

Spring 2016

Design and Implementation of a Virtual Team Collaboration System

Syed Ehsan
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/emse_etds

 Part of the [Engineering Commons](#)

Recommended Citation

Ehsan, Syed. "Design and Implementation of a Virtual Team Collaboration System" (2016). Master of Science (MS), thesis, Engineering Management, Old Dominion University, DOI: 10.25777/1h0e-dq93
https://digitalcommons.odu.edu/emse_etds/3

This Thesis is brought to you for free and open access by the Engineering Management & Systems Engineering at ODU Digital Commons. It has been accepted for inclusion in Engineering Management & Systems Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

**DESIGN AND IMPLEMENTATION OF A VIRTUAL TEAM COLLABORATION
SYSTEM**

by

Syed Ehsan

B.Sc. Computer Science, May 2007, North South University, Bangladesh

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

ENGINEERING MANAGEMENT

OLD DOMINION UNIVERSITY

May 2016

Approved by:

Pilar Pazos (Director)

Holly A Handley (Member)

Mamadou Seck (Member)

ABSTRACT

DESIGN AND IMPLEMENTATION OF A VIRTUAL TEAM COLLABORATION SYSTEM

Syed Ehsan
Old Dominion University, 2016
Director: Dr. Pilar Pazos

This paper analyzes a newly developed virtual team collaboration site and compares this application with the currently in use team collaboration tool created using a Google site. Various tests that are essential for a web service to function properly are performed to compare the applications. The tests are aimed to find any issues with the applications and identify if Teamboard ODU, the newly developed application, can perform better than the Google site. We test for speed, performance, usability, functionality and also explore the methodology used to develop the application. These test results allow us to make an informed assumption about the effectiveness of Teamboard ODU and help us understand whether this can be a replacement for the Google site. The study hypothesizes that Teamboard ODU performs better in terms of the HTTP transaction, site availability and load time. It also has functionalities comparable to the Google site. Future enhancement and maintenance can also be performed if needed. Several scenarios are developed to test the hypotheses, and the results will be used to compare the two sites and identify whether Teamboard ODU can be an effective replacement.

ACKNOWLEDGEMENTS

I would thank Dr. Pilar Pazos. I greatly appreciate her work, guidance and feedback while preparing this document. Also, special thanks to <https://loadimpact.com> and www.dotcom-tools.com for providing me with free services used to perform some of the necessary tests.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	vi
LIST OF TABLES	vii
Chapter	
1. INTRODUCTION	1
1.1 Background of the problem.....	1
1.2 Problem statement	2
1.3 Purpose of the project.....	2
1.4 Scope of the study	3
1.5 Importance of the study.....	3
1.6 Definition of Terms	4
2. LITERATURE REVIEW	5
2.1 Stages of Software Development	5
2.2 Choice of Language	7
2.3 Methodologies	10
3. METHODOLOGY	13
3.1 Primary Objective	13
3.2 Secondary Objectives	14
3.3 Hypotheses	15
3.4 Data Analysis	16
4. RESULTS	18
4.1 User Load test.....	18
4.2 Availability test (ping test).....	20
4.3 Browser load test.....	22
4.4 Module processing time	24
4.5 Usability and functionality	26
4.6 Development method	30
5. CONCLUSION.....	33
LIMITATIONS AND BENEFITS	34
Maintenance	34
Modifiability.....	34
Hardware resources	34
Outsourcing vs In-house development.....	34
FUTURE WORK.....	36
REFERENCES	37
APPENDIX.....	38
A.....	38
B	43

C	45
D	46
VITA	60

LIST OF FIGURES

Figures	Page
2.1: Generic software development life cycle.....	5
2.2: Completeness of solutions	8
2.3: Total size of the solutions	9
2.4: Summary of the robustness.....	10
2.5: Waterfall lifecycle model.....	11
2.6: Iterative spiral model.	12
4.1: Load time for Teamboard ODU with 50 users	18
4.2: Load time for the Google site with 50 users	19
4.3: Load time for Teamboard ODU with 25 users	19
4.4: Load time for the Google site with 25 users.....	20
4.5: Screen shot of a ping test from a machine.	21
4.6: Ping test result comparison between the two sites.....	22
4.7: Browser's load time comparison.	23
4.8: Processing times of each functionality.	25
4.9: Architecture of Teamboard ODU	31

LIST OF TABLES

Tables	Page
4.1: Summary of Test result for the Admin section in Teamboard ODU.....	27
4.2: Summary of Test result for the User section in Teamboard ODU.	29

CHAPTER 1

INTRODUCTION

Rapid advancements in technology have facilitated virtual collaboration among members working within a team. Individuals are no longer expected to conduct the majority of their work face-to-face, even if they are located in the same building. Employees today often prefer to communicate using virtual work spaces that combine various collaboration functionalities, such as email, video conferencing, versioning control, and data sharing (Boughzala, Vreede and Limayem 2013). In general, a virtual workspace can be created for any team working on projects that involve some form of collaboration. The same concept can be used to facilitate collaboration in teams of students working on collaborative class projects that are supervised by course instructors. However, every organization trying to utilize a virtual collaboration system may have different requirements; hence, assuming the necessary resources are available, it is beneficial to build an application to cater to a particular team's virtual collaboration goals.

This thesis examines an application that has been developed and implemented at Old Dominion University (ODU). This research will determine whether the tool meets the technical requirements and provides better performance than an alternative tool that has been used for a similar purpose.

1.1 Background of the problem

This section describes the current system used by some instructors at Old Dominion University (ODU) to support collaborative project-based work. For each team, a custom made Google site with embedded Google Apps has been used to support a group of remotely located students to work on a team project. The team sites help the users to perform the following major tasks:

- Send/receive messages and updates
- Upload and store project related files
- Add/modify tasks
- Track progress

Some preparatory team building and planning activities are also accomplished using the site, including posting individual profiles and pictures, developing the team charter and conducting synchronous meetings via video conferencing.

The instructor also has the ability to use the site to monitor the progress of each team and can communicate by posting messages on the team's site. The collaboration tool currently being used requires significant time investment to create and manage the sites. At the beginning of each semester, the instructor needs to replicate a Google site for each team project and invite the students to join each team. The students are required to use a Google account in order to access the site.

1.2 Problem statement

In order to increase the performance and usability of the tool from the student and instructor perspectives, a new collaboration tool was developed. This new system should be a significant improvement for both students and instructors over the prior system, currently in use, Google site.

This team site will aid in communication and task tracking and will reduce the need for team members to meet face-to-face. This could lessen the time required to complete a project. The site will also include a dashboard with indicators of team activity and progress. It provides the ability to monitor the team's progress and provide individualized feedback at the individual, group and course levels.

This project is aimed at building, deploying and preliminarily testing a virtual team application (called TeamBoard ODU) that will support collaborative work in teams while providing the instructor with a dashboard to manage teams.

1.3 Purpose of the project

This study is aimed at developing and testing a collaboration tool to determine whether it represents a substantial improvement over the tool currently in use. The analysis will evaluate whether the tool can fulfill the technical requirements outlined in 1.1 to support remote collaboration in student teams as well as provide the ability to manage large numbers of teams more effectively. To summarize, the outline below states the major project requirements that should be met before the collaboration tool can be adopted.

- Admin Section (Instructor section)
 - R1.3.1: Access using a Gmail ID.
 - R1.3.2: Dashboard presenting details of each team.
 - R1.3.3: Display details related to project activity of each team and its members.
 - R1.3.4: Create and/or modify project.
 - R1.3.5: Modify number of team members in a project.
 - R1.3.6: Add automated email notification for a team's members.
 - R1.3.7: Add team charter and project requirement functionality (Admin must be able to modify team charter questions and project requirement if needed).
 - R1.3.8: Add/remove admins.
- Student Section
 - R1.3.9: Gmail account should be used to be log in.
 - R1.3.10: Team members should be able to navigate through the site easily.
 - R1.3.11: Team profile page should hold biographies, and students must be able to edit their own biography.
 - R1.3.12: Separate team charter section that can be edited by team members.
 - R1.3.13: Must contain a project repository section that members can use to store project related information such as files and links to other sites.
 - R1.3.14: Files in project repository section should have a version number.
 - R1.3.15: A discussion section where team members can add and discuss topics.

1.4 Scope of the study

The objective of the study is to understand how Teamboard ODU as an application performs compared to the currently used Google site with embedded applications. Our study will include a comparison of the functionalities, performance and usability of the two collaboration tools.

1.5 Importance of the study

The proposed development, implementation and testing of TeamBoard ODU will explore whether the tool being developed meets the technical requirements while providing additional affordances from the team management perspective. These new affordances include the dashboard to facilitate assignment and access to the collaboration tool for student teams as well

as project tracking and communication between instructor and teams. TeamBoard ODU was developed using established software development methodologies that can simplify any future development and maintenance issues. The existing system based on Google applications had to rely on a third party for actualizations and updates, providing little control to the site administrator.

1.6 Definition of Terms

Design Document Specification (DDS): This document provides the information about the properties and characteristics of a project. This document includes information and description of all the necessary elements related to the software and its development.

Software Requirement Specification (SRS): This is used to describe the intended purpose of the software and its development environment.

Source lines-of-code (SLOC): This is a software metric for measuring the size of the program by counting the number of lines within the program's source code.

Hypertext transfer protocol (HTTP): This is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol that can be used for many tasks beyond its use for hypertext through extension of its request methods, error codes and headers (w3.org). These tasks include being used as name servers and distributed object management systems.

Internet Protocol version 6 (IPv6): This is the latest version of Internet Protocol. It's a communication protocol used to route traffic across the web.

CHAPTER 2

LITERATURE REVIEW

Since the objective is to create and implement an application that will be preferred over the current system within the ODU environment, a systematic management approach to development and deployment of the software is necessary. This section will discuss some well-established methodologies for software development as a foundation for the development of the new software tool proposed in this study.

2.1 Stages of Software Development

According to Christian Dawson and Ray Dawson (Dawson and Dawson 2014), at the highest level, software processes can be represented by the generic SDLC shown in Fig. 2.1.

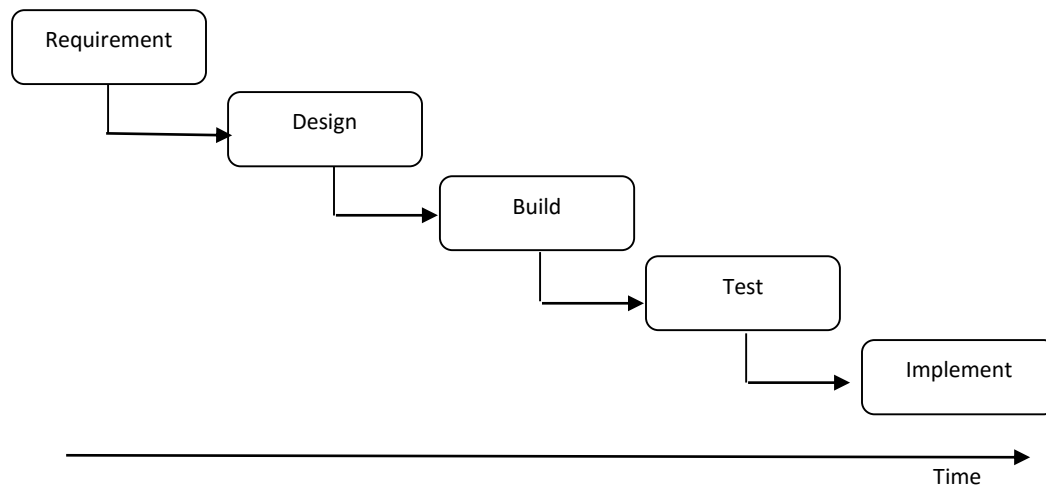


Fig. 2.1: Generic software development life cycle

Figure 2.1 shows the basic stages in software development projects. Every software development process will have some form of requirements capture, design, system building, testing and implementation (Stoica, Mircea and Ghilic-Micu 2013). These five stages will be further described in the next section.

Requirements

The first step of a software development project is requirement analysis. Failure to properly identify and analyze the requirements may lead to failure in achieving the ultimate goal. This helps us design a basic project plan and carry out a feasibility study from economic, operational and technical points of view. A software development team is typically responsible for identifying all the project related factors such as quality requirements, success and failure conditions and any possible project risks.

