Bournemouth University

FACULTY OF SCIENCE & TECHNOLOGY

BSc (Hons) Computing
July 2017

A Usable Usability Data Visualisation Tool

by

Stefan Horne

Faculty of Science & Technology
Department of Computing and Informatics
Final Year Project

Abstract

The study of usability plays an increasingly important and profound role in today's society, given that billions of people use the internet on a daily basis. The effects of improving internet usability would be incredibly important and far-reaching.

This project attempts to aid in that process by removing a needlessly time-consuming step from the daily routine of researchers and practitioners of usability. By developing a web-based data-visualisation tool, it will provide a quick and dirty way to analyse experimental data, providing on-the-fly insight, and potentially, an immediate indication of problem areas. This would obviate the requirement that several routine steps are carried out before any insight can be gained.

However, to achieve this, it will be important to set clearly defined goals and success criteria for evaluation. It will also require thorough research of the problem context, and investigation into potential methodologies for management and development, which will produce a viable roadmap for the project.

Not only does this have the potential to solve an important problem, but this will prove to be a profoundly challenging and valuable experience for the author.

Dissertation Declaration

I agree that, should the University wish to retain it for reference purposes, a copy of my dissertation may be held by Bournemouth University normally for a period of 3 academic years. I understand that once the retention period has expired my dissertation will be destroyed.

Confidentiality

I confirm that this dissertation does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. In particular any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history or sex life has been anonymised unless permission has been granted for its publication from the person to whom it relates.

Copyright

The copyright for this dissertation remains with me.

Requests for Information

I agree that this dissertation may be made available as the result of a request for information under the Freedom of Information Act.

Signed: ____.

Name: Stefan Horne

Date: 12/07/2017

Programme: Computing

Original Work Declaration

This dissertation and the project that it is based on are my own work, except where stated, in accordance with University regulations.

Signed: ____.

Acknowledgments

Firstly, I would like to thank Dr Nan Jiang for putting up with, what I imagine to be, one of the more difficult students he has had to supervise in his career. I cannot express enough gratitude for his advice, assistance, and patience throughout this challenging time.

Likewise, I wish to express my gratitude towards Richard Gordon and Lauren Jarrad, my employers, whom I also consider friends, and without whose support I would surely not have made it this far.

Finally, I would like to thank the academic staff, whose guidance and support has been a boon on my journey over these past 4 years. Your roles will be remembered.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Problem Background and Context

1.1.1 Context

According to We Are Social (2017), the majority of the developed world uses the internet. In particular, this is true for over 90% of the population in the UK. To compete for users' attention a website should always strive to provide the best user experience possible. Nielsen (2012), explains that, if users find a site difficult to use, they leave; if users find it difficult to identify what a website has to offer, they leave; if the information presented to the user by a website is hard to read or does not answer the right questions, they leave.

With such a significant portion of the world's population using the internet, the effects of usability research and evaluation are, potentially, incredibly profound and far-reaching, potentially propagating to billions of people worldwide.

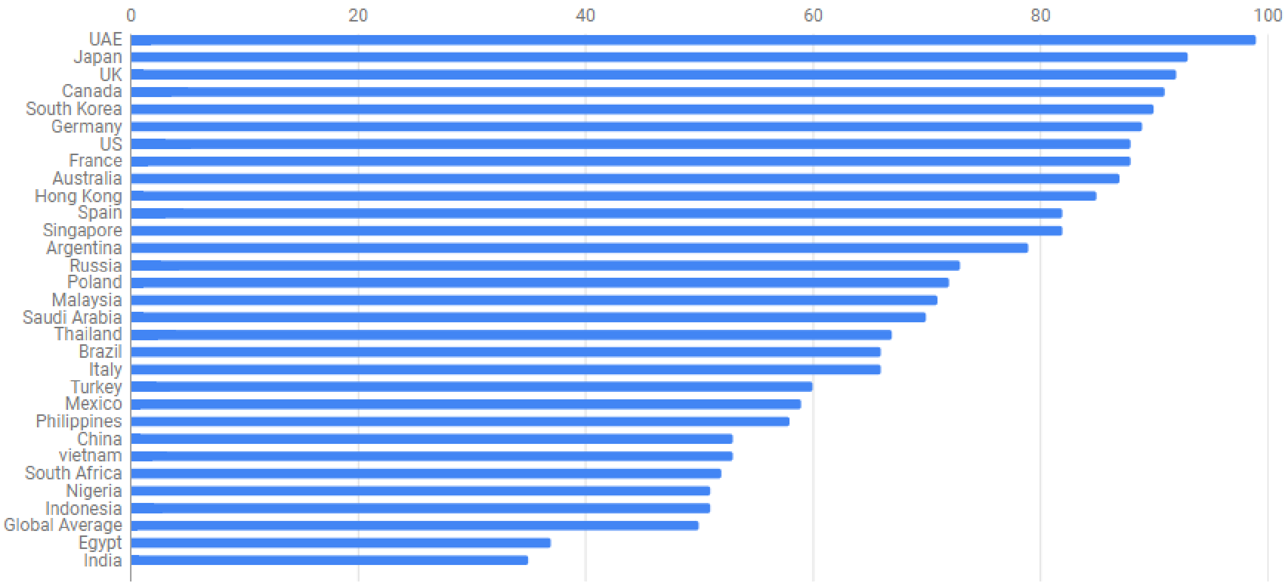


Figure 1. World internet penetration (We Are Social, 2017)

In an environment where users are likely to abandon a website in favour of another which is easier to use, measuring the usability of a system and being able to illustrate this in a meaningful way is an important step in creating a more user-friendly internet. A tool which can achieve this would be very valuable, and this paper details the process of researching and developing such a tool, with the goal of removing a needlessly time-consuming step from usability research and study.

This project will not only test the knowledge of this student, but also their ability to solve problems, deal with unforeseen circumstances, critically evaluate research materials, apply methodologies learned throughout their time at University, select an appropriate methodology, effectively elicit requirements, design effective solutions that meet these requirements, and finally, implement those solutions logically, all of which should be done in a timely manner.

1.1.2 Problem

Analysing usability is an involved process which starts with collection of data, followed by the calculation of several usability metrics based on that data. According to (Nielsen, 2001), usability data collected in a typical study should at least consist of: Success rate – whether the user succeeded or not, task time – the start and end time of the attempt, error rate – how many errors ocurred during the attempt, and some measure of user satisfaction.

This data is collected for each user attempt, for each task, and for each website. After collecting the data, the researchers and practitioners may wish to visualise this data during analysis. Azzam & Evergreen (2013) notes that visualisaing data is an important step in the analysis of quantitative data, and that it ‘enhances the evaluator’s grasp of the data’. This would typically involve the creation of improvised, ad-hoc spreadsheets or online graph generators; plotting a graph manually per metric, per task, per website. This can be a tedious process and, naturally, this is exacerbated when comparing and analysing multiple, inter-related data sets. This process can be time consuming, labour-intensive, and error-prone.

Furthermore, before any insight can extracted from this experimental data, these traditional methods must be followed to the letter. This means there is no shortcut to meaningful insight, as several time-consuming must first be carried out.

Thus, the problems identified are:

- Inevitable occurrence of errors when carrying out repetitive manual tasks
- A need for a shortcut to valuable insight – ability to analyse data on-the-fly

1.2 Proposed Solution

This project attempts to solve these problems through the research and development of a web-based tool which accepts and evaluates experimental usability data, visualising it in the form of various graphs, each of which relates to a quantifiable principle of usability. These graphs will ideally be able to provide multiple ways of viewing the data, that should be easily understood and always meaningful.

The ideal solution will also be responsive, allowing for use on mobile devices and handheld computers such as iPads. While the meaningfulness of data is reduced as the screen size decreases, the tool will support exporting the graph as a full-size image file.

The solution, developed using HTML5, CSS3 and JavaScript, will include multiple open-source JavaScript libraries, providing robust, well-tested and well-documented functionality, and allowing efforts to be devoted elsewhere.

Also of note is that, as an open-source project, hosted on GitHub, it will allow for future collaboration, promoting continuous feedback and iteration.

The tool will also feature a new usability metric, which more accurately indicates the relative effectiveness of a given system, the details of which may be found in the white paper: Understand System's Relative Effectiveness Using Adapted Confusion Matrix (Jiang & Haibin, 2013).

Finally, usability will be a very important factor which will be considered throughout development of the tool. Ideally the tool will be highly intuitive to use, and will not require an instruction manual to operate, although examples of how to use the system may be included if deemed necessary.

1.3 Aims and Success Criteria

1.3.1 Aims

The following high-level goals have been set in accordance with SMART (Specific, Measurable, Attainable, Realistic, Timely) (Blaine Lawlor & Hornyak, 2012):

1. Conduct background study, covering appropriate context, including usability, relevant technologies, and existing solutions.
2. Select appropriate methodologies.
3. Analyse and elicit requirements
4. Develop the application which meets the requirements
5. Evaluate and assess the extent to which the application solves the problem

1.3.2 Success criteria

1.3.2.1 Usable and robust

The tool should be easy and intuitive to use, as well as pleasing to the eye. In order to assess this, Nielson’s heuristics (1995) will be used as a model.

Furthermore, the tool should function efficiently and correctly with no unexpected behaviours.

Finally, the tool should function in a performant manner, with little to no noticeable delay for the user.

1.3.2.2 Effective

The tool should solve the problems outlined in section 0. The extent to which these have been achieved will be explored in section **Error! Reference source not found.**.

1.3.2.3 Achieve aims

The aims detailed above should be met. The extent to which this is the case will be discussed in section 6

1.4 Risk Analysis

Please refer to the risk assessment table **APPENDIX A**

1.5 Overview

With context established, and the problem defined, the sections that follow will detail the work carried out over the course of the project, starting with the background study that has been conducted, which will look at how usability is measured and visualised, followed by an appraisal of potential existing solutions. Furthermore, the requirements, as well as the implementation itself, will be discussed in detail, and finally, the conclusion section will discuss the extent to which the solution meets the requirements and solves the problem outlined in chapter 1, as well as areas with potential for future development and improvement.

2 BACKGROUND STUDY

2.1 OVERVIEW

The following section looks at the definition of usability and how it can differ between contexts. It will also look at the metrics used to measure and quantify usability, how they are analysed, and how they are presented. Furthermore, the section will examine data visualisation, and what it means in the context of this project. Finally, potential existing solutions will be appraised.

2.2 DEFINITION OF USABILITY

There are several valid definitions for the term “usability”. Nielsen (2012), an authority on the subject, defines it in terms of five quality components, namely: Learnability, Efficiency, Memorability, Errors, and Satisfaction.

The ISO 9241-11 standard elaborates on this, defining it as:

“*The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*” (Jokela, Iivari, Juha, & Minna, 2000).

Essentially, usability is the measure of how easily a user can reach their end goal using a system. It answers the questions: How easily was the user able to find the information they were looking for? Does the system produce the desired output, how well and how efficiently? And finally, how easy was it to learn to use the tool?

While the above definitions are adequate, and are generally considered valid, it is the ISO definition which best encapsulates its nuances, as it can be heavily context-dependant, and somewhat nebulous. Indeed, Speicher (2015) notes that both the above definitions leave much room for interpretation as usability is a ‘difficult-to-grasp concept’.

Thus, quantifying and measuring usability requires the establishment of a frame of reference. In other words, for the results to be at all meaningful, the product, users, goals, and use must be contextualised accordingly (Speicher, 2015), which is achieved through the analysis of usability metrics.

2.3 Metrics

2.3.1 Success rate

Also known as Completion rate, success rate speaks to the efficiency of a system, and is calculated by expressing the proportion of tasks that have been successfully completed as a percentage. This metric is a staple in usability performance study, as it is easy to collect the required data, easy to understand and always meaningful (Nielsen, 2001).

