



Full Length Research Article

RECOMMENDED PRACTISES FOR DEVELOING WEB-BASED SYSTEMS

***¹Ahmed Hassan Mohamed Ali and ²Dr. Mohammad Nazir Ahmad**

¹Faculty of Computer Science and Information Technology, University of Medical Sciences and Technology, Khartoum, Sudan

²Faculty of Computing, Universiti Teknologi Malaysia, Malaysia

ARTICLE INFO

Article History:

Received 15th October, 2014
Received in revised form
23rd November, 2014
Accepted 04th December, 2014
Published online 26th January, 2015

Key words:

Web Based Systems,
Software Engineering Models and
Web Engineering Models

ABSTRACT

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 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 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. This paper aims to introduce the recommended practices and features that each methodology should contains in order to develop web based system successfully.

Copyright © 2015 Ahmed Hassan Mohamed Ali and Dr. Mohammad Nazir Ahmad. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

In the beginning, the internet and the WWW were seen as a means of communicating 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). 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). Web based systems are those systems that deliver to users through Internet, intranets, and extranets. 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).

Advantages of web based systems

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):

***Corresponding author: Ahmed Hassan Mohamed Ali**

¹Faculty of Computer Science and Information Technology,
University of Medical Sciences and Technology, Khartoum, Sudan

Web based systems are more accessible: The HTTP protocol used in web based systems is a standard protocol that

can travel across corporate firewalls. The only client software a user needs 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.

Traditional Software Development and Web Based Systems Development

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).

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 that make Web development different and more difficult than traditional software development. Also many of the web based system development activities are business oriented (Eldai 2008) for example web application are sales- oriented, web application and intranets are content- oriented. (Mendes and Mosley 2006) grouped the differences between Web and software development into 12 areas, which are as follows:

Application Characteristics: Web applications are created by integrating several different elements, such as fine-grained components (e.g. DCOM, OLE, ActiveX), interpreted scripting languages, components off the shelf (COTS) (e.g. customized applications, library components, third-party products), multimedia files (e.g. audio, video, 3D objects), HTML/SGML/XML files, graphical images, mixtures of HTML and programs, and databases (Reifer 2000; Deshpande and Hansen 2001; Offutt 2002). In contrast, conventional software applications can also be developed using a wide variety of components (e.g. COTS), generally developed using conventional programming languages such as C++, Visual Basic, and Delphi.

Primary Technologies Used: Web applications are developed using a diverse technology such as the many flavored Java solutions, HTML, JavaScript, XML, UML, databases, and much more. In contrast, the primary technology used to develop conventional software applications is mostly represented by object- oriented methods, generators, and languages, relational databases, and CASE tools (Reifer 2000).

Approach to Quality Delivered: For Web development, quality is often considered as higher priority than time to market, with the mantra "later and better" as the mission statement for Web companies which wish to remain competitive (Offutt2002). Within the context of conventional software development, software contractors are often paid for their delivered application regardless of its quality. They are also often paid for fixing defects in the delivered application, where these failures principally exist because the developer did not test the application thoroughly (Mendes and Mosley 2006).

Development Process Drivers: The dominant development process drivers for Web companies are composed of seven quality criteria (Offutt 2002): Reliability, Usability, Security, Availability, Scalability, Maintainability and Time to Market. With regards to conventional software development, the development process driver is time to market is not quality criteria (Offutt 2002).

Availability of the Application: Customers who use the Web expect applications to be operational throughout the whole year (24/7/365). Any downtime, no matter how short, can be detrimental (Offutt 2002). Customers of conventional software applications do not expect these applications to be available 24/7/365.

Customers (Stakeholders): Web applications can be developed for use of a single organization (intranet), a number of organizations (extranets), or for use by people anywhere in the world. The implications are that stakeholders may come from a wide range of groups where some may be clearly identified and some may remain unknown, which is often the case (Mendes and Mosley 2003). As a result, Web developers are regularly facing the challenge of developing applications for unknown users, whose requirements and behavior patterns are also unknown at development time. With regards to conventional software applications, it is usual for stakeholders to be explicitly identified prior to development.

