University of Huddersfield

MEng Group Project

Cryptic Crossword Solver

Software Specification

Authors:
Mohammad Rahman
Leanne Butcher
Stuart Leader
Luke Hackett

Supervisor:
Dr. Gary Allen
Examiner:
Dr. Sotirios Batsakis
Moderator:
Dr. Colin Venters

Friday, 9^{th} May 2014

Contents

List of Figures

List of Tables

Chapter 1

Development Methodologies

In order to ensure all objectives and goals that have been set within this project are completed to the highest quality and upon time, a software development methodology will need to be chosen. Dividing a larger project into a set number of defined processes may seem like additional unnecessary work, but the advantages of this process far outweigh the disadvantages (Knott and Dawson, 1999).

The defined processes combine together to form part of a process model. The process model will allow for the following achievements (Knott and Dawson, 1999):

- Adding an element of control and planning
- Allowing for progress to be mapped visually
- Providing a structured approach to development
- Allowing for a higher quality of code and documentation to be produced

The Systems Development Life Cycle (SDLC) was one of the first formalised methodologies for building software. The SDLC utilises a methodical and structured approach to analysing, designing, building and testing software, to which many methodologies follow this rigid structure (Elliott, 2004).

1.1 Waterfall

One of the main aspects to the waterfall model is the fact that the project is expected to progress down the primary path (Cadle et al., 2010).

The waterfall model takes the major components of any project (requirements, design, implementation, testing and maintenance) and assigns each component a stage of its own. Each component is delivered as the flow down the primary path is completed (Cadle et al., 2010).

The waterfall model also supports backtracking (i.e. reverting back to a previous deliverable). This allows for project managers to check that the project has not expanded its defined scope, and to also ensure that each deliverable flows into the next correctly. It also allows for slight modifications to be made, however making many large changes might affect the project in the long run (Cadle et al., 2010).

1.1.1 Advantages

Cadle et al. (2010) states that the waterfall model houses a number of advantages, including:

- Provides a rigid project structure, that is easy to follow and review
- Deliverables are delivered in project order, one at a time
- Can work well for smaller projects, or for projects where by the requirements will not change.

1.1.2 Disadvantages

However Cadle et al. (2010) also goes on to state that the waterfall model houses a number of potential problems, including:

- Changes are difficult to implement the further a project is down its primary path
- Large projects may not benefit from the rigid structure
- A working piece of software is not delivered until late into the project

1.2 Spiral

A common feature found in the waterfall model is that all requirements are stated at the start of the project. It is these requirements that will form the basis of all work, along with any project planning (Cadle et al., 2010).

The spiral model forms its basis around iteration and prototyping to try to explore the requirements and develop the solution. During each turn around the spiral, a set of requirements are analysed and developed using prototyping (Cadle et al., 2010).

1.2.1 Advantages

Cadle et al. (2010) states that the spiral model houses a number of advantages, including:

- A high amount of risk analysis is conducted, and thus risk is more likely to be avoided
- The model allows for approval from clients, and large amounts of documentation to be produced
- Software can start to be produced earlier, in comparison to the waterfall methodology
- Additional functionality can be added on at any time during or after the project

1.2.2 Disadvantages

The spiral model allows for a high level of control, without too much restriction. However Cadle et al. (2010) states that this can cause difficulties such as:

- A thorough investigation into all of the requirements cannot be achieved early, therefore some requirements (and their priorities) may get completely missed
- The spiral model is based upon the clients knowing exactly what they want, which is unlikely
- A risk analysis must be conducted, and requires highly specific expertise to complete. If a risk analysis is not completed, then the project may completely fail

1.3 Agile

The agile software development methodology is designed to "reduce risk by delivering software systems in short bursts or releases" (Dawson, 2009).

Each release (sometimes referred to as iterations) will involve minimal planning and will cover all the major SDLC components: analysis, design, implementation and testing. The agile development model also heavily promotes collaboration/development between team members (Dawson, 2009).

