LAMK
Lahden ammattikorkeakoulu
Lahti University of Applied Sciences

SOFTWARE MAINTAINABILITY IN WEB DEVELOPMENT FOR A PERIODICAL SPORTING EVENT

Case: SELL Games 2020 hosted by Lahti

LAHTI UNIVERSITY OF APPLIED SCIENCES
Bachelor of Business Administration
Degree Program in Business Information Technology
Autumn 2019
Tuan Phan

Abstract

Author(s)	Type of publication	Published
Phan, Tuan	Bachelor’s thesis	Autumn 2019
	Number of pages 33, 0 pages of appendices	

Title of publication

SOFTWARE MAINTAINABILITY IN WEB DEVELOPMENT FOR A PERIODICAL SPORTING EVENT

Case: SELL Games 2020 hosted by Lahti

Name of Degree

Bachelor of Business Administration

Abstract

The primary goal of the thesis is to find out and discuss good solutions for optimizing the maintainability of periodical software. The research may contain discussion around the relevant topic. Some theories will be covered as long as it involved in the solutions.

In this case, there is a team of eight people will aim to finish a first stage application that possesses a scalable and maintainable platform. Consequently, the following team will able to develop smoothly later. The research includes concepts, features of technique which could improve application stabilization. There are some comparisons between applied technology as well to point out to answer for search topics.

This is a real-life project, the scope of the thesis could be a selection of techniques, tools and which reason makes its benefits to the project purpose. In addition, which downfalls after the project and how they would have been copied in order to be used efficiently.

Keywords

Node.js, React.js, Express.js, SCSS, refactoring, maintainability

CONTENTS

1 INTRODUCTION ABOUT THE CASE AND AIM OF RESEARCH 1
1.1 Introduction about the case 1
1.2 Aim of the thesis 1
2 METHODOLOGY AND RESEARCH DESIGN 3
2.1 Research questions 3
2.2 Research method 3
2.2.1 Deductive and Inductive 3
3 MAINTAINABILITY IN WEB DEVELOPMENT 5
3.1 Definition of maintenance and evolution 5
3.2 Differences 5
3.3 Why software development need maintainability? 6
3.3.1 Should or should not invest into maintainability 6
3.3.2 A software longevity 6
3.4 Which factors could affect maintainability? 7
4 REFACTORING 8
4.1 Definition 8
4.2 How to refactor 8
4.2.1 Following five steps of Tripathy and Naik 8
4.2.2 Setting refactoring as a periodic task 8
4.2.3 Testing and refactoring after a new upgrade 8
4.2.4 Testing and refactoring before a new upgrade 9
4.3 When and why to refactor? 9
4.3.1 When 9
4.3.2 Why 9
4.3.3 Adding up comprehensive 10
4.4 What not to do when refactoring? 10
4.5 Automated Testing 10
4.5.1 Turn into failure by refactoring 10
4.5.2 Get fixed by refactoring 11
4.5.3 How to apply refactoring on automated testing 11
5 ESSENTIALS FACTORS TO CONSTRUCT WEB DEVELOPMENT 12
5.1 Methodology for development 12
5.1.1 Different between waterfall(traditional) and agile method 12
5.1.2 Which method are chosen for this project? 12

5.2 Tools....12
5.2.1 Version control....13
5.2.2 Tools for supporting teamwork....13
5.3 Deployment....14
5.4 Frameworks and programming languages....15
5.4.1 Node.js....15
5.4.2 React.js....17
5.4.3 MongoDB....18
6 WHAT HAVE DONE....19
6.1 Front-end....19
6.2 Back-end and Security....19
6.3 Using patterns....19
7 WHAT SHOULD IMPROVE....20
7.1 Front-end....20
7.2 Back-end and Security....20
7.3 Working in a team....21
7.3.1 Tracking development progress....21
7.3.2 Attending....21
7.3.3 Dividing tasks....21
8 RESULTS....23
8.1 Question 1....23
8.2 Question 2....23
8.2.1 Question 3....23
9 CONCLUSIONS....24
9.1 General....24
9.2 Failure....24
9.3 Limitation....25
9.4 Backlogs....25
10 LIST OF REFERENCES....26
APPENDICES....29

1 INTRODUCTION ABOUT THE CASE AND AIM OF RESEARCH

1.1 Introduction about the case

In sports events, nowadays a website can handle a lot of tasks from SEO, manipulation of customer data, handling customer support. Building a functional website took a certain amount of resources such as time, money, human. Etc. Additionally, the sports event often hosts in an interval of times which means the application will be needs for a period but in a certain loop of time.

Moreover, technology develops dramatically which may make some software out of date quickly and then cannot/hard to expand anymore. In other words, companies/organizations ought to omit them and reinvest to build new ones. Thus, to avoid/lower the investment fee, companies/organizations tend to look for a way to reuse their application.

In this case, within approximately 3 months, there is a team that is 8 students building the platform which could be called the first stage of SELL games 2020. It will be subsequently developing by the next developer group. The requirement is as same as one of the primary goals which are creating a maintainable platform. Nevertheless, a scalable platform. Hence, the next team will have rooms to grow the software.

Moreover, there is a loop in website usage; in the other words, the life cycle of this website should be taken into account, thus it will bring quite remarkable benefits in long term for a sports event in general and SELL games to be specific. Therefore, this thesis is going to include discussions about and around how to make how to optimize the maintainability in any factors of software development.

1.2 Aim of the thesis

The ideas of the thesis come from SELL games sports event where a lot of countries send their candidates in order to compete in a chosen country. The event will be periodically hosted in turn. Hence, there is obviously a need for an information system in order to control the flow of news and also the ability to display the result of the event. That system may contain at least a social network, website, mobile app, etc. Thus, the website traffic will be regular high in a period of operational time, and then it finished its goal. It may still stay there but does not need to handle that much connection at that time. Therefore, choosing which technology, the methodology is suitable for this purpose is the main target and how they are used in order to make those achievements. This topic is discussed its efficiency to the project. In addition, the matters around the website will be mainly discussed in the thesis.

Moreover, the aim of the thesis is not only to seek the technology which has high performance in rush hour but also a solution to make a better life cycle for software development. As this event will be hosted in the same interval of time. It will be wasted if the website needs to rebuild from scratch every time. Hence, searching for solutions for a better lifecycle application is another primary purpose as well.