For a given task with three participants, two of which completed the task successfully, the success rate would be calculated as 2/3 = 66.67%.

2.3.2 Completion time

Task completion time speaks to the efficiency of the system and is calculated by averaging the time users took to complete a given task. This metric is not concerned with whether a task was completed successfully or not. For unsuccessful attempts, capitulation time is used instead.

Given a task with three participants who took 30, 60 and 50 seconds respectively to complete the task, the completion time is calculated simply as (30+60+50)/3 = 46.67 seconds.

2.3.3 Average errors

Average number of errors is calculated by averaging the number of errors made by participants while performing a given task.

For a given task, with three participants who made twenty errors collectively, the average errors would be calculated as 20/3 = 6.6.

2.3.4 Error rate

Error rate is calculated by expressing the proportion of actions linked to errors compared to the total number of actions taken.

For a given user who performed 8 actions, 3 of which resulted in errors, the error rate would be calculated as 3/8 = 37.5%

2.3.5 Task-level satisfaction

This metric aims to provide a measure of the difficulty experienced by the user when attempting a particular task. After the task is attempted, the user is given a questionnaire which consists of up to five questions, which is done regardless of whether the task was completed successfully or not.

A widely used (Sauro, 2010) example of such a questionnaire, is the Single Ease Question (SEQ), consisting of a single question and a scale with seven points, pictured below.

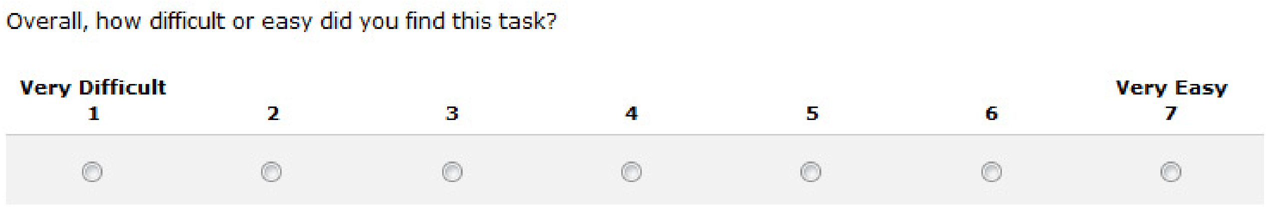


Figure 2. The Single Ease Question (Sauro, 2012)

The SEQ is simple but effective and in a study conducted by Sauro & Dumas (2009) it was concluded that it is at worst equivalent and at best an improvement over more complicated task-difficulty measures.

Sauro (2010) suggests that this kind of metric provides a measure of what he terms *performance satisfaction*, providing an immediate indication of the problem-areas in a user interface and where to focus efforts for improvement. This is because the questionnaire is completed shortly after the task was attempted, while the user's thoughts and sentiments surrounding the task are fresh in their mind.

2.3.6 Test-level satisfaction

This metric aims to provide a measure of the overall difficulty experienced by a user over the course of multiple tasks. After the test session, the user is issued a questionnaire which consists of up to fifty questions.

The most widely used questionnaire for this metric is the System Usability Scale (Sauro, 2010), which consists of the following ten questions:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

A Likert scale accompanies each question as pictured below:

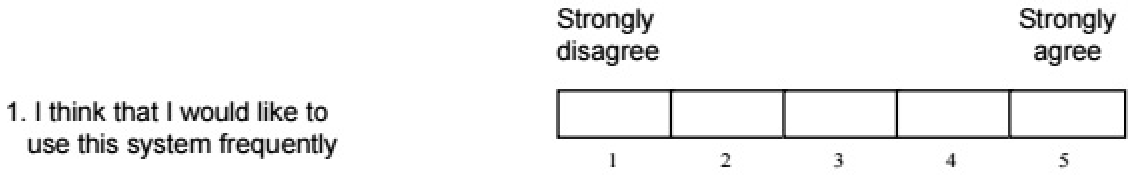


Figure 3. Likert scale (Thomas, 2015)

This metric is not as telling as the previous one, but it can still play a key role in assessing the usability of a website, as it can highlight problems which are not as obvious and would not have been uncovered otherwise.

2.3.7 Adapted confusion matrix technique

This technique, proposed by Jiang & Haibin in 2013, provides a measure of the relative effectiveness of a system based on a prediction of its own performance.

By correlating the measures of success and errors, it dispels any ambiguity in the relationship between the two, which can vary drastically between cases (Jiang & Haibin, 2013). This is achieved, in part, by using the number of errors as a classifier, and a binary success/failure for the classes.

This works by establishing an estimated upper-bound of errors, which serves as a benchmark. If, the number of errors largely falls below this threshold, it indicates that critical errors mostly occur when the number of errors exceeds this threshold (Jiang & Haibin, 2013). The system performance can then be expressed based on the number of critical errors which occurred at this threshold.

A confusion matrix is used to record and analyse the system's predicted and actual performances, pictured below.

		Prediction	
		Completion	Incompletion
Actual	Completion	*true positive cases (tp)*	*false negative cases (fn)*
	Incompletion	*false positive cases (fp)*	*true negative cases (tn)*

Figure 4. Adapted confusion matrix (Jiang & Haibin, 2013)

The performance is expressed using three measures, namely: recall, precision, and FScore.

Recall represents the accuracy of the predicted outcomes and is calculated as:

$$recall = \frac{tp}{tp + fn}$$

Figure 5. Recall formula (Jiang & Haibin, 2013)

Precision represents the accuracy of the actual outcomes and is calculated as:

$$precision = \frac{tp}{tp + fp}$$

Figure 6. Precision formula (Jiang & Haibin, 2013)

Finally, an F1 score is calculated to indicate at which upper-bound the system performed best. This is calculated as:

$$Fscore = 2 \times \left(\frac{tp \times fp}{tp + fp}\right)$$

Figure 7. F1 Measure (Jiang & Haibin, 2013)

2.3.8 Simplified Usability Score

In some situations, it may be desirable to represent the usability of a system using a single standardized score. In this case, some combination of the above metrics is used to calculate this.

A prominent example of one such metric is the Single Usability Metric (SUM), which is calculated by combining measures of completion, time, satisfaction, errors and clicks

This is useful for comparing systems on a high level but ultimately cannot be used as a substitute for its constituent metrics. (Sauro & Kindlund, 2005) echo this, stating: '[a single usability score] can never replace all the information inherent in the component metrics'. This is because valuable contextual information is lost in the process of combining the metrics, and as a result, so is much of its meaning.

2.4 Data Visualisation

Data visualization is a term which can encompass many things, depending on the context of use. In the context of this project, a fitting definition is offered by Azzam & Evergreen (2013), defining it as a process which is human-readable, based on raw qualitative or quantitative data, which produces an image representation of said data, and facilitates "exploration, examination and communication of the data".

Representation of the data in this way enables a different kind of analysis, owing to the fact that humans are visually-oriented animals. A visual format enables us to use our intuition, in addition to our cognitive abilities, when analysing the data. It employs a different part of our brain, the visual cortex, which handles all incoming visual information (Few, n.d.). This allows us to percieve potential hidden patterns, trends, and correlations which would otherwise go unnoticed. This is due to the fact that the visual cortex is much faster and more efficient at processing data than the cerebral cortex, which is responsible for cognition (Few, n.d.).

However, in order to take advantage of this fact, data visualisation tools should be designed in accordance with human-perception-centric design principles (Few, n.d.). This aligns not only with the area of study this tool attempts to aid, but also with usability goals, implying that, necessarily, the interface should be user friendly for the tool to be effective.

Furthermore, tools should be designed with purpose in mind. Few (2009) proposes that, based on their purpose, analytics tools can be divided into three categories; namely, Exploratory, Custom, and Customisable. Exploratory analytics tools allow for unhindered, unconstrained analysis of the data. They are typically designed with several purposes in mind. Custom analytics tools on the other hand, are designed with a specific purpose in mind and lend themselves to efficient and convenient completion of a routine task, obscuring extraneous functionality from the user and focusing on ease of use. Thus, the proposed solution belongs to this category, owing to the fact that it aims to provide an easy-to-use interface, with a single purpose in mind; visualise this data.

2.5 EXISTING SOLUTIONS

2.5.1 Google Fusion Tables (GFT)

Launched in 2009, GFT is a web-based system which provides a facility for uploading tabular data to the cloud, where it may be combined with other data, filtered, and visualised in the form of graphs, of which there are several types to choose from. This includes several interactive charts which allow manipulation of the data and views of differing levels of granularity.

With a focus on collaboration, the tool allows for the combination of data from multiple sources, including data hosted in the public dataset (Halevy, 2009). The tool also supports multiple file formats, including spreadsheets and CSV files, up to 100 megabytes in size.

While this tool sports a plethora of useful features, it is overly complicated for the problem this project attempts to solve. With reference to section 2.4, it becomes clear that this was designed as an exploratory analytics tool. Most damningly, however, the tool is still currently in its experimental phase, which means that it is liable to change in the future, perhaps drastically. This also implies that the system would not have been adequately tested.

Finally, GFT is only able to visualise data which has already been analysed, whereas the tool proposed in this project has been designed specifically to accept raw data, analyse it, and provide on-the-spot insight into the data.

2.5.2 Plotly

Plotly offers several products, including database connectors, dashboards, an open-source JavaScript library, and, the product in question, a web-based graphing tool which features a rich functionality set. File types supported include excel, csv files, and .sql files, and like GFT, it focuses on collaboration with the community.

The tool itself is very versatile, allowing users free-reign over the data, which can be modified conveniently using the excel-like interface. Many chart types are also supported, including science, statistics, cartographic, and 3D plots. These charts can be styled using several of the available styling options, including the ability to change the colours of bars, lines, text, axes, and data-points.

As with GFT, Plotly was designed to provide users a large degree of freedom and customisability, with many purposes in mind. As such, it would not provide the convenient experience proposed by this project, as the graph requires some configuration before the chart can be plotted. More importantly, however, it lacks the ability to analyse the raw data, which is a core requirement of this tool.

Notwithstanding that, it does support incredibly convenient importation of graphs from the plotly.js JavaScript library with the click of a button. This could potentially allow users to export their data from the tool proposed in this project into the Plotly tool with ease, allowing users to take advantage any relevant functionality, if they so desire.

Although Plotly is largely ill-suited for the problems proposed, the fact that utilising its JavaScript library would essentially extend the functionality of this tool unobtrusively and without additional required effort makes it an attractive candidate for addition to this project.

3 REQUIREMENTS AND ANALYSIS

3.1 OVERVIEW

The purpose of this chapter is to delineate the scope of the project. This is an important step in any development project, as it ensures that efforts are not misguided, and serves as a benchmark for evaluation down the road. Candidate development methodologies will be examined in this chapter, and the rationale behind the chosen methodology will be discussed.

The following was considered when selecting an appropriate methodology:

- The requirements of this project are well understood
- These requirements are unlikely to change
- The problem at hand is relatively simple

3.2 METHODOLOGIES

3.2.1 Scrum

Scrum is a framework which lends itself to problems which may not be perfectly understood, and promotes a high degree of quality and efficiency (Sutherland & Schwaber, 2016). Its use is not limited to the software development world, but for the sake of this project it will be discussed in this context only.

Scrum is a popular methodology for use in agile projects, as it facilitates frequent iterations of the system, and constant feedback. This is because development is carried out in stages, or “sprints”. Planning a sprint involves designating the time-period, and specifying the tasks to carry out during this period. The tasks are typically prioritised and developed accordingly. Once a sprint is complete, any tasks that were not completed are added to a backlog, to be included in the next sprint where it will be prioritised against any new tasks on the list. This ensures that efforts are focused on the appropriate areas.