Design

Software Requirement Specification (SRS) is the basic reference from which the architects set out to create the ideal architecture for the product. Usually, at least one product architecture approach is proposed, and it is documented in a Design Document Specification (DDS), which is revised by all relevant parties. It is based on parameters like: risk evaluation, product robustness, design method, budget and time constraints. The best approach is selected. A design approach clearly defines all architectural modules of the product, along with communication and data flows to and from external modules provided by third parties. Internal design of all modules in the proposed architecture must be presented in clear detail by the DDS.

Build

In this stage, product development starts, and the source code is generated. The objective is to generate this code in a detailed and organized manner. Developers must follow the guidelines of their organization, which often include a particular approach regarding development.

Test

After the build stage, the application faults are reported, tracked, fixed and reanalyzed until a product achieves the necessary quality requirements. In reality, testing begins during the build phase when developers work in the individual modules and test to check for erroneous code. The testing is also done to identify any usability or performance issues.

Implement

Once the application has been tested, it is ready to be implemented and launched for operation in a real user environment. It can be launched on a limited segment and tested in a real business environment. Then, based on feedback received, it can be completely launched as the

final product. The software from this point enters into the maintenance phase where operations such as bug fixes or feature upgrades are performed.

2.2 Choice of Language

During the build phase, selecting suitable platforms such as programming language, framework, and libraries can ease the development process. Prior research has addressed the methodological topics; however, few high quality studies were found describing specific technologies or approaches. In this study, a decision had to be made with regards to the technology (or programming languages) such as JAVA and PHP that can be used for developing Web-based applications.

This section will rely on the work of Lutz Prechelt (Prechelt 2010) to identify the most appropriate programming language to use for the new tool. This will allow us to make a more informed decision when selecting the programming language to be used.

Prechelt's (2010) research used teams of top class programmers (9 three-member teams) who were asked to implement the same requirements for a Web-based system within 30 hours, each team using a different technology platform (Java EE, PHP, Perl, etc). The expectation was to provide new insights into the real (rather than purported) pros, cons, and emergent properties of each platform. The evaluation focused on the following areas in the test process: systems completeness, implementation size, modifiability and robustness.

System completeness

JAVA provided the most complete solution even though scripting languages such as Perl or PHP tend to have higher productivity; however, uniform results were observed for Perl and PHP. Fig. 2.2 (Prechelt 2010) shows the number of functional requirements completed and correctly implemented for each programming language. There were 108 such requirements overall, each of them marked with priority MUST, SHOULD, or MAY. Each bar represents the solution of one team. Two reviewers first independently judged each requirement for each solution and then discussed any discrepancies until they reached agreement." (Prechelt, 2010)

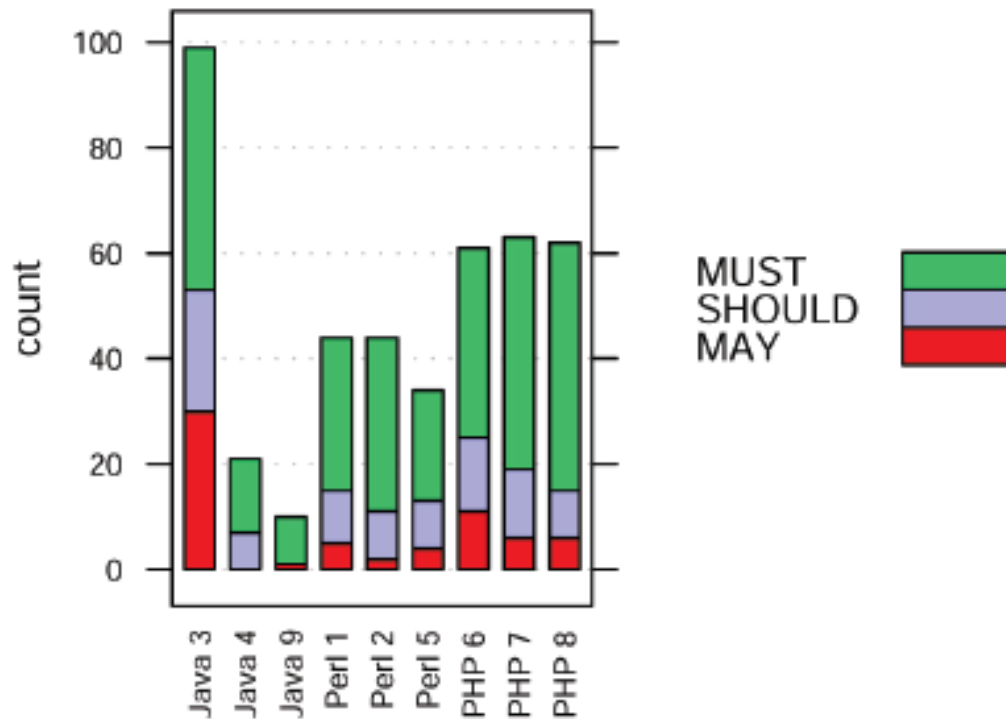


Fig. 2.2. Completeness of solutions by GUI requirements implemented.

Implementation size

Figure 2.3 (Prechelt 2010) compares how much code and other files the teams wrote from scratch during the contest relative to the amount of functionality they realized including the Web service requirements. In this case, we can observe that JAVA solutions are lengthier. Each digit represents the respective team (black for Java, red for Perl, green for PHP); the line is the overall trend line. The SLOC count ignores all files that were reused or automatically generated, even if they were modified manually, and includes only files written entirely by hand.” (Prechelt, 2010)

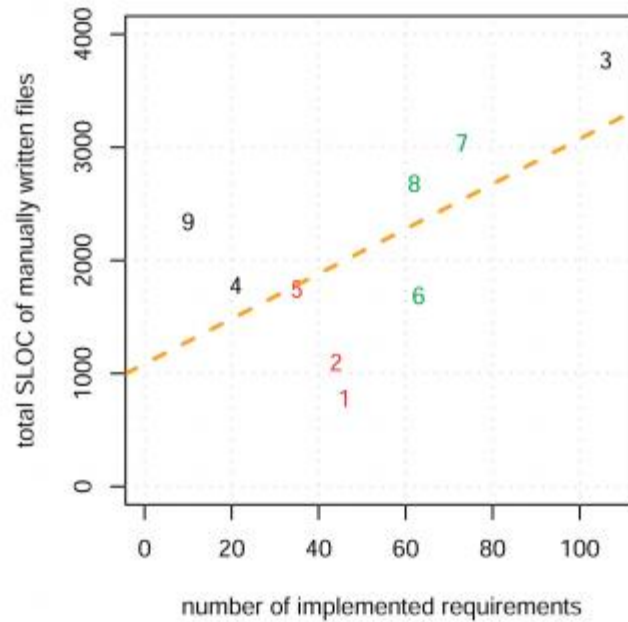


Fig. 2.3: Total size of the solutions in SLOC by functional requirements implemented

Modifiability

The test also suggested that modifiability is easiest to attain for Perl and is hardest for JAVA. The test was done by implementing two simple extensions for each of the solutions obtained by the teams.

Robustness/security

Simple, black-box tests of robustness and security issues were carried out to check for failures. The tests handled elements such as HTML tags, long inputs and Chinese ideograms. Other tests such as email address validation, SQL injection and operations with cookies turned off were also performed. Fig. 2.4 (Prechelt 2010) shows the results of these tests. For all nine teams, each column represents one robustness test. Green color “OK” means correct, yellow “(OK)” means acceptable, light red “!” means broken, and bright red “!!!” means security risk. White areas indicate results that could not be evaluated due to gaps in functionality.” (Prechelt 2010). Overall, PHP results were solid, and JAVA test results were better for cross-site scripting.

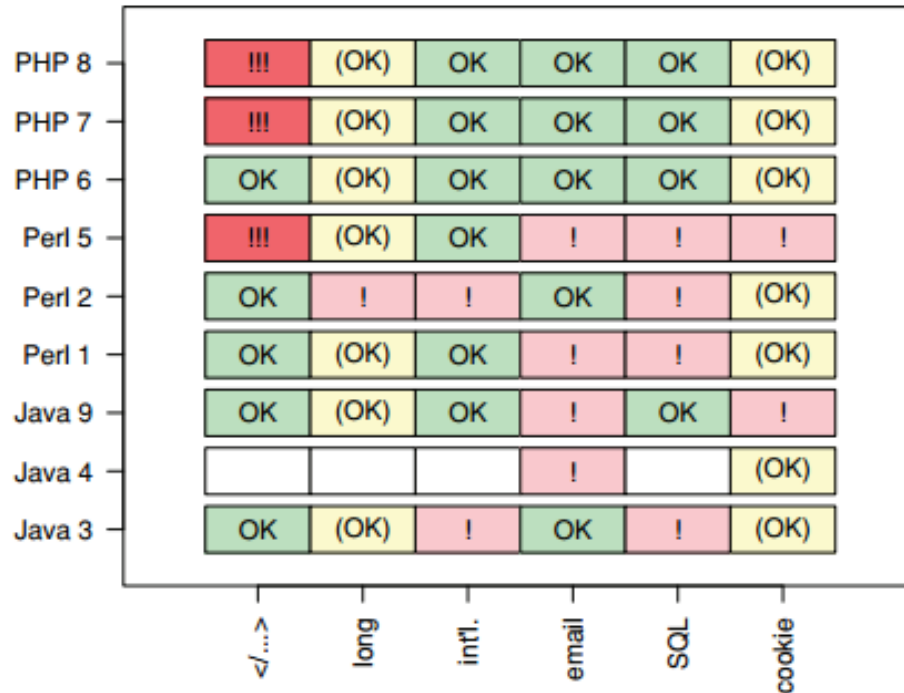


Fig. 2.4: “Summary of the robustness”

From Lutz’s study, it can be observed that, if utilized correctly, all technologies/platforms will yield good outcomes. However, depending on the requirement, we might lean towards using a particular technology. For example, if we are looking for a highly productive programming language, Lutz’s study suggests that JAVA would be the suitable choice.

2.3 Methodologies

Significant effort in identifying the right methodology for application development is a necessity. Failure to do so will result in an expensive and painstaking development process with a high chance of failure. We will briefly discuss 3 major methodologies in this section. These methodologies are used to manage the stages of the development lifecycle discussed in 2.1.

Waterfall

According to Amudha (2010), the waterfall methodology primarily takes care of the analysis phase, with the designing phase and then the implementation phase to follow. Testing is carried out during the entire process. Fig. 2.5 shows the steps of the waterfall lifecycle model.

This involves a series of cascading steps that cover the development process with a small level of iteration between each stage and works well for smaller projects with clearly defined objectives.

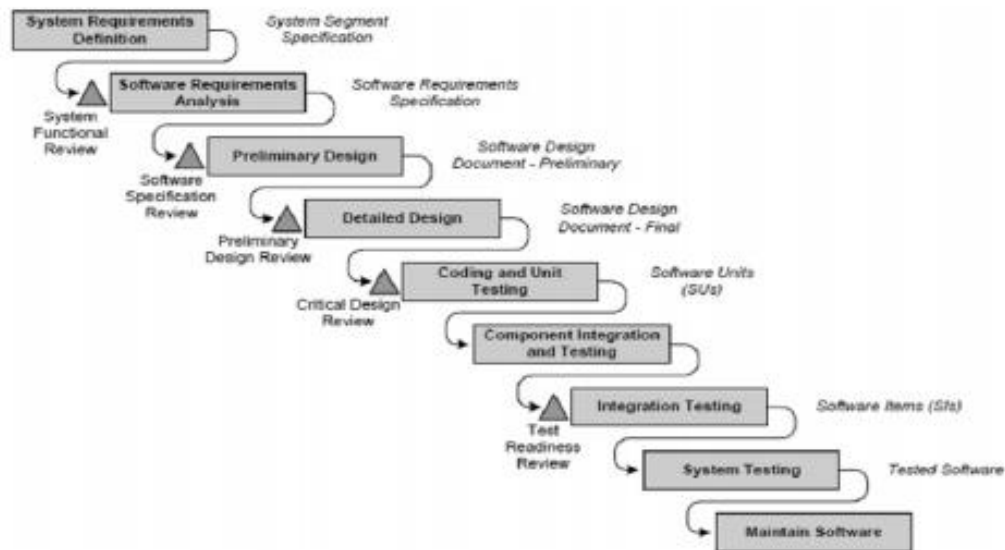


Fig. 2.5: Waterfall lifecycle model.

