Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Phases of Testing An Online Application

Sandeep Singh , Shivani Goel
Computer Science and Engineering Department
Thapar University, Patiala, India

Abstract— Online or web applications are used in every domain of life these days. Before proceeding to develop and online application, testing should be included at various levels of its development. Testing an online application is a tedious task. In this paper, various types of testing techniques applicable at lifecycle phases of developing an online application are presented. A sample case study of a project on Online Movie Store is taken to apply the proposed techniques.

Keywords— Application, Online Application, Software Testing, SDLC Testing, Feasibility Study Chart

I. INTRODUCTION

Application:
Application is nothing but just a small program (software) which runs on machine like computers, mobiles etc. Application are abbreviated as apps, this shortened version of application became popular with the Apple iphone, Android phones and Facebook, which allowed developers to use toolkits to develop applications for their products. A popular example of how Apple uses this term is the apple app store, which is an online location anyone with an apple product can go and purchase and install an application.

Online application:
An online application is an application that is accessed by users over a network such as the internet. The term may also mean a computer software application that is coded in a browser-supported programming language (such as JavaScript, combined with a browser-rendered markup language like html) and reliant on a common web browser to render the application executable. An online application therefore is an application where there are two or more components in different places that talk to one another. Online applications can refer to software or programs which you can use through the internet and that don't need to be installed on your home computer. An example of this is Google docs or an online calculator or game, like the games in Facebook are known as online application.

Software testing:
Computer software is a major component of an information system whose reliability is critical to the performance of an organization. The reliability of an information system has four facets : people, hardware, software and data. Among these four facets, the reliability of software is a joint responsibility of computer science and information system professionals. The former handles technical software and the latter application software. Regardless of the types of software involved, the reliability of software must be achieved through a continuous effort of planning, analysis, design, programming, testing, installation and maintenance. Most of the software errors detected during the testing phase originate in the early analysis phase. Therefore, software testing should start early in the system development process. Software testing is essential to ensure software quality.Software testing is iterative process which consists of
1. Designing tests
2. Executing tests
3. Identifying problems
4. Getting problems fixed

Difference between desktop application and web application:
The following table will explain the main difference between a desktop application (Calculator program, Yahoo Messenger etc) and a web application (Meebo, Avairy, Google Docs etc).

Table 1. Difference between desktop application and web application [9]

S. No.	Functionality	Desktop Application	Web Application
1.	Performance	Faster	Slower
2.	Network Congestion	Depending o the data transfer and connections made to the server from various clients.	Depends
3.	User Interfaces, data binding etc.	Easy to build	Difficult to build

4.	Deployment and Maintenance	Complex. New versions of assemblies, configuration files and other required files must be deployed on all clients' machines. Usually user interaction required.	Easy. Need to deploy assemblies and configuration files on the server only. Transparent to the client.
5.	Robustness and Reliability	One client machine goes down, other users are still live.	Usually web servers are never down. However if the server goes down, all users are affected.
6.	Resources	Runs on the client machine.	Runs on a Web server.
7.	Catastrophic failure	User interaction required.	Usually user interaction not required.
8.	Framework dependency	All client machines have to install required versions of .NET framework and other required libraries.	Only server needs to have .NET framework and other required libraries.

II. OBJECTIVE OF SOFTWARE TESTING

The objective of software testing is to find problems and fix them to improve quality. Software testing typically represents 40% of a software development budget. There are four main objectives of testing [1]:

1. **Demonstration:** It show that the system can be used with acceptable risk, demonstrate function under special conditions and show that products are ready for integration or use.
2. **Detection:** It discovers defects, errors and deficiencies. Determine system capabilities and limitations quality of components, work products and the system.
3. **Prevention:** It provides information to prevent or reduce the number of errors clarify system specifications and performance. Identify ways to avoid risks and problems in the future.
4. **Improving quality:** By doing effective testing, we can minimize errors and hence improve the quality of the software.

III. PHASES IN SOFTWARE TESTING

Although many test teams use test tools or scripts to automate testing activities, there's a lot about testing which is just simply labour intensive [2]. Here are just some of the activities involved:

1. Planning and developing test cases: Writing test plans an documentation, prioritizing the testing based on assessing the risks, setting up test data, organising test teams [2].

2. Setting up the test environment: An application will be tested using multiple combinations of hardware and software and under different conditions. Also, setting up the prerequisites for the test cases themselves [2].

3.Writing test harnesses and scripts: Developing test applications to call the API directly in order to automate the test cases. Writing scripts to simulate user interactions [3].

4. Planning, writing and running load tests: Non-functional tests to monitor an application's scalability and performance. Looking at how an application behaves under the stress of a large number of users [4].

