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

List of Figures

List of Tables

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, 30^{th} September 2013 and ended on Friday, 9^{th} 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
- 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 (21^{st} 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

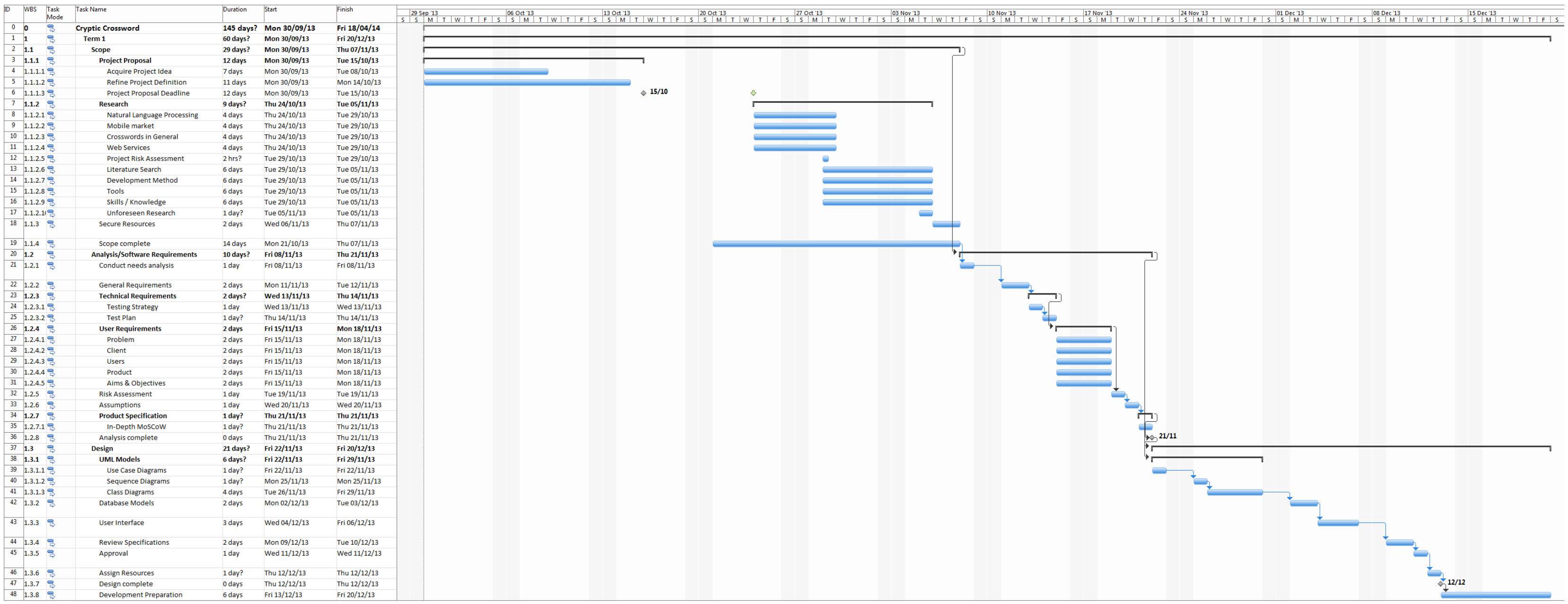


Figure 1.2: Gantt Chart for Term one (21^{st} October 2013)