Scrum would be a good candidate methodology for this project, given that it works especially well in small teams. Indeed, in this case the team consists of a single developer, obviating the constraints imposed by communication. However, the requirements of this project are fairly well understood, eliminating the need for an incremental approach.

3.2.2 Dynamic systems Development Method (DSDM)

DSDM is a variation of the agile development methodology which promotes the use of the MoSCoW prioritisation technique (Hatton, 2008). It employs the use of a technique referred to as “Timeboxing”, where the project is divided into several time periods, each with a set of aims.

It is worth nothing that, to account for possible changes in requirements, the scope for each task may be modified to ensure that the deadline is met.

While this methodology could be applied to this project, due to the requirements being well-known up-front, it is not well suited for this task. However, the MoSCoW prioritisation method has some value, and can be used in conjunction with other methodologies more suited to it.

3.2.3 Waterfall

The waterfall model is a development methodology which separates development into distinct, linear stages, each of which is concerned with a different facet of the project. The stages, first described by Royce (1970), are:

1. Requirements specification
2. Design
3. Implementation
4. Testing
5. Maintenance

This methodology is often faulted for its inability to respond to changing requirements. However, if requirements are known from the start, it offers some advantages

Firstly, development progress is much easier to measure, as stages can serve as communicable milestones. Secondly, if requirements are known up-front, less time is spent on elicitation, and can be used on other stages of the development, increasing quality significantly.

3.2.4 Rationale for chosen methodologies

After conducting the research, and much deliberation, it was decided that this project does not necessitate any sophisticated project management methodologies, and that an ad-hoc approach would be used instead, in conjunction with the MoSCoW prioritisation technique. The latter would serve to guide efforts during development, ensuring an accurate trajectory.

Furthermore, given the relatively simple nature of this project, and that the requirements are well understood and not liable to change, it was decided that a linear development approach would be taken.

3.3 Requirements Prioritisation Using MoSCoW

MoSCoW		
ID	**Requirement**	**Priority**
M1	File input	**Must Have**
M2	Recognise and analyse research data	
M3	Completion time graph	
M4	Success rate graph	
M5	Average errors graph	
M6	Error rate graph	
M7	F-Measure graph	
M8	Recall graph	
M9	Usable	
M10	Drag and drop file input	**Should Have**
M11	Offer multiple ways to input data	
M12	On-the-fly data modification and real-time graph updates	
M13	Good performance	
M14	Compatbility with modern browsers	
M15	W3C compliant	
M16	Compatibility with older browsers	**Could Have**
M17	Responsive	
M18	Support multiple systems	
M19	Collaboration	

Figure 8. MoSCoW table

3.4 Summary

This chapter has looked at some of the methodologies that have been researched, and their applicability to this project. This information has in turn been used to justify the chosen methodology. And finally, the requirements have been listed and prioritised.

4 DESIGN

4.1 OVERVIEW

The design phase is a key component in the systems development life cycle. This chapter will look at some of the research conducted into the chosen design methodology, which will be examined and justified. Furthermore, it will look at the rationale behind decisions made during the interface design process. This will include examples of subtle techniques used to influence the user's experience of the system, with the purpose of improving their experience.

4.2 HEURISTICS FOR USER INTERFACE DESIGN

According to Nielsen & Molich (1990), four methods exist for evaluating a user interface: Formally by analysis, automatically by a computer process, empirically by experiments involving test users and data collection, or heuristically, by judging the interface based on one's own experience and opinion.

Although heuristic evaluation works better with multiple evaluators (Sauro, 2010), it can be carried out without much need for preparation (Nielsen & Molich, 1990), meaning that it can be used from the very early stages, moulding the interface design throughout development of the tool, and can therefore be used not only to evaluate the finished interface, but also to during the design process.

Thus, Nielsen's (1995) ten heuristics were chosen to serve as a guideline during the design and evaluation of this tool. Their role will be examined in **Section 6.2**.

4.3 INTERFACE DESIGN CONSIDERATIONS

In order to facilitate the best user experience possible, a minimalist approach was chosen for the interface design. According to (Nielson, 2015):

> "A minimalist web-design strategy is one that seeks to simplify interfaces by removing unnecessary elements or content that does not support user tasks".

This aligns directly with the usability requirement featured in section 3, and its influence should be obvious in the following paragraphs which go into more detail.

When the user first visits the site, their eyes are directed to a singular object; a blue box with a perforated border, pictured in **Figure 9. Landing "page"**. This is achieved using two principles of visual organisation; contrast, and dominance. The former is achieved through colour, and serves to emphasise the box, while its lack of proximity to the rest of the page makes it the dominant feature, and immediately grabs the user's attention (usability.gov, 2017). By isolating and contrasting the box from the rest of the page, the user is given implicit instructions on how to proceed. Such principles are used throughout the rest of the interface, to similar effect.

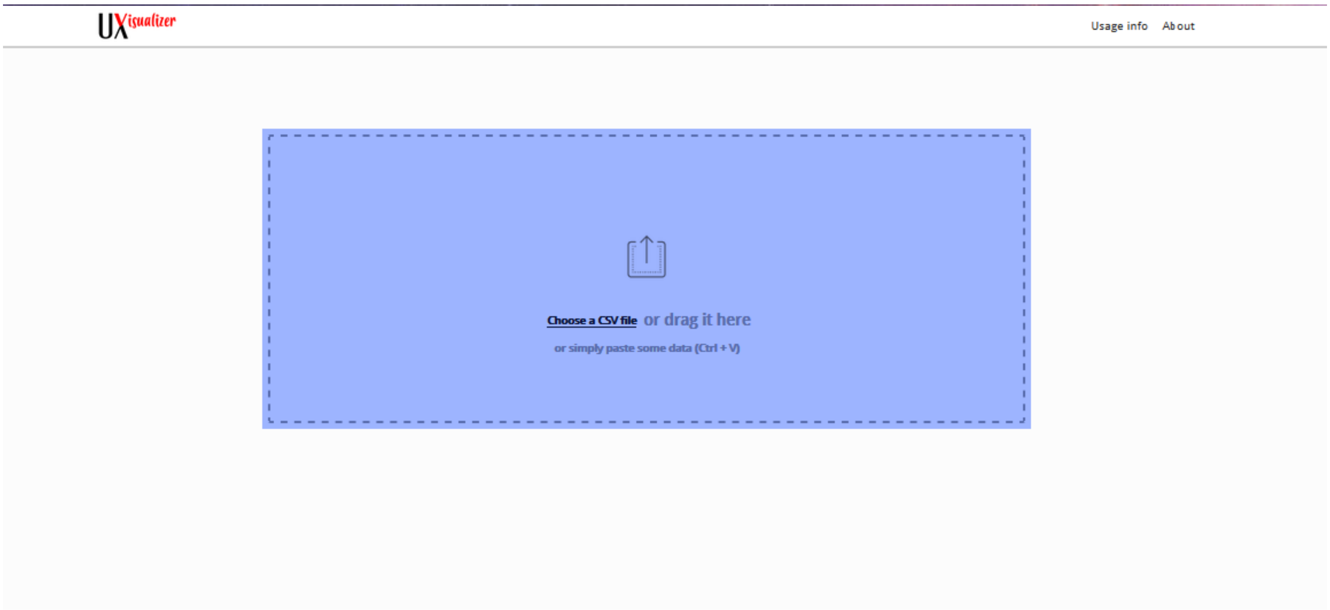


Figure 9. Landing "page"

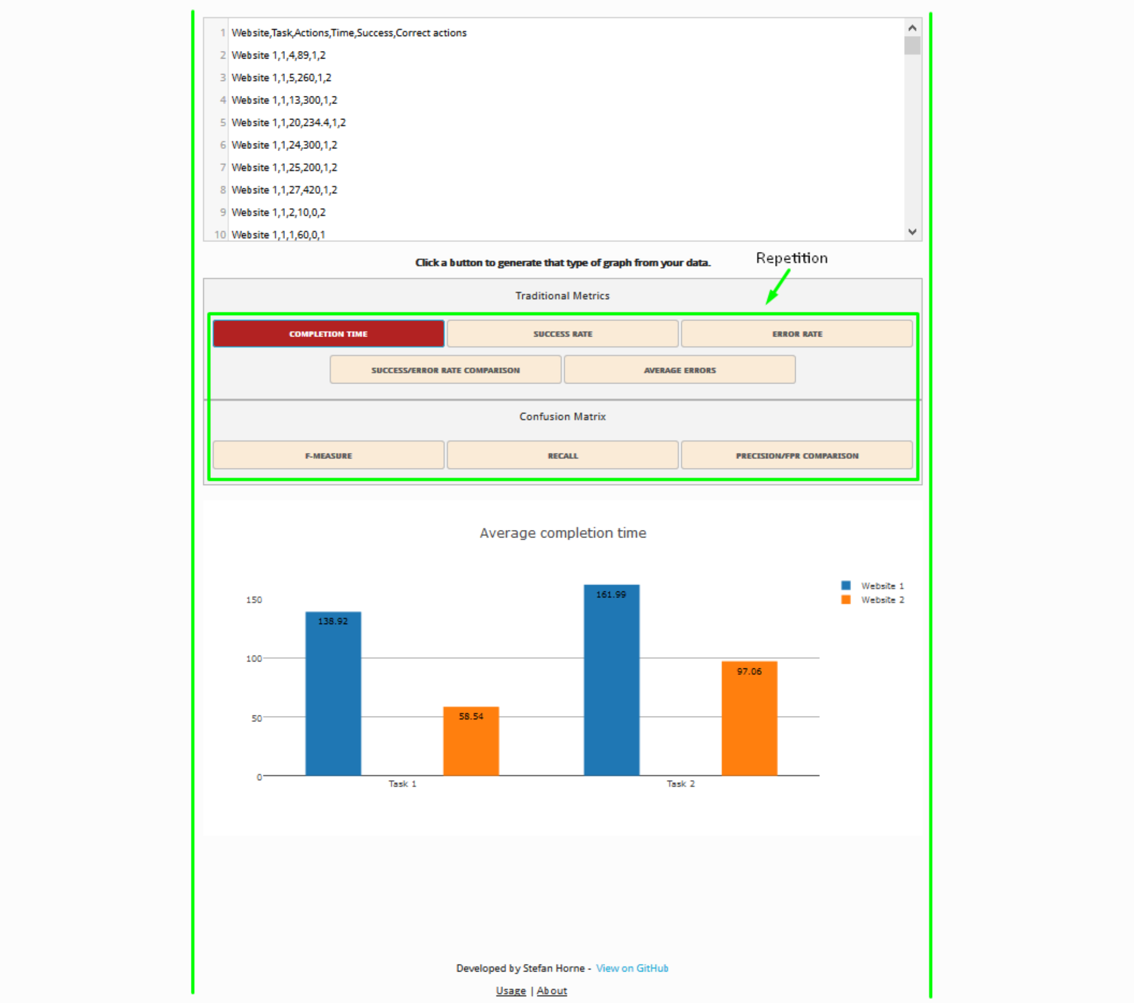


Figure 10. Alignment and repetition

In addition to the concepts mentioned above, the page shown above employs several principles of visual design and organisation to tailor the experience for the user.

For example: unity, giving the impression that elements belong together (usability.gov, 2017), is achieved through the use of alignment with invisible lines, and repetition of a particular style. However, sometimes it is impossible to direct a user's eye in such a subtle way, and so an alternative technique should be used.

