



UNIVERSITY OF HUDDERSFIELD

MENG GROUP PROJECT

Cryptic Crossword Solver

PROJECT MANAGEMENT

Authors:

Mohammad RAHMAN

Leanne BUTCHER

Stuart LEADER

Luke HACKETT

Supervisor:

Dr. Gary ALLEN

Examiner:

Dr. Sotirios BATSAKIS

Friday, 9th May 2014

Contents

| | | |
|----------|--|-----------|
| 1 | Project Management | 3 |
| 1.1 | Project Overview | 4 |
| 1.2 | Planning | 5 |
| 1.3 | People Involved | 14 |
| 1.4 | Ethics and Professionalism | 15 |
| 1.5 | Psychology of Software Development | 17 |
| 1.6 | Estimation and Costing | 18 |
| 1.7 | Allocation of Work | 20 |
| | Glossary of Terms | 21 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Project Timeline (21 st October 2013) | 6 |
| 1.2 | Gantt Chart for Term one (21 st October 2013) | 7 |
| 1.3 | Gantt Chart for Term two (21 st October 2013) | 8 |
| 1.4 | Project Timeline (30 th January 2014) | 9 |
| 1.5 | Gantt Chart for Term one (30 th January 2014) | 10 |
| 1.6 | Gantt Chart for Term two (30 th January 2014) | 11 |
| 1.7 | Final Gantt Chart of Project Summary (7 th May 2014) | 12 |
| 1.8 | Final Gantt Chart for Term two (7 th May 2014) | 13 |

Chapter 1

Project Management

The management of projects has evolved to become more complex and diverse as technology grows. One common characteristic shared by all projects is the projection of activities and ideas. The management of a project allows the explicit control of resources and budget as well as a coordinated path for a successful project (Lock, 1986).

This chapter consists of the management resources that were deployed within the project. The chapter includes the initial plans developed for the project as well as the accurate and concise plans of the project which were changed on a constant basis to apprehend the need of the clients, the users, and the development team of the project. Further more the psychology of the development is briefly discussed to illustrate the change in the life cycle of the project.

1.1 Project Overview

The project began on Monday, 30th September 2013 and ended on Friday, 9th May 2014 allowing the team 32 weeks in order to plan, develop and deploy the project as well as allowing sufficient time to produce the academic and technical reports to accompany the product.

The planning of the project has been produced to incorporate all the academic and technical deliverables. The major academic deliverables are:

- Project Proposal
- Presentation of mid-point progress
- Submission of reports
- Final product demonstration

and the project milestone deliverables are:

- Scope
- Software Requirements
- Design
- Development
- Unit Testing
- Integration Testing
- Documentation
- Deployment

The following sections within this chapter show the progress of the project where each of the deliverables were met accompanied by their predicted completion dates to the actual dates when they were completed. There have been a number of factors that have impacted the plans and these will be discussed in the following sections.

1.2 Planning

This section contains the project plans which have been developed in Microsoft Project 2010. There are three sub sections which define the initial, interim and final progress of the project. Jalote (2005) refers to software engineering as two domains:

1. Student Systems - Programs that people build to illustrate something or for hobby.
2. Industrial Strength Software - Software that can lead to significant direct or indirect loss.

A software project that is planned thoroughly is a more successful project than a project that is developed through extreme programming. The Cryptic Crossword project initially started through the Waterfall life cycle model and gradually grew to become an Iterative model. Therefore the project plans had to be updated constantly to reflect the change in which the project was being undertaken.

Although the overall project is developed in the Waterfall model due to the time constraints applied by University deadlines the project consists of an iterative development outside of the deliverables which are to be produced for academic purposes.

1.2.1 Timeline



Figure 1.1: Project Timeline (21st October 2013)

Figure 1.1 shows the initial time scale that was drawn up while planning the project. This was produced to show the major deliverables and their dates. In the next section the gantt Chart that was produced with all the initial tasks to be completed are shown.