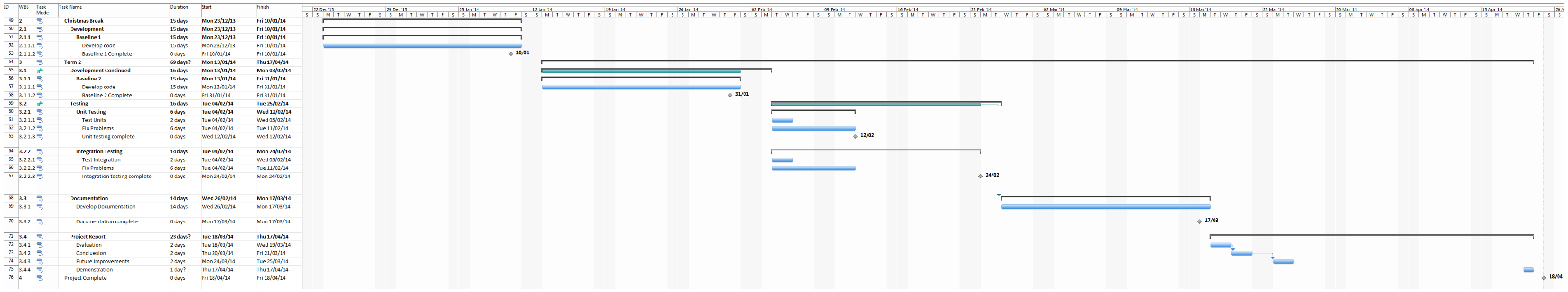


Figure 1.3: Gantt Chart for Term two (21^{st} October 2013)

Figure 1.2 and 1.3 represent the initial life cycle of the project. The Gantt charts have been developed using the Waterfall model to allow the team members to work in flowing rhythm to ensure that all deliverables for the academic report are met. This includes the research and design for the project. It was at this point the team had a clearer understanding of the life cycle the project was to incorporate.

1.2.3 Timeline Revisited

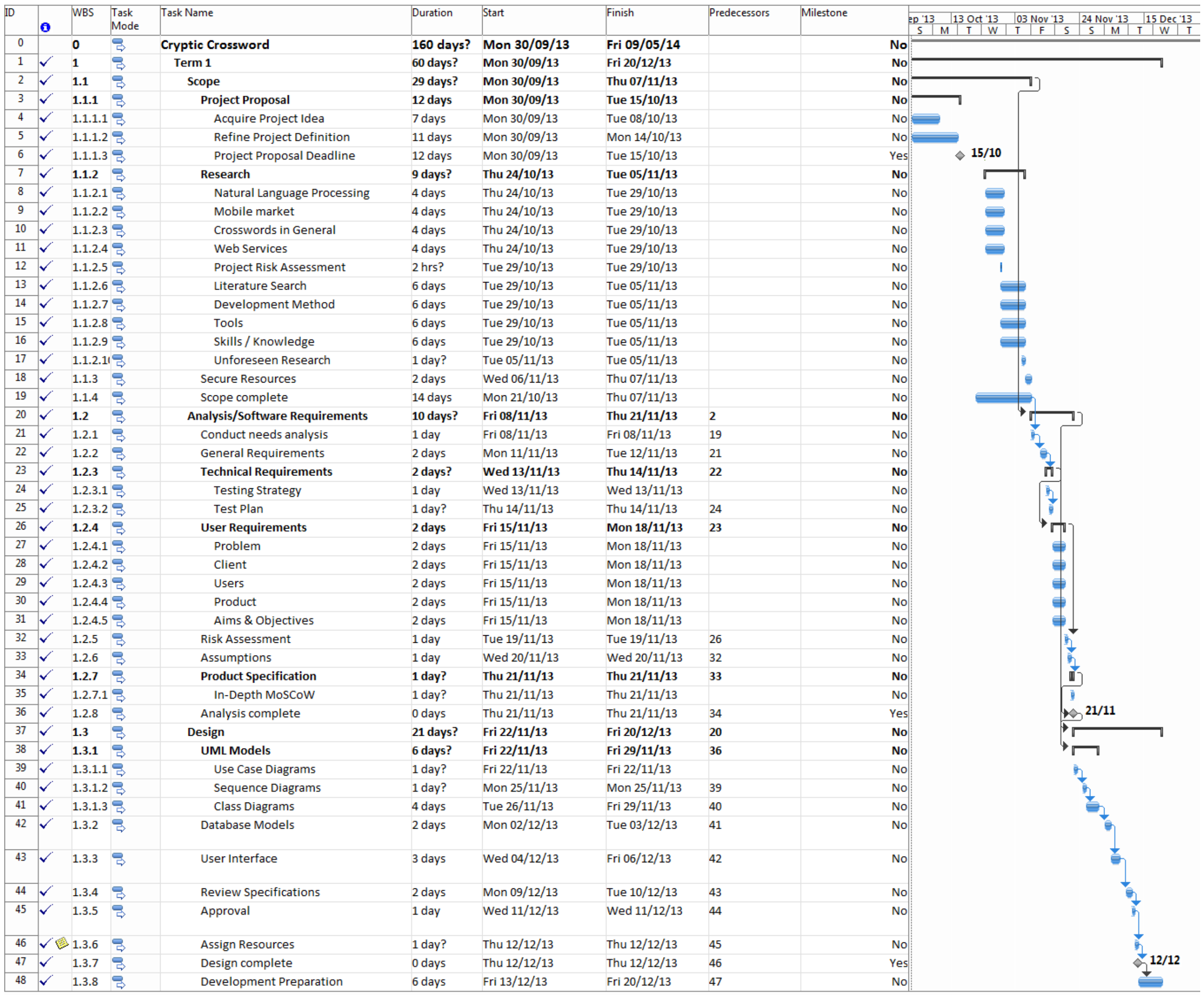
Figure 1.4: Project Timeline (30^{th} January 2014)