When a user clicks a button to generate a graph, it appears outside of the browser boundary and outside of their view. In order to view it in its entirety the user would then have to scroll down, breaking their immersion. To counteract this, a mechanism was implemented that automatically focuses the graph once it has been generated. This maintains continuity, by avoiding disruption to the user's experience.

Several more of examples of such design principles were employed to maximise the ease of use, without the user's knowledge whenever possible. It's not about how much information can fit on the page, but how much can be removed without compromising the quality of the outcome. This may be a tenuous relation, but it was Pascal (1656) who once echoed a similar sentiment: 'I would have written a shorter letter, but I did not have the time'.

This speaks to the fact that less is more, and that reaching equilibrium between ease of use, effective functionality, can be challenging.

4.4 System Design

4.4.1 General design considerations

4.4.1.1 Compatibility

One of the must have requirements is that the system be compatible with all modern browsers. However, another one of the must-haves is that it supports a drag and drop interface, which is only supported by 40% of the browsers globally and 43% of browsers in the UK (according to caniuse.com).

Thus, it is necessary to implement a function which detects whether the user's browser supports this functionality, and modify the user interface to reflect this. This is illustrated in **Appendix C**

4.4.1.2 Raw data format

In order for data to be usable by the system, it must be arranged in a format which is recognised by the system. This required the design of unique data configurations which can be intuitively understood by users while also being suitable for parsing and analysis by the system.

An example of this can be seen in **Figure 19. Confusion matrix data accompanied by respective reference**. Headings serve as reference points for the algorithm while simultaneously increasing the readability of the data.

In turn, this necessitates the design of an algorithm which can parse this data consistently, while alerting the user to any parsing errors, and being able to distinguish between the different sets of data amongst the data proffered. In some cases, the algorithm also interprets the user's desired output based on the format of their data.

4.5 SUMMARY

This chapter has examined several design techniques, their effects, and how they contribute to a better experience for the user. This includes a minimalist design, which removes unnecessary elements from the interface, allowing users to better focus on important information. Subtle techniques for visual design have also been explored, illustrating that the best techniques are often the least obvious ones. General design considerations have also been discussed, showing how they too contribute to the goal of improved user experience. Finally, the chapter detailed the value of the chosen design methodology, and how it can be used for evaluation as well as during the design stages.

5 IMPLEMENTATION

5.1 OVERVIEW

This section will detail the implementation of the functionality that meets the requirements discussed in section 3. It will outline a non-exhaustive list of features, each of which will be examined and explained to provide insight into the system's inner-workings and how it achieves the desired outcomes.

5.1.1 Third-party libraries used in this project

- **jQuery** – incredibly well tested and documented, feature-rich JavaScript library
- **Modernizr** – Feature detection. Improves user experience and compatibility by detecting features offered by user's browser.
- **Lodash** – very well-tested and documented JavaScript utility belt tailored for performance and modularity
- **CodeMirror** – highly versatile text editor with a rich set of functionalities
- **Plotly.js** – their own website says it best; Plotly is a high-level, declarative charting library
- **Papa Parse** – a simple yet robust CSV parser
- **Skeleton.css** – a lightweight, responsive css boilerplate

5.2 USER INPUT

As noted in the requirements, it was important that users are offered multiple ways to submit data. This makes the tool more versatile and conducive to a good user experience. As such, the tool offers the ability to submit a file via drag and drop, traditional file selection, or the data may simply be pasted into the browser window from the clipboard. The way this was achieved is detailed below.

5.2.1 CodeMirror plugin

This plugin is a. It was chosen for its simplicity and ease of use and enables the user to inspect the data, as well as modify it on-the-fly.

This plugin operates just like a `<textarea>` html element, but includes additional functionality such as events and formatting options.

Bound to the ‘blur’ event of the CodeMirror container is an update function. This facilitates real-time updates after users make changes and click outside the plugin, or the plugin loses focus. This function is pictured below.

```
        this.$codeMirror.on("blur", function(cm, change) {
            $('#codeMirrorMain').css('border', 'none');
            if (self.$codeMirror.doc.getValue() === '') {
                self.reset();
            } else {
                self.buttonClicked({
                    id: self.lastOp
                });
            }
        });
```

Figure 11. CodeMirror 'blur' event

When a user performs an operation, this operation is stored in a variable. This is so that when the above event fires, it can perform the same operation again. The `if` statement distinguishes between the user updating an aspect of the data, and the user completely clearing the contents. In the latter case it simply resets, awaiting new data.

Finally, a blue border is used to indicate focus status. The above snippet shows the removal of this blue border when the plugin loses focus.

5.2.2 File input

Files containing experimental data can be submitted in .csv format. These files must be pre-formatted to match the required data format outlined in the ‘Example usage’ section. Once the file is submitted, the contents are parsed and sent to the CodeMirror plugin for display to the user. This function is pictured below.

```
        function fileToCodeMirror(file) {
            if (config.ALLOWED_FILETYPES.indexOf(file.type) !== -1) {
                var csv;
                config.papaConfig.complete = function(results, file) {
                    csv = _.reduce(results.data, function(sum, n) {
                        return sum + n + "\n";
                    }, "");
                    self.$codeMirror.doc.setValue(csv);
                };
                Papa.parse(file, config.papaConfig);
                fileSubmitted();
            } else {
                functions.showError('Unsupported file type');
            }
        };
```

Figure 12. File to CodeMirror function

The file type is compared to a list of allowed filetypes and if found to be valid (line 760), its contents are retrieved using the Papa Parse parsing library (line 768). The CodeMirror plugin is populated upon completion (line 766).

Finally, the `fileSubmitted` function performs several styling changes which serves to provide feedback to the user, indicating that the data has been parsed and they may proceed. This includes unhiding the CodeMirror text input field and operation buttons.

5.2.3 Text input

The tool also supports pasting data from the clipboard, provided the data is in the correct format. This allows the user to simply copy data from an excel file and paste it straight into the application. If this is done on the index page before a file is submitted, the data from the clipboard is sent to the CodeMirror plugin (Figure 13. on 'Paste' binding). If this is done after a file has already been submitted, it is handled differently but the result is the same. In this case the data is simply pasted into the CodeMirror element, the same way it would if a `<textarea>` element was used.

```
$(window).on('paste', function(e) {
    fileSubmitted();
    self.$codeMirror.doc.setValue(e.originalEvent.clipboardData.getData('text'));
});
```

Figure 13. on 'Paste' binding

5.2.4 Operation selection

Once the data has been submitted using one of the above methods, the user must choose an operation to perform on the data. A list of operations is provided in the form of buttons, with captions indicating their operation. These buttons are separated into two categories depending on the data they require.

```
$(".toggle").click(function() {
    self.buttonClicked(this)
});
```

Figure 14. operation button bindings

```
627     buttonClicked: function(caller) {
628         var self = this;
629 
630         Plotly.purge('chart');
631 
632         // If the button was actually clicked toggle the classes; otherwise it is redoing the previous operation
633         if (caller.nodeType === 1) {
634             $(".toggle").removeClass('button-primary');
635             $(caller).addClass('button-primary');
636         } else {
637             $("#" + caller.id).addClass('button-primary');
638         }
639         self.lastOp = caller.id;
640 
641         switch (caller.id) {
642             case "ct":
643                 self.completionTime();
644                 break;
645             case "sr":
646                 self.successRate();
647                 break;
648             case "er":
649                 self.errorRate();
650                 break;
651             case "ae":
652                 self.averageErrors();
653                 break;
654             case "serc":
655                 self.successErrorComparison();
656                 break;
657             case "fm":
658                 self.fMeasure();
659                 break;
660             case "rc":
661                 self.recall();
662                 break;
663             case "pfc":
664                 self.precisionFPRComparison();
665                 break;
666         };
667 
668         document.getElementById('chart').scrollIntoView()
669     },
```

Figure 15. buttonClicked function

When a button is clicked, the element object which was clicked is passed to the `buttonClicked` function. The if statement determines whether the caller is an actual button, or the update function mentioned in the section 5.2.1, and updates the button style accordingly.

As mentioned before, the operation is saved (line 639), and finally the appropriate operation is determined using the switch statement and caller id.

5.3 Plotting the Charts

Once the appropriate operation has been triggered, the data, which is takes the form of a single large string, is parsed using Papa Parse, converting it into a multidimensional array.

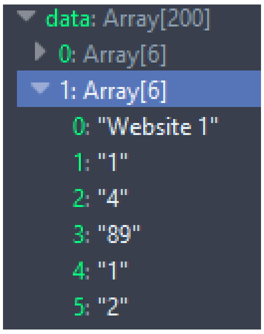


Figure 16. Papaparsed data

5.3.1 Analysing and processing the data

After the initial parsing, the way this data is processed from here depends on the graph being plotted.

For most of the graphs in the 'Traditional Metrics' section, the data is processed once again using the `analyseData` function, sorting the data by site and task.

5.3.1.1 Traditional Metrics

For most of the traditional metric graphs, the data is sent to the `analyseData` function. The data is sorted using two lookup arrays, as well as an array which is populated with the sorted data. When a website or task is first encountered, it is pushed into the appropriate lookup array. Each subsequent appearance is then tested for existence in the array, and the resulting index used to determine the appropriate index location where the data should be placed in the sorted array. This is a kind of ad-hoc comparison sort, which sorts by two inter-dependant properties while maintaining the original order of the data. Only a single iteration of each value is required to sort the whole array, despite the two-dimensional nature of the desired result. This, coupled with the fact that the JavaScript implementation of an array is such that accessing and inserting is achieved in constant time, results in a time complexity of O(n).

Note, however, that the analysis function performs other O(n^2) operations which are unavoidable, such as the operation which checks for alphabet characters (line 61-65).

```
var sites = [];
var tasks = [];
var sorted = [];
var valid = true;
_.each(data, function(val, index) {
    _.each(val, function(v, i) {
        if (i > 0 && v.match(/[a-z]/i)) {
            valid = false;
        }
    });
    var i = sites.indexOf(val[0]);
    var j = tasks.indexOf(val[1]);
    if (i === -1) {
        sites.push(val[0]);
        i = sites.length - 1;
    }
    if (j === -1) {
        tasks.push(val[1]);
        j = tasks.length - 1;
    }
    if (typeof sorted[i] === 'undefined') {
        sorted[i] = []
    }
    if (typeof sorted[i][j] === 'undefined') {
        sorted[i][j] = [];
    }
    sorted[i][j].push(val);
});
```

Figure 17. Sorting algorithm

Once this is complete, lodash `_.zip` is used to transmute the data into a format more conducive to calculating the usability metrics.

The resulting object contains three arrays and a boolean flag to indicate whether any alphabet characters were found, pictured in figure 12.

```
▼ data: Array[2]
  ▶ 0: Array[2]
  ▶ 1: Array[2]
    length: 2
  ▶ __proto__: Array[0]
▼ sites: Array[2]
    0: "Website 1"
    1: "Website 2"
    length: 2
  ▶ __proto__: Array[0]
▼ tasks: Array[2]
    0: "1"
    1: "2"
    length: 2
  ▶ __proto__: Array[0]
  valid: true
```

Figure 18. Sorted for metric calculations

Finally, the operation-specific calculations are carried out. For example, for a completion time graph, for each website and task, the sum of all the completion times is calculated using lodash `_.reduce`, and divided by the length of the array, resulting in the average completion time for each respective website and task.

5.3.1.2 Confusion matrix

As with the traditional metrics, the confusion matrix graphs all use their own variation of an otherwise similar algorithm.

The F-Measure algorithm has been chosen as an example.