1.2.2 Initial Gantt Chart

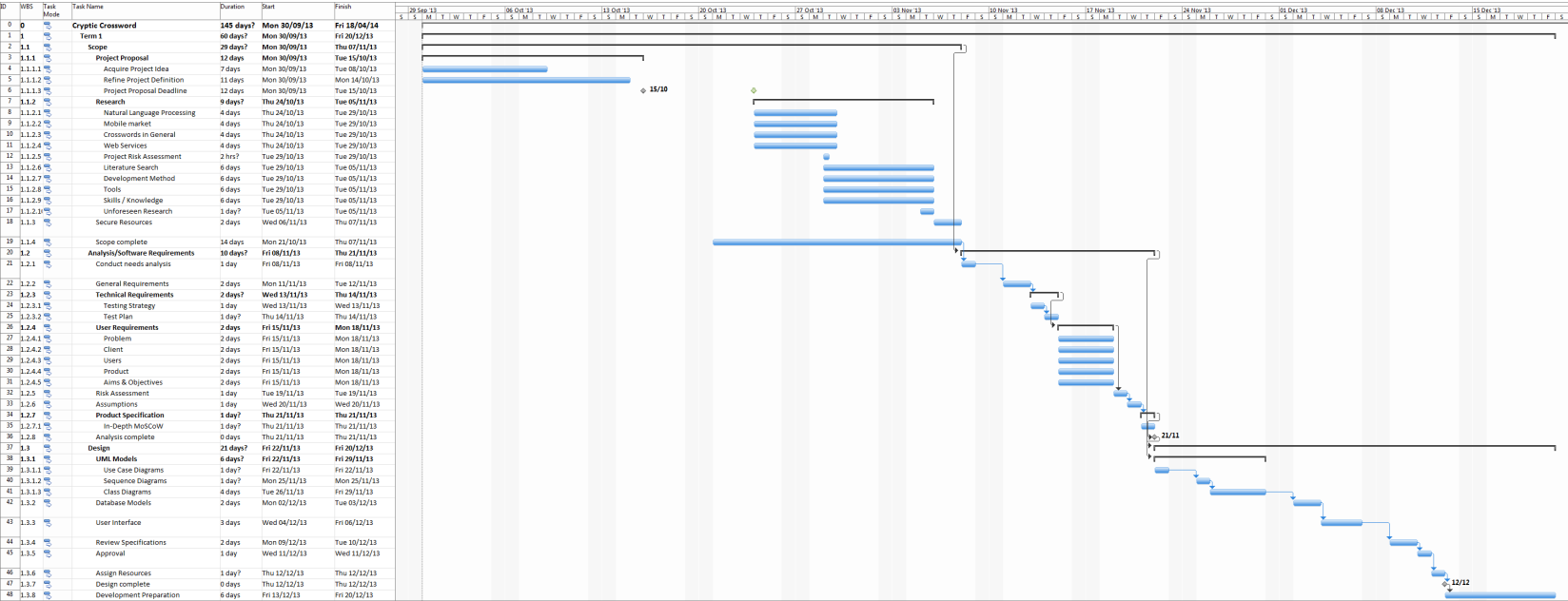


Figure 1.2: Gantt Chart for Term one (21st October 2013)

1.2.3 Timeline Re-Visited

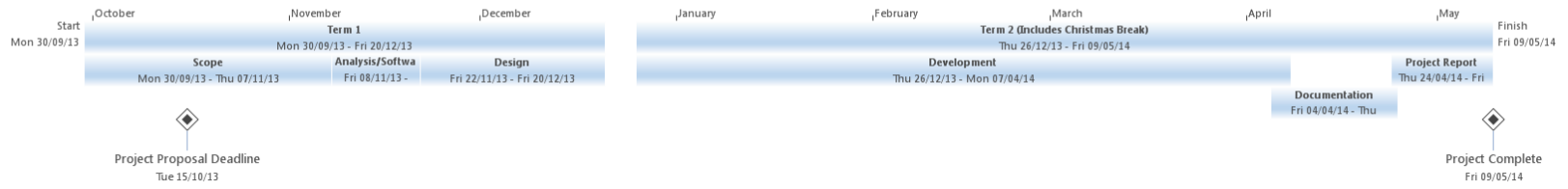


Figure 1.4: Project Timeline (30th January 2014)

6

By the end of term 1 the members of the group had accomplished all the back end research and acquired all the materials to start development of the project. At this point of the project new specifications were created and the research had implied that to develop such a system the best approach would be to take an iterative approach where some code can be developed and fully tested and deployed and then revisited to improve an another iteration. To ensure that a working system was developed and comprehensively tested it was decided that there was only sufficient time for three iterations. Although there would be three iterations it would allow a product to be submitted by the end of the second term and in the case the project ended early further iterations could be added in. Figure 1.4 represents the new timeline which only focuses a change in the second term of the academic year and future tasks.