Besides, as indicated previously, stability plays a certain role in this project as well. It will be another primary objective.

In brief, there are discussions related to these primes:

- Maintainability
- Lifecycle
- Essential factors for building this web application.

Last but not least, there is a chapter to cover analytics about which goals have been achieved and which can be improved all matters after the project development stage.

2 METHODOLOGY AND RESEARCH DESIGN

In this section, research questions are introduced. Additionally, the research method is chosen with an explanation.

2.1 Research questions

One of the solutions to manage my statements that answering the question base on book ref, others perhaps are solved by snippets code which supports for better declaration clarification. The topic is handled as answering related questions by debates revolve around:

-" What is maintainability, lifecycle in software development why it is important in this case?"

-" What exactly can benefit to have better maintainability and lifecycle in development, especially in SELL games 2020?"

-" Which tool, technology, implementation, etc. can be solutions for all previous question?"

2.2 Research method

The research method will be chosen after the introduction and comparison between deductive and inductive methods.

2.2.1 Deductive and Inductive

According to Miessler (2019), both deduction and induction method is meant to persuade people about statements of a subject. However, there is a significantly different between deductive and inductive methods. The deductive method uses theories, test them by applying into a case and then watch and evaluate the result in order to prove or answer a question. The inductive method is contrary, it uses evidence, results to consolidate theories or statements.

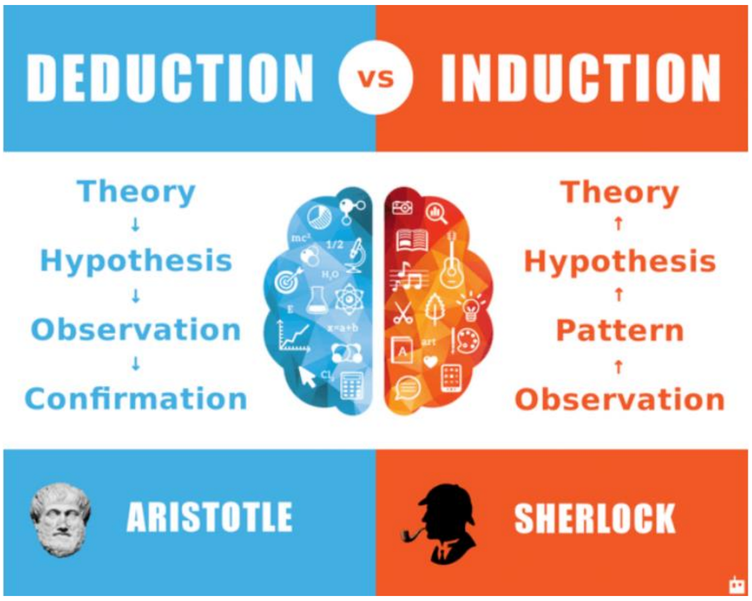


Figure 1 The Difference Between Deductive and Inductive Reasoning (Miessler 2019)

This thesis is made by collecting data in order to increase maintainability in a real-life application. Hence, it uses statements and theories in order to apply to a real product and then gets tested. After that, a result is observed how efficient it is. Therefore, the deductive method is obviously suitable as a research method for this thesis.

3 MAINTAINABILITY IN WEB DEVELOPMENT

3.1 Definition of maintenance and evolution

Before answering the question “What are maintainability and its importance in software development”. There is a need to clarify the diverse maintenance and evolution.

According to Tripathy and Naik (2015, 2), maintainability in software increases the chance which it could avoid failures from the delivery stage. Additionally, boosting maintainability means that it could a debugging, adding comprehensive or improving the quality of the stage.

Software maintenance plays an essential role to make the working process smoothly and level down interruptions chances. Meanwhile, software evolution is a stage that will be implemented after a completely built application due to the need for an upgrade.

Additionally, Stephens (2015, 9) stated that as long as the application is still being used, the users will find out the issues. When the bug is found, it also needs to be fixed but this process could lead to other issues and the maintenance stage will go repeatedly like this until everything gets fixed.

As seen in the solid role of maintenance, it can take place anytime during before or after the new feature implementation or even when the users report issues.

3.2 Differences

Maintenance and evolution are quite similar to each other, the meaning could be unclear because they all came up to a stage that an engineer or maintainer need to handle a couple of tasks on a completed built or on-going building software. However, they are different because of their initialization time and usage purpose. On one thing, maintenance, the process is initialized periodically and compulsory. It solves problems like bugs, crash, error on the software.

As Naik and Tripathy (2015, 2) have stated meanwhile software evolution puts efforts in order to improve functionalities, external behaviors or improving some features for the soft-ware. Maintenance tasks focus on how to make the software stable, smooth performance without trying to adjust any its functionalities or design. In addition, this makes no changes to its design, neither big changes to the architecture. For instance, refactoring could be considered as a part of maintenance and it focuses on rearranging files or folders, removing duplicate code but not changing the whole client-side or database system. In this topic, there will be a section to discuss more refactoring later.

This one only occurs when and where there is a need to enhance or add up more functions. It does a couple of adjustments on the software such as building new functions or make an application able to works across devices which transforms it into the better version. Within the software evolution, stage, not only the system or application improve but also the user's capability lifts up over thoroughly during the process.

3.3 Why software development need maintainability?

In software development generally, there may be needs to reuse or even extend a former application, especially it happens quite a lot in sporting events that are hosted in interval time. With this type of project, it is developed and use for a period of time and then takes the rest until next event which could be some months or years; and this process happens repeatedly over time. Hence, it is a must that the companies ought to reinvest again on the next cycles. However, it might give rise to an extraordinary problem. Those problems could be the website is outdated, the applied technology is no longer getting supported or just simply the infrastructure is too complicated to do whether software maintenance or evolution. To do so, there are a lot of things that are applied such as constructing a good data structure, writing a good format code, doing concise configurations in the file system, following variables naming rules, etc. should be taken into account carefully (Stephens 2015, 241.)

3.3.1 Should or should not invest into maintainability

After listing these concepts of maintains. It may lead to a questionnaire problem that is “What is the point of spending quite much time on boosting the value of life cycle.?”. If that amount of time is used for the implementation process, will it be a better result? Based on the 5s method, for maximizing maintainability, the team/organization who handles the project should invest time to build a collection of habits which is named 5s. The 5s contains: 5 pillars named prospectively “sort(seiri), set in order(seition), shine(seiso), standardized or visual control (seiketsu) and sustain (shitsuke)”. According to the 5s author, it is stated that although it takes time for constructing 5s habits, it is an obvious good investment for long term improvement. (Santos et al. 2006, 148.)