5. Writing bug reports: Communicating the exact steps required to reproduce unexpected behaviour on a particular configuration. Reporting to development team with the results [2].

Testing web application: According to *"Kota K"* there are ten quick steps to test your web application. Their names are: Objectives, Process and Reporting, Tracking Results, Test Environment, Usability Testing, Unit Testing, Verifying the HTML, Load Testing, User Acceptance Testing and Testing Security [5].

IV. APPLY TESTING ON SDLC PHASES

A Software Development Life Cycle (SDLC) is essentially a series of steps, or phases that provides a model for the development and lifecycle management of an application or piece of software. The methodology within the SDLC process can vary across industries and organizations, but standards such as ISO/IEC 12207 represent processes that establish a lifecycle for software, and provide a mode for the development, acquisition, and configuration of software systems. The intent of a SDLC process it to help produce a product that is cost-efficient, effective, and of high quality. Once an application is created, the SDLC maps the proper deployment and decommissioning of the software once it becomes a legacy. The SDLC methodology usually contains the following g stages: Feasibility study, Analysis (requirements and design), construction, testing, release, and maintenance (response). Testing should not be only one phase of SDLC. Actually this can be treated as an umbrella activity that can be applied at all phases of SDLC. Testing code is not the only form of testing. There are various techniques used as testing techniques like regression testing, integration testing, requirements testing using prototyping along with all other techniques for testing code. Figure 1 shows various phases on an online application development and corresponding techniques that can be applied for testing in that phase. The next sections show the results of applying these all in SDLC of an Online Movie Store System. The system allows the users to select a movie from various categories for rent purpose or buying online. Then the user can

submit the feedback about the movie in a form online. The ratings of the movie are selected on the basis of the analysis of the feedback given by various users.

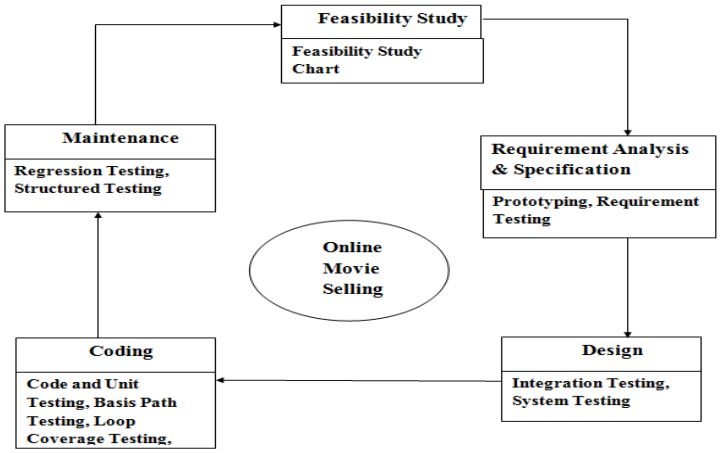

Fig 1. Testing applied on SDLC phases

1. Feasibility Study Chart:

Table 2. Feasibility Study Chart

Sr.No	*Requirements*	*Operational Feasibility*	*Technical Feasibility*
1.	*Search using movie language and type*	✓	✓
2.	*Manage movies i.e. add new movies, delete old movies, edit etc*	✓	✓
3.	*Login/Register*	✓	✓
4.	*Buy/Rent movies up to 20*	✓	✓
5.	*Search using Google*	✓	✓
6.	*Contact us through phone*	✓	✓
7.	*Contact us through Email*	✓	✘
8.	*Online pay*	✘	✘
9.	*Frequently asked questions*	✓	✓
10.	*Get details about the movies i.e. releasing date, actor name, director name etc*	✓	✓

2. Requirement Analysis & Specification

a. Prototyping: A prototype is an initial version of a multimedia product. The prototype is tested to make sure it is fit for the audience and purpose. If there are any errors or problems, the prototype is improved and tested again. This goes on until the product is considered to be fully functional and suitable. This process is known as prototyping in Figure 2 [6]. The screens for Online Movie Store System were also discussed with various stakeholders before final development.

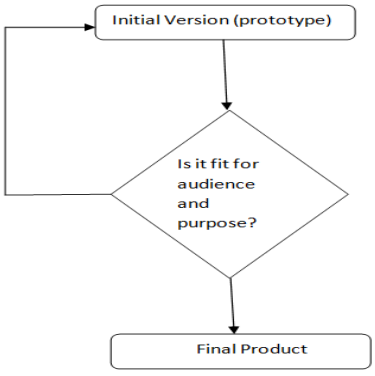

Fig 2. Process of prototyping

b. Requirement Testing: Requirements seem to be ephemeral. They fit in and out of projects; they are capricious, intractable, unpredictable and sometimes invisible. We throw out a net and try to capture all these criteria. Using Blitzing, Rapid Application Development (RAD), Joint Application Development (JAD), Quality Function Deployment (QFD), interviewing apprenticing, data analysis and many more other techniques, we try to snare all of the requirements in our net [1]. Interviews were used as medium for testing requirements from all stakeholders in the system developed.