1.2.4 Interim Gantt Chart

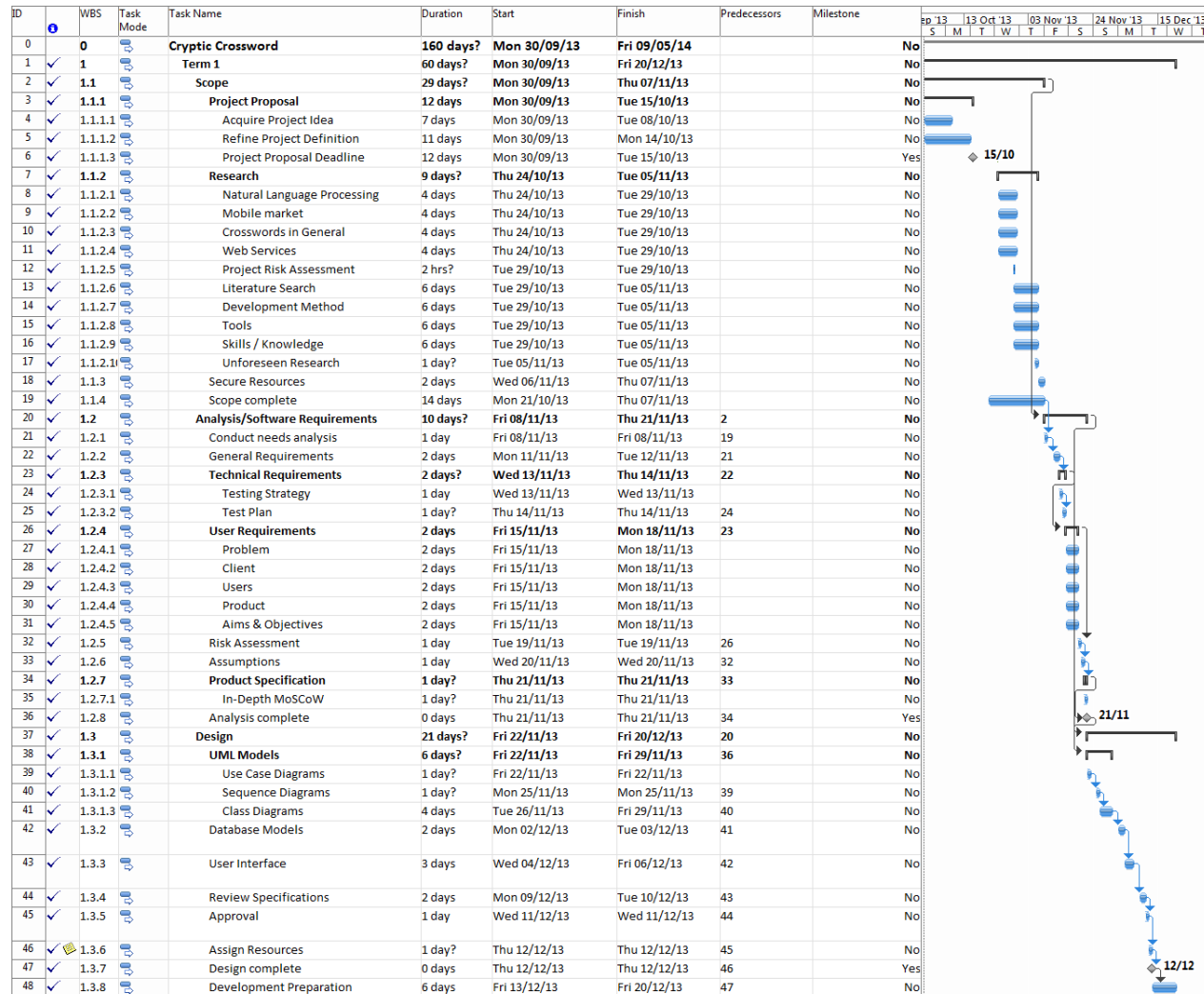


Figure 1.5: Gantt Chart for Term one (30th January 2014)