Iterative Spiral

According to Amudha (2010), iterative and incremental development (Fig. 2.6) is a product development and management methodology that permits iterative project development and cyclic progress assessment. The project cycle is sub divided into vertical segments, called 'slices,' and each slice calls for a deliverable. Each slice is developed individually using the waterfall model. Developers will analyze, design, code and test in a rigid loop.

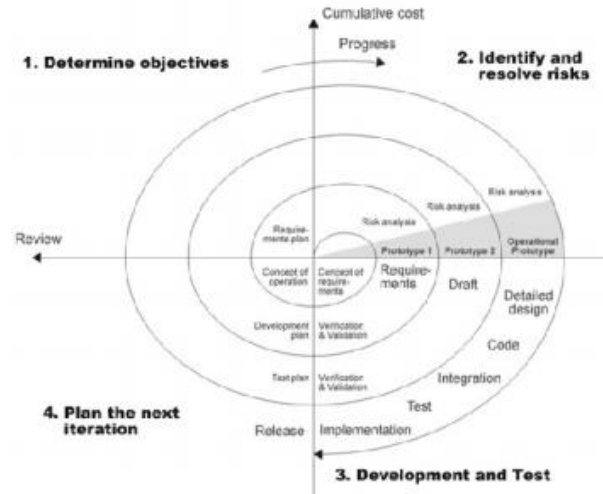


Fig. 2.6: Iterative spiral model.

Agile

“Agile” is a collective term for methodologies (and practices) that have emerged over the past two decades to increase the relevance, quality, flexibility and business value of software solutions (Cooke 2012). An Agile approach is ideal for projects where the requirements frequently change. The objective of Agile is to assist all essential aspects to ensure project success. These include: development, change requests, quality control, and customer associations. This allows us to evaluate a project throughout the development lifecycle using an iterative cycle to build and test followed by an assessment by the stakeholders. Agile methodology can be portrayed as iterative and incremental. In Agile development, customers work in small teams with developers as active team members. For example, customers and developers jointly determine the system features to be implemented in each development cycle. Change of this nature in the user's role suggests that successful acceptance of Agile methodology is concerned not only with software developers and organizations but also with customers who are expected to be collaborative, representative, authorized, committed, and knowledgeable (Chan and Thong 2009).

The review of the literature included a synthesis of the research on software development in the context of web applications. The next section presents the methodology used in the study.

CHAPTER 3

METHODOLOGY

3.1 Primary Objective

The primary objective of this paper is to determine whether TeamBoard ODU constitutes a significant improvement over the currently used application that utilizes Google app's functionalities. For this purpose, both applications will be analyzed and compared with respect to load/stress testing, website performance, site availability, functionality, and usability.

Load Testing or Stress Testing

Load testing can be defined as a performance testing technique conducted to determine how a system behaves under normal and peak load conditions in a specific time frame using a set of different tools and technologies (Nagy and Chis 2015). We will utilize some free testing services to identify how these two applications behave under load.

Website performance

There is evidence suggesting that a website's performance (speed) correlates directly to its success (Meenan 2013). We will perform speed testing on both applications and analyze the results. This process will rely on tools that compute the load time of each of the applications from an independent machine.

Site availability (Ping test)

Since this study is aimed at developing a virtual team collaboration tool, the testing methodology should include an availability test comparing both applications. Servers from different locations will be used to test accessibility or availability of the web applications.

Functionality comparison

The Google site is already in use and is an established application that can be used for virtual team collaboration. As a primary requirement, Teamboard ODU must include all the functionalities of the application currently in use and produce expected results. Our study will include an analysis of the functionalities of Teamboard ODU and compare the results with the Google site currently in use.

User based usability Testing

This category includes usability evaluation methods that involve users in the process of identifying usability problems (Hasan and Abuelrub 2013). Since Teamboard ODU is newly developed and the functionalities in the Google site are already established, usability testing will be performed on Teamboard ODU only. The usability testing included building some test cases, performing the tests, and comparing the outcome with the Google site in use. At the moment, the tests will be limited to 5 users. This may not provide a comprehensive set of results, but it will give us a fair idea of the actual outcome of the application.

3.2 Secondary Objectives

As a secondary analysis, additional elements of Teamboard ODU will be explored. This analysis will provide insight into the development phases and methodology used to build Teamboard ODU.

Development process

We will break down the process that was followed and identify whether it coincides with the development processes discussed in section 2. We will try to understand where Teamboard ODU stands with respect to the development phase, use of methodology and the choice of programming language. A study of the process will determine whether the application is maintainable and expandable.

Unit analysis

From a technical perspective, each unit of Teamboard ODU must produce the intended results efficiently. Even though it cannot be guaranteed that no further improvement is required, at a minimum the analysis of the units must suggest that there are no design flaws. For example, the code section that controls the upload functionality should complete its intended operation within a specific timeframe proportional to the size of the file that is being uploaded. A design flaw in the upload system would cause it to either take a significant amount of time or to fail. This study will evaluate the time it takes for each major unit of Teamboard ODU to complete and determine whether the units are working as expected.

The reason to use a secondary analysis is to assert whether the system is capable of performing to the expected level even if in some areas the existing application may yield better results.

3.3 Hypotheses

As mentioned earlier, we are looking to evaluate whether Teamboard ODU is a significant improvement over the currently used system. Therefore, our tests should indicate that Teamboard ODU provides better results in the points mentioned earlier in this section.

Based on the test data available, the following hypotheses are proposed:

Hypothesis 1: Automated load testing using virtual agents suggests that Teamboard ODU takes less time to perform the initial HTTP transactions.

Hypothesis 2: Teamboard ODU provides faster response time that suggests better availability.

Hypothesis 3: Functionalities of Teamboard ODU can be compared to the functionalities of the Google site. Also, any missing functionalities can be developed and integrated into the system.

Hypothesis 4: Each unit is fully functional with little processing time that indicates there are no glitches in the system.

Hypothesis 5: Website performance comparison indicates that Teamboard ODU requires less time to perform data processing and loading.

Hypothesis 6: Secondary analysis supports the feasibility of any future development.

Free tools for load and availability tests were used to analyze areas related to performance. We also temporarily built external modules that calculated the processing time of each unit of Teamboard ODU. Functionality checks and usability tests were conducted comparing both systems following a predefined testing protocol.

3.4 Data Analysis

For analysis purposes, we will rely on both primary and secondary data. One source of the primary data will be the calculation of processing time of each of the units of Teamboard ODU. For example, when a user logs in and the home page loads, we will calculate the time it takes for the entire process to complete. In this particular case, our system will determine whether the individual is a valid user and then will load all the groups that the individual may belong to. For the scope of our paper, we will not debug each step of this process; however, testing this scenario with multiple login attempts and analyzing the processing times will help us understand whether there are potential issues.

For usability and functionality testing, primary data were used. Test cases for the key functionalities were developed to determine whether the application was behaving in the expected manner. For example, to login into Teamboard ODU, a user must use a Google account. We will also perform a comparison test between the two sites and check for the applications' capabilities.

It must be mentioned that it is very difficult to perform some of the tests that we have discussed so far on the Google site because we cannot see the underlying architecture. For example, we cannot calculate the processing time from the code level; however, we will be able to check the loading time from within the browser (as part of the secondary data analysis). The secondary data includes results from the following tests: browser load time, virtual user load testing and availability test.

Browser load time

This test will determine how a machine's browser performs while loading both sites. There are built in tools available (within the browser) that compute the sites' loading times. Comparing the times using different browsers on different machines will provide insight into how the browsers are handling the sites.

Virtual user load testing

This test is aimed at comparing the capability of both tools to handle load. This test will make use of a free load testing tool that uses virtual agents and checks how the applications are responding as the number of users increases. It should be mentioned that load testing can be performed in many ways, using many different applications. Different applications use different

algorithms and techniques to perform the load testing. We will use a similar freely available load testing tool, and since we will be using the same tool for testing both applications, we can consider this test unbiased.

Availability test

One of the major requirements of any virtual collaboration site is that the tool is accessible and available from any location in the world. There are many free online tools that allow us to determine whether the sites are reachable from different locations. We will use such a tool to ping both Teamboard ODU and the Google site to see if they are responding (which suggests they are available) and how fast they are responding.

CHAPTER 4

RESULTS

The purpose of the study was to find out if the newly developed Teamboard ODU is a significant improvement over the currently used application. For this, we conducted a number of testing scenarios (mentioned in chapter 3) to determine whether Teamboard ODU meets its intended objective.

4.1 User Load test

We need to understand how both systems will behave under load. The idea is to simulate a scenario where a number of concurrent users will be loading the websites. The best approach for this uses any automated tool that will perform this load test.

The load test was conducted using a freely available online service (<https://loadimpact.com/>). This tool loads the sites using virtual users and calculates the time required for the sites to respond. The load test was performed for 5 mins, gradually increasing the number of virtual agents. In the first instance we used 50 virtual agents, and in the second instance we used 25 agents. Figs. 4.1, 4.2, 4.3 and 4.4 below show the results of the tests.