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 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 code can be developed, fully tested, 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. If a case arose where by the project ended earlier than expected, further iterations could be added in.

Figure 1.4 represents the new timeline which shows a change in the second term of the academic year by adding known future tasks.

1.2.4 Interim Gantt Chart



ID		WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Milestone
0		**0**		**Cryptic Crossword**	**160 days?**	**Mon 30/09/13**	**Fri 09/05/14**		**No**
1	✓	**1**		**Term 1**	**60 days?**	**Mon 30/09/13**	**Fri 20/12/13**		**No**
2	✓	**1.1**		**Scope**	**29 days?**	**Mon 30/09/13**	**Thu 07/11/13**		**No**
3	✓	**1.1.1**		**Project Proposal**	**12 days**	**Mon 30/09/13**	**Tue 15/10/13**		**No**
4	✓	1.1.1.1		Acquire Project Idea	7 days	Mon 30/09/13	Tue 08/10/13		No
5	✓	1.1.1.2		Refine Project Definition	11 days	Mon 30/09/13	Mon 14/10/13		No
6	✓	1.1.1.3		Project Proposal Deadline	12 days	Mon 30/09/13	Tue 15/10/13		Yes
7	✓	**1.1.2**		**Research**	**9 days?**	**Thu 24/10/13**	**Tue 05/11/13**		**No**
8	✓	1.1.2.1		Natural Language Processing	4 days	Thu 24/10/13	Tue 29/10/13		No
9	✓	1.1.2.2		Mobile market	4 days	Thu 24/10/13	Tue 29/10/13		No
10	✓	1.1.2.3		Crosswords in General	4 days	Thu 24/10/13	Tue 29/10/13		No
11	✓	1.1.2.4		Web Services	4 days	Thu 24/10/13	Tue 29/10/13		No
12	✓	1.1.2.5		Project Risk Assessment	2 hrs?	Tue 29/10/13	Tue 29/10/13		No
13	✓	1.1.2.6		Literature Search	6 days	Tue 29/10/13	Tue 05/11/13		No
14	✓	1.1.2.7		Development Method	6 days	Tue 29/10/13	Tue 05/11/13		No
15	✓	1.1.2.8		Tools	6 days	Tue 29/10/13	Tue 05/11/13		No
16	✓	1.1.2.9		Skills / Knowledge	6 days	Tue 29/10/13	Tue 05/11/13		No
17	✓	1.1.2.1(		Unforeseen Research	1 day?	Tue 05/11/13	Tue 05/11/13		No
18	✓	1.1.3		Secure Resources	2 days	Wed 06/11/13	Thu 07/11/13		No
19	✓	1.1.4		Scope complete	14 days	Mon 21/10/13	Thu 07/11/13		No
20	✓	**1.2**		**Analysis/Software Requirements**	**10 days?**	**Fri 08/11/13**	**Thu 21/11/13**	**2**	**No**