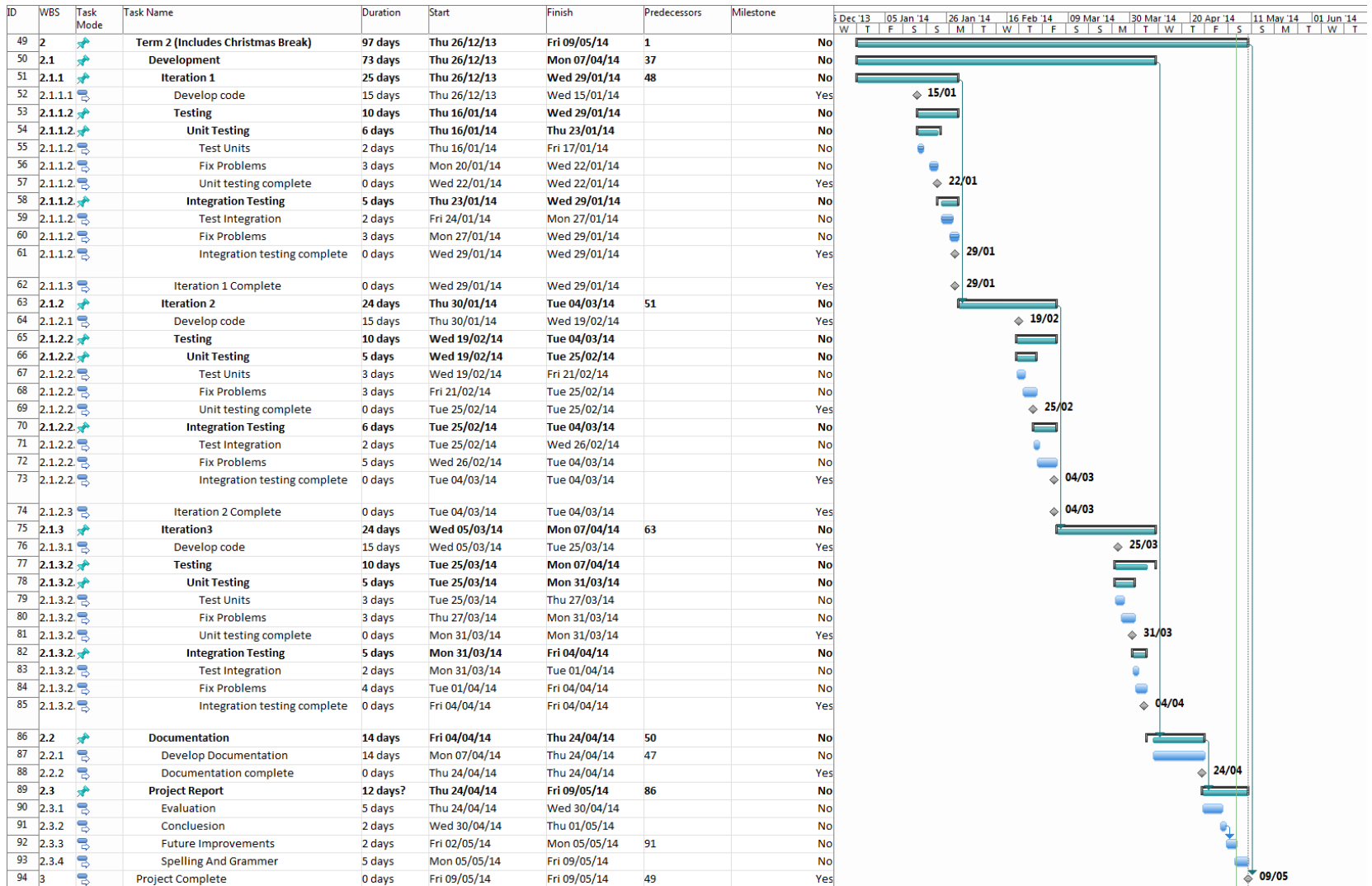
Figure 1.6: Gantt Chart for Term two (30th January 2014)

Figure 1.6 shows the flow of the new tasks that incorporate the Iterative development model and the remaining tasks of the second term. The Work Breakdown Structure(WBS) ensures that tasks are started only when tasks are completed and have a must finish on constraint to ensure that work is completed on time. Weekends have been excluded from the work hours to allow team members to work on other modules from their university studies as well as time off for social activities.