Instead of iterating over each row of data, this function employs a simple `for` loop which increases the counter by three with each iteration. This is so that the counter always refers to the heading of the current matrix, serving as a reference point, avoiding necessitation of a more complicated method. The current operation is then simply able to reference the information in terms of the current counter. This is illustrated below:

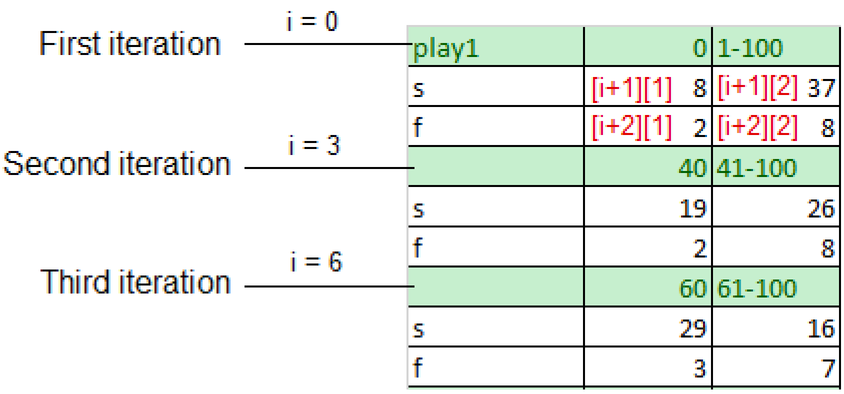


Figure 19. Confusion matrix data accompanied by respective reference

With a reference point established, the F-Measure is calculated and pushed into the results array used to draw the graph.

5.3.2 Drawing the plot

The graphs are drawn using a library called Plotly.js. Plotly was built on top of D3.js, a well-known graphing library. Plotly is very well documented and although it is not as versatile, and does not offer as much low-level functionality as D3, it is much easier to use and, otherwise, suits the needs of this tool.

Drawing the graph is trivial once the data has been formatted correctly. To do so, the `newPlot` function is passed the html div id, and the data in the following format:

```
x: xaxis, // array
y: values, // array
mode: 'lines+markers',
type: 'scatter',
name: analysedData.headings[i],
marker: {
    size: 8
}
```

Figure 20. Object passed to plotly draw function

5.4 Responsiveness

Responsiveness in the context of interface design refers to the ability to respond and adapt depending on device screen size. This is done to maintain a good user experience across several types of device.

In the case of this tool it was achieved through the combined use of a responsive CSS framework (skeleton.css), and custom media queries.

```
@media (max-width:768px) {
    nav {
        max-width: 100%;
    }
    .logo-icon {
        text-align: left;
        width: 50%;
    }
    .toggle {
        width: 100%;
    }
    .button-wrapper {
        display: none;
    }
    .footer-nav ul {
        width: auto;
    }
    .footer-nav li {
        float: none;
    }
    .file_select {
        width: 100%;
    }
    .file_select strong {
        font-size: 1.2em;
    }
}
```

Figure 21. Media queries

The above styles are applied only when the screen width shrinks below 768 pixels.

The effects of these media queries are illustrated in **APPENDIX B**.

5.5 SUMMARY

This section has taken a detailed look at the fundamental functions of the system, giving a better understanding of some of the rationale behind some of the design choices, and a better understanding of how the more prominent features have been implemented. The flow of data has been followed from the moment of input up until the repeated data transmutations culminate into a form that is radically different and significantly more meaningful.

6 CONCLUSIONS

6.1 SUMMARY

The ultimate goal of this project was to produce a solution to productivity problem commonly experienced by researchers. This has been achieved, and the evaluation section will illustrate that this is the case. The solution is an effective, usable, and compatible web-based tool which accepts and analyses experimental data, visualising it in the form of user-friendly graphs, which are displayed in a user-friendly manner, and can operate across multiple devices. This provides a shortcut to insight for researchers and practitioners alike. Most notable of these graphs are those which are generated based on the confusion technique. As a relatively recent development, this tool can boast that it is one of, if not the first tool to provide this type of analysis, and a new form of insight.

This could not have been achieved without an investigation of the problem space, which included the areas of usability research, quantitative data visualisation, and similar off-the-shelf solutions. Equipped with this knowledge, it was possible to tailor the solution to the exact needs of the target audience.

Furthermore, the chosen methodologies played a key role in its success, ensuring that time and efforts were devoted appropriately. This is because they allowed clear design goals to be established.

This project has proven to be a profoundly challenging and valuable learning experience, having had to consider every stage of the software development life cycle in its entirety. This has developed and honed the student’s technical and time-management skills, which will aid them as they issue forth into the next stage of their career.

6.2 EVALUATION

6.2.1 Success criteria defined in Section 1.3

6.2.1.1 SMART Objective 1 – Background Study

The background study in **Section 2** was conducted, covering key areas in this project, including the study and measurement of usability, data visualisation, and potential existing solutions.

Usability was defined, and some of the methods for its measurement were examined in detail. This included detailing several fundamental usability metrics, as well as the newly-proposed adapted-confusion-matrix technique. This was an important step in (a) learning how to analyse the raw data, and (b), deciding how best to visualise this data in a meaningful way. The data visualisation research further aided the latter.

Research into data visualisation also focused the design efforts of the project, assuring that the proposed solution would be designed for its specific purpose, a key consideration explained in **Section 2.4.**

6.2.1.2 SMART Objective 2 – Select appropriate methodology

Section 3.2 details the research relating to, and rationale behind the chosen methodologies. The research covered two contemporary agile methodologies, which were determined to be unsuited to this project, given that requirements were well known and unlikely to change.

This turned out to be pivotal in the development of the artefact, as, during development, due to unforeseen personal circumstances, work had to cease for an inordinate amount of time. However, the chosen development methodology, a linear waterfall approach, coupled with effective risk mitigation, meant that the analysis had already been completed at this stage, making it possible to hit the ground running upon return.

However, one time during development, unrelated to the situation above, progress had to be halted due to the need for a re-evaluation of some of the requirements. This did not prove detrimental, however, as, due to effective risk management, additional time had been allotted for this eventuality (refer to **Appendix A**).

This does, however, pose the question of whether the chosen methodology was indeed the right choice and perhaps an iterative approach would have been more suitable.

6.2.1.3 SMART Objective 3 – Analyse and elicit requirements

This may be considered one of the weaker aspects of this project. Due to its relative simplicity, and the fact that the core requirements were offered by my supervisor, it was decided that the project would forego the formal requirements elicitation stage.

6.2.1.4 SMART Objective 4 – Develop the application which meets the requirements

The following section refers to items listed in **Figure 8. MoSCoW table**.

Requirements M1, M10, M11, and M12:

These requirements have all been met, the implementation details of which may be found in **Section 5.2**. Users can utilise any of the three methods for uploading data, and can modify data on-the-fly, which is instantly reflected in the graphs.

Requirements M2 – M8:

The tool offers the ability to analyse raw data and produce all the specified graphs. Some of these graphs, which are representative of the others, have been examined in **Section 5.3.** The representative few were chosen to avoid creation of an exhaustive list.

Nevertheless, it can be said definitively that the tool is able to visualise the data and present it in a meaningful way.

Requirement M9 – Usability

Jakob Nielsen's heuristics have been used to evaluate the usability of the system. Below is a table which lists each heuristic, and how it has been satisfied.

Heuristic	Satisfied by
Visibility of system status	Dynamic hiding of elements ensures that system status is always implicitly known. i.e On landing page, only the file input and instructions are visible, providing an intuitive sense of the system status.
Match between system and the real world	English, the most commonly used language on the internet, is used. Commonl phrases are used to instruct the user. File input uses a metaphor commonly used to indicate that files can be dropped.
User control and freedom	Text input control supports common word-processor functions such as ctrl + x and ctrl + z, to cut and undo, respectively. Furthermore, the system has been designed in such a way that no dead-ends exist. Users can always select another graph, or submit new data.
Consistency and standards	The page navigation is consistent with best-practice and standards. The page is generally layed out like a user would expect a web page to be.
Error prevention	The system will always try to generate a graph where possible. By doing so it leaves it up to the user to decide whether there is a problem with the data. this may seem wrong, but the user will have a better sense of whether the data being presented has been analysed correctly. If data displays wrong, they can adjust the data accordingly. This is much better than trying to tell the user that their information is wrong
Recognition rather than recall	There are no phase transitions in the system. All options are always clearly visible to the user, but guidance is clearly available if needed
Flexibility and efficiency of use	Word-processor functions (ctrl+a, ctrl+c, ctrl+x, ctrl+v, ctrl+z) are available to the more tech-savvy user.
Aesthetic and minimalist design	This is explored in detail in section 4.2
Help users recognize, diagnose, and recover from errors	Error messages are presented to the user in certain unavoidable cases, such as when a user attempts to draw a graph without any data submitted.
Help and documentation	Usage information, containing example data and general information, is available on the main page.

Figure 22. Heuristics satisfied

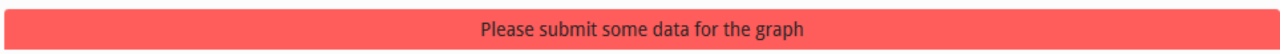


Figure 23. Error message

Requirements M13, M14, M15

According to Miller (1968) there are three response times to take into consideration when designing for human perception. A tenth of a second is the limit for giving the impression of an **instant** reaction, one second is the limit for maintaining uninterrupted thought, and ten seconds is the limit for maintaining the user's attention.

Thus, in order for the application to meet this requirement, all graph operations would have to fall between the first two thresholds; 0.1 and 1 second. In order to assess this, peformance benchmarks were performed on each graph operation, the results of which can be found in **Appendix D**.

This benchmarking found that every graph operation was performed in less than half a second, falling well within the threshold of maintaining uninterrupted thought.

Compatibility and compliance were tested using an online tool.

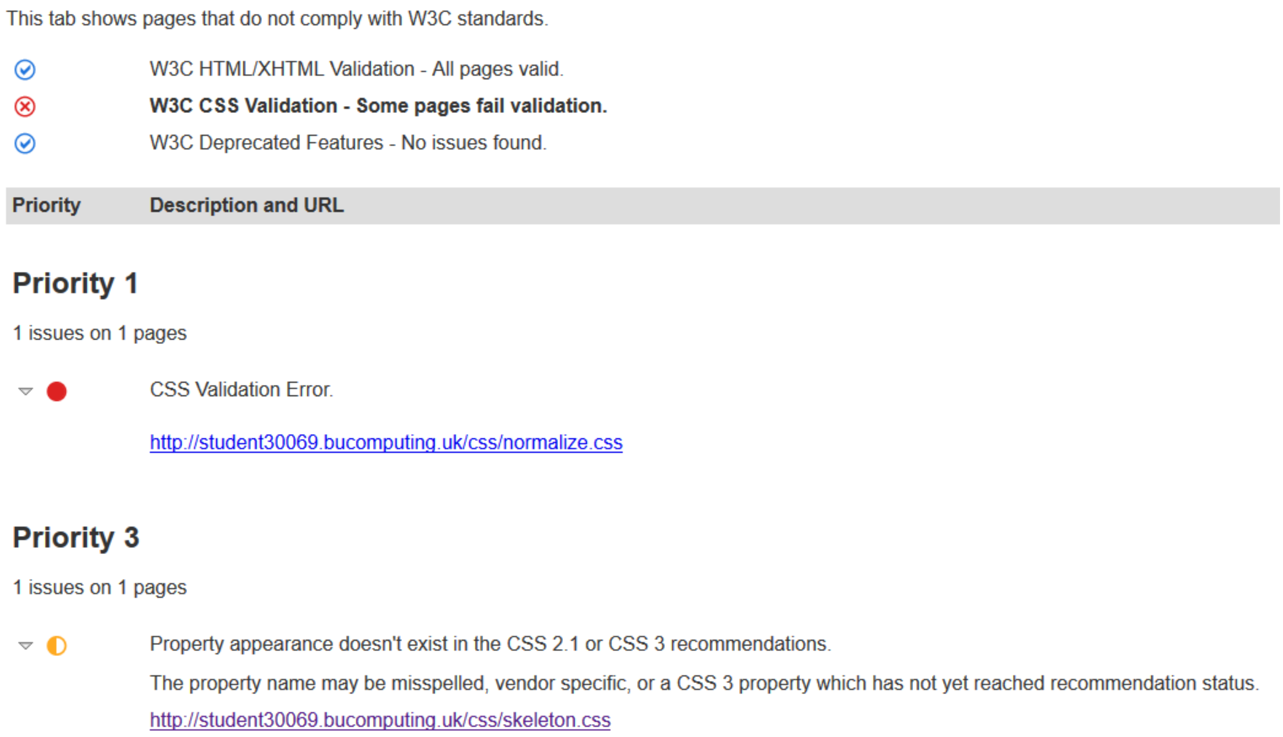


Figure 24. Standards compliance results

The site was found to be fully W3C standards compliant, with the exception of two issues found in .css files belonging to third-party libraries.