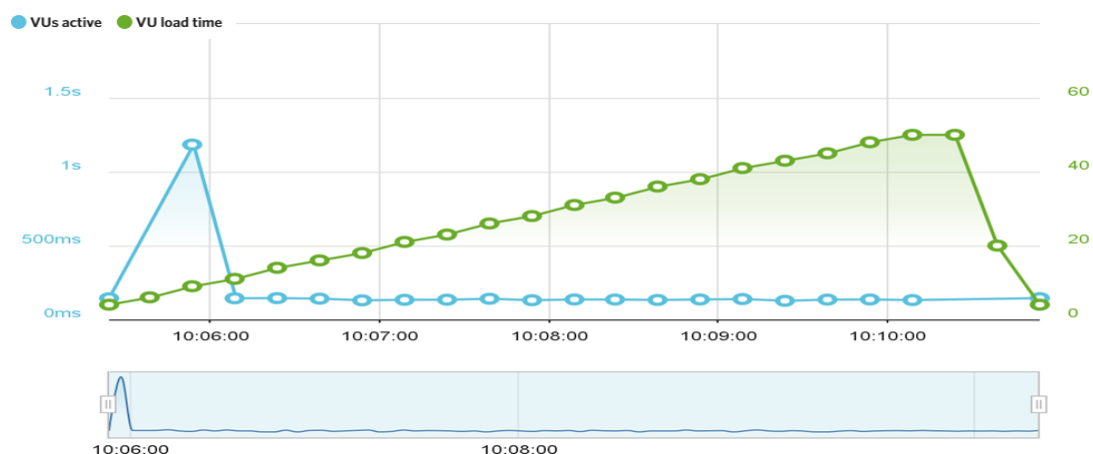


Fig. 4.1: Load time for Teamboard ODU with 50 users

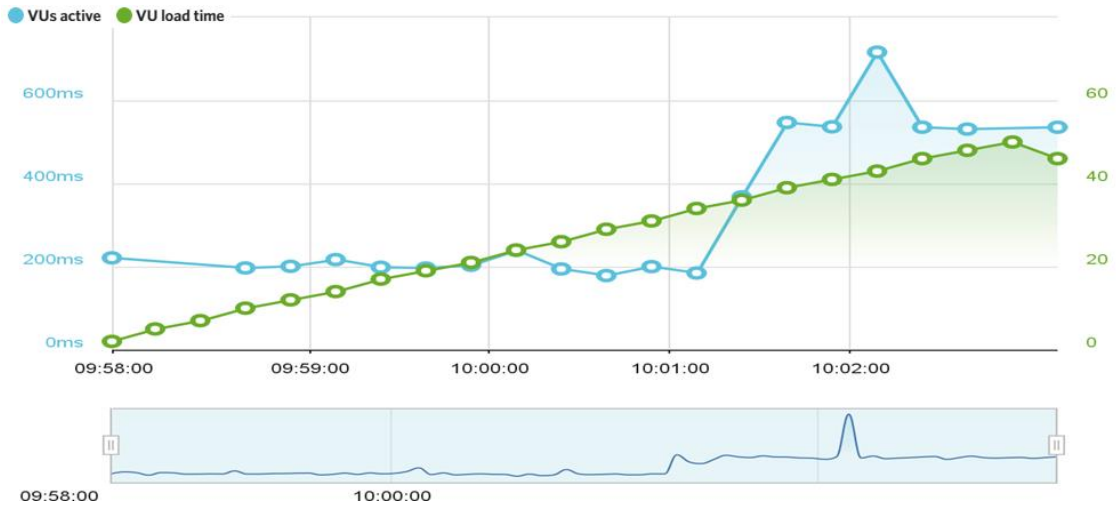


Fig. 4.2: Load time for the Google site with 50 users

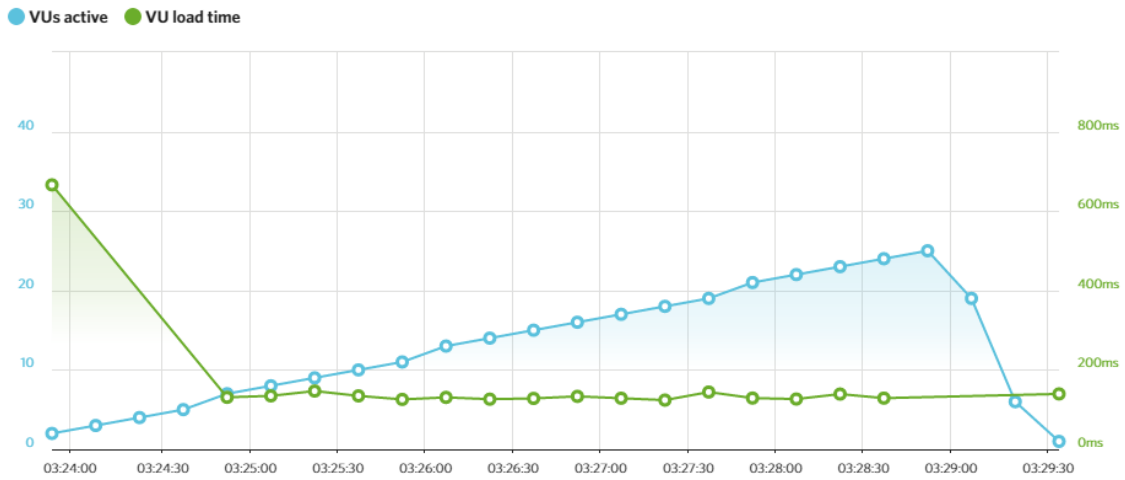


Fig. 4.3: Load time for Teamboard ODU with 25 users

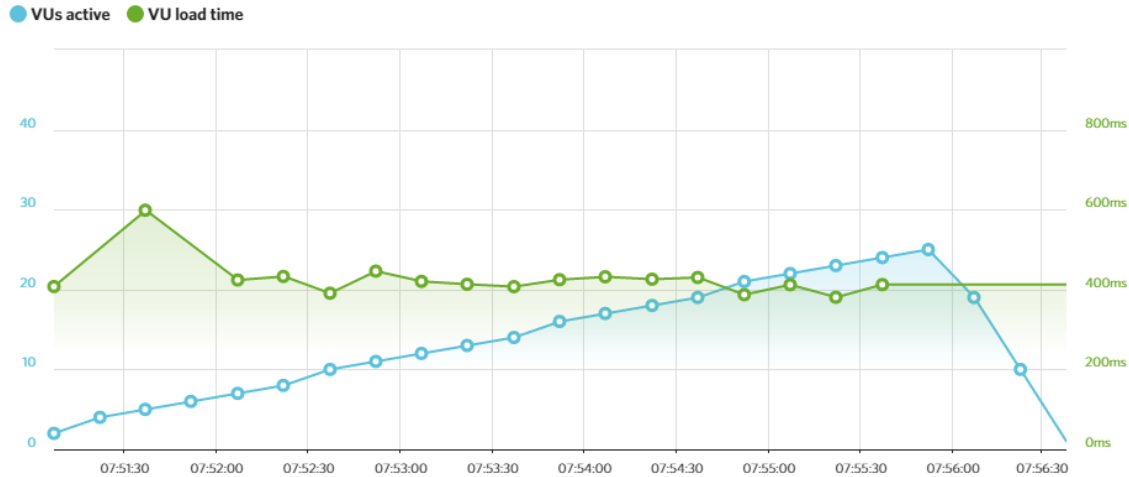
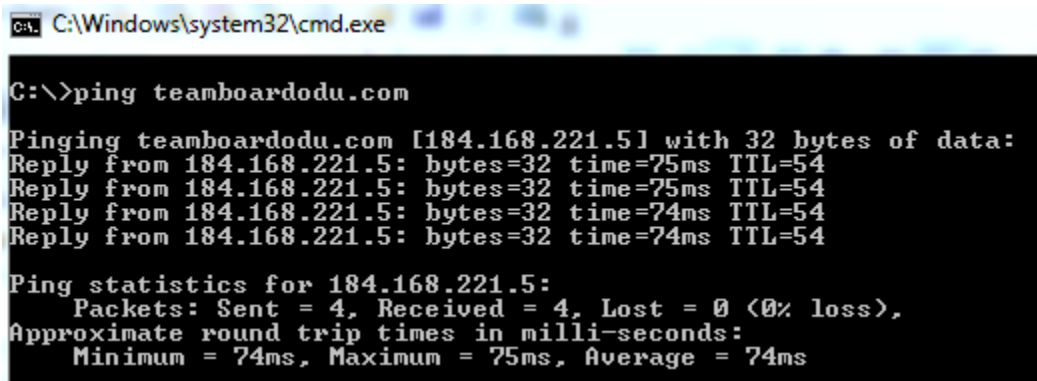


Fig. 4.4: Load time for the Google site with 25 users.

Looking at the figures above, we can conclude that the time needed for Teamboard ODU was less for both test cases; this suggests that Teamboard ODU should perform better under load.

4.2 Availability test (ping test)

As a virtual team collaboration site, Teamboard ODU must be visible from any location on the globe. The best way to determine if this is true is to check whether the server is responding in a timely manner from different locations. The method is to send an echo request packet to the intended address and wait for a reply. The test is successful if the destination address replies back to the echo request in a timely manner; this proves that the destination server is alive and acknowledging any request from the client machine. As an example, if we want to check whether www.teamboardodu.com (which is our web address for Teamboard ODU) is available and responding properly, we can simply use any machine and send a ping request. The screenshot below (Fig. 4.5) shows the results of such a ping request.



```
C:\Windows\system32\cmd.exe
C:\>ping teamboardodu.com

Pinging teamboardodu.com [184.168.221.5] with 32 bytes of data:
Reply from 184.168.221.5: bytes=32 time=75ms TTL=54
Reply from 184.168.221.5: bytes=32 time=75ms TTL=54
Reply from 184.168.221.5: bytes=32 time=74ms TTL=54
Reply from 184.168.221.5: bytes=32 time=74ms TTL=54

Ping statistics for 184.168.221.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 74ms, Maximum = 75ms, Average = 74ms
```

Fig. 4.5: Screen shot of a ping test from a machine.

To perform the intended test, we can use one of many free online services and perform this test with very little effort. In this case, we have used such an online service (<https://www.dotcom-tools.com/web-server-performance-test.aspx>) to test the response time of both Teamboard ODU and the Google site application. Five ping tests for each application were used, and the results are plotted in Fig. 4.6.

Note that for presentation purposes, two locations for Teamboard ODU and one location for Google were not included in the chart above (timed out: response time exceeded 20000 ms). Also, Teamboard ODU was not deployed in IPv6, which is the most recent version of Internet Protocol; hence, the CA, USA location response should be ignored at this stage. The complete test results can be found in the appendix.

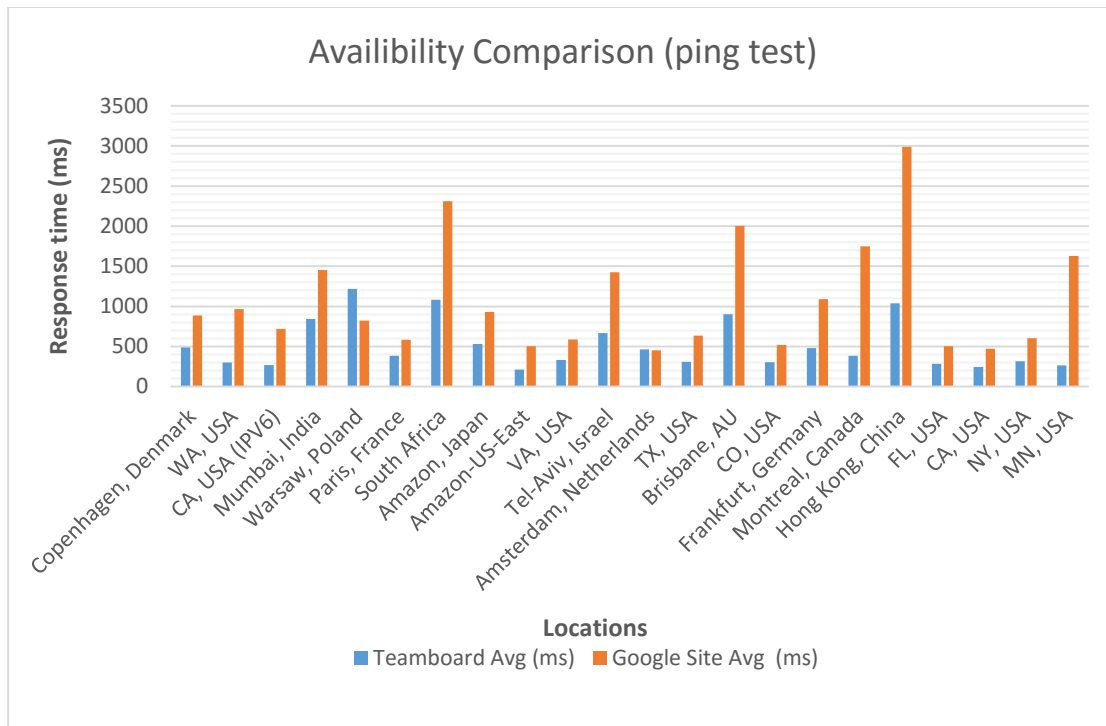


Fig. 4.6: Ping test result comparison between the two sites.

Based on the results, we can see that average response time from almost all locations was higher for the Google site, suggesting that the application will respond slower. Therefore, ping tests confirm that Teamboard ODU will be a significant improvement over the currently used application with regards to response time. This indicates that the overall response time of the new system will be faster and users will have a better browsing experience. It is worth noting that there may not be a significant difference in a single user's usability experience when the number of concurrent users is low. However, in a scenario where the system has to respond to a large number of users simultaneously, browsing time will decrease due to a lower ping time.

4.3 Browser load test

For proper functionality and user experience, the sites must load in a client's browser(s) without issues. This test is intended to compute the time that a browser needs to load the sites. Note that this is dependent on the test machine itself, as theoretically a slow machine can

significantly affect the load time. Thus, in order to ensure accurate results, we will use one machine to test the browser's load time. Also, both Teamboard ODU and the Google site may have browser compatibility issues, so our test was performed on Google Chrome, Version 48.0.2564.109 m, which successfully loaded the pages that were tested.

For this test, we utilized a free Chrome extension called Page load time v1.2.4. This tool calculates the time starting from page load request initiation to completion. The test method was to first stop any process that may affect the computer's processing time and then load both the Google site's and Teamboard ODU's major components (home page, project repository, etc.) and record the load time. Fig. 4.7 shows the average of 5 load times for each major component.