1.2.5 Final Gantt Chart

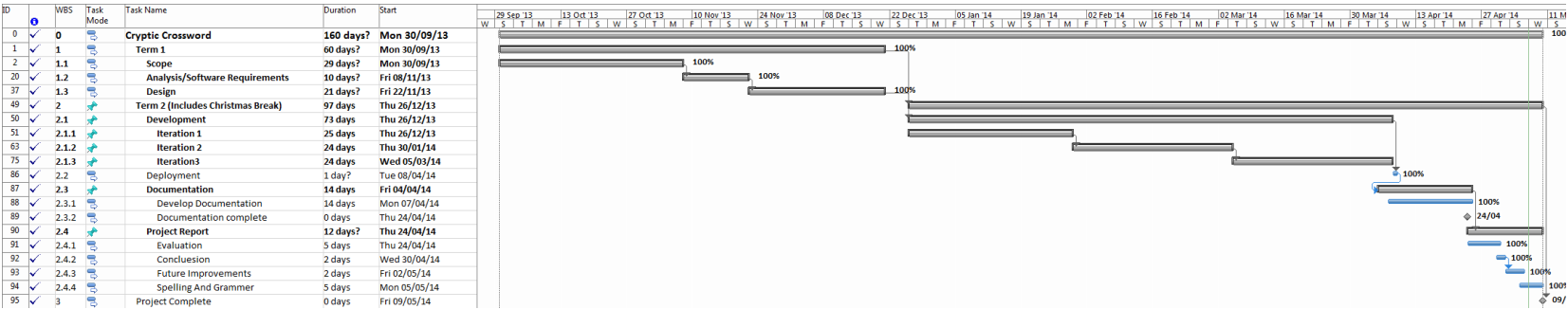


Figure 1.7: Final Gantt Chart of Project Summary (7th May 2014)

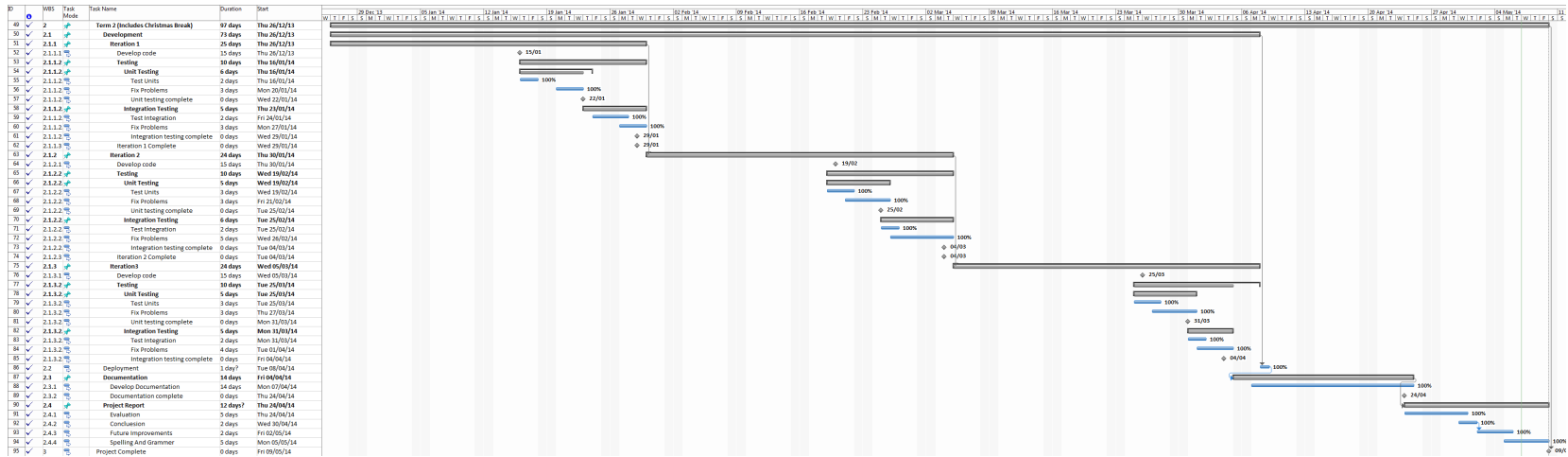


Figure 1.8: Final Gantt Chart for Term two (7th May 2014)

Figure 1.8 is the final gantt chart for the second term. All tasks have been completed on time and a product has been deployed on the web server. Improvements have been identified but there is not enough time for a fourth iteration. Figure 1.7 represents an overview of the two terms and the milestones completed.

1.3 People Involved

A number of project stakeholders have already been outlined with the initial problem analysis. Within this section the stakeholders that are involved with the project will be redefined upon a more formal basis.