3. **Design:**
   a. **Integration Testing:** While software modules may function well by themselves when they are developed, getting them to work together efficiently and correctly is another matter. Eventually, all software modules and integrated and debugged so they function correctly as a whole. Integration testing is activity of software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing [1]. Not much integration testing is done for current web application.
   b. **System Testing:** When the software, hardware and other subsystems are complete, they in turn are integrated and tested as a system. This is final development testing. Any problems or errors discovered during systems testing are analysed to determine which subsystems are at fault, then those subsystems are sent back for debugging, with its code, unit and integration testing [3]. Approximately 10 errors were reported while system testing and were corrected one by one.

4. **Coding:**
   a. **Code and Unit Testing:** Code and unit testing is a procedure used to validate that individual units of source code are working properly. In object-oriented programming, the smallest unit is a method; which may belong to a base or super class, abstract class or derived class [7].
   b. **Basis Path Testing:** The testing mechanism was proposed by *McCabe*. Its aim is to derive a logical complexity measure of a procedural design and use this as a guide for defining a basic set of execution paths [8].
   c. **Loop Coverage Testing:** The goal of loop coverage testing is to test while-do, repeat-until, (or do-while) and any other loops in a program thoroughly; by trying to ensure that each is executed at minimal, typical, and (if this is defined) maximal values - and to try to ``break'' the program, by trying to have a loop executed with a fewer than minimum, as well as a larger than maximal, number of iterations.
   All these testing were done for the current web application at hand by the developer.

5. **Maintenance:**
   a. **Regression Testing:** Regression testing is any type of software testing which seeks to uncover regression bugs. Regression bugs occur whenever software functionality that previously worked as desired. A run-time error takes place during the execution of a program and usually happens because of adverse system parameters or invalid input data [10]. There were around 5 changes made to the current web application. Based on these test cases were re-executed to check the effect of the changes made.
   b. **Structured Testing:** Structured testing uses cyclomatic complexity and the mathematical analysis of control flow graphs to guide the testing process. Structured testing is more theoretically rigorous and more effective at detecting errors in practice than other common test coverage criteria such as statement and branch coverage [11]. Required changes in structure were tested based on structured testing for the given application.

V. CONCLUSIONS

An online application demands a lot from the developers. This is due to the fact that a large number of users are to visit the link to the application. The access can be made at any time during 24 hours. A large number of users may access the webpage simultaneously. This requires testing an online application thoroughly before making it online. Various testing techniques applicable at various phases of development of an online application are explained with the help of a case study. All users who are developing online applications of any sort can benefit from this research.

ACKNOWLEDGEMENT

We would like to thank all the users who have given feedback for Online Movie Store Application.

REFERENCES

[1] Jain Deepak, “Software Engineering Principle and Practices” First edition by Oxford University Press, ISBN-13:978-0-19-569484-0, (2009).
[2] Chilarege, Ram, “Software testing techniques”, Second edition, (1990).
[3] Joe W. Duran, Semeon, C.Ntafos, “An Evaluation of Random Testing”, IEEE Transactions on Software Engineering, Vol.SE-10,No.4, pp438-443 (July 1984).
[4] Beizer, Boris, “Black-Box Testing Technique for Functional Testing of software and system” New York Wiely, ISBN: 0471120944 Physical description:xxv,194 p.ill; 23cm(1995).
[5] Kota K., “ Testing Your Web Application”, A Quick 10-Step Guide, © 2005 Krishen Kota
[6] “Program Manager`s Guide for Managing software”, 0.6, (29 June 2001).
[7] IEEE” Standards for a Software Quality Metrics Methodology” (IEEE standard P-1061/D21). New York, N.Y.: Institute of Electrical and Electronic Engineers. Inc. (1990).
[8] IEEE,“ANSI/IEEE Standard 1008-1987 , IEEE Satandard for Software unit Testing” (1986)
[9] Chand M, c-sharpcorner.com/Blogs/416/, Oct 03, 2007.
[10] Jorgensen,Paul C, “Software Testing A Craftman’s Approach”, CRC Press, p.3(1995)
[11] Watson, A., Structured Testing: Analysis and Extensions, Ph.D. dissertation, Princeton University, in preparation.

