A Survey- use of Software Quality Attributes for Web Based Software Applications

[1]Ms. Preeti Jain (Asst. Prof.), Acropolis Institute of Technology & Research

[2]Mr. Anurag Punde (Asst. Prof.), Acropolis Institute of Technology & Research

Indore (MP)-452001, INDIA

[1]preetijjain@acropolis.in
[2]anuragpunde@acropolis.in

Abstract: In older days the main purpose of the web application is to surf by means of trouble-free sites that consists simple hypertext documents. The use of web application on internet is limited and expensive. The demands of the users are increasing. Because of that, the IT industries have changes their requisite to serve numerous activities such as distribution of information, entertainment, E-commerce, survey, online shopping, core banking and many more. Web application has very high requirements for numerous quality attributes. In this paper we have discusses some of the technological challenges for building up high quality software based application and their unique quality requirements.

Keywords: SQA, web application, software engineering, reliability, scalability.

I. Introduction

The software that powers Web applications is distributed, is implemented in multiple languages and styles, incorporates much reuse and third-party components, is built with cutting edge technologies, and must interface with users, other Web sites, and databases. Although the word "heterogeneous" is often used for Web software, it applies in so many ways that the synonymous term "diverse" is more general and familiar, and probably more appropriate. The software components are often distributed geographically both during development and deployment, and communicate in numerous distinct and sometimes novel ways (diverse communication). Web applications consist of diverse components including traditional and non-traditional software, interpreted scripting languages, plain HTML files, mixtures of HTML and programs, databases, graphical images, and complex user interfaces. As such, engineering an effective Web site requires large teams of people with very diverse skills and backgrounds. These teams include programmers, graphics designers, usability engineers, information layout specialists, data communications and network experts, and database administrators. This diversity has led to the notion of Web site engineering. [1]

II. Aspect of Web application software

To develop the web based software is not an easy task for the software developers. When they actually developed the software, they have to integrated many components which affect the quality of the web application. To ensure high quality for Web systems composed of very loosely coupled components, we need novel techniques to achieve and evaluate these components' connections. Web-based software offers the significant advantage of allowing data to be transferred among completely different types of software components that reside and execute on different computers. However, using multiple programming languages and building complex business applications complicates the flow of data through the various Web software pieces. When combined with the requirements to keep data

persistent through user sessions, persistent across sessions, and shared among sessions, the list of abilities unique to Web software begins to get very long. Thus, software developers and managers working on Web software have encountered many new challenges. Although it is obvious that we struggle to keep up with the technology, less obvious is our difficulty in understanding just how Web software development is different, and how to adapt existing processes and procedures to this new type of software.

III. Economic changes

We evaluate software by measuring the quality of attributes such as reliability, usability, and maintainability, yet academics often fail to acknowledge that the basic economics behind software production has a strong impact on the development process. Although the field of software engineering has spent years developing processes and technologies to improve software quality attributes, most software companies have had little financial motivation to improve their software's quality. Software contractors receive payment regardless of the delivered software's quality and, in fact, are often given additional resources to correct problems of their own making. So-called "shrink wrap" vendors are driven almost entirely by time to market; it is often more lucrative to deliver poor-quality products sooner than high-quality products later. They can deliver bug fixes as new releases that are sold to generate more revenue for the company. For most application types, commercial developers have traditionally had little motivation to produce high-quality software. Web-based software, however, raises new economic issues. When I recently surveyed a number of Web software development managers and practitioners, I found that companies that operate through the Web depend on customers using and, most importantly, returning to their sites. Thus, unlike many software contractors, Web application developers only see a return on their investment if their Web sites satisfy customer needs. And unlike many software endors, if a new company puts up a competitive site of higher quality, customers will almost immediately shift their business to the new site once they discover it. [4]

IV. Criteria for web application

Of course, this is hardly a complete list of important or even relevant quality attributes, but it provides a solid basis for discussion. Certainly speed of execution is also important, but network factors influence this more than software does, and other important quality attributes such as customer service, product quality, price, and delivery stem from human and organizational rather than software factors. [1, 5–8] suggesting wide agreement that successful Web software development depends on satisfying these quality attributes.