1.3.1 Advantages

One of the main advantages of using the agile development model is that software is developed in rapid cycles, which ultimately results in smaller constant incremental releases of software. As well as this major advantage, Dawson (2009) states the following advantages of using the agile development model:

- The methodology surrounds the concept of regular face-to-face meetings as opposed to in-depth documentation
- Utilises a close working relationship between the client and the developers, thus providing continuous delivery of useful software
- Uses shorter, iterative time scales (usually weeks rather than months or years), which results in working software being delivered frequently
- Easily able to change the requirements at any stage (however late the changes are)

1.3.2 Disadvantages

However many of the disadvantages of agile development model are surrounded by the lack of a rigid documentation, as Dawson (2009) also suggests the following disadvantages:

- There is often a lack of emphasis on necessary documentation (user documentation, design documentation etc.), which is normally skipped to save time
- The uncertainty of a specification may lead to poor code and/or structure
- The project can become confused if the original specification is not clear from the start
- Some software deliverables can be difficult to allocate the correct amount of resources (time, effort etc.) at the start of the project

1.4 Rapid Application Development

The Rapid Application Development (RAD) model is an extension to the incremental development methodology. The RAD model states that all requirements should be treated as mini projects, and that they should be completed in parallel. Each of the mini projects are ran like a normal project, and hence time scales need to be adhered to (ISTQB Exam Certification, 2013).

Upon completion of the mini project, the customer is able to review the output, and provide value feedback regarding to the delivery and the requirements. RAD will follow a somewhat simpler primary path, allowing for business modelling, data modelling, process modelling, application generation, testing and turnover (ISTQB Exam Certification, 2013).

1.4.1 Advantages

ISTQB Exam Certification (2013) states that there are many advantages of adopting the RAD model within a team:

- A reduced development time, due to the fact that the business modelling and data modelling processes should cover all aspects
- The combination of Data modelling and Process modelling should allow for the increased ability to reuse components
- Reviews of delivered outputs are constantly reviewed by the customer, allows for early feedback to be gained
- Parts of the system are integrated at an earlier stage, which allows for fewer integration issues towards the end of the project

1.4.2 Disadvantages

However, ISTQB Exam Certification (2013) also states that there can be disadvantages of adopting the RAD model within a team:

- There is a high dependency upon an overall strong team and strong individual performances for identifying business requirements
- The model will only work for systems that can be modularised
- The model assumes that the team members are highly skills designers and developers, with an even higher dependency upon modelling skills

1.5 Summary

In order to achieve the best possible product, it is clearly evident that the project should be developed utilising a feature driven approach. This will allow for any revisions, modifications, and changes to be considered and implemented with as little delay as possible, as well as little impact upon the rest of the project.

The projects requirements are not set directly by an external client, and hence it is possible for the requirements to be changed. It is because of these uncertainties that an agile development methodology would be best adopted by this project.

This methodology will not only allow for the requirements to change, but can allow for substantial research to be able to take place upon new topic areas if needed. Agile development methodologies allow for multiple releases of software, which fundamentally means that the team is able to use prototyping techniques to find the best outcome to a given problem.

Chapter 2

Software Specification

Robertson and Robertson (2013) state that 'a requirement is something the product must do to support its owner's business, or a quality it must have to make it acceptable and attractive to the owner.' The aim of gathering the requirements is to ensure that all ambiguities are removed before a product is developed.

Within this chapter the project's aims and objectives will be outlined, as well as project constraints and assumptions.Software requirements will be categorised as functional and non-functional requirements, to which a full MoSCoW analysis will be conduced.

Finally a risk assessment into the various project related risks will be conducted to help identify any potential risks. The risk assessment will also identify how the risks can be reduced.

2.1 Aims & Objectives

2.1.1 Aims

- Produce a software application that accepts the input of cryptic crossword clue from a user
- Produce a software application that will present a set of results to a user for the cryptic crossword clue which has been input
- Effectively adhere to a project plan as a team
- Effectively abide by the guidelines of a chosen software methodology
- Construct systematic and comprehensive documentation for the academic report

2.1.2 Objectives

- To research cryptic crosswords themselves and the various types of cryptic crossword clues
- To study relevant software architectures associated with the group project
- To investigate the field of artificial intelligence, in particular natural language processing and justify the need for it within the project to solve cryptic crossword clues
- To research the various software methodologies which exist and select the most appropriate for the project
- To understand and effectively follow a collection of requirements
- Evaluate the progress of the project at numerous instances throughout the life cycle of the project

2.2 Constraints

The following section describes the constraints that effect the design of the product. The product that is to be developed cannot be successful unless these constraints have been accomplished.