21	✓	1.2.1		Conduct needs analysis	1 day	Fri 08/11/13	Fri 08/11/13	19	No
22	✓	1.2.2		General Requirements	2 days	Mon 11/11/13	Tue 12/11/13	21	No
23	✓	**1.2.3**		**Technical Requirements**	**2 days?**	**Wed 13/11/13**	**Thu 14/11/13**	**22**	**No**
24	✓	1.2.3.1		Testing Strategy	1 day	Wed 13/11/13	Wed 13/11/13		No
25	✓	1.2.3.2		Test Plan	1 day?	Thu 14/11/13	Thu 14/11/13	24	No
26	✓	**1.2.4**		**User Requirements**	**2 days**	**Fri 15/11/13**	**Mon 18/11/13**	**23**	**No**
27	✓	1.2.4.1		Problem	2 days	Fri 15/11/13	Mon 18/11/13		No
28	✓	1.2.4.2		Client	2 days	Fri 15/11/13	Mon 18/11/13		No
29	✓	1.2.4.3		Users	2 days	Fri 15/11/13	Mon 18/11/13		No
30	✓	1.2.4.4		Product	2 days	Fri 15/11/13	Mon 18/11/13		No
31	✓	1.2.4.5		Aims & Objectives	2 days	Fri 15/11/13	Mon 18/11/13		No
32	✓	1.2.5		Risk Assessment	1 day	Tue 19/11/13	Tue 19/11/13	26	No
33	✓	1.2.6		Assumptions	1 day	Wed 20/11/13	Wed 20/11/13	32	No
34	✓	**1.2.7**		**Product Specification**	**1 day?**	**Thu 21/11/13**	**Thu 21/11/13**	**33**	**No**
35	✓	1.2.7.1		In-Depth MoSCoW	1 day?	Thu 21/11/13	Thu 21/11/13		No
36	✓	1.2.8		Analysis complete	0 days	Thu 21/11/13	Thu 21/11/13	34	Yes
37	✓	**1.3**		**Design**	**21 days?**	**Fri 22/11/13**	**Fri 20/12/13**	**20**	**No**
38	✓	**1.3.1**		**UML Models**	**6 days?**	**Fri 22/11/13**	**Fri 29/11/13**	**36**	**No**
39	✓	1.3.1.1		Use Case Diagrams	1 day?	Fri 22/11/13	Fri 22/11/13		No
40	✓	1.3.1.2		Sequence Diagrams	1 day?	Mon 25/11/13	Mon 25/11/13	39	No
41	✓	1.3.1.3		Class Diagrams	4 days	Tue 26/11/13	Fri 29/11/13	40	No
42	✓	1.3.2		Database Models	2 days	Mon 02/12/13	Tue 03/12/13	41	No
43	✓	1.3.3		User Interface	3 days	Wed 04/12/13	Fri 06/12/13	42	No
44	✓	1.3.4		Review Specifications	2 days	Mon 09/12/13	Tue 10/12/13	43	No
45	✓	1.3.5		Approval	1 day	Wed 11/12/13	Wed 11/12/13	44	No
46	✓	1.3.6		Assign Resources	1 day?	Thu 12/12/13	Thu 12/12/13	45	No
47	✓	1.3.7		Design complete	0 days	Thu 12/12/13	Thu 12/12/13	46	Yes
48	✓	1.3.8		Development Preparation	6 days	Fri 13/12/13	Fri 20/12/13	47	No