Update Rate (Maintenance Cycles): Web applications are updated frequently without specific releases and with maintenance cycles of days or even hours. In addition, their content and functionality may also change significantly from one moment to another, and so the concept of project completion may seem unsuitable in such circumstances (Mendes and Mosley 2006). The maintenance cycle for conventional software applications complies with a more precise process (Hughes 2007).

People Involved in Development: Web applications usually require a team of people with diverse skills and experience (Lang 2009). Such teams consist of Web designers and programmers, graphic designers, librarians, database designers, project managers, network security experts, and usability experts. The development of conventional software requires IT professionals where knowledge of programming, database design, and project management is necessary.

Architecture and Network: Web applications are typically developed using a simple client-server architecture (two-tier), represented by Web browsers on client computers connecting to a Web server hosting the Web application, to more

sophisticated configurations such as three-tier or even n-tier architecture (Maxwell 2002). Conventional software applications either run in isolation on a client machine or use two-tier architecture whenever applications use data from database systems installed on a separate server (Mendes and Mosley 2006). The type of networks used by the stakeholders is usually known in advance since most conventional software applications are limited to specific places and organizations.

Disciplines Involved: To develop large and complex Web applications adequately, a team of people with a wide range of skills and expertise in different areas is required. These areas reflect distinct disciplines such as software engineering (development methodologies, project management, tools), hypermedia engineering (linking, navigation), requirements engineering, usability engineering, information engineering, graphics design, and network management (performance measurement and tuning) (Conte 2007; Gray 2007; Hughes 2007). Building a conventional software application involves contributions from a smaller number of disciplines than those used for developing Web applications, such as software engineering, requirements engineering, and usability engineering.

Legal, Social, and Ethical Issues: The Web as a spread environment enables a huge amount of structured (e.g. database records) and unstructured (e.g. text, images, audio) content to be easily available to a multitude of users worldwide. This is often cited as one of the greatest advantages of using the Web. However, this environment is also used for the purpose of dishonest actions, such as copying content from Web applications without acknowledging the source, distributing information about customers without their consent, infringing copyright and intellectual property rights, and even, in some instances, identity theft (Mendes and Mosley 2006). Conventional software applications also share a similar chance to that of Web applications, although to a smaller extent, since these applications are not so readily available for such a large community of users, compared to Web applications.

Information Structuring and Design: As previously mentioned, Web applications present structured and unstructured content, which may be distributed over multiple sites and use different systems (e.g. database systems, file systems, multimedia storage devices) (Finnie 2007). In addition, the design of a Web application, unlike that of conventional software applications, includes the organization of content into navigational structures by means of hyperlinks. These structures provide users with easily navigable Web applications. Well-designed applications should allow for suitable navigation structures (Conte 2007), as well as the structuring of content, which should take into account its efficient and reliable management. Another difference between Web and conventional applications is that Web applications often contain a variety of specific file formats for multimedia content (e.g. graphics, sound, and animation). These files must be integrated into any current configuration management system, and their maintenance routine also needs to be organized as is likely that it will differ from the maintenance routine used for text-based documents (Briand 2005). Conventional software applications present structured

content that uses file or database systems. The structuring of such content has been addressed by software engineering in the past so the methods employed here for information structuring and design are well known by IT professionals (Mendes and Mosley 2006).

Unique characteristics of Web Based Systems

The unique characteristics of web based systems 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).

The look and feel of web based system is very important. When an application has been designed to market or sell products or ideas, aesthetics may have as much to do with success as technical design. Creating good look and an easy-to-use user interface is just one of the many challenges that the developers of a web based system face during their work (Atterer 2005). Due to the many factors which influence the user experience of the application (ranging from differences in the user's input and output devices to latency issues and the

request/response paradigm of the HTTP protocol), an application with the same functionality typically has a quite different interface when implemented as a web application instead of a GUI application for a desktop computer (Atterer 2006).