- The project will have four developers operational throughout the life cycle
- The user interface to be developed for the software application must be developed with considerations regarding its simplicity for the usage on mobile phones
- Resources to test the software application on mobile phones will cover the following platforms:
  - Android
  - BlackBerry
  - iOS
- The product shall require an Internet connection
- The product shall make use of the Apache OpenNLP Library
- Possible external resources to be used for the project, such as a dictionary and a thesaurus, may have limits on the usage
- The product shall be completed on or before the 9th May 2014
- The product shall be developed using all freely available tools and the only constraint of budget shall be the time the project resources put in to the product

2.3 Facts and Assumptions

- Answers to cryptic crosswords are usually published the following day
- Two clues that are identical, may not lead to the same solution
- Equipment needed to test the application on different platforms will be available from the University
- A server to host the web service will be available from the University
- An external resource (The Guardian) has given permission to use their cryptic crossword data to use for our test data
- Four team members working on the project throughout the year at approximately 25% of their academic study time
- Project scope will remain the same throughout the project

2.4 Scope

2.4.1 Organisation Structure

The project team is based largely upon democratic discussions and decisions, however to ensure that team deadlocks do not occur a project leader has been chosen. Figure 2.1 reflects the hierarchy of the team.

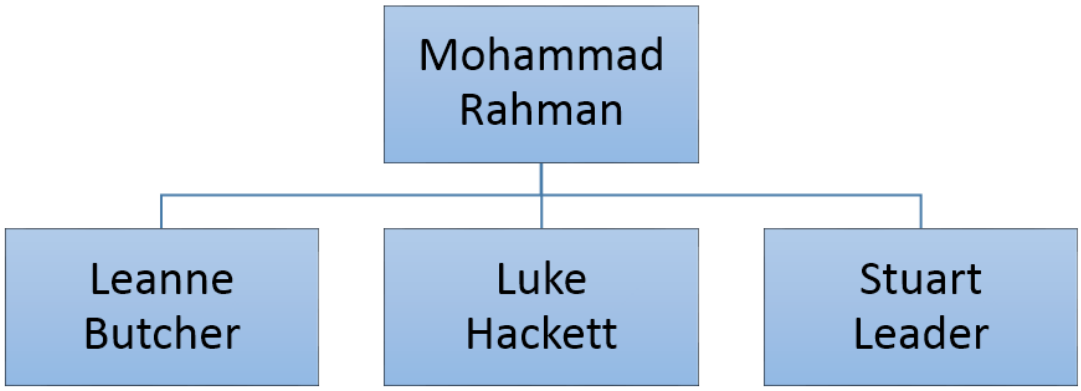

Figure 2.1: Hierarchical Structure of the team

2.4.2 Methodology

It has been decided that the team will follow an Agile software development model. This will allow the team to split the larger task down into smaller, more manageable 'chunks', that allows for a good quality analysis, evaluation, development and planning (on to the next 'chunk'). An iterative approach is best suited for this project due to the nature of changes and updates the product will require in future builds.

This method of development also allows for a more feature-driven approach to the project. Ultimately this allows for more important features and aspects of the project to be completed first.

2.4.3 Meetings

A weekly meeting will take place between all project members 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.

2.4.4 Product

Within the cryptic crossword area, there is a need for a cryptic crossword clue solver in the form of a software application. This has been justified previously within the document through research and investigating existing applications within the field. The main purposes of the software deliverable are to allow the input of a cryptic crossword clue by a user and output a potential result or number of results.

Once the project has been completed by the team, the following deliverables will have been accomplished:

- A software application which accepts input from the user and outputs appropriate solutions.
- A written report that:
  - document the entire software development process from a group perspective and analyses and evaluate the project as a whole

Furthermore the subsequent internal components will be implemented to aid with the goals of the project:

- A storage area to collect and store the cryptic crossword clues and their details used for test data as well as, clues successfully solved by the application which a user has input.
- A service which will allow connections from web browsers and potentially mobile phones to ensure the running of the software application.

