highest respondents of 60% partially solved the problem of lack of documentation during integration phase.

Which problems do you think are most critical and you want the new methodology to solve during the integration phase?

Table 4.37: Most critical problems during integration phase

Problem	rank
Lack of product test	7
Difficulties with frequent changes in requirements	8
Total	1 5

Table 4.37 shows that the frequent changes in requirements is the most critical problem during integration phase that the new methodology should solve.

What are the problems you face during evaluation phase and how often do they occur?

Table 4.38 Problems faced during evaluation phase and occurrence

Problem	Always F (%)	Often F (%)	Rarely F (%)	Never F (%)	Total F (%)
Lack of good planning	-	-	3(20.0)	12(80.0)	15 (100.0)
Insufficient estimating size of web project	-	-	3(20.0)	12(80.0)	15 (100.0)
Difficulties with configuration management and documentation	-	-	10(66.7)	5(33.3)	15 (100.0)
Budget and time-frame exceeds	-	1(6.7)	3(20.0)	11(73.3)	15 (100.0)
No maintenance or evolution planning	-	1(6.7)	2(13.3)	12(80.0)	15 (100.0)
Low usability	1(6.7)	1(6.7)	1(6.7)	12(80.0)	15 (100.0)
Low performance	1(6.7)	-	3(20.0)	11(73.3)	15 (100.0)
Low security	1(6.7)	-	5(33.3)	9(60.0)	15 (100.0)
Low availability	1(6.7)	-	4(26.7)	10(66.7)	15 (100.0)
Low scalability	1(6.7)	1(6.7)	3(20.0)	10(66.7)	15 (100.0)
Low maintainability	1(6.7)	1(6.7)	3(20.0)	10(66.7)	15 (100.0)
Difficulties with project and product measurement and evaluation	1(6.7)	1(6.7)	3(20.0)	10(66.7)	15 (100.0)

Table 4.38 shows that with high problem faced on 80% of respondents in lack of planning, insufficient estimating size of web project, low usability and performance, whereas with lower problem faced of 6.7% respondents only on low usability, performance, security, availability, scalability, maintainability and difficulties with project and product measurement and evaluation.

What problems can the methodology you use solve (completely, partially, unable to solve) during the evaluation phase?

Table 4.39 Problems solve on methodology used during evaluation phase

Problem	Completely F (%)	Partially F (%)	Unable to solve F (%)
Lack of good planning	7 (47)	8(53)	0
Insufficient estimating size of web project	8(53)	7 (47)	0
Difficulties with configuration management and documentation	8(53)	5(33)	2(13)
Budget and time-frame exceeds	8(53)	7 (47)	0
No maintenance or evolution planning	10(67)	4(27)	1(7)
Low usability	12(80)	2(13)	1(7)
Low performance	11(73)	2(13)	2(13)
Low security	11(73)	2(13)	2(13)
Low availability	11(73)	3(20)	1(7)
Low scalability	11(73)	3(20)	1(7)
Low maintainability	12(80)	2(13)	1(7)
Difficulties with project and product measurement and evaluation	12(80)	2(13)	1(7)

F=frequency

Table 4.39 shows that lack of good planning, insufficient estimating size of the project and time frame exceed have high percentage as solved partially, whereas other problems distributed between completely or unable to solve problems.

Which problems do you think are most critical and you want the new methodology to evaluation phase?

Table 4.40 Most critical problem during evaluation phase

Problem	Frequency	%
Lack of good planning	8	53
Low usability	1	7
No maintenance or evolution planning	6	40
Total	1 5	

Table 4.40 shows that lack of good planning, evaluation plan and low usability are the most critical problem that the new methodology should solve.

Appendix 3: Project aim and objectives report
<Project Name>

Project Aim and Objectives

Executive Team <<16 bold>>:

- Name #1 (Team leader) <<14 bold>>
- Name #2 (Developer)
- Name #3 (System administrator)

- Project aim
- Problem statement
- General business process diagram

- Project objectives
- Objective name
    - Description
    - Development priority
- Objective name
    - Description
    - Development priority
- Objective Name
    - Description
    - Development priority
- Project Gantt chart

Objective	Start date	Duration / days
Objective 1	dd/mm/yyyy	Days
Objective 2	dd/mm/yyyy	Days

Technical Details

- Application type
- Used technologies
- Programming languages
- Database engine
- Tools
- HW & SW requirements

Appendix 4: Project objectives requirements report

<< Project Name >>

Project Objectives Requirements

Objectives Details: Objective name

- Objective business process diagram
- Objective users
    - User #1 name
- Use case diagram
    - User #2 name
- Use case diagram
    - User #3 name
- Use case diagram
- Class diagram
- Objective requirements
- Requirement name

Description

- Users
- Pre-condition
- Validation

- Post-condition
- User interface design
- Requirement name
  - Description
  - Objective
  - Users
  - Pre-condition
  - Validation
  - Post-condition
  - user interface design

****