Finally, compatibility was tested using the same site. The results are shown below.

Browser	Internet Explorer				Edge	Firefox	Safari		Opera	Chrome	iOS			Android	
Version	8	9	10	11	15	54	≤ 9	10	45	59	≤ 8	9	10	≤ 3	4*
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	◕	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	◐	◐	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Key

- ● Missing content or functionality
- ◕ Major layout or performance problems
- ◐ Minor layout or performance problems

These results indicate that the site is compatible with all modern browsers, which account for over 95% of market share.

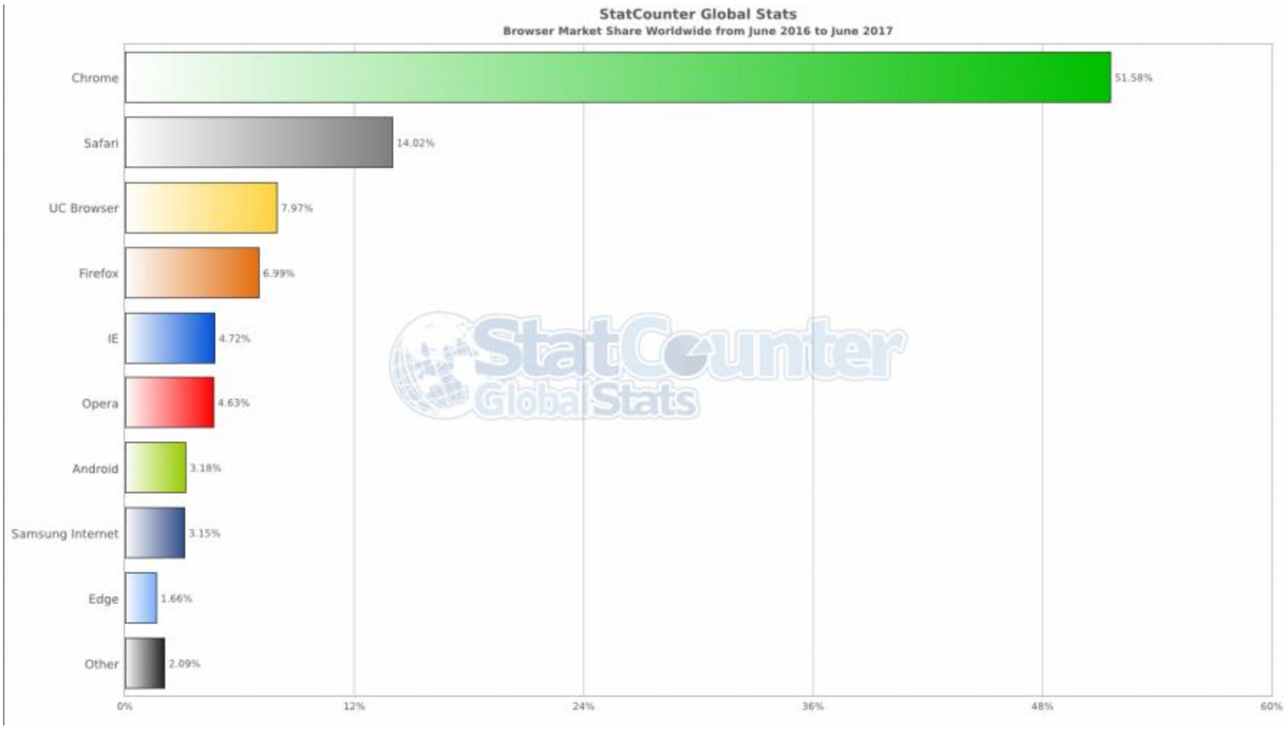


Figure 25. Global browser market share (statcounter.com)

Requirement M17 & M18

Noted in **Section 5.4**, and illustrated in **APPENDIX B** - Responsiveness Showcase, the tool is fully responsive, enabling use on most web-capable devices.

6.2.1.5 SMART Objective 5 – Evaluate and assess the extent to which the application solves the problem

This chapter attempts to achieve this objective by addressing each requirement, and each criterion for success. If done correctly, this should give a good indication of how well the requirements have been met, and whether problem has been solved.

6.3 FUTURE WORK

The future of this project involves collaboration; not only as a feature of the tool, but collaboration in its ongoing improvement.

The source code has been uploaded to a public GitHub repository at **https://github.com/Shaffan/UXVisualizer**. This establishes a platform for potential collaboration between students, instructors, and especially, those members of the target audience, usability researchers, who are interested in improving the tool.

Some future improvements include, adding additional metrics, as well as some utility graphs for quick comparison of data which has already been analysed; adding collaboration facilities which allow for easy sharing, similar to those available in tools such as JSFiddle and Pastebin.

Word count: 7709

7 REFERENCES

Azzam, T. & Evergreen, S., 2013. *J-B PE Single Issue (Program) Evaluation Ser. : Data Visualization, Volume 1 : Part 1, New Directions for Evaluation, Number 139 (1).* 1 ed. Somerset: Jossey-Bass .

Blaine Lawlor, K. & Hornyak, M. J., 2012. Smart Goals: How the application of smart goals can contribute to the achievement of student learning outcomes. *Developments in Business Simulation and Experiential Learning,* Volume 39, pp. 259-267.

Few, S., 2009. *Fundamental Differences in Analytical Tools: Exploratory, Custom, or Customizable.* [Online]
Available at:
http://www.perceptualedge.com/articles/visual_business_intelligence/differences_in_analytical_tools.pdf
[Accessed 20 June 2017].

Few, S., n.d. *Data Visualization for Human Perception.* [Online]
Available at: https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception
[Accessed 20 June 2017].

Halevy, A., 2009. *Google Fusion tables.* [Online]
Available at: https://research.googleblog.com/2009/06/google-fusion-tables.html
[Accessed 16 June 2017].

Hatton, S., 2008. *Choosing the "Right" Prioritisation Method.* Crawley, IEEE computer society.
Jiang, N. & Haibin, L., 2013. *Understand System's Relative Effectiveness Using Adapted Confusion Matrix,* Bournemouth: Springer.

Jokela, T., Iivari, N., Juha, M. & Minna, K., 2000. *The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11.* Rio de Janeiro, ACM, pp. 53-60.

Miller, R. B., 1968. *Response time in man-computer conversational transactions.* Poughkeepsie, ACM.

Nielsen, J., 1993. *Response Times: The 3 Important Limits.* [Online]
Available at: https://www.nngroup.com/articles/response-times-3-important-limits/
[Accessed 20 June 2017].

Nielsen, J., 1995. *10 Usability Heuristics for User Interface Design.* [Online]
Available at: https://www.nngroup.com/articles/ten-usability-heuristics/

Nielsen, J., 2001. *Success Rate: The Simplest Usability Metric.* [Online]
Available at: https://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/

Nielsen, J., 2001. *Usability Metrics.* [Online]
Available at: https://www.nngroup.com/articles/usability-metrics/
[Accessed 16 June 2017].

Nielsen, J., 2012. *Usability 101: Introduction to Usability.* [Online]
Available at: https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Nielsen, J. & Molich, R., 1990. *Heuristic Evaluation of User Interfaces.* Seattle, ACM.

Nielson, J., 2015. *The Characteristics of Minimalism in Web Design.* [Online]
Available at: https://www.nngroup.com/articles/characteristics-minimalism/
[Accessed 20 June 2017].

Pascal, B., 1656. *The Provincial Letters, by Blaise Pascal.* [Online]
Available at: https://ebooks.adelaide.edu.au/p/pascal/blaise/p27pr/part17.html
[Accessed 20 June 2017].

Royce, W. W., 1970. *Managing the development of large software systems.* s.l., TRW, pp. 1-9.

Sauro, J., 2010. *6 Things You Didn't Know About Heuristic Evaluations.* [Online]
Available at: https://measuringu.com/he/
[Accessed 20 June 2017].

Sauro, J., 2010. *Can You Use The SUS For Websites?.* [Online]
Available at: https://measuringu.com/sus-websites/
[Accessed 16 June 2017].

Sauro, J., 2010. *If You Could Only Ask One Question, Use This One..* [Online]
Available at: https://measuringu.com/single-question/
[Accessed 17 June 2017].

Sauro, J., 2010. *Performance Satisfaction And Perception Satisfaction.* [Online]
Available at: https://measuringu.com/test-task-sat/
[Accessed 17 June 2017].

Sauro, J., 2012. *10 Things To Know About The Single Ease Question (SEQ).* [Online]
Available at: https://measuringu.com/seq10/
[Accessed 17 June 2012].

Sauro, J. & Dumas, J. S., 2009. *Comparison of Three One-Question, Post-Task Usability ,* Denver: Oracle Corporation.

Sauro, J. & Kindlund, E., 2005. *A Method to Standardize Usability Metrics Into a Single Score.* Portland, s.n., pp. 401-409.

Scrum.org, 2016. *The Scrum Guide.* [Online]
Available at: http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf
[Accessed 30 June 2017].

Speicher, M., 2015. What is Usability? A Characterization based on ISO 9241-11 and ISO/IEC 25010. *CoRR,* Volume abs/1502.06792.

Thomas, N., 2015. *How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website.* [Online]
Available at: http://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/
[Accessed 16 June 2017].

usability.gov, 2017. *Visual Design Basics.* [Online]
Available at: https://www.usability.gov/what-and-why/visual-design.html
[Accessed 20 June 2017].

Usability.gov, n.d. *System Usability Scale.* [Online]
Available at: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html
[Accessed 16 June 2017].