There are specific criteria which have been identified as important features which must be completed to deem the project as adequately finished:

- A document outlining the full software development life cycle of the project
- A software application which can be accessed via either:
  - A web browser **and/or**
  - A portable device (e.g. smartphone or tablet)
- A service which can be connected to by either a mobile phone or a web browser which will sufficiently solve a cryptic crossword clue and output a result or number of results
- A storage area accessible by the service which stores the cryptic crossword clues and their associated details

Furthermore, there are specific criteria which have been deemed as out of the scope of the project:

- The user will not be able to access the storage area to browse through data

- The software application will not be implemented to have the ability to generate cryptic crosswords from the data stored

The restrictions listed below are the justifications for the project scope:

- The time scale of the project
- External priorities from the academic year such as other module assignments and examinations
- Limited resources for testing the software application on restricted platforms which the software application will be fully compatible with

2.5 Functional Requirements

2.5.1 MoSCoW Analysis

Functionality	**Must**	**Should**	**Could**	**Won't**
Retrieve and process cryptic clues and output possible solutions	Yes	-	-	-
Obtain characteristics of the clue's solution (e.g. number of words, word lengths and known letters) and match these against possible solutions	Yes	-	-	-
Provide a web interface for users to interact with the system	Yes	-	-	-
Provide a mobile-friendly interface for users to interact with the system	-	Yes	-	-
Store clues and their corresponding solutions for future retrieval, including the clue type and solution length	-	Yes	-	-
Take feedback from a user on a solution's accuracy and use this to rank solutions for a given clue	-	Yes	-	-
Relay the process (solution trace) followed to arrive at a proposed solution to the user	-	Yes	-	-
Holistically determine a confidence score for each proposed solution, and relay this to the user	-	Yes	-	-
Document and store the process (solution trace) followed to solve a given clue	-	-	Yes	-
Solve clues which take clues from the clue's orientation or clue number in the crossword	-	-	Yes	-
Provide real-time feedback of the processes being followed by the application in an attempt to calculate the correct solution	-	-	Yes	-
Take annotations provided by the user of the known aspects of a clue, such as the definition word(s), fodder or indicator word(s)	-	-	Yes	-
Store a clue's orientation and clue number in the database	-	-	Yes	-
Use stored data to generate complete cryptic crosswords	-	-	-	Yes
Provide a personalised service, including a history of a user's interactions with the system	-	-	-	Yes

Table 2.1: MoSCoW analysis of the project's functional requirements

2.6 Non-Functional Requirements

2.6.1 Performance

The performance of the system should be good, however it must be noted that the system is intended to run as a web service upon a server. The quality of the connection between the end user and the web service will be out of the scope of this project.

2.6.2 Legal and Access

Within the project a number of data sources will be used. Firstly an English dictionary will need to be accessed in order to allow the system to deduce if a given arrangement of characters formulates an English word.

In order to achieve this, an offline dictionary and an online dictionary will need to be used. The offline dictionary is a short dictionary and is provided as part of the GNU/Linux operating system under the GNU Licence Agreement.

A number of cryptic crossword clues and solutions have been obtained from the Guardian's website. The data will be used as a training set of data, when testing the final software. The Guardian has given us permission to use their data as long as the project remains within an academic environment, and that no profit is made from the project.

2.6.3 Backup and Recovery

The project will make use of the git revision control system over a secure shell connection. The service provider is github, who offer a distributed setup to ensure that data is always available. Using a secure revision control system will ensure that:

1. A copy of the project is stored upon a remote, secure server;
2. Changes are able to be tracked;
3. Issues and comments are able to be raised in a secure environment.

The above measures will in effect be able to reduce against the loss, damage or theft of all project information.

2.6.4 Archiving and Retention

The University will retain a copy of the project (software product and the final report) for assessment and evaluation purposes.

2.6.5 Maintainability

The project – including the software product – will be maintained up until the project hand in deadline: 9th May 2014.

2.6.6 Availability

The software will not contain any request or time limitations for users, and thus should be available at all times. However it must be stated that the server availability that will be hosting the web service is out of the scope of the project.

2.6.7 Capacity

The end product should be able to handle a number of requests from different users at the same time. As the project is only intended as an academic project, large scale user support will be out of the scope of this project.