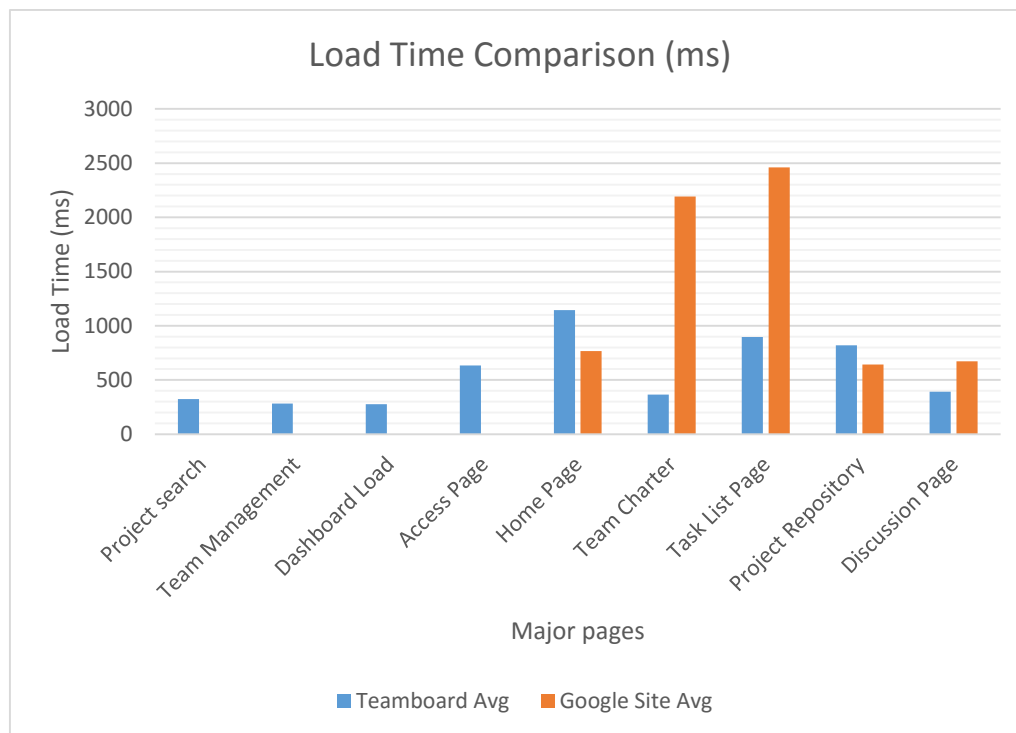


Fig. 4.7: Browser's load time comparison.

The four administrative components of the new tool (Project Search, Team Management, Dashboard Load, Access Page) are not available in the Google site application. Therefore, the load time could not be compared for test purposes; however, these functions are included here to provide an overall picture of both applications with respect to load times. Similar administrative functions that Google site does have are not relevant to our study as they are not integrated as a part of the application. Later in this chapter, we will look at these functionalities from the perspective of usability and functionality.

Based on the results, we can see that for the common components, the browser load time is much higher for 2 cases in the Google site. Overall, Teamboard ODU has a faster browser load time. Thus, we can conclude that Teamboard ODU loads faster in the browser suggesting a better user experience.

4.4 Module processing time

This test is to verify that there are no glitches in the system. Since we cannot perform this test on the Google site, we will not be able to compare both sites. For this test, we will rely on primary data.

The objective is to calculate the processing time of each module. For example, if the user clicks the homepage, we will check the time it takes for the system to acknowledge the click, populate the data and return the result to the browser. If there are any anomalies in this process, the processing time is expected to be very high, so we have performed 5 tests for each major area and checked for the processing time. Fig. 4.8 shows the results obtained for each test.

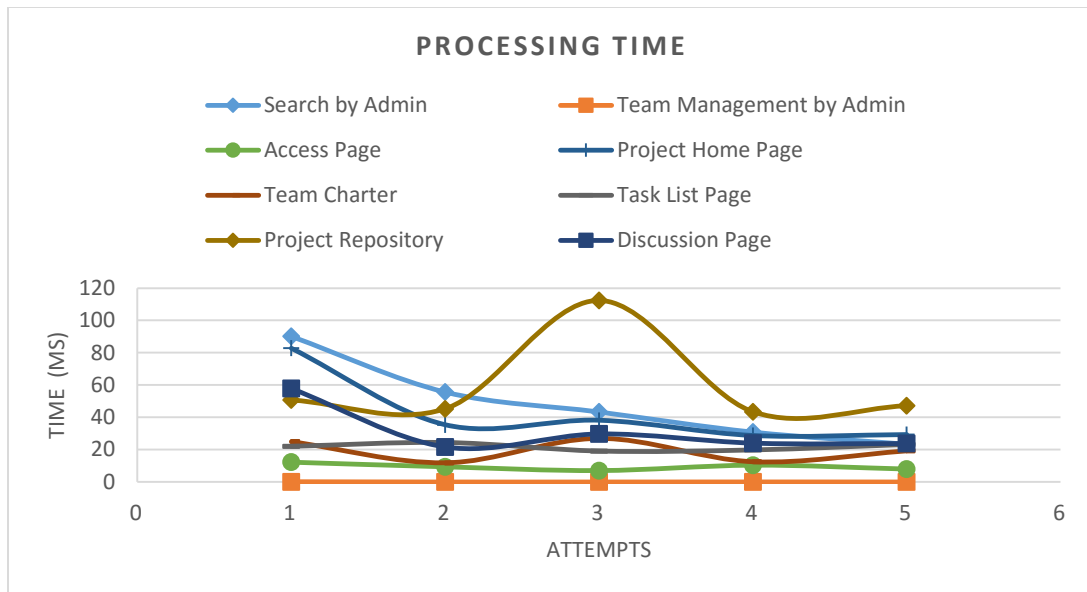


Fig. 4.8: Processing times of each functionality.

The results indicate that none of the processing times showed any unusual behavior, suggesting that modules are well written. Therefore, by looking at the processing time we can conclude that the probability of an error occurring is very small. The overall processing time, however, may increase as team projects start receiving more input from users (files, messages, etc). For example, a discussion page with 10 topics is expected to load faster than a page with 20 topics as more data processing will be required to populate the results page for viewing in the browser.

Observation

The processing time is higher during attempt 1, and gradually falls during the succeeding attempts. This is expected as after the first test, some of the related data gets loaded into the system's cache. If we clear the system's cache after every attempt we will have higher load time, but since we are investigating whether there are flaws in the modules, we are not clearing cache and attempting to replicate the expected result. A major variation at this point would have indicated a flaw in the code.

“Project repository” involves a lot of data processing before it populates the results. This data is fetched from the database during every load. At the moment, Teamboard ODU does not

have an internal process to cache this data. In the third attempt, we can see that the loading process took some extra time. Considering the data processing time, we can suggest this happened due to the read speed of the storage drive where the database resides. Thus, even though we can suggest that this is not a factor to consider in identifying an error in our internal modules, we have discovered one area that can be improved for better performance.

4.5 Usability and functionality

From the usability perspective, we need to understand how the Google site and Teamboard ODU are comparable. In order to provide a comprehensive assessment, test cases were developed and tested to isolate the differences in the applications. To summarize, we have identified the following differences between the two applications (full results with the complete test steps are given in the appendix). A testing expert was utilized to provide an unbiased comparison of the tools. The results are presented in the following section. The reference numbers next to the test objectives in Tables 4.1 and 4.2 highlight whether this validates any of the requirements mentioned in section **1.3**.

Administrators' section

Table 4.1 identifies the differences in the admin functionalities of the two applications. In the left column, we have the test objectives, and observations of each test are listed in the corresponding right column. Complete test data are given in Appendix D.

TEST OBJECTIVE FOR COMPARISON	Differences between Teamboard ODU & Google site
Navigate to teamboardodu.com	Teamboard ODU can be navigated by typing address in a browser. Google site requires instructor's invitation.
Application's user interface	Google site has better look and feel.
Access the site teamboardodu.com (R1.3.1)	Teamboard ODU can be accessed easily. Google site needs invitation.
<i>Usability of the application</i>	<i>Google site has Site map for easier navigation within the site.</i>
E-mail account access to the application.	Access request process can be confusing in Teamboard ODU. Google site does not have access request option.
Log out/Sign out" functionality.	Teamboard has faster logout process.
Verify the functionality of the link "Dashboard" (R1.3.2)	Visibility of features works better in Teamboard.
Search project" functionality (R1.3.3)	Teamboard displays ease of access in searching for a project. Google site does not exhibit this feature.
Create new project" functionality (R1.3.4)	Fully functional project creation is easier in Teamboard ODU.
Functionality of "Assign team members" to a team (R1.3.5)	Assigning team members to a project is simpler in Teamboard ODU than Google site.
Verify if e-mail notification was automatically sent to the after creating a new team (R1.3.6)	Google site does not have automatic notification feature, Teamboard ODU does.
Verify if e-mail notification was automatically sent to the students who were added to a team	Teamboard ODU has automatic notification feature.
Modify Project" functionality (R1.3.4) (R1.3.5)	Teamboard ODU allows modifying project after they have been created. This cannot be done in Google site.
Delete team" functionality.	Similar functionalities and results in both the application.
Team Charter" functionality.	Team charter modification is easier in Teamboard ODU. Google site requires a file upload.
Validate the functionality of "Modify Team Charter" (R1.3.7)	Much simpler to edit team charter in Teamboard ODU than Google site.

Table 4.1: Summary of Test result for the Admin section in Teamboard ODU.

Table 4.1 Continued...

TEST OBJECTIVE FOR COMPARISON	Differences between Teamboard ODU & Google site
<i>Deleting Team charter.</i>	<i>Team charter cannot be deleted in Teamboard ODU.</i>
<i>Add Comments" under Team Charter page.</i>	<i>Comments section is not available in Teamboard ODU, but is present in Google site.</i>
Assign Task" functionality.	Google site gives liberty to the administrator to assign/upload task list to users. Teamboard ODU lacks this functionality.
Administrator's access to add a new administrator.	Both the sites have similar functionality.
To Remove the Administrator access of existing administrator (R1.3.8)	Both the sites have similar functionality.
Project Requirements functionality.	Teamboard ODU has better visibility of project requirements unlike Google site.
To verify the Project tracking functionality.	Teamboard exhibits all the required features, whereas, Google site does not
Feedback functionality.	Google site has the Add comments functionality. Teamboard does not.
Site updates from administrators' perspective.	Google site has the Activity list visible to both user & Administrator. Teamboard Activity list is visible only from the user's perspective of the site.

User's section

Similarly, Table 4.2 identifies the differences in the user functionalities of the two applications. The left column lists the test objectives, and the observations are in the right column. Full test results are given in Appendix D.

TEST OBJECTIVE FOR COMPARISON	Differences between Teamboard ODU & Google site	
Ease of navigation	User can access TeamBoard ODU site easily by typing in the browser. Google site link needs to be provided by the administrator.	
Validate authentication to the site (R1.3.9)	Can be accessed with any synced Gmail site, whereas, Google site needs to have ODU mail to access.	
GUI of the home page (R1.3.10)	Both sites exhibit same kind of structure.	
Functionality of Team profile (R1.3.11)	Can edit bio directly in home page in Teamboard ODU. Google site has a Team profile page.	
Functionality of Team Charter (R1.3.12)	User can type answers in Team charter in Teamboard which looks much more appealing than Google site, where user has to upload answers in a file.	
Rigidity of Team Charter	It is easier to modify in Teamboard ODU unlike in Google site.	
<i>'Add comments' in Team Charter</i>	<i>Google site has provided Add comments section.</i>	
Functionality GUI of Project repository (R1.3.13)	Teamboard ODU has better usability than Google site.	
Functionality of 'Add' link in Project repository	Both sites have similar add link functionalities.	
Creating a folder	Repository section has minor differences in look and feel.	
Move to Tab	Both sites have same structure in this aspect	
<i>(1) Functionality of Manage Files Tab</i>	<i>Not present in Teamboard ODU.</i>	Refer to Appendix for the differences in these 'Files tab' functionality tests.
(2) Functionality of Manage Files Tab	Both sites have same structure in this aspect.	
(3) Functionality of Manage Files Tab	Google site has this option.	
<i>Functionality of View and Download</i>	<i>Google site has option to view the specified file without downloading.</i>	
Functionality of Expand arrow of a folder	Both sites have same structure in this aspect.	
Validity of 'Version no. (R1.3.14)	Both sites have same structure in this aspect.	
Functionality of 'Discussion' link (R1.3.15)	Better usability in Teamboard ODU than in Google site.	
Feasibility of accessing older posts in Discussion page (R1.3.15)	Both sites have same structure in this aspect.	
Availability of site overview	Google site has this feature which allows usability.	
Project deadline view on home page	Preferable to have it in Teamboard ODU. This is present in the Google site.	

Table 4.2: Summary of Test result for the User section in Teamboard ODU.

The user and admin functionalities comparison tables demonstrate how the two applications differ. Some of the functionalities in the Google site (the *italicized* lines marked RED in both Tables 4.1 and 4.2) are not present in Teamboard ODU as they were not part of the requirement. Therefore, even though the usability comparison test considers these to be positive attributes of the Google site, we will not consider these as drawbacks for Teamboard ODU.