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

ID	WBS	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Milestone
49	**2**		**Term 2 (Includes Christmas Break)**	**97 days**	**Thu 26/12/13**	**Fri 09/05/14**	**1**	**No**
50	**2.1**		**Development**	**73 days**	**Thu 26/12/13**	**Mon 07/04/14**	**37**	**No**
51	**2.1.1**		**Iteration 1**	**25 days**	**Thu 26/12/13**	**Wed 29/01/14**	**48**	**No**
52	2.1.1.1		Develop code	15 days	Thu 26/12/13	Wed 15/01/14		Yes
53	**2.1.1.2**		**Testing**	**10 days**	**Thu 16/01/14**	**Wed 29/01/14**		**No**
54	**2.1.1.2.**		**Unit Testing**	**6 days**	**Thu 16/01/14**	**Thu 23/01/14**		**No**
55	2.1.1.2.		Test Units	2 days	Thu 16/01/14	Fri 17/01/14		No
56	2.1.1.2.		Fix Problems	3 days	Mon 20/01/14	Wed 22/01/14		No
57	2.1.1.2.		Unit testing complete	0 days	Wed 22/01/14	Wed 22/01/14		Yes
58	**2.1.1.2.**		**Integration Testing**	**5 days**	**Thu 23/01/14**	**Wed 29/01/14**		**No**
59	2.1.1.2.		Test Integration	2 days	Fri 24/01/14	Mon 27/01/14		No
60	2.1.1.2.		Fix Problems	3 days	Mon 27/01/14	Wed 29/01/14		No
61	2.1.1.2.		Integration testing complete	0 days	Wed 29/01/14	Wed 29/01/14		Yes
62	2.1.1.3		Iteration 1 Complete	0 days	Wed 29/01/14	Wed 29/01/14		Yes
63	**2.1.2**		**Iteration 2**	**24 days**	**Thu 30/01/14**	**Tue 04/03/14**	**51**	**No**
64	2.1.2.1		Develop code	15 days	Thu 30/01/14	Wed 19/02/14		Yes
65	**2.1.2.2**		**Testing**	**10 days**	**Wed 19/02/14**	**Tue 04/03/14**		**No**
66	**2.1.2.2.**		**Unit Testing**	**5 days**	**Wed 19/02/14**	**Tue 25/02/14**		**No**
67	2.1.2.2.		Test Units	3 days	Wed 19/02/14	Fri 21/02/14		No
68	2.1.2.2.		Fix Problems	3 days	Fri 21/02/14	Tue 25/02/14		No
69	2.1.2.2.		Unit testing complete	0 days	Tue 25/02/14	Tue 25/02/14		Yes
70	**2.1.2.2.**		**Integration Testing**	**6 days**	**Tue 25/02/14**	**Tue 04/03/14**		**No**
71	2.1.2.2.		Test Integration	2 days	Tue 25/02/14	Wed 26/02/14		No
72	2.1.2.2.		Fix Problems	5 days	Wed 26/02/14	Tue 04/03/14		No
73	2.1.2.2.		Integration testing complete	0 days	Tue 04/03/14	Tue 04/03/14		Yes
74	2.1.2.3		Iteration 2 Complete	0 days	Tue 04/03/14	Tue 04/03/14		Yes
75	**2.1.3**		**Iteration3**	**24 days**	**Wed 05/03/14**	**Mon 07/04/14**	**63**	**No**
76	2.1.3.1		Develop code	15 days	Wed 05/03/14	Tue 25/03/14		Yes
77	**2.1.3.2**		**Testing**	**10 days**	**Tue 25/03/14**	**Mon 07/04/14**		**No**
78	**2.1.3.2.**		**Unit Testing**	**5 days**	**Tue 25/03/14**	**Mon 31/03/14**		**No**
79	2.1.3.2.		Test Units	3 days	Tue 25/03/14	Thu 27/03/14		No
80	2.1.3.2.		Fix Problems	3 days	Thu 27/03/14	Mon 31/03/14		No
81	2.1.3.2.		Unit testing complete	0 days	Mon 31/03/14	Mon 31/03/14		Yes
82	**2.1.3.2.**		**Integration Testing**	**5 days**	**Mon 31/03/14**	**Fri 04/04/14**		**No**
83	2.1.3.2.		Test Integration	2 days	Mon 31/03/14	Tue 01/04/14		No
84	2.1.3.2.		Fix Problems	4 days	Tue 01/04/14	Fri 04/04/14		No
85	2.1.3.2.		Integration testing complete	0 days	Fri 04/04/14	Fri 04/04/14		Yes
86	**2.2**		**Documentation**	**14 days**	**Fri 04/04/14**	**Thu 24/04/14**	**50**	**No**
87	2.2.1		Develop Documentation	14 days	Mon 07/04/14	Thu 24/04/14	47	No
88	2.2.2		Documentation complete	0 days	Thu 24/04/14	Thu 24/04/14		Yes
89	**2.3**		**Project Report**	**12 days?**	**Thu 24/04/14**	**Fri 09/05/14**	**86**	**No**
90	2.3.1		Evaluation	5 days	Thu 24/04/14	Wed 30/04/14		No
91	2.3.2		Concluesion	2 days	Wed 30/04/14	Thu 01/05/14		No
92	2.3.3		Future Improvements	2 days	Fri 02/05/14	Mon 05/05/14	91	No
93	2.3.4		Spelling And Grammer	5 days	Mon 05/05/14	Fri 09/05/14		No
94	3		Project Complete	0 days	Fri 09/05/14	Fri 09/05/14	49	Yes