3.3.2 A software longevity

According to Stephens (2015, 242), a life expectancy of software might be surprised, it could be years or even more than that. There is some example of the code can live for decades such as Y2K. In that case, the application was created in the 1950s and after 2000 that application is still in used. In fact, a typical business does not prefer to rebuild its application but keep using it, so the Y2k does. As a consequence, they have to ask help from developers who have experiences with BASIC and COBOL even the retired ones all over the world. Since, some projects might last longer than expectations like Y2K one, in that case, it might still be fixed by event retirement human resources. However, the main thing is that without good maintainability, fixing the decades year old would be remarkable trouble. In this project, SELL Games 2020, it is known that it will last at least one year but this event takes place yearly. Hence, the next host event could keep using it if they would like to. As a software with good maintainability, they do not need to rebuild anything but keeps developing from this one and makes some modifications upon their needs.

3.4 Which factors could affect maintainability?

One of the problems that software sustainability is cut down over the development. This method supports to maintain the work environment with a little try by building a collection of habits for workers. At first, before implementation anything, the working environment needs to be arranged systematically. In this case, soft-ware development, need to organize file and folder orderly which can help others understand the system or structure with a small effort. In addition, 5s pillars mentioned previously which is one of the methods could focus on boosting the working environment. (Santos et al. 2006.)

Refactoring is a process to make some changes to the structure. In addition, it eases development process in the future and level down the cost and resources without losing functions or features. It is also called restructuring.

4 REFACTORING

4.1 Definition

Refactoring is considered as modifying internally software factors without changing how its external behaviour. The purposes of doing this stage are putting an effort to decrease the ratio of software crashing, keep the code up to date with the latest technology, raise production smoothing and increase software extensively. (Tripathy & Naik 2015, 13.)

4.2 How to refactor

4.2.1 Following five steps of Tripathy and Naik

According to Tripathy and Naik (2015, 14), there are 5 steps to refactor. “(I) add a class, method, or attribute; (II) rename a class, method, or attribute; (III) move an attribute or method up or down the hierarchy; (iv) remove a class, method, or attribute; and (v) extract chunks of code into separate methods. ”. Refactoring aims to changes the code such as moving code to a better place, adjust the name of the element or remove the duplicate one. The main point is the code should be cleaner after refactoring unless it is not refactored yet.

4.2.2 Setting refactoring as a periodic task

As stated by Martin Fowler (2019), refactoring is not some special task that needs to be planned but regular. Whenever a new feature has to be implemented, it is recommended to check up the codebase whether this implementation can be initialized immediately or not. Therefore, unless it will need to be refactored before the implementation. By doing so, it will take one more step, but it is faster than trying to upgrade or enhance the software with the messy codebase. Moreover, if this step keeps being considered before new implementations, it seems to guarantee software development in an effortless way.

4.2.3 Testing and refactoring after a new upgrade

In addition, according to Fowler (2019), after adding new features. The code is tested to make sure it works properly. Since then, there is a recommendation that the new code needs to be taken a look and then make it into a good shape. This step aims to not only have a good effect now but later on. Assuming that after some weeks of successful production, the application needs to be enhanced; within a good and comprehensive codebase, the programmers will effortlessly start doing their job right away without spending time on figuring out how it works.

4.2.4 Testing and refactoring before a new upgrade

Furthermore, before building new features, Martin Fowler states that the new features or part of it might be already created and located somewhere in the software. Thence, it is suggested to look for the existing solution. Then, checking if it is in a good location to use or not. Unless, moving it into an easy to use place such as a class, a constructor, etc. Additionally, refactoring it if there is not comprehensive enough. Hence, later on, there will no struggle to use it. (Hughes 2019, 43.)

4.3 When and why to refactor?

4.3.1 When

There will be a time all the features are successfully implemented but as a software, it always needs to get upgraded. Therefore, refactoring could affect this process. First, you need to add some code inside to build new features. As time passing by, the more features it has, the more complexity it is. In addition, the new features are usually built over the top of another feature. Moreover, refactoring is the intensification of software extensity. It restructures the code, makes it more comprehensive. Hence, it supports for maintenance stage and also when there is a need for new functions, it reduces the effort for this stage. (Fowler 2019, 62.)

4.3.2 Why

At first, according to Fowler (2019, 58) refactoring does not mean that it makes this state invulnerable to any matters at all. However, it plays a solid role in software development. Additionally, it is a tool that is helping a lot to development state.

Moreover, if the developers keep doing software evolutions in order to raise quality of external, the application would go so far from the beginning design. This might take the comprehensive out of the application, make the documentation out of trustworthy and wasting a maintenance cost. (Naik & Tripathy 2015, 255.)

And this chance for software evolutions happens is high because as long as there are users using it, it needs to evolve to satisfy it’s the needs of users. Therefore, in this subchapter, the discussions get around these matters.

4.3.3 Adding up comprehensive

Refactoring is helpful to add up comprehensive since it is able to reduce complexity after or before new implementations. By the time, the code looks easier to understand for developers. (Tripathy & Naik 2015, 13.) As usual, demanding a computer processor what to do is the main purpose of code. Hence, when writing lines of code, a developer primarily put an effort to make the computer to do some specific tasks. Nevertheless, not only the machine needs to understand what the developer wrote in but a future developer who will read this code as well. It means a computer might only take some moments to compile the code, meanwhile, it may be a week for a human who tries to understand before making a change. However, within understanding the written code, it might only take one hour. (Fowler 2019, 59.)

4.4 What not to do when refactoring?

There is no doubt about how valuable refactoring is in software development. Nevertheless, no matter how good the tool is, how it is used will matter. Tripathy and Naik (2015,15) stated that there is no functionality that should be implemented during the refactoring stage. It makes the process harder. Additionally, as explained, everything has passed the testing stage need to pass the refactoring stage.

However, it does not mean that refactoring will slow down production performance. A new implementation just needs to be refactored after the testing stage. As stated before, the larger the update is, the harder the refactor stage is. Hence, the good time to put the code in the refactoring stage is after small adjustments or production.