The comparison tables can be used as a guideline for future work on Teamboard ODU. Nevertheless, based on the results to date, Teamboard ODU's functionalities can be compared to those of the Google site and can serve as an effective replacement. The new tool also has enhanced functionalities from the team management perspective (dashboard, team tracking, etc.).

4.6 Development method

In order to justify the feasibility of future development of Teamboard ODU we need to look at its design elements. This is a theoretical approach to affirm that all development and implementation guidelines were followed when building our application. Even though following the guidelines does not justify a successful integration of additional modules (which will occur in future development), we can predict that such future enhancements of Teamboard ODU will be possible.

In chapter 2, we discussed some important aspects of software development. We will try to understand how Teamboard ODU meets these characteristics with respect to its development lifecycle.

Requirement Analysis

The current application (Google site) was used to help determine the capabilities and specifications expected of Teamboard ODU. This was an effective tool for understanding the specific expectations of the application's features. Every effort was made to reduce any possible ambiguity through frequent communication with users.

Design

Based on the requirement analysis, we focused on designing an architecture that allows development of independent modules/features and integrating them into our system. The idea is presented in Fig. 4.1 below. If explained in simple steps, the user initially requests some data or operation (for example clicks on the task list). The request is transferred to the modules by the

front end of Teamboard ODU (for example, a request is sent to the backend of the application to fetch all task list related information). After the necessary data processing, the result is sent back to the user in a meaningful browser readable format.

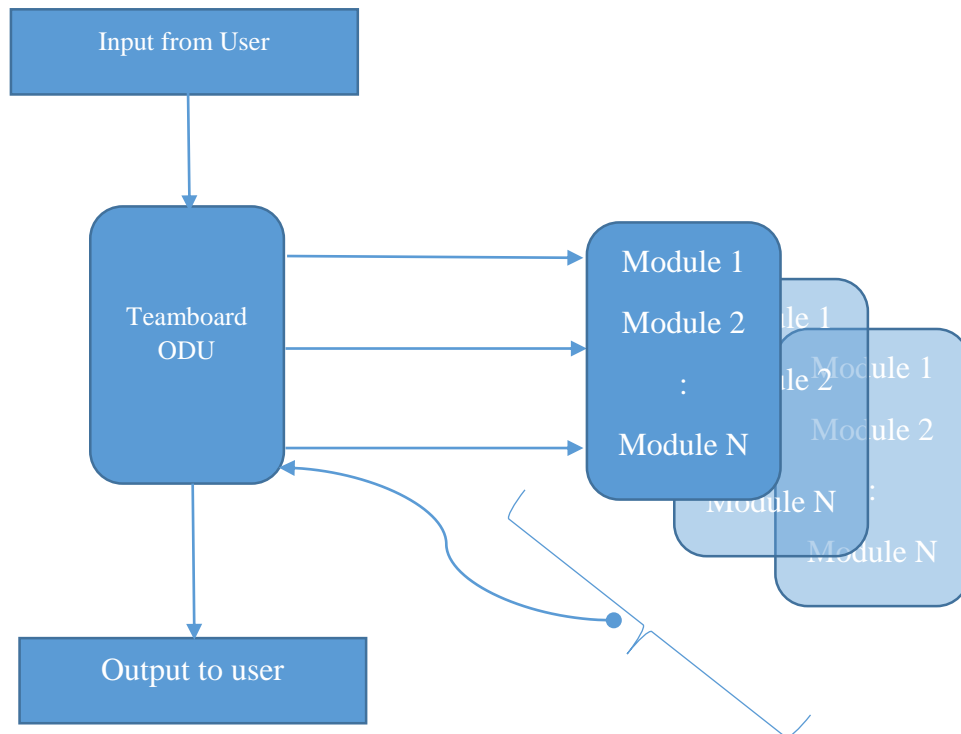


Fig. 4.9: Architecture of Teamboard ODU

With this design architecture, little effort is required to build additional functionalities if needed. This design approach has helped us in responding to factors (such as performance and add-ons) that affected our system's modularity. For instance, if we want to add another module for video chatting, we can simply develop this as a separate entity that will be treated as a new module.

Build

Once a clear set of requirements and architecture were established the development process started. The process followed an established development architecture and framework for web services. The programming language used was JAVA, and the necessary machine configuration was set up to host web services using JAVA. In chapter 2, evidence was provided that JAVA outperforms other languages in several areas; hence, this project used JAVA as the preferred approach.

This project was relatively small in the large context of software development projects, so it used a waterfall technique as the main project lifecycle method. A description of the general waterfall approach was discussed in Chapter 2. This approach afforded flexibility in developing the modules as the objectives were clear.

Test

Tests were performed at every stage of the lifecycle to identify and track errors in the system. We did not have a defined quality requirement, and our main objective was to make the process of each individual module work. Quality checks and system testing were performed and if needed, a fix in the code was applied until the modules were producing the intended results. Special care was taken to ensure that these fixes did not affect any other part of the application.

Implement

After the testing phase, the project was implemented in a real user and live environment. The application is fully functional and usable; however, since we did not officially launch this application, many bugs may be discovered when full deployment takes place. This research was aimed at preliminary testing with the understanding that further massive usability testing will be necessary for full deployment of the tool.

CHAPTER 5

CONCLUSION

After evaluating the findings, results indicated that Teamboard ODU can be implemented as a collaboration tool for student groups and will be a good replacement for the currently in use Google site. The results of our tests suggest that Teamboard ODU meets all the necessary objectives. It is capable of handling load better, which means concurrent users will not have any issues with the site's usability. In comparison to the current Google site, the new tool can be seen from different locations in the world and responds faster to requests from these locations. Teamboard ODU has the required functionalities, and it produces the expected output. Teamboard ODU also has additional functionalities in terms of team management that are not available with the current tool. The tests did not reveal any issues in the coding, suggesting that the probability of any code related failure is very low. End user tests confirmed that Teamboard ODU loads faster and performs better in the browser. A study of the development approach shows that Teamboard ODU followed all the guidelines and steps needed for a proper web application development. Additional modules can be integrated, and theoretically maintenance can be easily performed. Overall, all our tests indicate that Teamboard ODU was properly built and deployed and can be used as a good and effective substitute.

LIMITATIONS AND BENEFITS

Maintenance

Teamboard ODU is a custom developed web application. As a result, during the maintenance phase, we will need a team with enough system knowledge and skills to be able to maintain the system. In the future, resources could become a factor during any maintenance or development. However, because this is custom built and owned by ODU, the system is not dependent on any third party applications, and any modifications, updates or enhancements can be done easily provided we effectively manage our resource constraint.

Modifiability

The Google site is free, can be modified and requires very little effort to build the site. Teamboard ODU, on the other hand, requires a hosting location and has limited functionalities. However, hosting a system on any external location is relatively cheap, and most of the functionalities in the Google site can also be integrated into Teamboard ODU.

Hardware resources

The tests we performed were dependent on machine configuration. It is reasonable to expect that machine configuration will change if we move this to a separate hosting location. This will effectively change the performance of Teamboard ODU with respect to its processing and response times. This, in reality, offers the flexibility to choose the hardware for the system. Depending on the load and the number of users, we can increase or decrease the system's resources. For example, if we host our application in a shared environment, we may move to a dedicated environment to increase performance.

Outsourcing vs In-house development

As stated in 6.1, if we do face issues with managing system resources, at some point the question of outsourcing may arise. Outsourcing will provide additional options if an enhancement is to be implemented, but this will drastically increase our cost of operation. We will receive a quality product in a very short period of time, but our dependency on external developers may backfire in the long run. On the other hand, if Teamboard ODU is viewed as an application that provides a real life opportunity to improve the programming skills of students enrolled at ODU, steps should be taken to facilitate its development.

In either scenario, we will need an additional environment for user acceptance testing (UAT) and quality assurance (QA) testing. We will also require monitoring to ensure the code written by students meets necessary coding standards as the students may not have enough experience with software development.

FUTURE WORK

It is expected that Teamboard ODU will grow over time even though we do not have tangible data to predict future additions or changes in requirements. Hence, in the future, effort will be required to develop additional functionalities. We also need to conduct a usability survey once the system is fully in use. This will give us the opportunity to identify any area that may require further development.

A thorough analysis should also be performed before fully implementing Teamboard ODU. We will need to identify the operating expenses and usability issues that may surface if hosted in a shared environment. Initially, the objective should be to at least match the resources currently being used; in the future additional resources might be required in order to increase the application's capabilities.

REFERENCES

- Amudha, T.K. 2010. "Agile-Software Development Methodology." *Advances in Computational Sciences & Technology* 3 (2): 257-265.
- Boughzala, Imed, Gert-Jan de Vreede, and Moez Limayem. 2013. "Team Collaboration in Virtual Worlds." *Journal of the Association for Information Systems* 13: 714-734.
- Chan, Frank K.Y., and James Y.L. Thong. 2009. "Acceptance of agile methodologies: A critical review and conceptual framework." *Decision Support Systems* 46 (4): 803-814.
- Cooke, , Jamie Lynn. 2012. *Everything You Want to Know About Agile : How to Get Agile Results in a Less-than-agile Organization*.
- Dawson, Ray, and Christian Dawson. 2014. "Software Development Process Models: A Technique for Evaluation and Decision-Making." *Knowledge and Process Management* 21 (1): 42-53.
- Hasan, Layla, and Emad Abuelrub. 2013. "Usability Testing for IAJIT OpenConf Journal Management System." *JOURNAL OF SOFTWARE* 8 (2): 387-396. doi:10.4304/jsw.8.2.387-396.
- Meenan, Patrick. 2013. "How Fast is Your Website?" *Communications of the ACM* 56 (4): 49-55. doi:10.1145/2436256.2436270.
- Nagy, Alexandra, and George Sebastian Chis. 2015. "Sensitivity Analysis and Patterns Implementation on Load Testing Software." *Journal of Software* 10 (12).
- Prechelt, Lutz. 2010. "Plat_Forms Is there one best Web development technology?" *Communications of the ACM* 53 (8): 146-149.
- Stoica, Marian, Marinela Mircea, and Bogdan Ghilic-Micu. 2013. "Software Development: Agile vs. Traditional." *Informatica Economica* 17 (4): 64-76.
- w3.org. n.d. *Hypertext Transfer Protocol -- HTTP/1.1*.
<https://www.w3.org/Protocols/rfc2616/rfc2616.html>.

APPENDIX

APPENDIX A

Figure A1

Full availability comparison results

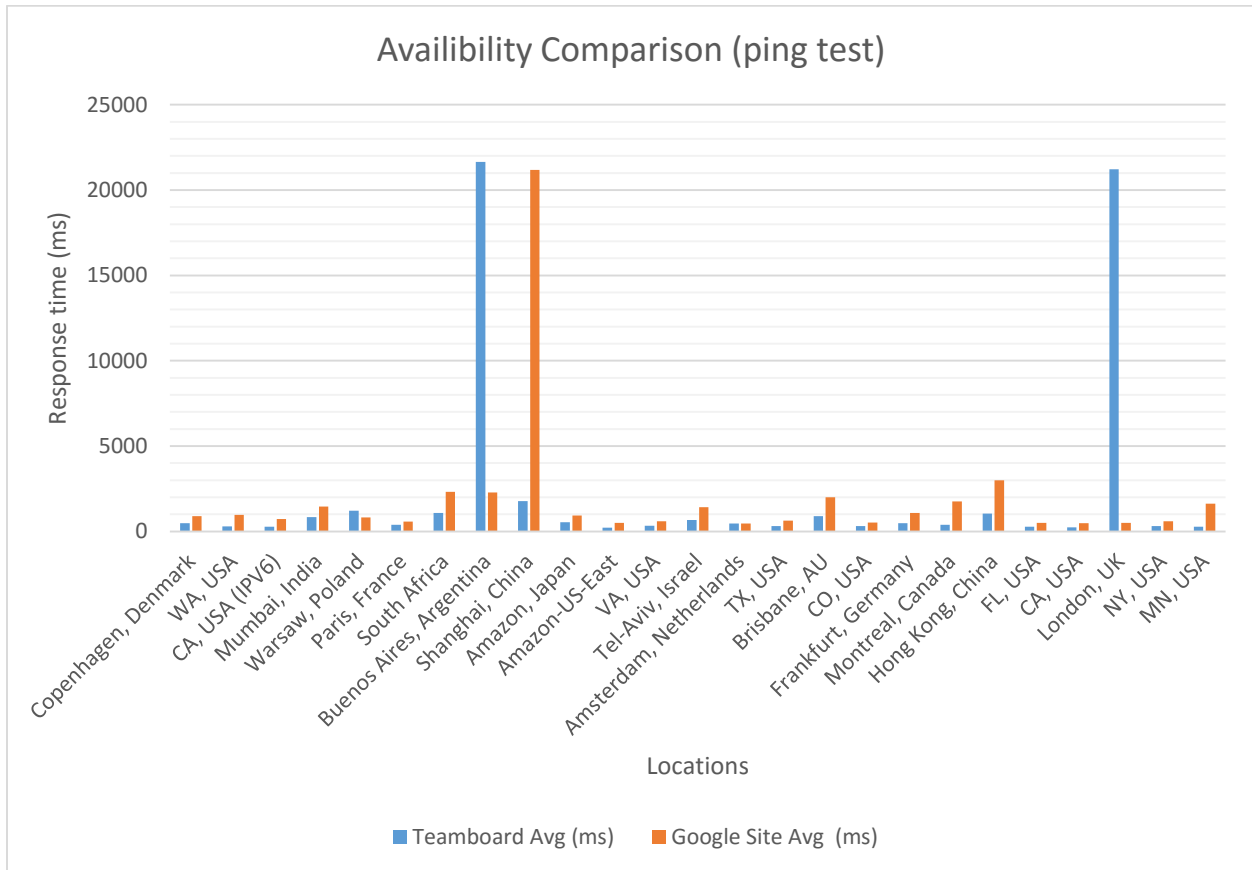


Table A1

Complete Test Result of Availability tests for www.teamboardodu.com

S = Success, F = Failure, RED = Failed, YELLOW = Ignoring IPv6 results.