2.6.8 Usability Requirements

Functionality	**Must**	**Should**	**Could**	**Won't**
Provide a text field for the user to input the cryptic clue to be solved	Yes	-	-	-
Provide a drop-down box allowing the user to input the number of words in the solution, with an initial upper-limit of 10 words	Yes	-	-	-
Dynamically provide individual text boxes for each word of the solution, which allow the length of the words to be specified. Also provide check-boxes between these text-boxes to define whether they are separated by a space or a hyphen	Yes	-	-	-
Dynamically provide text boxes to represent each character of each word of the solution, allowing the user to input any known characters	Yes	-	-	-
Provide a table of results which allow the user the to view the possible solutions	Yes	-	-	-
Alert the user to required fields with red asterisks	Yes	-	-	-
Have a consistent layout avoid unnecessary scrolling	-	Yes	-	-
Alert the user to invalid input through validation checks with error messages	-	Yes	-	-
Provide a confidence rating in the table of possible solutions	-	Yes	-	-
Allow the user to select a proposed solution in the corresponding table and mark this as correct	-	Yes	-	-
Display help buttons to indicate to the user the purpose and use of each control	-	-	Yes	-
Provide a button to submit the user input to the application for processing	-	-	Yes	-
Provide a group of radio buttons for the user to select the clue's orientation in its containing crossword	-	-	Yes	-
Provide a text box to input the clue's number within its containing crossword	-	-	Yes	-
Provide mechanisms to accommodate for users with difficulties, such as colour-blindness or poor eyesight	-	-	Yes	-

Table 2.2: MoSCoW analysis of the project's usability requirements

2.7 Risk Assessment

An investigation into the potential risks that may present themselves during the course of this project will allow the group to effectively minimise their impact should they become a reality.

Many facets of risk management contribute to the potential for a project that will be completed on time and with little interruption from unexpected occurrences. According to McManus (2003), not only will risk management improve the flow through the project due to a reduced overall project risk, but fewer surprises will allow for a more accurate schedule, and hence a greater likelihood of finishing on time.

2.7.1 Risk Identification

Risk (Threat)	**Description**	**Prob (%)**	**Impact (/10)**	**P-I** Value
Workload constraints	Fluctuations in the workload from university modules mean that certain times of the year are typically busier than others, as time and effort have to be delegated to the tasks at hand. This is especially applicable when assignment deadlines are imminent.	80	7	5.6
Underestimated system complexity	Without a thorough investigation into the project's requirements and comprehensive designs, it may become evident that the team has not accurately ascertained the complexity of the task at hand.	60	6	3.6
Project scope incorrect	A misunderstanding of the scale of the project may present significant problems and have adverse effects on the team's ability to produce a quality product on time.	40	6	2.4
Lack of understanding	The team may simply not understand a given task and be able to continue with the project in the expected fashion.	40	4	1.6
Inadequate facilities/resources	Potential resources that may be required include web servers, database management systems, phone app development kits and dictionaries / thesauruses. Seamless access to each and every resource may prove difficult.	70	2	1.4
Ineffective team structure	Explained elsewhere in this document, the team have adopted a largely democratic model. As pressures and workloads increase, conflicts may arise and the team structure may hamper the group's ability to resolve any issues effectively.	40	3	1.2
Skills mismatch	The team may find that the skills required to complete the project are beyond those which are currently held. This has been identified as a risk as the group's collective experience is relatively low.	15	4	0.6
Workforce reduction	The most likely cause for this risk to become a reality is if a team member leaves the university course. As there are only 4 members working on the project, this would equate to a 25% reduction in the workforce.	5	8	0.4