Don’t mix refactoring and direct the development of new features. Try to separate these processes at least within the confines of individual commits. The code needs to be refactored peri-odically unless the structure will be impaired due to regular maintenance. Moreover, in an-other perspective, refactoring may evolve the internal system and structure. (Fowler 2019, 67.)

4.5 Automated Testing

4.5.1 Turn into failure by refactoring

There is a relation between refactoring and automated testing that after refactoring stage the automation code might be a failure. This occurs because of the modification in the code which makes by the refactoring stage. As mentioned previously, refactoring might remove or restructure the code. That is how it turns the automated testing into failure. Hence, automated testing is pretty easily turning into failure by a refactoring stage. (Softwaretestinghelp 2019.)

4.5.2 Get fixed by refactoring

In addition, there is another relation between refactoring and automated testing. As automated testing could be easily affected by a refactoring stage, it could get affected in reverse. According to Zhan (2009), automated testing is hard to maintain but as long as it proves it remarkable benefits in a project. Therefore, it needs to get refactoring to improve maintainable and readable during the development.

4.5.3 How to apply refactoring on automated testing

As stated by Merrill (2019), besides features like adding maintainability, comprehensive refactoring also has great performance on testing. There is a feature in refactoring called restructuring and this can make a solution smaller and easier to understand and be used. This step repeatedly occurs until that solution can be obviously seen exactly its function. In the automation testing itself, this method can be applied as well. It makes a test into smaller pieces and keeps splitting those smaller pieces into smaller ones until they are maintainable, understandable enough.

5 ESSENTIALS FACTORS TO CONSTRUCT WEB DEVELOPMENT

5.1 Methodology for development

5.1.1 Different between waterfall(traditional) and agile method

In agile methods, the user plays a solid role regarding almost every stage during development such as requirements, design. In addition, they might contribute partly to the debugging stage as they mainly use or check up the product in the reviewing stage after each cycle. After that, they will notify the errors of the developing team. (Turker et al. 2011, 28.)

In agile methods, the main different compare to the traditional way is the same steps in short cycles will occur repeatedly. One of those steps is refactoring which adds up efficiently comprehensive and maintainability into the project. (Marmol 2015.) However, this factor is not considered as it needs.

On the other side, the waterfall method has certain stages that begin with a specific stage and only go to the next stage when followed the previous one gets done. In addition, there must be solid requirements beforehand. In the other words, the aim of the project is surely confirmed before it starts. Moreover, it should not be occurred by majority adjusts. Additionally, it would be ideal if the team used to have experience with this kind of project previously. (Stepehens 2015, 270.)

There are 2 primary development methodology in software development named Agile and Waterfall method. They are suitable separately to different projects. While waterfall method has fixed design at the beginning and needs a highly specific requirement; the agile methods requires frequently interactions, flexible planning within short cycle. (Casteren 2017.)

5.1.2 Which method are chosen for this project?

In this case, the customer knows what they want as a result which is an upgradeable platform for the next team. In addition, hence there is not much change during the development as the customer points out that they want an upgradeable and maintainable platform as a result. This makes the waterfall method suitable for the project.

5.2 Tools

How much the tools support the case needs to consider carefully. For example, in the development process, the company or organization prefers to keep the product in private until a certain time and the team is at least 8 people. The tools must support at least these requirements. To clarify which tools should be chosen in this case, the next paragraphs make a comparison between those tools and figure out which one of them will be more suitable for the team in this project.

5.2.1 Version control

Another essential factor in development stage: Version control tool. As there would be a lot of change, modifications, upload, revers happen during development process, the version control toll is a must to have which has ability to manage this process. During the working process, when a developer uses this tool to upload code, it is able to record the time, the content and also the one who just make that action which is called a commit. In case something goes wrong and the application is broken, this version control tool is able to reserve to the previous working version without any problem. Moreover, confliction might happen quite a lot while working in the team when 2 or more try to upload an initial part with different version. At that time, it cannot be merged. Since, the tool provides the function named branch which help a developer can start an own experiment without touching to the main code-based area. If the result looks good, later it might be merged back into the main code. There are many variety version control tools around the market and some of them are open sources. However, regardless they are different, the different between mentioned features above is not much. (Tucker et al. 2011, 72.)

In this project, the chosen one is a popular service named Github. Both free and pro version available on this tool. On free one, Github limits the maximum amount of people in as 5 within free tier. In this project, it required space for at least 8 people who has accessed to the application. On the pro one, Github provides unlimited members in a team as default account. However, the student can get the paid one via student email according to Github (2019). To be specific, Github has its own UI application on 2 most popular Windows and MacOS named “Github desktop”.

5.2.2 Tools for supporting teamwork

Trello is an application which supports for project management. At first, the Trello has been developed after agile methodology information radiator boards.

According to Nocarrier (2008), the information radiator is a big board which contains relevance team information such as design, in progress task, plans, etc. This information keeps updating ceaselessly and also is located where the whole team can observe. Here is an example:

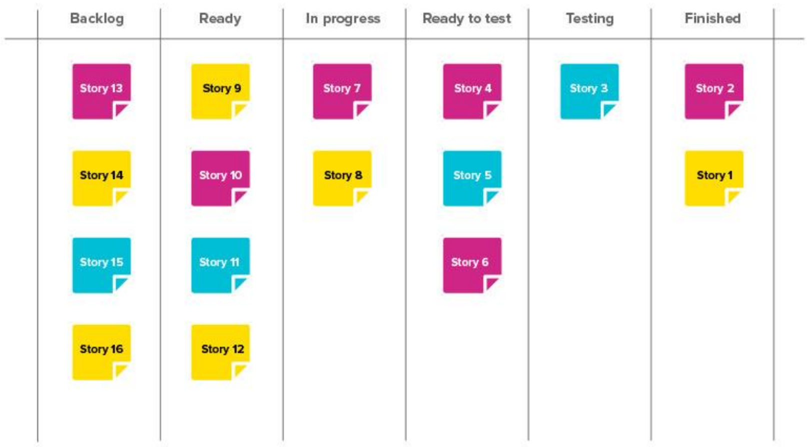


Figure 2 How Agile transparency reduces project risk (Donaldson 2018)

There are stories on the board which will be moved from the left at the beginning to the right side depending on its stage until it has been done. It is where the Trello model from, later on, Trello is strongly influenced by lean methods introduced by Toyota that being used to boost productivity and flexibility in their workflow. It basically provides a multifunction blank which can be edited or move around the column or setting a role for a member in the group. In another word, this function looks like an advance to-do app. Moreover, it is a cloud-based platform that can be saved on the Trello server and synchronously changes depending on each interaction. In the project, it helps the team keep on tracks by setting goals and mark the things done. It is able for everyone in the team can take the preferred tasks. (Johnson 2017.)