The ‘group’ will be actively referred to throughout the project, and therefore the group will be defined. The group will have four members throughout the project, and these members are:

- Leanne Butcher
- Luke Hackett
- Stuart Leader
- Mohammad Rahman

The project group is based largely upon democratic discussions and decisions, however to ensure that group deadlocks do not occur Mohammad has been selected as the group leader.

The group will be assigned a project supervisor directly from the University. The project supervisor will help to deal with an technical and non-technical queries, whilst also ensuring the group remains upon a general focus. The project supervisor may also suggest features and improvements to work that the group managed to produce. The group’s project supervisor is Dr. Gary Allen.

A weekly meeting will take place between all group members and the project supervisor to discuss all aspects of the project. This includes (but is not limited to) project issues, software development issues, research findings, possible improvements and code reviews.

As the project is a formal assessment, the project assessor and moderator will be classified as indirect stakeholders within the project. Dr Sotirios Batsakis and Dr Colin Venters have been assigned the roles of project assessor and project moderator respectfully. Both the project assessor and moderator will have a limited amount of contact time with the group, but will be present at formal presentations that are scheduled throughout the year.

The project formally does not have any clients associated with the work, however Dr Hugh Osborne has offered to fill a client-like role within the project. Hugh has a keen interest in cryptic crosswords and the problem area that the group intends to solve. The role of the client for the group project will be to input ideas and potential requirements which Hugh, as an experienced solver of cryptic crosswords, would consider to be necessary.

1.4 Ethics and Professionalism

In software engineering ethics is the study of right and wrong in relation to human actions (Bott, 2001). The participants of the group project have looked at all aspects of ethical thinking as well as professionalism throughout the course of the project. The group members adopted the ethical codes in order to demonstrate their professionalism within the academic institution and to re-enforce these skills in the working industry. The seven main functions of ethical thinking that were looked at are:

- Professionalism
- Protection of group interests
- Etiquette and inspiration
- Education
- Enforcement
- Principles, ideals and rules
- Rights - Obligations and duties as well as the rights of members and the obligation of the professional body to its members.

1.4.1 Professional Codes of Conduct

As students of an engineering degree each participant is a member of the British Computing Society(BCS). As a member of this professional body each member adheres to the rules which cover four main areas:

- Public interest
- Professional competence and integrity
- Duty to relevant Authority
- Duty to the profession

BCS - The Chartered Institute for IT (2014)

For the purpose of the project the duties of each member of the team has been to ensure that the all work, research has been conceptual, empirical and fully technical. As well as sticking to the BCS code of conduct the team members are aware of the Student Handbook at the University of Huddersfield and are obliagted to follow these rules and regualtions set out in this handbook.

1.4.2 Intellectual Property Rights

The cryptic crossword solver is an academic application which is formally a prototype to a real world problem. The decision has been taken by each team member that there will be no claims for rights to the application and its documents and after the marks have been awarded from the examining board the application is to be published as an open source project.

The project does not contain any confidential information which are internal to the software engineering team and external to the clients. The members of the group have also adhered to the Copyright, Patents and Designs Act 1988 and ensured that all material used within the project is of their own and permission has been granted from the relevant authorities such as the Guardian newspaper to ensure that there are no conflicts in the code, design and documentation of the project.

1.4.3 Computer Misuse

The members of the team are aware of the Computer Misuse Act 1990 and have followed the correct procedures to ensure that all material that has been generated has been lawfully generated under the correct and legal environments of the University grounds and at the students home. Again mentioning the Guardian Newspaper and the data that was collected from their website the team members ensured to make sure that permission was obtained and this can be found in the appendix. To provision the project the credentials for access to the Helios server and the Amazon EC2 server have been maintained by the all users ensuring not to share this information to anyone other than the team members them selfs.

1.5 Psychology of Software Development

Jones (1986) states that the psychology of software development is often referred to as the psychology of programming which reflects its primary orientation to the coding phenomena that constitute rarely more than 15 percent of a large project effort. The need to share information on a software project by people is not enough and the communication needed to support this lack in many organisations. Larger teams and more members are required to produce systems which are rapidly growing in the current market in order to develop these systems quickly (Boehm, 1981). The ability to manage large software systems not only consists of empirical research but depends on the social and organisational aspects of the team.