We Are Social, 2017. *Digital in 2017: Global Overview.* [Online]
Available at: https://wearesocial.com/special-reports/digital-in-2017-global-overview

APPENDIX A

Risk	Description	Likelihood	Impact	Severity	Mitigation
Data loss	Data deleted accidentally or lost due to external factors. E.g. power failure	Low	High	Medium	Use source control with remote repo (GitHub) and push changes frequently
Insufficient time for research and development	Insufficient time allotted to research and development	Medium	Medium	Medium	Allow more time than required for these phases and update Gantt chart periodically
Unforeseen development problems	Emergence of unforeseen technical problems	Medium	Medium	Medium	
Limited time	Time is limited since it is split between the project and a job	Medium	Medium	Medium	Ensure that time is managed effectively, and project plan allows for movement
Sole developer	Success of the project relies on a single person; a single point of failure	Low	Medium	Medium	Seek assistance and advice from supervisor/online sources
Lack of expertise	Technical challenges encountered are beyond capabilities	Low	Medium	Medium	

APPENDIX B - RESPONSIVENESS SHOWCASE

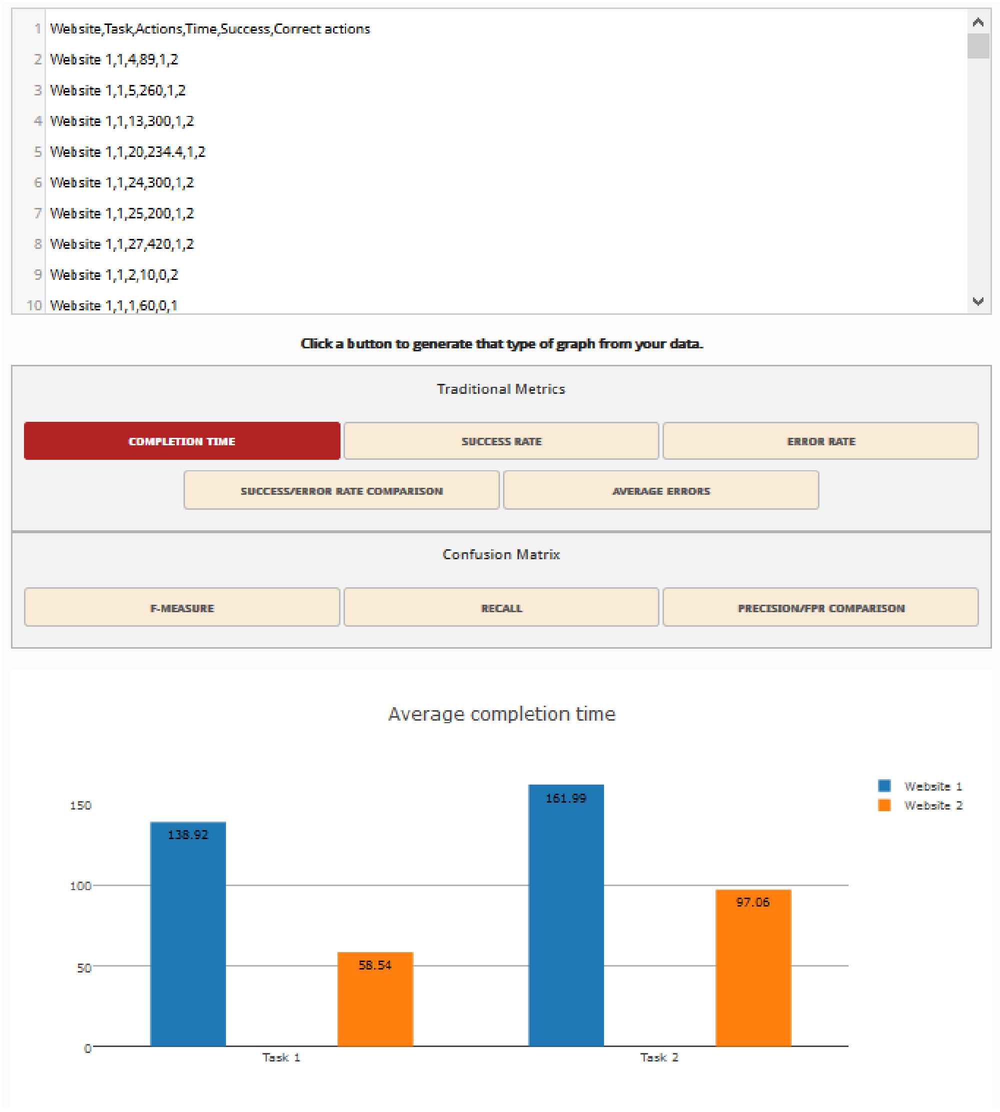


Figure 27. 1080 pixels wide

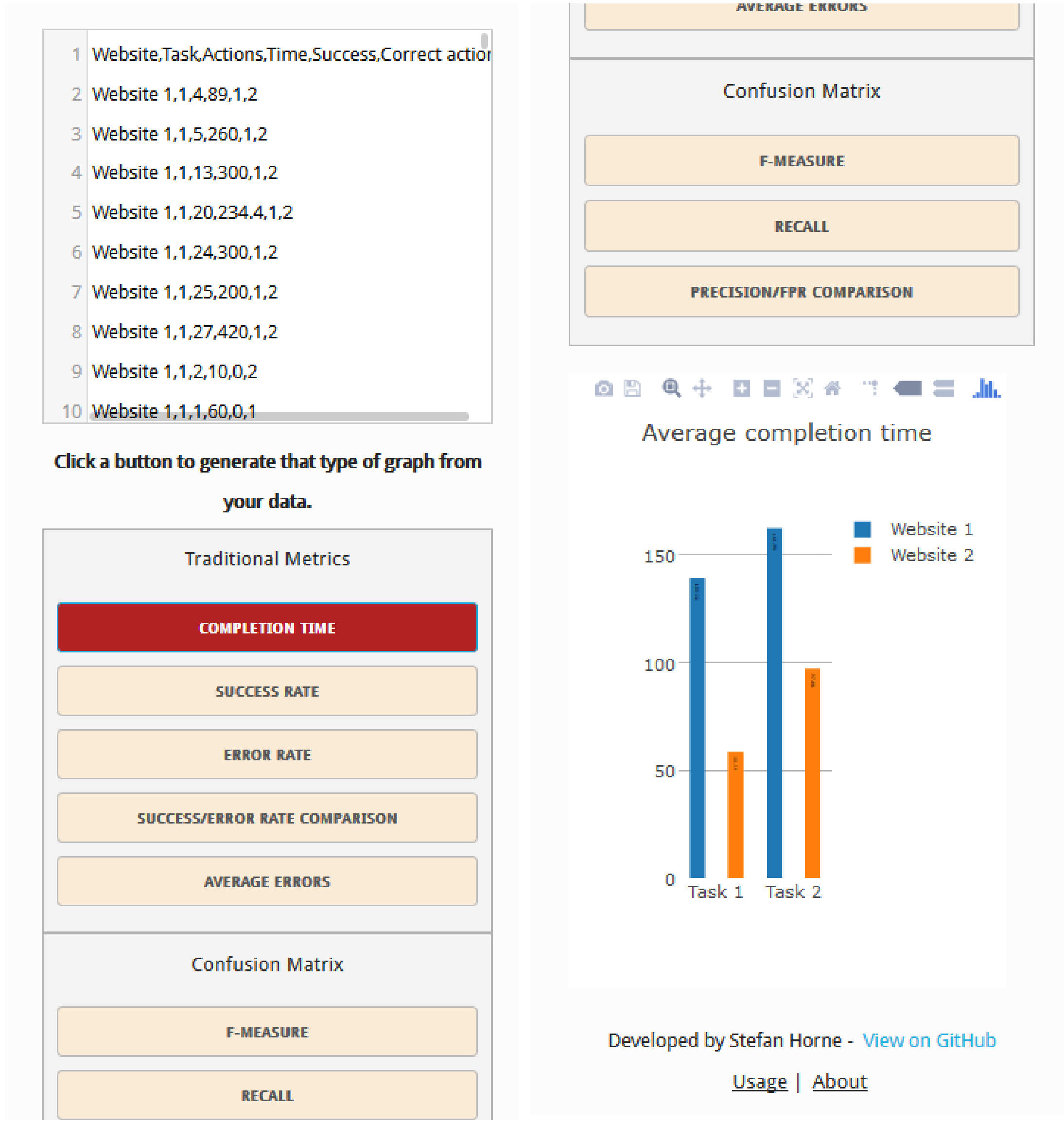


Figure 26. 360 pixels wide

APPENDIX C

Figure 28. Drag and drop supported

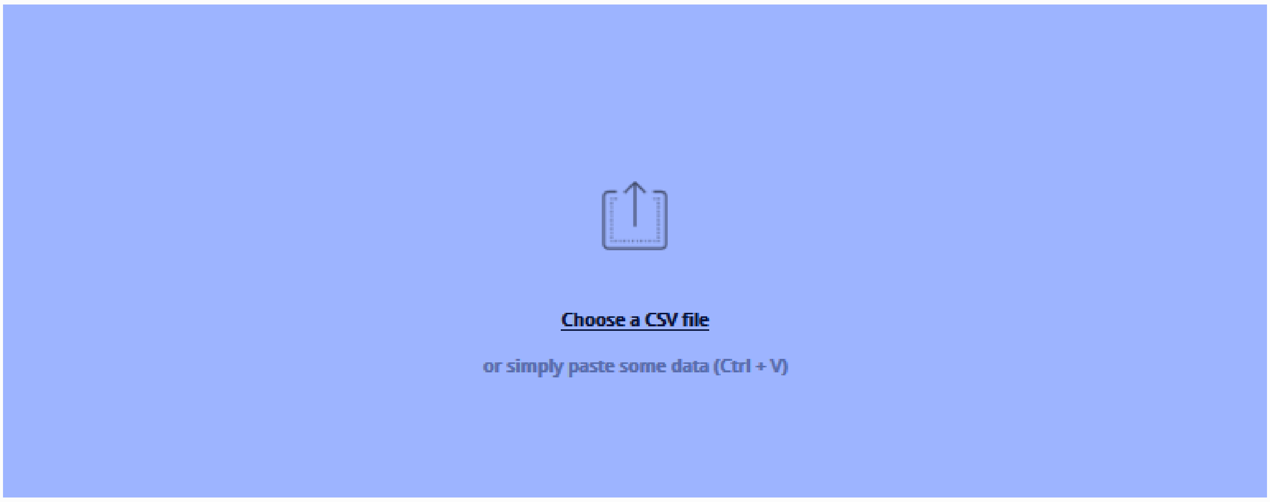


Figure 29. Drag and drop not supported

APPENDIX D

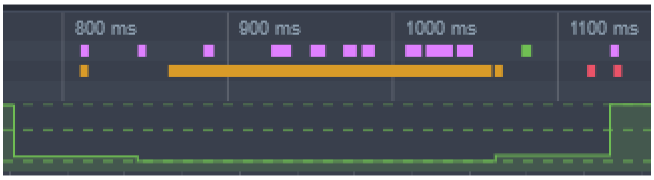


Figure 30. Completion time performance

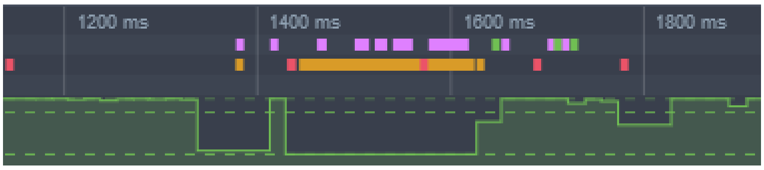


Figure 31. Success Rate performance

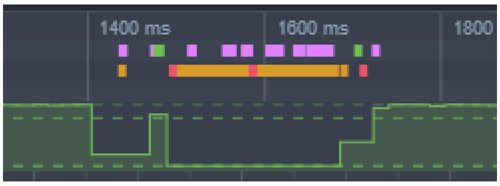


Figure 32. Error rate performance

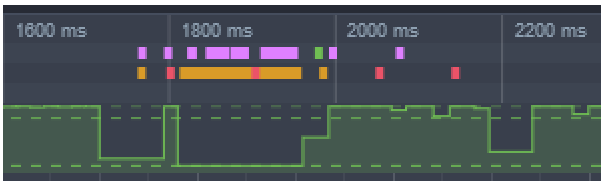


Figure 34. Average errors performance

Figure 33. Success/Error rate comparison performance

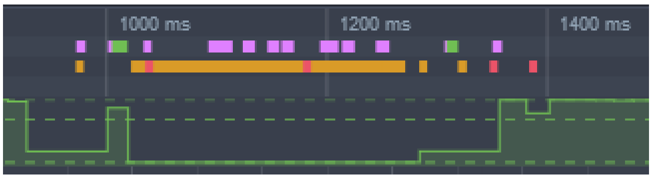


Figure 35. F-Measure performance

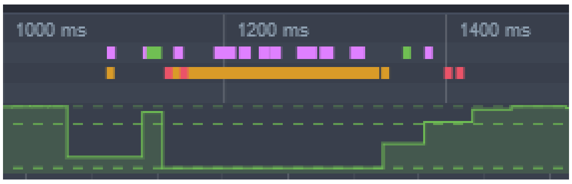


Figure 36. Recall performance

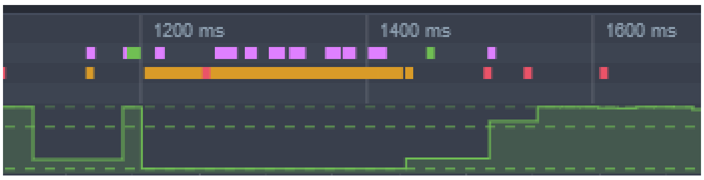


Figure 37. Precision/FPR Comparison performance

APPENDIX E – Project proposal Form

BU Computing Programmes 2016-2017

Undergraduate Project Proposal Form

Please refer to the **Project Handbook Section 4** when completing this form

Degree Title: Computing	**Student's Name:** **Stefan Horne**
	Supervisor's Name: **Nan Jiang**
	Project Title/Area: **Online data visualisation tool**

Section 1: Project Overview

1.1 Problem definition - use one sentence to summarise the problem:

Researchers and practitioners typically make use of improvised spreadsheets to capture experimental data and plot graphs using this data themselves, which can be time consuming, labour intensive and error-prone.