In addition, the agile method requires high interaction between members which make a communication platform is a must. In this case, Slack is chosen for that role. Slack is a cloud-based platform that has a function like a chat room for a group of people. Moreover, it does not only handle conversations but also able to share the files, documents or make a group call. A user can join different workspace in order to connect different subjects or information. The comparative advance of this application is that it goes over multi-platforms, hence every member can interact comfortably regardless of whichever device they use. (Standuply 2019.)

5.3 Deployment

After the finished the development stage, it is about the time to deploy the application and show it to the world. In some cases, if the client totally started a new project which means they have not hosting, domain, etc before and it is as same as this case for SELL games

2020. According to Weikel (2019), in this case, at first, a domain is a must as a domain like a home address that customers know where to visit. There are some popular service domain providers like Namecheap, GoDaddy, etc. After that, it needs a hosting server where the files are located and configuration steps need to be done before the application can be launched. In fact, the LAMK IT department handled this section.

5.4 Frameworks and programming languages

The SELL games 2020 stage one has been built with React.js, SCSS, Node.js, Express.js, and MongoDB. Hence, there is a discussion in this chapter about why these techniques are chosen and there also might be some comparison during the analysis.

5.4.1 Node.js

Node.js is a JavaScript and It is run by Chrome's V8 engine which is Google open source, written in C++. It can be run alone or as an embedded application. It can run on different OSes such as Windows, macOS and Linux.

Normally, JavaScript is used for handling the client-side through manipulate Document Object Model (DOM). Meanwhile, the V8 provides this feature as well, plus it allows JavaScript to be able to access objects and functions in a C++ application. (V8 2019.)

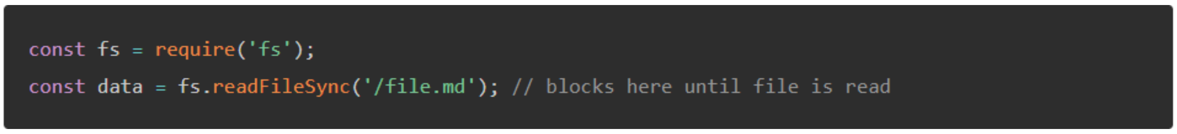
As mentioned before, Node.js was born based on the V8 engine. These are remarkable changes in the usage of JavaScript. At first, only handling for front-end of websites, which mainly make the static web application to a dynamic one. However, within the release of Node.js, it becomes a performance back-end tool as well.

Node.js is running in contrast to a common model. In common one, most of them are regularly lack of performance and has barrier usage. On the other hand, within Node.js, there are certain features for handling these problems. First, in contemporary Node.js has the ability to manage many connections. For every single connection, the call back is executed. However, in case, there is no job, it will go hibernation mode. In addition, barely any functions in Node goes straight to perform I/O but call back function. Hence, there is no lock, the process is able to gradually handle those requests, responses on the website. (Nodejs 2019.)

There is nothing miracle but the Node.js features itself as asynchronous programming, call-back pattern. Without Node.js, JavaScript can still be standalone, but it has to face the blocking issues. Since there are a lot of jobs running on the website, a blocking operation will force JavaScript to as long as the other task still in process. Meanwhile, within a nonblock feature, Node.js only slows down because CPU power is not fast enough to handle recent tasks.

Here is a comparison example between, synchronous programming and asynchronous programming.

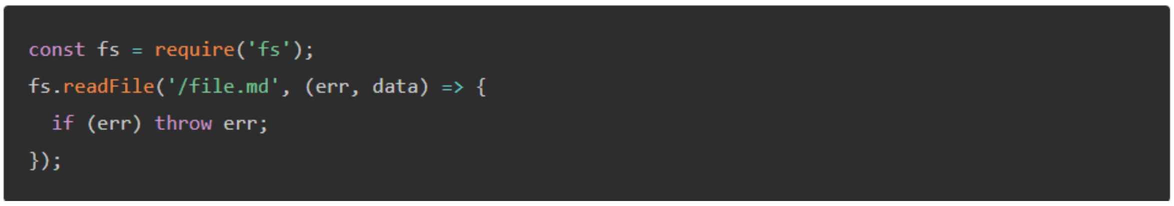
Synchronous:



```
const fs = require('fs');
const data = fs.readFileSync('/file.md'); // blocks here until file is read
```

Figure 3 Overview of Blocking vs Non-Blocking (Nodejs 2019)

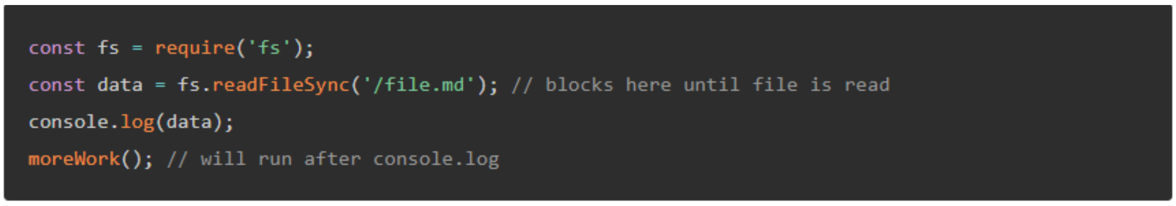
Asynchronous:



```
const fs = require('fs');
fs.readFile('/file.md', (err, data) => {
  if (err) throw err;
});
```

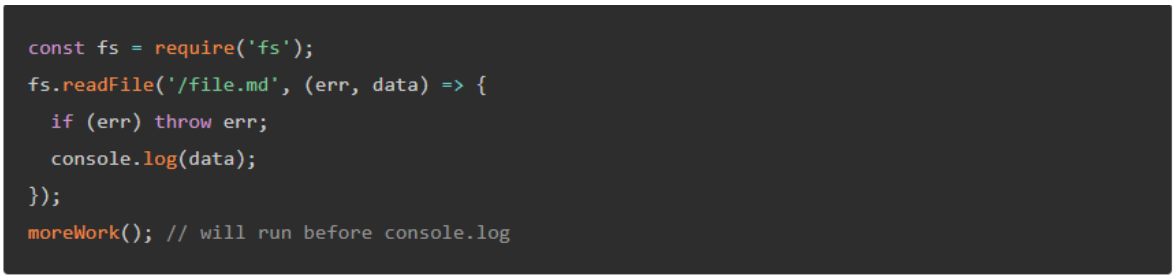
Figure 4 Overview of Blocking vs Non-Blocking (Nodejs 2019)