1. Reliability
2. Usability
3. Security

Additional important criteria include

4. Availability
5. Scalability
6. Maintainability

1. Reliability

Express the ability of the component to maintain a specified level of performance, when used under specified Conditions.

2. Usability

Express the ability of a component to be understood, learned, used, configured, and executed, when used under specified conditions.

3. Security

We have all heard about Web sites being cracked and private customer information Distributed or held for ransom. This is only one example of the many potential security flaws in Web software applications. When The Web functioned primarily to distribute online brochures, security breaches had relatively small consequences. Today,

however, the breach of a company's Web site can cause significant revenue losses, large repair costs, legal consequences, and loss of credibility with customers. Web software applications must therefore handle customer data and other electronic information as securely as possible.

4. **Availability**

This attribute measures the ratio of the total time to the time which a service is capable of being operable. Typically availability objectives are specified either in decimal fractions or percentage.

5. **Scalability**

The ability of the component to accommodate major data volumes without changing its implementation.

6. Maintainability

One novel aspect of Web-based software systems is the frequency of new releases, or the *update rate*. Traditional software involves Marketing, sales, and shipping or even personal installation at customers' sites. Because this process is expensive, software manufacturers usually collect maintenance modifications over time and distribute them to customers simultaneously. For a software product released today, developers will start collecting a list of necessary changes. For a simple change (say, changing a button's label), the modification might be made immediately. But the delay in releases means that customers won't get more complex (and likely important) modifications for months, perhaps years. Web-based software, however, gives customers immediate access to maintenance updates— both small changes (such as changing the label on a button) and critical upgrades can be installed immediately. Instead of maintenance cycles of months or years, Web sites can have maintenance cycles of days or even hours.

Architecture

The technology has changed because the old two-tier model did not support the high quality requirements of Web software applications. It failed on security, being prone to crackers who only need to go through one layer of security on a computer that is, by definition, open to the world to provide access to all data files. It failed on scalability and maintainability because as Web sites grow, a 2-tier model cannot effectively separate presentation from business logic, and the applications thus become cumbersome and hard to modify. It failed on reliability: whereas previous Web software generations relied on CGI programs, usually written in Perl, many developers have found that large complex Perl programs can be hard to program correctly, understand, or modify. Finally, it failed on availability because hosting a site on one Web server imposes a bottleneck: any server problems will hinder user access to the Web site. Figure 1 illustrates current Web site software. Instead of a simple client-server model, the configuration has expanded first to a three-tier model and now more generally to an *N*-tier model. Clients still use a browser to visit Web sites, which are hosted and delivered by Web servers. But to increase quality attributes such as security, reliability, availability, and scalability, as well as functionality, most of the software has been moved to a separate computer—the application server. Indeed, on large Web sites, a collection of application servers typically operates in parallel, and the application servers interact with one or more database servers that may run a commercial database. [11, 12]