Teamboard ODU											
	Test 1		Test 2		Test 3		Test 4		Test 5		
	Time	state	Time	state	Time	state	Time	state	Time	state	Avg.
	ms	S/F	ms	S/F	ms	S/F	ms	S/F	Ms	S/F	(ms)
Copenhagen, Denmark	493	S	497	S	560	S	374	S	510	S	486.8
WA, USA	350	S	193	S	368	S	337	S	260	S	301.6
CA, USA (IPV6)	373	F	247	F	294	F	196	F	234	F	268.8
Mumbai, India	727	S	781	S	896	S	933	S	885	S	844.4
Warsaw, Poland	3992	S	497	S	577	S	528	S	492	S	1217.2
Paris, France	341	S	425	S	352	S	387	S	409	S	382.8
South Africa	888	S	962	S	1375	S	1008	S	1181	S	1082.8
Buenos Aires, Argentina	21698	F	21660	F	21734	F	21718	F	21465	F	21655
Shanghai, China	1515	S	1273	S	1528	S	2148	S	2414	S	1775.6
Amazon, Japan	418	S	518	S	617	S	578	S	518	S	529.8
Amazon-US-East	197	S	226	S	220	S	222	S	187	S	210.4
VA, USA	286	S	570	S	253	S	216	S	324	S	329.8
Tel-Aviv, Israel	667	S	712	S	691	S	543	S	729	S	668.4
Amsterdam, Netherlands	412	S	424	S	496	S	549	S	439	S	464
TX, USA	304	S	368	S	271	S	262	S	336	S	308.2
Brisbane, AU	987	S	949	S	956	S	851	S	759	S	900.4
CO, USA	354	S	358	S	241	S	198	S	367	S	303.6
Frankfurt, Germany	518	S	479	S	419	S	431	S	558	S	481
Montreal, Canada	313	S	316	S	433	S	346	S	519	S	385.4
Hong Kong, China	1150	S	1072	S	815	S	1025	S	1125	S	1037.4
FL, USA	247	S	356	S	298	S	306	S	205	S	282.4
CA, USA	267	S	269	S	252	S	263	S	159	S	242
London, UK	21243	F	21329	F	21288	F	21112	F	21121	F	21218.6

Table A1 Continued...

	Test 1		Test 2		Test 3		Test 4		Test 5		
	Time	state	Time	state	Time	state	Time	state	Time	state	Avg.
	ms	S/F	ms	S/F	ms	S/F	ms	S/F	Ms	S/F	(ms)
NY, USA	267	S	263	S	272	S	512	S	268	S	316.4
MN, USA	240	S	283	S	405	S	180	S	215	S	264.6

Table A2

Complete Test Result of Availability tests for the Google site

S = Success, F = No response found from that location within ~2000 ms, RED = Failed,

YELLOW = Ignoring IPv6 results.

Google site											
	Test 1		Test 2		Test 3		Test 4		Test 5		
	Time	state	Time	state	Time	state	Time	state	Time	state	Avg
	ms	S/F	ms	S/F	ms	S/F	ms	S/F	ms	S/F	(ms)
Copenhagen, Denmark	715	S	805	S	821	S	932	S	1160	S	886.6
WA, USA	824	S	921	S	1018	S	898	S	1164	S	965
CA, USA (IPV6)	796	S	575	S	550	S	560	S	1116	S	719.4
Mumbai, India	1468	S	1474	S	1362	S	1412	S	1547	S	1452.6
Warsaw, Poland	678	S	515	S	851	S	815	S	1262	S	824.2
Paris, France	523	S	508	S	488	S	449	S	946	S	582.8
South Africa	2284	S	2456	S	2392	S	2171	S	2257	S	2312
Buenos Aires, Argentina	1687	S	2826	S	2279	S	2582	S	2037	S	2282.2
Shanghai, China	21103	F	21462	F	21181	F	21137	F	21062	F	21189
Amazon, Japan	978	S	1065	S	900	S	867	S	843	S	930.6
Amazon-US-East	429	S	509	S	324	S	911	S	348	S	504.2
VA, USA	296	S	948	S	364	S	1021	S	316	S	589
Tel-Aviv, Israel	1283	S	1497	S	1476	S	1396	S	1470	S	1424.4
Amsterdam, Netherlands	432	S	481	S	415	S	465	S	471	S	452.8
TX, USA	548	S	551	S	457	S	1002	S	616	S	634.8
Brisbane, AU	1962	S	2197	S	1970	S	2152	S	1728	S	2001.8
CO, USA	576	S	497	S	402	S	487	S	629	S	518.2
Frankfurt, Germany	1096	S	1063	S	1155	S	982	S	1146	S	1088.4
Montreal, Canada	1721	S	1756	S	1853	S	1761	S	1652	S	1748.6
Hong Kong, China	1297	S	10224	S	1247	S	986	S	1190	S	2988.8
FL, USA	452	S	453	S	538	S	517	S	559	S	503.8
CA, USA	451	S	447	S	421	S	676	S	361	S	471.2

APPENDIX B

Table B1

Browser Load test result for www.teamboardodu.com

Action	Time (ms)	Teamboard ODU				Teamboard Avg
		Attempt 1	Attempt 2	Attempt 3	Attempt 4	
Admin						
Project search	316	313	309	393	285	323.2
Team Management	272	269	273	293	311	283.6
Dashboard Load	277	264	296	278	272	277.4
Student						
Access Page	578	770	600	670	551	633.8
Home Page	1030	1065	1285	1327	1021	1145.6
Team Charter	371	386	350	342	378	365.4
Task List Page	1425	1073	607	792	586	896.6
Project Repository	814	812	858	820	801	821
Discussion Page	406	368	380	398	402	390.8

Table B2
Browser Load test result for Google site

Action	Time (ms)		Google site				Google site Avg	
Admin	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5			
Project search	Not Applicable							
Team Management								
Dashboard Load								
Student								
Access Page	Not Applicable							
Home Page	1499	705	521	589	516		766	
Team Charter	2651	2498	2136	1821	1857		2192.6	
Task List Page	3336	3837	1824	1986	1322		2461	
Project Repository	514	910	551	539	693		641.4	
Discussion Page	1217	771	464	490	421		672.6	

APPENDIX C

Table C1

Full result of the processing times of the major functionalities.

Action	Time (ms)					Average
Admin	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5	
Search by Admin	90.146062	55.657025	43.311111	30.838717	23.31538	48.653659
Team Management	0.044179	0.006257	0.006028	0.014843	0.006812	0.0156238
Dashboard Load	0.101954	0.092185	0.096547	0.089439	0.085617	0.0931484
Student						
Access Page	12.197414	9.33114	7.006183	10.338711	7.94404	9.3634976
Project Home Page	82.763073	35.338411	38.137418	28.66159	29.234023	42.826903
Team Charter	24.918626	11.721732	26.989957	12.409877	19.183992	19.0448368
Task List Page	21.999248	24.376592	19.07865	19.776098	23.155636	21.6772448
Project Repository	50.633362	45.250481	112.397228	43.435165	47.118704	59.766988
Discussion Page	57.775186	21.577783	29.647464	23.956736	23.68348	31.3281298

APPENDIX D

Table D1

Usability test: Full comparison test result of Admin Section

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Navigate to the site www.teamboardodu.com	A browser should be opened	1) Type teamboardodu.com in the browser 2) Press Enter key	Positive: teamboardodu.com TeamBoardOdu.com teAMBOaRdODu.cOm Negative: team_boardodu.com	Browser should navigate to the site teamboardodu.com	Browser is navigating to teamboardodu.com as expected	Pass	Negative test data resulted in the error "Server not found"	Teamboard ODU can be navigated by typing address in a browser. Google site requires instructor's invitation.
Application's user interface	Application should be opened	1) Verify the page layout, Spell check, Font and color of the application		1) ODU logo at the top left corner of the screen 2) COLLEGE TITLE should be at the top center of the application page 3) Should have the link to "Sign in with google"	Page layout is simple, no spelling errors.	Pass		Google site has better look and feel.

Table D1 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Access the site teamboardodu.com	Need to have a google mail account to access the application	1) Click on the "sign in with google field" 2) Choose Gmail 3) Enter login info 4) Click "Sign in" button	Positive: salla001@odu.edu Purnesh_11 Negative: N/A	To navigate into the site with login credentials	Action resulted in navigation with google mail login credentials	Pass	1) Associated email account should be opened to access the site. 2) Application is not compatible with non-Gmail accounts	Teamboard ODU can be accessed easily. Google site needs invitation.
Usability of the application	Should have access to the application	1) Login to the application 2) Click on Site map		A Site map link should be available on the home page with the basic overview of all the functionalities	No Site map/Site overview link provided	Fail	A site map is suggested on the Home page to have ease of access of the tool	Google site has Site map for easier navigation within the site.
E-mail account access to the application	Need to have a google mail account to access the application	1) Sign in with a random email that is not associated with the application 2) Click on Enter	Purnesh.konathala@gmail.com	Should not have access to the application	Navigated to the main page, however, a message displayed "Click here to request access to the team"	ok	Does the administrator give access automatically to every student in the course or does each student has to request for it?	Access request process can be confusing in Teamboard ODU. Google site does not have access request option.

Table D1 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Log out/Sign out" functionality	Need to have access to the application	1) Click on Logout link		Should be logged out from the application	Worked as expected	Pass		Teamboard has faster logout process.
Verify the functionality of the link "Dashboard"	Should be logged into the application as an "Administrator"	1) Click on the "Dash Board"	N/A	Should navigate "Search project" page	Working as expected	Pass	Does previous year's projects will still be active in current year?	Visibility of features works better in Teamboard.
Search project" functionality	Should be logged into the application as an "Administrator"	1) Select the Term and Year 2) Click a Search		Should display the list of all existing projects in that particular term and year	Working as expected	Pass		Teamboard displays ease of access in searching for a project. Google site does not exhibit this feature.
Create new project" functionality	Should be logged into the application as an "Administrator"	1) Select the year and term 2) Click on "Create Team"	Positive: 2014, Fall Negative : 2020, summer	A new project should be created with correct term and year specified	Working as expected	Pass		Fully functional project creation is easier in Teamboard ODU.