As be viewed, the Node.js, the asynchronous one looks a bit longer with a call-back function. However, when there is an error, the call back function will just show an error. Mean-while, the synchronous one will block anything other things until that command will be executed or the operation crash.



```
const fs = require('fs');
const data = fs.readFileSync('/file.md'); // blocks here until file is read
console.log(data);
moreWork(); // will run after console.log
```

Figure 5 Overview of Blocking vs Non-Blocking (Nodejs 2019)



```
const fs = require('fs');
fs.readFile('/file.md', (err, data) => {
  if (err) throw err;
  console.log(data);
});
moreWork(); // will run before console.log
```

Figure 6 Overview of Blocking vs Non-Blocking (Nodejs 2019)

In this SELL games 2020, at least more than one thousand participants come across over 10 countries to compete with each other. The web application will be served for multiple purposes, reviewing result for competitors, looking up for schedule for the audience as well, etc. Therefore, the number of sequential connections could be thousand or more. But this number will not stay as same as all over the time, sometimes it could be too high, others could be low. Within a common solution, the website could be crash if the traffic is too high. The solution is paid more for the server, but it will be a waste when the traffic is in a low time.

This problem is able to be solved by Node.js, whether the performance or crashing opportunity. Referring to my previous argument, Node.js has the advantage to handle a lot of connections at the same time based on its asynchronous features, the process is no longer blocked. Moreover, when there is an upcoming connection, it will work, when else it goes to sleep, it is an advantage if the deployment is running on AWS which you only paid how much you use. Therefore, within this case, the Node.js a highly optimal solution.

Express.js can create a local web server in a snippets 3 lines of code.

5.4.2 React.js

React.js, a framework is created by Facebook, the most popular social network recently. In 2013, Facebook made it an open-source under license to the developer community. Although it is open-source, it is still mostly developed by Facebook. (Hughes 2017.)

Within React.js, a developer can create websites effortlessly. The UI will be rendered immediately after every single change of data. In addition, the component has highly reusability and flexibility, which can be manipulated by its own state, plus the components are handled by JavaScript language as well. Hence, it is an advantage for a developer using one language to pass data or debug in both front-end, back-end. (Reactjs 2019.)

In this project, the maintainability has a priority above other requirements. And the React.js' components feature just to satisfy it. Moreover, with this pick, the project could have both back and front end in JavaScript which could be used later to generate some patterns design.

There is a usage of a mobile app as well and the reaction has a mobile application framework named React Native. This react has ability to rendering the UI for mobile devices such as Android, iOS platforms via react-native. (React Native 2019.)

5.4.3 MongoDB

Almost every project needs a database, in this project, MongoDB is chosen. It is an open-source database that is free to use for everyone. MongoDB stores record as JSON format, which is pretty much familiar with developers, especially with developers in this project. To be more specific, as mentioned previously, every other stack in this project are JavaScript framework or library and JSON stand for JavaScript Object Notation. In other words, the database is using the JSON format which is effortless for JavaScript developers to get used to. In addition, MongoDB is one of the most efficiency NoSQL recently which contains collections and documents and greatly flexible, an application only needs to focus on its task and make as many as records it needs without worrying about the dynamic of the database. (Intellipaat 2019.)

In this case, IT department hosted the database for SELL games 2020. However, there might still another choice as well. Cloud software is getting popular, so MongoDB does. Almost developers who use MongoDB know about MongoDB Atlas. This cloud-based service is originally from MongoDB which handles essentials things such as security, hosting patching, etc. However, because of that previously mentioned features, MongoDB Atlas should be used in some case such as the application runs offline, the application does not allow 3rd party software, etc. (Gray 2018.)

There is no doubt about useful of MongoDB with the role of the database. However, this project is worked under a team of 8 people, it will shorten and synchronize working time between the team when using Atlas rather than just MongoDB. If just MongoDB is in usage, there would be a remarkable waste of time in the development process, especially when passing the project for another team. Moreover, later on, the database will be always on the cloud service, there are no worries about preserving the files, upgrading the version, etc but just focus on the coding and design. Last but not least, in fact, the application is available all the time but during 3 days of the event, the users will go sky-high. MongoDB Atlas totally can handle this because according to Gray (2018), the service offers quite many options such as shared clusters, dedicated development clusters and dedicated production clusters. Each choice has a different performance which can be used strategically. While in off-event days the application just can choose a standard package because there is not much access at that time. However, in 3 days of events, it can switch to the much prettier package which provides enough performance for thousands of users. In this way, it can both be saving waste and doing great performance necessarily.

6 WHAT HAVE DONE

6.1 Front-end

The project has been used React.js, one of the most popular frameworks which are highly supported by the developer’s community. This framework helps the maintenance task easier later on. Thus, there is a bit less worry about maintainability. Although, by time passing, a part of software might be deprecated React.js community will provide a better version within more powerful function. Moreover, within the group of developer for example 8 people in this project, the React has technique named components which can be used for rendering the UI and this just makes the workflow smoother since it allows as many as possible UI elements can be built at the same time. However, this technique requires a good plan for infrastructure at the beginning before it can be expanded largely.

Every front-end part has to have at least 2 things, one is responsible for building the UI elements, another one handle for the stylesheet. It is just like the bone and skin on the body human. In this case, they are respectively React.js and SCSS. SCSS has the as same purpose as React.js since it can be written every part simultaneously. Once again, within a good infrastructure, it is effortlessly to control modify for later on. In addition, it is completely compatible to CSS and also its syntax is indeed comprehensive.

6.2 Back-end and Security

In this project, the aim is building an easily maintainable and expandable platform for the following group developers. It is essential which can host both news and the administration could control the content system. Unfortunately, the backend is not invested much time and human resources into but the application still able to do certain things. It has connected to MongoDB which tested by some sample models and it is ready for future implementation. The registration form is implemented, and it can add up a user or more into the database.