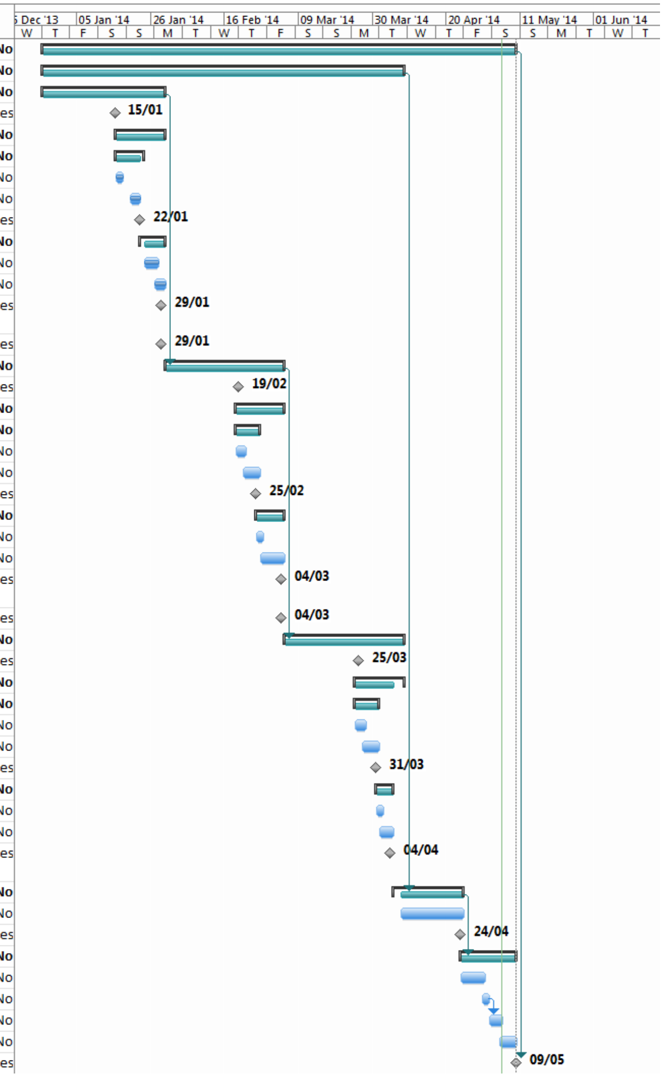


Figure 1.6: Gantt Chart for Term two (30^{th} January 2014)