Recommended practices for developing web based systems

The differences listed in the previous section, result in some Development practices that are special for web based systems. Those practices are:

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 requirements evolving.

Conclusion

Web based system is different from traditional software in many areas. This difference lead to failure of software methodologies in developing web based systems and make developing web based systems requires specific practices. The recommended practices that can deal with special nature of web based systems development are:

- Parallel development
- Release orientation
- Tool dependence
- Customer involvement
- Prototyping, increments and iterative development

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

- 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.
- Atterer, R. 2005. Where Web Engineering Tool Support Ends: Building Usable Websites. The 20th Annual ACM Symposium on Applied Computing. Santa Fe, New Mexico, ACM.
- Atterer, R. 2006. "Extending web engineering models and tools for automatic usability validation." *Journal of Web Engineering*, 5(1): 43–64.
- Bansler, J. 2000. "Corporate Intranet Implementation: Managing emergent technologies and organizational practices." AIS 1: 1-39.
- Brandon and Dan 2008. "Software engineering for modern Web applications: methodologies and technologies". Hershey, PA, Information Science Reference. "This book presents current, effective software engineering methods for the design and development of modern Web-based applications"--Provided by publisher.
- Briand, L. 2005. "An Assessment and Comparison of Common Cost Estimation Modeling Techniques". ICSE, Los Angeles, USA.
- Conte, S. 2007. "Software Engineering Metrics and Models". Benjamin/Cummings.
- Deshpande, Y. and S. Hansen 2001. "Web engineering: creating a discipline among disciplines." *IEEE Multimedia* 8(2): 8-87.
- Eldai, O. I. 2008. "Towards a New Methodology for Developing Web-Based Systems." *World Academy of Science, Engineering and Technology*, 46.
- Finnie, G. 2007. "A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models." *Systems and Software*, 39: 281-289.
- Gray, R. 2007. "Factors Systematically associated with errors in subjective estimates of software development effort: the stability of expert judgement". the IEEE Metrics Symposium.
- Holck, J. 2003. "4 Perspectives on Web Information Systems". 36th Hawaii International Conference on System Sciences, Big Island, Hawaii, USA, IEEE Computer Society.
- Hughes, R. 2007. "An Empirical investigation into the estimation of software development effort". Dept. of Computing, University of Brighton. PhD.
- Ingle, D. and D. B. B. Meshram, 2012. "Hybrid Analysis and Design Model for Building Web Information System." *IJCSI International Journal of Computer Science Issues* 9(4).
- 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).
- Lam, M. 2011. "Methodologies, tools, and techniques in practice for web application development." *Journal of Technology Research*, 3.

- Lang 2009. "Web-based Systems Development: The Influence of Disciplinary Backgrounds on Design Practices." JIOS 33(1).
- Lavie, T. and N. Tractinsky, 2004. "Assesing dimensions of perceived visual aesthetics of web sites." *International Journal Human- Computer Studies*, 60: 269-298.
- Maxwell, K. 2002. "Applied Statistics for Software Managers." Englewood Cliffs.
- Mendes and Mosley, 2003. "A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation". ACM and IEEE International Symposium on Empirical Software Engineering, Rome, Italy.
- Mendes and Mosley, 2006. "Web engineering". Berlin ; New York, Springer.
- 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.
- Offutt, J. 2002. "Quality attributes of Web software applications." *IEEE Software* 19(2): 25-32.
- Pressman, R. S. and D. B. Lowe, 2009. "Web engineering : a practitioner's approach". Boston, McGraw-Hill Higher Education.
- Reifer, D. 2000. "Web development: estimating quick-to-market software." *IEEE Software*: 57-64.
- 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.