6.3 Using patterns

According to Tucker (2015, 29), both development and deployment stages get boosted when patterns are being used while building infrastructure. Following the patterns, tasks are easier to be separated into each member. In fact, this part is planned carefully at the beginning and it gets tested after every implementation. Equally important, it also uses a certain pattern naming to create architecture.

7 WHAT SHOULD IMPROVE

7.1 Front-end

Nowadays, human does not only use a personal computer to access the internet but quite many different devices such as laptop, phone, tablet and so on. Therefore, a good structure and catchy design are not enough but eventually, there is a need for a multiple across devices product which is technically called responsive. Unfortunately, we do have enough people who have this skill. As consequence, the website might look good on the computer screen but in other devices, the components messed around each other and in some cases, the picture in the computer screen is too big for the phone screen or some UI elements are dis-appeared because meanwhile, others were going over them.

This issue did not happen at the beginning since there is one person can handle this. Until, the rest, 7 to be exact, spent time to build the UI concurrent. At this time, there were too many jobs for a person who is responsible for adding responsive to the software.

This problem is underestimated, a more correct way to deal with this probably is spending more human resource to handle this field. In this case, there should have been time for training at least half of the team to get this skill. Additionally, there should not be more people building UI than doing responsive UI, it would be nice if these people could handle both tasks at the same time but it might be a risk that they may be overwhelmed. Last but not least, the content should have been separated completely from the code. This helps the team more concentrates on the layout.

7.2 Back-end and Security

There is an important thing that the content needs to be separated from the back-end code. The project has not occurred this issue during development. This is one essentials factor which makes the project cannot become an official usage version. In fact, the code should be split out from the website content which provides freedom for the editor to adjust the in-formation easily and the developers keep up their tasks without disturbing each other. In addition, the separated content area also might help to improve application security. Be-sides, during the developing process, as the data is able to save into MongoDB, it is not a healthy way for the application life cycle. The other developer cannot use it concurrently, even so, they might share the database between each other, there is a high chance that the MongoDB database conflict itself. Finally, the password was not hashed, it is stored as same as how input was typed from the users. After fixing these things, the application can improve a certain level of security.

7.3 Working in a team

7.3.1 Tracking development progress

There might be a chance a or some members of the team are ill or take a vacation. Hence, time resources for each task or even a whole project should be extended more than expectations. (Stephens 2015, 45.) In fact, there are some members take days off at that time there is a holiday week between the project but it seems like not everyone willing to sacrifice their days-off for the project. If the team dedicated more to the project, maybe some issues should have been fixed in time.

Stephens also stated that in order to ensure the process is following the plan, there is a must to check every task’s process. If there is one of them is not as planned, the action should occur. Additionally, this checking progress task should not be postponed until the end of the project but it should occur after the amount of time. Then if something goes wrong, there would be enough time to make a decision (2015, 46). In fact, not all the members gave notices beforehand about their absences in weekly meetings. Hence, there should have been more interactions regards to tracking their task process after half of the task time. Then, it would improve the result or at least the team would have time to handle that issue. Therefore, a tracking time process is a must tool during development.

7.3.2 Attending

One of the failures in this project is that just a few members are willing to gather every week in a no-mandatory meeting. In addition, there is at least 6 hours every normal week in typical lessons in which the team could gain pretty much helpful assistance from a lecture. If all team members could attend frequently, a lot of problems could be solved because of that support.

7.3.3 Dividing tasks

The team has created some channels to work together. The Trello is chosen for diving the task in the group and keeping track of the process and the plan. Furthermore, the Slack application is used in order to keep contact with each member. Inside it, there are divided into 2 topics, one for the design team and other for developing a team which makes the conversation separated and comprehensive. It looks nice at first, but later on this cause a trouble which is difficulty in diving tasks for members. In design team, after finishing their job, they are waiting or keep creating more UI but there are not enough people to build the layouts based on this UI. This last for the whole project, it makes the important tasks have not enough time to focus on.

8 RESULTS

This section answers the research questions introduced in chapter 2. Those questions are:

-" What is maintainability, lifecycle in software development why it is important in this case?"

-" What exactly can benefit to have better maintainability and lifecycle in development, especially in SELL games 2020?"

-" Which tool, technology, implementation, etc. can be solutions for all previous question?"

8.1 Question 1

As discussed in chapter 3, maintainability and lifecycle have a solid role in this case. The event is hosted in the periodical time and event organizers usually remake a similar function application all the time. Therefore, good maintenance could solve this issue and save quite a lot of resources for organizers.

8.2 Question 2

In SELL games 2020, the development is divided into 2 stages, the first stage lasts until May in 2019 and then the second stage will be continuing. A maintenance application could provide highly upgradable which means the next team could jump into the project and develop the application in an effortless way.

8.2.1 Question 3

Version control, refactoring and full stacks JavaScript is chosen as main solutions for building the application. Their features are not unique in the market as there are a lot of similar function ones. Besides that, refactoring is a not tools or technology but a way to enhance a recent product to a better maintainability one. However, their performance is suitable and good enough for the project as boosting maintainability.

9 CONCLUSIONS

This chapter discusses which achievements have been reached, how the final product in general. Besides that, which important failure in the development process. Also arguing about limitations in the thesis. Finally, further studies might have done deeper in the future.

9.1 General

The main purpose is boosting maintainability into the project and able to pass it for the next developer team. There have been steps to prepare carefully from the stage of choosing methods, tools, and techniques to do and plan the different refactoring tasks during the development process.

As a consequence, a full-stack JavaScript application is created. In the beginning, the infrastructure is carefully built and afterward, the components were divided into pretty small pieces that are effortless to get fixed. Pattern designs are followed when building infrastructure, creating components or naming a class as well. This makes the workflow is pretty much smoothly. Although not everywhere in the application has responsive across multi-devices, essential one is taken quite carefully. The next team perhaps effortlessly continue to develop on this one.

9.2 Failure

One of the critical issues is in security the password is not hashed, it is stored exactly as same as what the users type. This is harmful for personal information belongs to the customer. However, the most important issue is the content sticks to the layout. It leads to maintenance solutions that could be applied to the project and might fix these issues.

In human resources, it was challenging for the whole team. This is a very first real-life project for almost every member, they did not experience enough on this and were struggling with time management. All member of the team has different programming background which leads the human resource to become a bit struggle at the beginning.