1.2 Background - please provide brief background information, e.g., client:

In order to retain users a website should always strive to create the best user experience possible. In an environment where users are likely to abandon a site in favour of another which is easier to use, quantifying the usability of a system and being able to illustrate this in a meaningful way is essential.

1.3 Aims and objectives – what are the aims and objectives of your project?

The aim of this project is to implement a tool which facilitates the following:

- effective communication with system stakeholders
- first-glance comparisons of multiple systems
- a clearer indication of the extent and severity of usability problems.

Objectives:

- Create an effective and user-friendly interface
- Ensure that the tool is as intuitive as possible to use
- Incorporate a new usability comparison method based on confusion matrix. [1]

BU Computing Programmes 2016-2017

Section 2: Artefact

2.1: What is the artefact that you intend to produce?

An online tool which accepts and evaluates experimental usability data, visualising it in the form of various graphs, each of which would relate to a quantifiable principle of usability and effectiveness.

2.2 How is your artefact actionable (i.e., routes to exploitation in the technology domain)?

The artefact being produced will be built and is therefore implicitly actionable.

Section 3: Evaluation

3.1 How are you going to evaluate your work?

In order for the tool to solve the problem it should meet the following requirements:

- The tool must be capable of providing visualisations of usability evaluations
- The tool must provide multiple input methods, including using a drag and drop interface
- The tool must support graphs based on, task completion time, success rate, average errors and error rate
- The tool must be able to provide graphs based on the confusion matrix
-

The tool must also be user friendly. This will be assessed during QAT with the aid of some helpful students

3.2 Why is this project honours worthy?

The creation of this tool, which will provide a new usability comparison method, can be highly beneficial, not only to researchers and practitioners, but also system stakeholders. Furthermore, the project involves the implementation of an artefact which is valid for the Computing course.

BU Computing Programmes 2016-2017

3.3 How does this project relate to your degree title outcomes?

This project is related to the following intended learning outcomes of the Computing course [2]:

- B3. Analyse, interpret, synthesise and evaluate information – analysis and evaluation of project requirements
- B4. Identify and solve problems
- B5. Select and apply appropriate design methods to the solution of problems
- C1. Retrieve, select and evaluate information from a variety of sources – research will be an ongoing process for the duration of the project
- C3. Design a solution to an IT problem
- C4. Implement a solution to an IT problem
- C6. Plan, monitor and evaluate the progress of an IT project
- D1. Structure and communicate ideas effectively in writing – project report
- D3. Work professionally as an individual to develop creative solutions to problems

3.4 How does your project meet the BCS Undergraduate Project Requirements?

This project attempts to solve the real-world problem outlined above. This problem is situated within the discipline of usability and user centred design, which forms an integral part of my computing degree. Furthermore, the project will test the analytical and technical skills I have developed on the course.

3.5 What are the risks in this project and how are you going to manage them?

The following risks have been identified:

Risk	Likelihood	Impact	Severity	Effect	Mitigation Actions
insufficient time due to part-time job	3	6	18	unable to implement all required functionality	Frequent progress checks
Use unfamiliar technology or tool	5	2	10	less time spent on "nice to haves"	take time to learn the technology/tool

BU Computing Programmes 2016-2017

Section 4: References

4.1 Please provide references if you have used any.

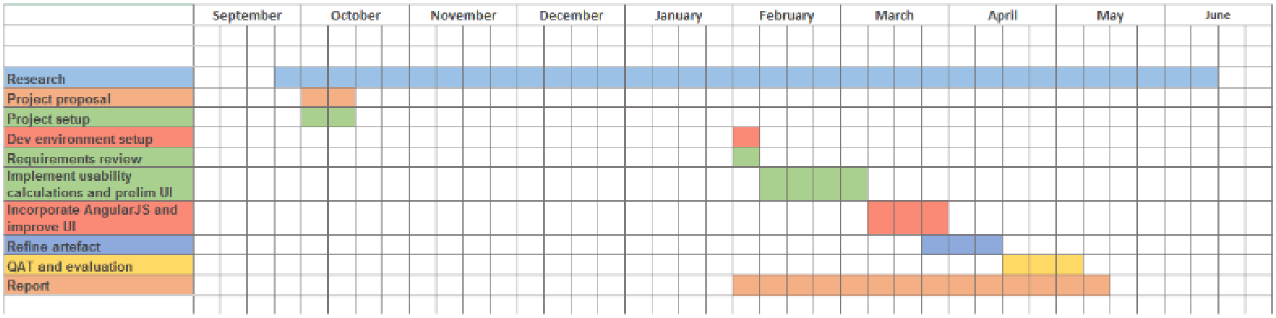
1. Nan Jiang, H. L., 2013. *Understand System's Relative Effectiveness Using Adapted Confusion Matrix,* Bournemouth: Springer.

2. https://intranetsp.bournemouth.ac.uk/progspecs/ug-computing-framework.pdf

Section 5: Ethics (please delete as appropriate)

5.1 Have you submitted the ethics checklist to your supervisor?	**No**
5.2 Has the checklist been approved by your supervisor?	**No**

Section 6: Proposed Plan (please attach your Gantt chart below)



APPENDIX F – RESEARCH ETHICS CHECKLIST

Bournemouth University

Research Ethics Checklist

Adapted for the use by Department of Computing and Informatics ONLY

1. Student Details

Name	Stefan Horne
School	Faculty of Science & Technology
Course	BSc Computing
Have you received external funding to support this research project?	No
Please list any persons or institutions that you will be conducting joint research with, both internal to BU as well as external collaborators.	

2. Project Details

Title	Creating a usable usability tool
Proposed Start Date	13-March-2016
Proposed End Date	
Supervisor	Nan Jiang
Summary (including detail on background methodology, sample, outcomes, etc.)	

The project aims to explore the viability of tool which aids in the area of usability research. This will be achieved by researching and developing a tool that accepts and evaluates experimental usability data and visualises this information.

A thorough investigation will be conducted into the problem context, which will aid in selecting an appropriate methodology. This will provide a framework for eliciting requirements and designing a system which is able to satisfy them.

Once the system has been designed and built, it will be evaluated, and the extent to which it has reached its goals will be discussed.

Research Ethics Checklist

Bournemouth University

Adapted for the use by Department of Computing and Informatics ONLY

3. External Ethics Review (Answer "Yes" go to 4, "No" go to 5)

Does your research require external review through the NHS National Research Ethics Service (NRES) or through another external Ethics Committee?	No

4. External Ethics Review Continued

Answered "Yes" to question 3 will conclude the BU Ethics Review so you do not need to answer the following questions. Note you will need to obtain external ethical approval before commencing your research.

5. Research Literature (Answer "Yes" go to 6, "No" go to 7)

Is your research solely literature based?	No

Research Ethics Checklist

Adapted for the use by Department of Computing and Informatics ONLY

6. Research Literature Continued (Either answer will conclude the review)

Will you have access to personal data that allows you to identify individuals OR access to confidential corporate or company data (that is not covered by confidentiality terms within an agreement or by a separate confidentiality agreement)?	Choose an item.
Describe how you will collect, manage and store the personal data (taking into consideration the Data Protection Act and the Data Protection Principles).	

7. Human Participants Part 1 (Answer "Yes" go to 8, "No" go to 12)

Will your research project involve interaction with human participants as primary sources of data (e.g. interview, observation, original survey)?	No

8. Human Participants Part 2 (Answer **any** "Yes" go to 9)

Does your research specifically involve participants who are considered vulnerable (i.e. children, those with cognitive impairment, those in unequal relationships—such as your own students, prison inmates, etc.)?	Choose an item.
Does the study involve participants age 16 or over who are unable to give informed consent (i.e. people with learning disabilities)? NOTE: All research that falls under the auspices of the Mental Capacity Act 2005 must be reviewed by NHS NRES.	Choose an item.

Bournemouth University

Research Ethics Checklist

Adapted for the use by Department of Computing and Informatics ONLY

Will the study require the co-operation of a gatekeeper for initial access to the groups or individuals to be recruited? (i.e. students at school, members of self-help group, residents of Nursing home?)	Choose an item.
Will it be necessary for participants to take part in your study without their knowledge and consent at the time (i.e. covert observation of people in non-public places)?	Choose an item.
Will the study involve discussion of sensitive topics (i.e. sexual activity, drug use, criminal activity)?	Choose an item.

9. Human Participants Part 2 Continued

Describe how you will deal with the ethical issues with human participants?

10. Human Participants Part 3 (Answer **any** "Yes" go to 11, **all** "No" go to 12)

Could your research induce psychological stress or anxiety, cause harm or have negative consequences for the participant or researcher (beyond the risks encountered in normal life)?	Choose an item.
Will your research involve prolonged or repetitive testing?	Choose an item.
Will the research involve the collection of audio materials?	Choose an item.
Will your research involve the collection of photographic or video materials?	Choose an item.
Will financial or other inducements (other than reasonable expenses and compensation for time) be offered to participants?	Choose an item.

Research Ethics Checklist

Adapted for the use by Department of Computing and Informatics ONLY

Please explain below why your research project involves the above mentioned criteria (be sure to explain why the sensitive criterion is essential to your project's success). Give a summary of the ethical issues and any action that will be taken to address these. Explain how you will obtain informed consent (and from whom) and how you will inform the participant(s) about the research project (i.e. participant information sheet). A sample consent form and participant information sheet can be found on the Research Ethics website.

12. Final Review

Will you have access to personal data that allows you to identify individuals OR access to confidential corporate or company data (that is not covered by confidentiality terms within an agreement or by a separate confidentiality agreement)?	No
Will your research take place outside the UK (including any and all stages of research: collection, storage, analysis, etc.)?	No
Please use the below text box to highlight any other ethical concerns or risks that may arise during your research that have not been covered in this form.	

APPENDIX G

Stefan_Horne_4525097_120717.pdf – An electronic copy of this document

Source Code – contains source code folders

Test Data – folder containing files which contain test data for convenient testing of the system

Instructions.txt