Table 2.3: Probability-impact table for the project's risks

2.7.2 Risk Classification

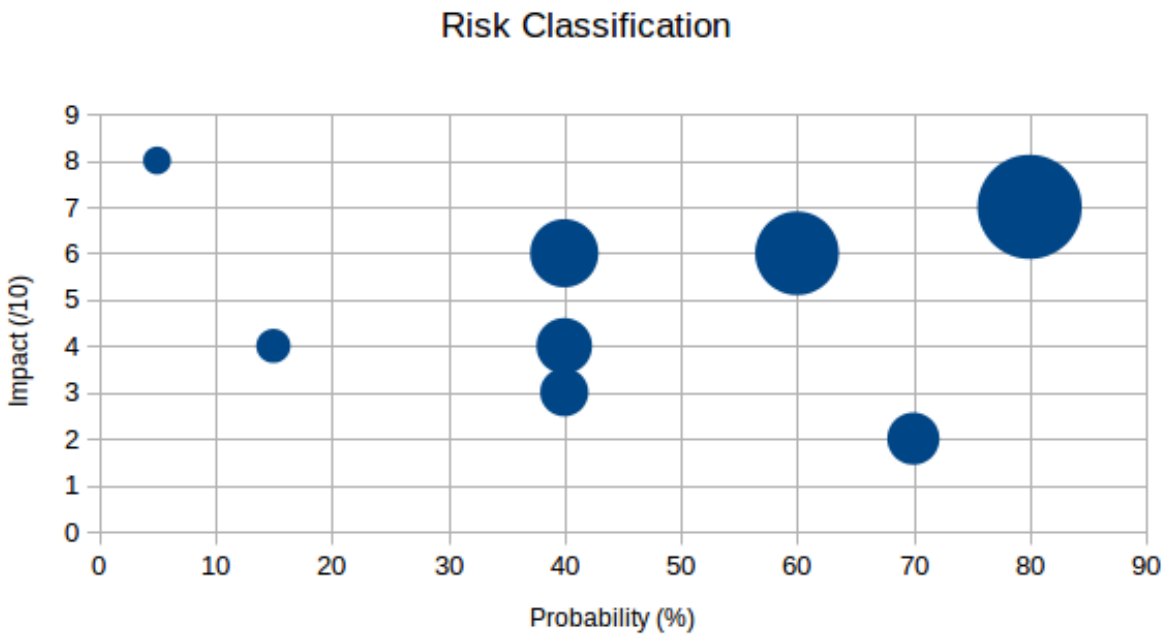

Figure 2.2: A visualisation of the identified risks of the project

Figure 2.2 demonstrates the impact and likelihood of the identified risks occurring. The risk with the greatest PI (*Probability / Impact*) score is the risk of constraints from other university commitments interfering with the expected progression of the project. This is further compounded by the fact that each member of the team is enrolled on the same course of study, and so the impact from such commitments will be magnified across each and every member.

In contrast, the risk with the lowest identified PI score is that of a group member leaving the course, and hence the project. While this has the largest impact score identified, the risk of such an occurrence is minimal. This is categorised by Stoeller (2003) as an *alligator*. This is defined as a risk with a high impact, but a low probability of becoming a reality.

2.7.3 Risk Prevention

Workload constraints The appearance of this risk balances on each team member's ability to effective manage their time. If a consistent effort is maintained throughout the course

of the academic year, there will likely be a greater chance of this risk being prevented. Otherwise an sudden increase in workload will detract from the team's collective ability to work on the project. Project planning should include contingencies for unexpected occurrences, especially around the time of pressures or deadlines from other course modules.

Underestimated system complexity A thorough set of design documents based on the gathered requirements will minimise the potential for this risk to show itself. A range of diagrams from use-case diagrams to sequence diagrams will allow for a greater understanding of the identified tasks, thus reducing the risk of encountering unexpected complexity.

Project scope incorrect & Lack of understanding A recurring theme of this project is that a lack of planning may turn into magnified issues further down the line. The team members should be able to gather a reasonable understanding to the project's scope during the earlier phases. Conclusions may be drawn from the research and requirements analysis phases which hold some indication to the potential size of the project.

Inadequate facilities/resources The team will need to draw on a host of resources in order to efficiently work on the project. Repository and versioning systems will be used to maintain documentation and a code base. Development environments and development kits will be used to aid the software development, and servers and frameworks may be required to host any applicable software. To prevent the risk of issues arising from any of these software packages, it may be best to forecast which software will be required along with an estimation of when this is likely to be. Consequently provisions can be made in advance to ensure the software is set up.

Ineffective team structure Despite a unanimous vote for a democratic structure within the group, a team leader has been selected (*Mohammad*) who will hold the capacity to make an overriding decision on the project if the team cannot come to an agreement unaided. It is also the responsibility of Mohammad to delegate work where he feels it necessary. Such actions may collectively contribute to minimising the impact of this risk, but they may not allow for it to be entirely prevented.

Skills mismatch To ensure the team possess all the skills required to create the proposed solution, the gathered requirements, and in particular the technical requirements will outline the technologies earmarked to be used. As this phase is positioned relatively early on in the project, there should be relatively little chance of realising we require skills we do not possess further on in the project.

Workforce reduction A decision to leave the course before completion will likely be one that will not be taken lightly be the respective student. Consequently this is a threat for which there is little that could possibly be done to prevent it, other than allowing for an amiable working environment amongst each team member.

2.7.4 Risk Mitigation

Workload constraints Effective project planning should prevent this risk from manifesting itself in the project, but should it occur, it may be possible to balance the project workload more effectively amongst the team members. It may be possible that the anticipated workload of each team member can be mapped, and planned work on the project can be temporarily postponed until these constraints no longer present an issue.