The team has looked at various aspects in regards to the structure of the organisation and found the most effective way to structure the team is to have a democratic layout in order to allow all members sufficient participation. Not only have the members looked at the organisation structure but also the technical ability of each member in order to be served tasks which will push members to gain new knowledge or become an expert in that field.

The path from a programmer to the end user can be very long and this is one of the many reasons in which the team took an iterative approach during development to ensure that there is a product by the end of the academic year allowing room for extra functionalities and enhancements to be added in the future. Although the initial Waterfall model suited well into the format of the academic year the requirements produced re-inforced that an iterative approach would be better for development to ensure that there were working and completed requirements. Other models were looked at and considered such as agile development and spiral but the main reason to why Iterative was chosen because it contained characteristics of the waterfall model and with the minimal amount of time the team had this would best fit in to the team structure.

1.6 Estimation and Costing

The costs of a software development project are primarily the costs of the effort involved. The three main areas which correspond to the cost of a software project include:

- Hardware and software costs including maintenance
- Travel and training costs
- Effort costs (Costs of paying software engineers).

The main cost that will be discussed in this section is the effort costs as this is the most adequate cost that can be calculated for this project. In order to estimate the project costs the volume of the software must be calculated. This can be done by calculating number of Source lines of Code(SLOC) also calculated as Thousands of lines of Code (KLOC). The most common estimating metric is the Constructive Cost Model (COCOMO) and the new COCOMO II. Function points is another metrics that can be used to estimate the project.

COCOMO was originally developed by Boehm (1981). It has since been re-developed as COCOMO II (Boehm et al., 2000).

The basic COCOMO formula is:

Effort Applied (Person Months)

$$E = a(KLOC)^b$$

Development Time(Months)

$$DT = cE^d$$

People Required

$$P = E/D$$

| Mode | a | b | c | d |
|---------------|-----|------|-----|------|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Table 1.1: Basic COCOMO

The cryptic solver best suits the Organic mode because the team is small consisting of four members who have experience working with requirements. On this basis and the time constraint factors the project can be estimated. The development time allocated for the project is 3.25 months which is the 98 days calculated from the project plan. This has been considered to exclude time for other academic deliverables and to mainly focus upon

the actual development. There fore according to Boehm et al. (2000) the project can be estimated as followed:

Effort Applied (Person Months)

$$E = 2.4(5)^{1.05} = 13$$

Development Time (Months)

$$DT = 2.5(13)^{0.38} = 6.5$$

People Required

$$P = 13/6.5 = 2$$

Now to evaluate this against the number of actual team members we can do the following.

Development Time (Months)

$$DT = 2.5(13)^{0.38}/2 = 3.25$$

People Required

$$P = 13/3.25 = 4$$

The estimated number of lines of code SLOC is therefore 5(thousand). Upon completion of the project the COCOMO model was applied to see how closely the original estimation was made. In total there are 5721 SLOC and therefore applying the same algorithms as before produced the following results.

Effort Applied (Person Months)

$$E = 2.4(6)^{1.05} = 15$$

Development Time (Months)

$$DT = 2.5(13)^{0.38}/2 = 3.5$$

People Required

$$P = 15/3.5 = 4$$

1.7 Allocation of Work

Glossary of Terms

The following section contains a glossary with the meanings of all names, acronyms, and abbreviations used by the stakeholders.

| Term/Acronym | Definition |
|--------------|---|
| The Guardian | A newspaper with a website featuring cryptic crosswords |
| Blackberry | A mobile phone platform by Blackberry |
| iOS | A mobile phone platform by Apple |
| Android | A mobile phone platform by Google |
| NLP | Natural Language Processing |
| SRS | Software Requirements Specification |
| App | Short for application |
| | |
| | |
| | |

Bibliography

- BCS - The Chartered Institute for IT (2014). Code of conduct. Retrieved from <http://www.bcs.org/codeofconduct>.
- Boehm, B. (1981). *Software Engineering Economics*. Prentice Hall, New Jersey.
- Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., and Reifer, D. J. (2000). *Software Cost Estimation with COCOMO II*. Prentice Hall, New Jersey.
- Bott, F. (2001). *Professional issues in software engineering*. Taylor And Francis Ltd, London.
- Jalote, P. (2005). *Planning a Software Project*. Springer US, Boston, MA.
- Jones, C. (1986). *Programming Productivity*. McGraw Hill, New York.
- Lock, D. (1986). *Project management*. Wildwood House, Aldershot.