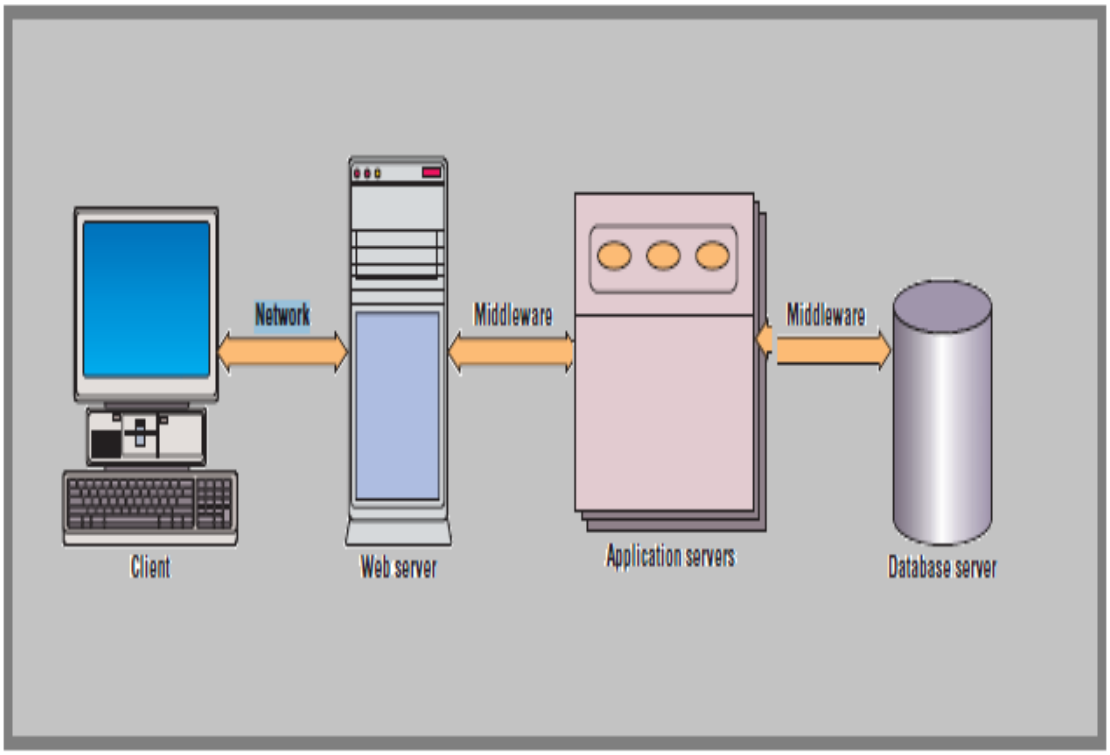

Figure 1: Modern Web sites following N-Tier Model

Conclusion

Even with the current economic downturn, the university output is not enough. If universities could double the production of computer scientists, we still could not put a dent in the need. (Most economists and business leaders currently believe last year's many layoffs and bankruptcies in the e-commerce sector resulted from temporary problems, and expect significant growth in the near future. I optimistically accept this prognosis; if it is wrong, then this article will be irrelevant anyway.) We can only meet this need by

- Retraining experienced engineers to work with the new technology
- Applying knowledge and technology to increase efficiency, thereby reducing the number of engineers needed.
- Finding ways to let people with less education and skills contribute.

References

1. T.A. Powell, *Web Site Engineering: Beyond Web Page Design*, Prentice Hall, Upper Saddle River, N.J., 2000.

2. D.A. Menascé, *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*, Prentice Hall, Upper Saddle River, N.J., 2000.

3. E. Dustin, J. Rashka, and D. McDiarmid, *Quality Web Systems: Performance, Security, and Usability*, Addison- Wesley, Reading, Mass., 2001.

4. L.L. Constantine and L.A.D. Lockwood, *Software for Use: A Practical Guide to the Models and Methods of Usage Centered Design*, ACM Press, New York, 2000.

5. S. Murugesan and Y. Deshpande, “Web Engineering: A New Discipline for Development of Web-Based Systems,” *Web Engineering 2000*, Lecture Notes in Computer Science 2016, Springer-Verlag, Berlin, 2001, pp. 3–13.

6. N. Kassem and the Enterprise Team, *Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition*, Sun Microsystems, Palo Alto, Calif., 2000.

7. J. Nielsen, *Designing Web Usability*, New Riders Publishing, Indianapolis, Ind., 2000.

8. M.E. Segal and O. Frieder, “On-the-Fly Program Modification: Systems for Dynamic Updating,” *IEEE Software*, vol. 10, no. 2, Mar. 1993, pp. 53–65.

9. Wrox Multi Team, *Professional Java Server Programming, J2EE edition*, Wrox Press, Chicago, 2000.

10. Scharl, *Evolutionary Web Development*, Springer- Verlag, Berlin, 2000.