Underestimated system complexity An assessment of the implications on the project should be conducted, and the effects investigated. If it would not be possible to incorporate the additional complexity using the time and resources available, the project scope may be reduced to allow for a smaller, but fully functional product. Research into development methodologies suggest that an agile methodology will be adopted. This will allow for a fast reaction to any change seen in the scope or structure of the project.

Project scope incorrect The group should draw on experiences from encountering this risk, and reflect on why it became a reality. A revised project scope should then be created, using these deductions to ensure the risk doesn't manifest itself twice.

Lack of understanding & Skills mismatch The obvious solution to mitigate this risk is to acquire the understanding necessary to carry forward with the work. Contingencies in the project plan may allow for such tasks to be carried out with little impact on the overall schedule. A variety of resources should be utilised to ensure the most effective acquisition of the knowledge necessary to proceed.

Inadequate facilities/resources In the case that a particular resource is unavailable, the team should conduct an assessment of why this is the case and what can be done to rectify the issue. If the issue is a delay in the time it takes to acquire a particular resource, the most viable option could be to search for alternatives or rearrange the schedule to complete other tasks which are non-dependent on this particular task.

Ineffective team structure If the chosen team structure is proving ineffective, it may be necessary to adopt a different set-up for the benefit of the group. With the team members

working with one another over a span of months, each members' working style in a group environment should become apparent, and this may help to shape any new structure.

Workforce reduction The time frame for the project's completion is fixed, but a reduction in workforce may necessitate a respective reduction in project scope or complexity. For the remaining three team members to attempt to complete the work set for four would possibly result in a product which is either unfinished or below standard. While it may prove frustrating to reduce the scope, this should allow for a subset of the complete solution, but which is just as refined.

2.8 Feasibility Study

A feasibility study will be carried out to identify the difficulty of the implementation needed for each type of cryptic crossword clue with the aid of the research carried out previously. The study will focus on the resources needed for each implementation which will contribute to the level of difficulty for each clue type. Furthermore, using the previous research resources within the cryptic crossword field, a measurement of how common each clue type is within a cryptic crossword will be presented to assist in the understanding of the necessity of the clue type within the project.

Aspects such as how common clue types are featured within research resources and how regular indicators are found in the project test data will contribute to the judgement made on how regular a type of clue is. The justification of the use of specific resources for a clue type will come from research resources and inspecting project test data.

Clue Type	**Possible Resources**	**Difficulty**	**Regularity**
Hidden	Dictionary	Low	Common
Anagrams	Dictionary	Low	Common
Acrostics	Dictionary	Low	Intermediate
Pattern	Dictionary	Low	Intermediate
Homophones	Homonym Dictionary	Medium	Common
Charades	Abbreviations, Dictionary, Thesaurus	Medium	Common
Deletions	Abbreviations, Dictionary, Thesaurus	Medium	Common
Reversals	Abbreviations, Thesaurus	Medium	Common
Palindromes	Dictionary, Thesaurus	Medium	Intermediate
Double Definition	Dictionary, Thesaurus	Medium	Intermediate
Substitutions	Abbreviations, Dictionary	Medium	Rare
Shifting	Dictionary, Thesaurus	Medium	Rare
Exchange	Dictionary, Thesaurus	Medium	Rare
Spoonerisms	Dictionary, Thesaurus	Medium	Rare
Containers	Abbreviations, Dictionary, Thesaurus	High	Common
Purely Cryptic	-	High	Common
& lit	Abbreviations, Dictionary, Thesaurus	High	Intermediate

Table 2.4: Feasibility Study for Clue Types

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

Cadle, J., Eva, M., Hindle, K., Paul, D., Rollaston, C., Tudor, D., and Yeates, D. (2010). *Business Analysis*. British Computer Society, second edition.

Dawson, C. W. (2009). *Projects in Computing and Information Systems: A Student's Guide*. Addison Wesley, second edition.

Elliott, P. G. (2004). *Global Business Information Technology: An Integrated Systems Approach*. Addison Wesley, first edition.

ISTQB Exam Certification (2013). What is rad model- advantages, disadvantages and when to use it? Retrieved from `http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/`.

Knott, R. and Dawson, R. (1999). *Software Project Management*. Group D Publications Ltd.

McManus, J. (2003). *Risk Management in Software Development Projects (Computer Weekly Professional)*. Routledge.

Robertson, J. and Robertson, S. (2013). *Mastering the requirements process*. Addison-Wesley, third edition.

Stoeller, W. (2003). What is risk management? *Globalization Insider*, XII(3.7).