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

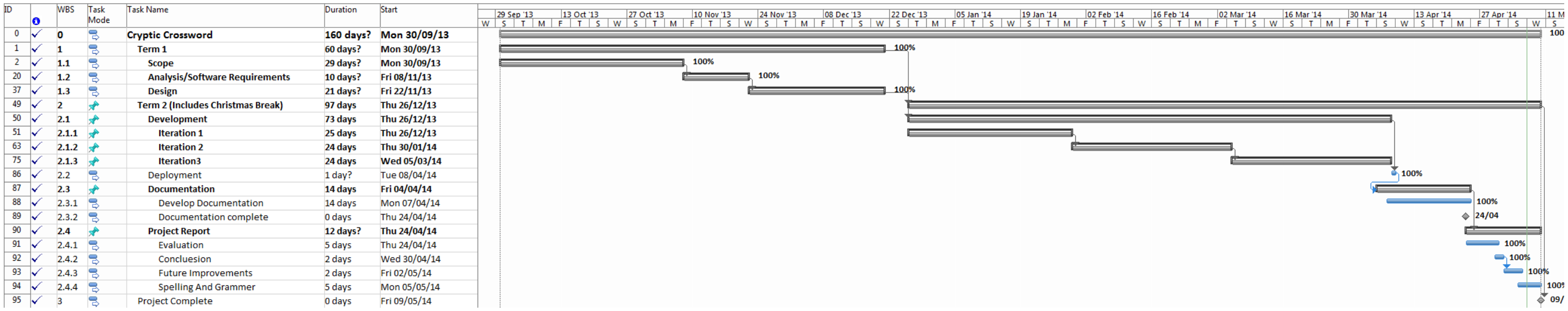


Figure 1.7: Final Gantt Chart of Project Summary (7^{th} May 2014)

Figure 1.7 represents an overview of the two terms and the milestones completed.

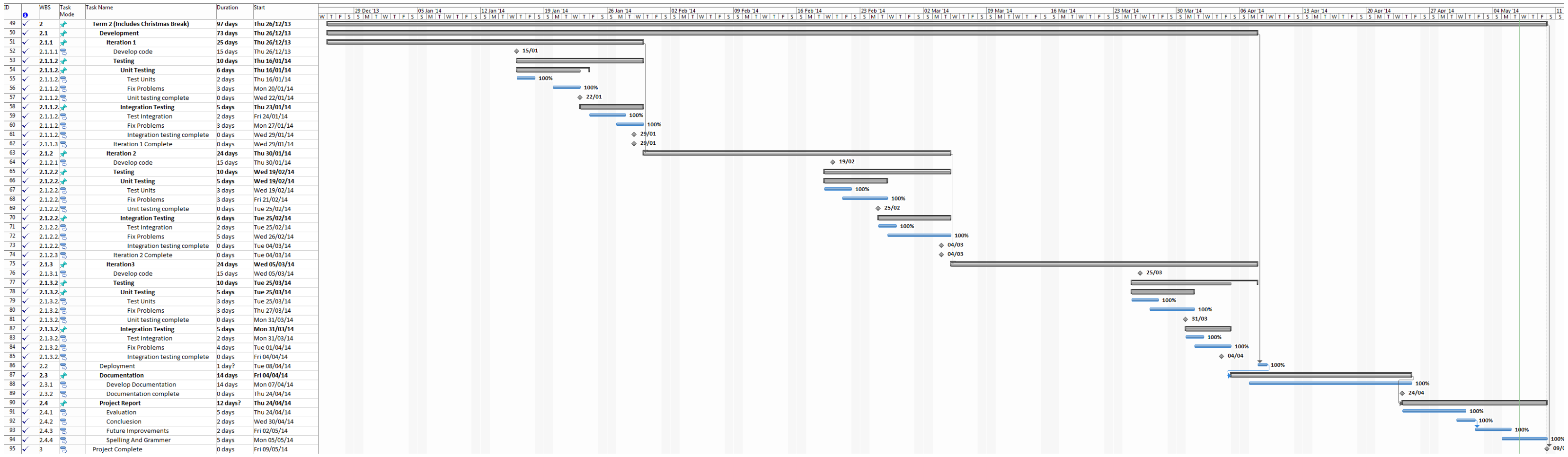


Figure 1.8: Final Gantt Chart for Term two (7^{th} May 2014)

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 upon the production web server. Improvements have been identified but there is not enough time for a fourth iteration.