Consequently, dividing tasks between the teams was not so successful which lead to some essential tasks are still missing. For example, the priority of heavy tasks should have been taken into account at first. After they are solved, smaller tasks can be handled and we will spend the rest of the time for them without losing the essential features.

Lastly, unfortunately, the failure in human resources leads to 2 critical issues and one of them makes obstacles for the customer to use an application. It means the content can only be edited from opening the codes but no other ways. It makes the customer cannot control the content based on their needs. This reason gets rid of the chance of the application to get in usage officially.

9.3 Limitation

The research has been done through discussions by theory references and practical from the project. However, the limitation is lacking statistics in the research, if the application is in used already, there would be more records about users' actions or getting to know about bugs reported by users.

Furthermore, the thesis could have collected data from the members and customers in the project. Unfortunately, the project is done and the team is disassembled. Besides that, the former customer representative moved to another job. Since the thesis could not get benefits from the customer opinion as well.

There are also ways that are mentioned and discussed to improve the software life cycle, the resource of time is not enough to make them available in the project. Unless they could be applied and perhaps there will be progress steps later. Hence, that could make the research more reliable.

9.4 Backlogs

Although there are some subjects that have discussed regard to maintainability in this thesis such as refactoring, frameworks, tools, etc. These things could be done deeper also there is still another could be researched further. Moreover, every case is unique, it would have its own solution. Hence, further study could be:

- The benefits of refactoring regard to technical matters in an interval event.
- Boosting maintainability in a large-scale application.
- Advantages of the maintainability in the LAMK web application.

10 LIST OF REFERENCES

Written References

Fowler, M. 2019. Improving the Design of Existing Code. The second edition. Pearson Education.

Santos, J. et al. 2006. Environmental Improvements and the 5S methodology. John Wiley & Sons, Incorporated.

Stephens, R. 2015. Beginning software engineering. John Wiley & Sons, Inc.

Tripathy,P. & Naik, K. 2015. Software evolution and maintenance. The first edition. John Wiley & Sons, Inc.

Tucker, A. et al. 2011. Software development an open source approach. CRC Press.

Electronic References

Casteren, W. 2017. The Waterfall Model and the Agile Methodologies: A comparison by project characteristics – short [accessed 09 November 2019]. Available at: https://www.researchgate.net/publication/313768860_The_Waterfall_Model_and_the_Agile_Methodologies_A_comparison_by_project_characteristics_-_short

CM, V. 2014. Architecture Issues Scaling Web Applications. Blog [accessed 05 November 2019]. Available at: http://venkateshcm.com/2014/05/Architecture-Issues-Scaling-Web-Applications/

Craig, W. 2019. 5 Fundamental Steps to Deploying a Website. Blog [accessed 09 November 2019]. Available at: https://www.webfx.com/blog/web-design/5-fundamental-steps-to-deploying-a-website/

Donaldson, N .2018. How Agile transparency reduces project risk. Blog [accessed 08 November 2019]. Available at: https://www.boost.co.nz/blog/2018/11/agile-transparency-reduces-project-risk

Github. 2019. Plans for every developer [accessed 10 June 2019]. Available at: https://github.com/pricing

Gray, T. 2018. MongoDB Atlas: what, why? Article [accessed 09 November 2019]. Available at: https://optimalbi.com/blog/2018/07/04/mongodb-atlas-what-why/

Hughes, M. 2017. Facebook's open source React library is increasingly worrying devs. Blog [accessed 15 June 2019]. Available at: https://thenextweb.com/dd/2017/09/19/should-developers-be-afraid-of-zuckerbergs-bearing-gifts/

Intellipaat. 2019. What is MongoDB? Article [accessed 09 November 2019]. Available at: https://intellipaat.com/blog/what-is-mongodb/

Johnson, H. 2017. Trello. Journal [accessed 03 November 2019]. Available at: https://wpcurve.com/trello-for-project-management/

Marmol, L. 2015. Agile Project Management for Non-Agile Professionals [accessed 03 November 2019]. Available at: http://www.agile-scrum.be/blog/agile-project-management-non-agile-professionals/

Merrill, P. 2019. Why you should refactor your test automation scripts. Article [accessed 09 November 2019]. Available at: https://techbeacon.com/app-dev-testing/why-you-should-refactor-your-test-automation-scripts

Miessler, D. 2019. The Difference Between Deductive and Inductive Reasoning. Blog [accessed 19 November 2019]. Available at: https://danielmiessler.com/blog/the-difference-between-deductive-and-inductive-reasoning/

Mongodb. 2019. What Is MongoDB? Official documentation [accessed 08 November 2019]. Available at: https://www.mongodb.com/what-is-mongodb

Nocarrier. 2008. Information Radiators. Blog [accessed 08 November 2019]. Available at: https://nocarrier.tistory.com/entry/Information-Radiators

Nodejs. 2019. About Node.js. Official documentation [accessed 10 June 2019]. Available at: https://nodejs.org/en/about/

Nodejs. 2019. Overview of Blocking vs Non-Blocking. Official documentation [accessed 10 June 2019]. Available at: https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/

Reactjs. 2019. The Component Lifecycle. Reactjs. Official documentation [accessed 17 April 2019]. Available at: https://reactjs.org/docs/react-component.html

Reactjs. 2019. Introducing JSX. Official documentation [accessed 10 June 2019]. Available at: https://reactjs.org/

React Native. 2019. Learn the Basics. Official documentation [accessed 29 October 2019]. Available at: https://facebook.github.io/react-native/docs/tutorial#whats-going-on-here

Romexsoft. 2017.How to Increase The Scalability of a Web Application. Blog [accessed 05 November 2019]. Available at: https://www.romexsoft.com/blog/improve-scalability/

Softwaretestinghelp.2019. Code Refactoring: What You Need To Know About It. Blog [accessed 09 November 2019]. Available at: https://www.softwaretestinghelp.com/code-refactoring/

Standuply. 2019. Learn How To Use Slack Effectively In 2019. Blog [accessed 09 November 2019]. Available at: https://standuply.com/how-to-use-slack

V8. 2019. What is V8?. Official documentation [accessed 10 June 2019]. Available at: https://v8.dev/

Zhan, Z. 2009. Refactoring Automated Functional Test Scripts with iTest2. Articles [accessed 09 November 2019]. Available at: https://www.infoq.com/articles/refactoring-test-scripts/

APPENDICES