Table D1 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Functionality of "Assign team members" to a team.	Should have a team created	1) Add all the Students name/UIN/ODU mail to the team 2) Click on "Assign team members"	Positive: Sujatha, Syed, Zikhai Negative: hvygyb*&^%, 2324%^&B B, *****	Should get updated with only positive results	Negative data resulted in error. Positive data works fine	Pass		Assigning team members to a project is simpler in Teamboard ODU than Google site.
Verify if e-mail notification was automatically sent to the after creating a new team.	Should have a team created and members assigned to it.	1) Check the mail inbox associated to this application 2) Verify what details are included in the mail		An automated notification should be sent to the corresponding mail IDs	Working excellent	Pass		Google site does not have automatic notification feature, Teamboard ODU does.
Verify if e-mail notification was automatically sent to the students who were added to a team	Should have a team created and members assigned to it.	1) Verify in the "Activity list" about the mail notifications 2) Alternatively, can ask a student to confirm automatic mail notification		An automated notification should be sent to the user on update event.	Working as expected	Pass		Teamboard ODU has automatic notification feature.

Table D1 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
"Modify Project" functionality.	Should have a project created and members assigned to it.	1) Select a project that was already created 2) Add new information 3) Click on Modify/Update		1) Automated notification should be sent to the corresponding IDs 2) Should include the basic details of modification (Ex: Project requirements modified)	Working as expected	Pass		Teamboard ODU allows modifying project after they have been created. This cannot be done in Google site.
"Delete team" functionality.	Should have a team created and members assigned to it.	1) Select a team that was already created 2) Click on "Delete Team" field		Selected team should get deleted from the system	Working as expected	Pass		Similar functionalities and results in both the application.
Team Charter" functionality.	Need to have "Administrators or access" to the application	1) Upload a Team charter contract		The Team charter file should get uploaded in specified format and should be editable by students	Working as expected	Pass	Are we taking team charter template from Google site or designing our own	Team charter modification is easier in Teamboard ODU. Google site requires a file upload.
Validate the functionality of "Modify Team Charter"	Need to have a "Team Charter" doc present in the dashboard	1) Edit the "Team Charter" doc 2) Upload it again to Team Charter	Positive : A Word or Excel file Negative: A .exe file	The Team charter file should get modified and get uploaded	Working as expected	Pass		Much simpler to edit team charter in Teamboard ODU than Google site.

Table D1 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Deleting Team charter.	Need to have "Administrator access" to the application	1) Select the "Team Charter" doc to be deleted 2) Click on the Delete/Erase tab 3) Confirm deletion		The selected Team Charter doc should be deleted from the application	Since, Team charter file is hardcoded it can be modified. So no need to delete	Pass		Team charter cannot be deleted in Teamboard ODU.
Add Comments " under Team Charter page.	Need to have "Administrator access" to the application	1) Type comments in the assigned area. 2) Click on Enter	Positive: Combination of alphabets, numbers and special symbols Negative: No.	Administrator should be able to give feedback on Team charter	No Feedback or Comments space provided	Fail	In my opinion Team Charter page should have a Comments section, so that administrator can give suggestions for the team	Comments section is not available in Teamboard ODU, but is present in Google site.
Assign Task" functionality.	Need to have "Administrator access" to the application	1) Click on "Project management" 2) Select a project from existing projects 4) Click on "Assign Task" 5) Click on Add task		A new task should be created and notification should be sent to all the students of that particular team	No Task space assigned for administrator	Fail	I think administrator needs to use Assign task space in order to post the guidelines that students need to surely abide	Google site gives liberty to the administrator or to assign/upload task list to users. Teamboard lacks this feature.

Table D1 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Administrator's access to add a new administrator.	Should be logged into the application as an "Administrator"	1) Go to Team Management page 2) Add the mail ID of the person who needs administrative access 3) Click on Enter		Should be able to add another administrator	Administrator was added	Pass		Both the sites have similar functionality.
To Remove the Administrator access of existing administrator.	Should be logged into the application as an "Administrator"	1) Click on Remove administrator 2) Select the person 3) Click on Apply		The person selected should be deleted should get a notification	Selected administrator deleted, but no notification was sent	Pass		Both the sites have similar functionality.
Project Requirements functionality .	Should be logged into the application as an "Administrator"	1) Enter the Dashboard page 2) Click on the Project Management 3) Click on Add files 4) Upload needed file		Project requirements should be uploaded in Read-only format	Project requirements hardcoded on Team Charter page	Pass		Teamboard has better visibility of project requirements unlike Google site.
To verify the Project tracking functionality .	Should be logged into the application as an "Administrator"	1) Select the Term and Year 2) Click a project		1) Should display Project name and project member's contacts 2) Start date & expected End date 3) Status of the project 4) Progress percentage	Exhibiting all required features	Pass		Teamboard ODU exhibits all the required features, whereas, Google site does not

Table D1 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Feedback functionality.	Should be logged into the application as an "Administrator"	1) Select a particular team 2) Verify the team progress		Administrator should be able to give feedback on project's progress and any suggestions at that particular point	No Feedback section found	Fail	Sometimes the administrator needs to interfere if the project is going off track	Google site has the Add comments functionality. Teamboard does not.
Site updates from administrators perspective.	Should be logged into the application as an "Administrator"	1) Create/Modify a project 2) Verify the activity list		Administrator should have complete record of activities done from administrator's side	No Activity record found	Fail	If we have more than one administrators for the site, it is often difficult to track the critical activities done by other administrators.	Google site has the Activity list visible to both user & Administrator. Teamboard ODU Activity list is visible only from the user's perspective of the site.

Table D2

Usability test: Full comparison test result of User Section

Test Objective for Comparison	Prerequisite site	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Ease of navigation	Browser needs to be used	1) Open the browser 2) Type teamboardodu.com 3) Click on enter		Page should navigate to the TeamBoardOdu.com	Working as expected	Pass		User can access TeamBoard Odu site easily by typing in the browser. Google site link needs to be provided by the administrator.
To validate the authentication to the site	Should have a google mail account synced to the site	1) Type the site name in the browser 2) Click on 'Sign in with google 3) Enter username and password 4) Click on Enter		User should have access to the site only with the linked Google site	Working as expected	Pass		Can be accessed with any synced Gmail site, whereas, Google site needs to have ODU mail to access.
GUI of the home page	Site has loaded	1) Sign in to the site 2) Observe the look and feel of the application		Site should be user friendly, easy to navigate between the pages, no spelling mistakes	Site instructions are clear and precise	Pass		Both sites exhibit same kind of structure.
Functionality of Team profile	Need to have a team assigned to the team members	1) Click on the Team profile on the home page 2) Verify the team Bios' update		Should have the accessibility to upload individuals' bios and pictures	Working well as expected	Pass		Can edit bio directly in home page in Teamboard. Google site has a Team profile page.

Table D2 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Functionality of Team Charter	Need to have a team assigned to the team members	1) Click on the Team Charter link on home page 2) Type answers to all the questions 3) Click on Post		Team Charter should get updated with the posted answers	Working well as expected and sending automatic notifications to team members	Pass	There seemed to be question mark symbols(?), in the place of other symbols	User can type answers in Team charter in Teamboard which looks much more appealing than Google site, where user has to upload answers in a file.
Rigidity of Team Charter	Team Charter should be already filled in	1) Click on the Team Charter link on home page 2) Try to modify the answers to some questions 3) Click on Post		Tool should allow the modification of Team charter	Working well as expected and sending automatic notifications to team members	Pass		It is easier to modify in teamboard unlike in Google site.

Table D2 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Functionality of 'Add comments' in Team Charter	Needs to be in the Team charter section	1) Enter the comments in the space provided. 2) Click on enter 3) Verify if comment has been posted		Administrator should be able to add the comments	No space provided to Add comments	Fail	I think administrator should be able to give feedback on the team charter of a team	Google site has provided Add comments section.
functionality GUI of Project repository	Need to have a team assigned to the team members	1) Click on the Project repository link 2) Verify the usability of the page		Should have links to 1) upload files from Computer, google drive, add link 2) Move to folder 3) Manage the files: Delete, Share, Edit 4) Details of the uploaded file	exhibit all the features mentioned in Test steps	Pass		Teamboard ODU has better usability than Google site.

Table D2 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Functionality of 'Add' link in Project repository	Need to have a team assigned to the team members	1) Click on the project repository 2) Click on the Add tab 3) Verify the functionalities of the tab Add		Add should be a drop down which should allow to add file from computer, Google drive, add link	Exhibits all the features but Add is not a drop down link	Pass		Both sites have similar add link functionalities.
Functionality of creating a folder	Need to have a team assigned to the team members	1) Click on the Project Repository 2) Click on New folder tab 3) Type the folder name and click on Create		A new folder should be created with specified name	Working as expected	Pass	Can we integrate Add folder in Move to section as in google docs ?	Repository section has minor differences in look and feel.
Functionality of Move to Tab	Need to have a team assigned to the team members	1) Select a file 2) Click on the move to 3) Select any existing folder		Selected file should move to specified folder	Working as expected	Pass		Both sites have same structure in this aspect
(1) Functionality of Manage files Tab	At least one file should already exist in the repository	1) Select a file 2) Click on Manage Files 3) Should contain drop down to- Edit, Delete, Subscribe to notifications		Selected file(s) should perform the action	No such option is available and Delete is present as an individual option	Fail		Not present in Teamboard ODU.
(2) Functionality of Manage files Tab	At least one file should already exist in the repository	1) Select a file 2) Click on Manage files 3) Click on Delete		Specified file should be deleted	Working as expected	Pass		Both sites have same structure in this aspect.

Table D2 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
(3) Functionality of Manage files Tab	Need to have a team assigned to the team members	1) Click on Manage files 3) Subscribe from changes		Members should receive mail notification regarding any changes or updates	Subscribe to notifications not available	Fail		Google site has this option.
Functionality of View and Download functionalities	At least one file should already exist in the repository	1) Click on view link of a file 2) Verify the appearance of the file 3) Repeat the process with Download link		Should be able to view and download file regardless of the file type	Not able to view without downloading a file	Fail		Google site has option to view the specified file without downloading.
Functionality of Expand arrow of a folder	At least one folder should be present with 2 files in it	1) Click on the Down arrow symbol at the left hand side of the folder 2) Verify the display of the details		Should display number of files upon expanding. Should contain name of the person uploaded, date and time and size and version of the file.	Working as expected	Pass		Both sites have same structure in this aspect.
Validity of 'Version no.'	At least one file should already exist in the repository	1) Click on the Version number of a link 2) Verify the details displayed		Should display Version number, Last edited time and person's name	Working as expected	Pass		Both sites have same structure in this aspect.

Table D2 Continued...

Test Objective for Comparison	Prerequisite	Test Steps	Test Data	Expected Results	Actual Results	Status	Observations	Differences
Functionality of 'Discussion' link	Need to have a team assigned to the team members	1) Click on Discussion tab/link 2) Click on New post 3) Type Title, subject and attachments if required 4) Click on save		A new post should be created showing the person's name, date and time posted and all the details of the post. Comments section should be enabled	Working as expected	Pass		Better usability in Teamboard ODU than in Google site.
Feasibility of accessing older posts in Discussion page	Should have existig posts in the Discussion	1) Click on any existing post's title link 2) Verify the Edit functionality		Should be able to edit the post or enter comments	Working as expected	Pass		Both sites have same structure in this aspect.
Availability of site overview	Should have an active team	1) Search for the Site map 2) Verify the overall view of the site		Upon clicking Site map, user should be able to see the accessible segments	No Site map available which enables easy navigation throughout site	Fail		Google site has this feature which allows usability.
Project deadline view on home page	Should have an active team	1) Click on any existing team 2) Verify the home page for project deadline		Upon clicking home page of a team, should display the no. of days left to complete the project	Home page does not display the no. of days left to submit the project	Fail		Preferable to have it on Teamboard ODU. This is present in the Google site.

VITA

Syed Ehsan

Norfolk, VA-23505. sehsa002@odu.edu

Education

Old Dominion University, Norfolk, Virginia

Master of Science in Engineering Management May 2016

North South University, Dhaka, Bangladesh

Bachelors of Science in Computer Science August 2007

Experience

Perfect Commerce, Newport News, VA

January 2015 – Present

Software Analyst

IMS Health, Dhaka, Bangladesh

Nov 2008 – April 2012

Software Engineer

Therap Services

August 2007 – Nov 2008

Software Engineer

Notable Projects

Controlling electrical equipment through cell phone text messages.

Project involved developing both a hardware and a software to interact with each other in order to control electrical equipment through cell phone text messages.

Tournament Management software

Project involved designing an application that was used to manage a chess tournament.