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 sole. 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 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 were:

- 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 computing discipline 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 obligated to follow these rules and regulations set out in this handbook.

1.4.2 Intellectual Property Rights

The cryptic crossword solver is an academic application which is formally a prototype solution 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 final project (documentation and the software product) 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 done in a lawful manner.

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.

The Guardian Newspaper was contacted with regards to their data upon their website by the team, to ensure that their data could be used within the project. A copy of the email, and the response can be found upon the next page.

From: Permissions Syndication <permissions.syndication@theguardian.com>
Sent: 13 November 2013 10:39
To: S.Leader U0955187
Subject: Re: Cryptic Crossword Data for an Academic Project

Dear Stuart,

Provided that the use is strictly non-commercial and educational I am happy for you to use Guardian articles as source material for you project free of charge.

Please can you ensure that the story is credited to Guardian News & Media Ltd (year of publication)

Best regards,

Helen

Helen Wilson
Content Sales Manager
Syndication
web | print | tablet | mobile

T: +44 (0) 20 3353 2367
M: +44 (0) 7717 807 973
LinkedIn: http://lnkd.in/XFcpCs

Guardian News & Media Ltd
Kings Place, 90 York Way, London, N1 9GU

On 12 November 2013 22:17, S.Leader U0955187 <U0955187@unimail.hud.ac.uk> wrote:

Hello,

I am currently a student of Software Engineering at the University of Huddersfield. As part of my studies I am required to complete a group project, and for this we have chosen to design and produce a piece of software that will attempt to provide the correct solutions to given cryptic crossword clues.

As part of the development process, we expect to need an amount of training data; that is collections of cryptic clues with their corresponding answers. We would like to ask the Guardian's permission to use a few of the clues and answers that are publicly available on the crossword section of the Guardian's website. This would help us tremendously in the success of our academic project.

Please let me know if you would like clarification of any aspect of our project.

Kind regards,

Stuart Leader - u0955187
MEng Software Engineering

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 constitutes rarely more that 15% 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 is lacking 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 to 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 long and this is one of the reasons why the team took an iterative approach during development. This was to ensure that there is a product by the end of the academic year, whilst also 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 reinforced that an iterative approach would be better for development. This would ensure that all members were working efficiently and were able to complete the various project requirements.

Other models were looked at and considers such as the agile development and spiral methodologies, but the main reason why the iterative development methodology was chosen was because it contained characteristics of the waterfall model. Not only this, it managed to fit best into the team structure with the minimal amount of time that was made available.

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 achieved 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 newer 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 redeveloped 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
Embeded	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 estimated number of lines of code SLOC has been set to 5 thousand. 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

$$P = 13/3.25 = 4$$

Upon completion of the project the COCOMO model was applied to see how closely the original estimation was made. In total there are 7731 SLOC and therefore applying the same algorithms as before produced the following results:

Effort Applied (Person Months)

$$E = 2.4(8)^{1.05} = 21$$

Development Time (Months)

$$DT = 2.5(15)^{0.38}/2 = 4$$

People Required

$$P = 21/4 = 5$$

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 national UK newspaper that prints daily cryptic crosswords
Android	A mobile phone software platform by Google Inc.
BlackBerry	A mobile phone hardware and software platform developed by BlackBerry Limited
iOS	A mobile phone software platform developed by Apple Inc.
iPhone	A smart phone developed by Apple Inc.
iPod	A portable digital music player developed by Apple Inc.
iPad	A tablet developed by Apple Inc.
NLP	Natural Language Processing
SRS	Software Requirements Specification
App	Shorthand 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.