University of Huddersfield

MEng Group Project

Cryptic Crossword Solver

Authors:
Mohammad Rahman
Leanne Butcher
Stuart Leader
Luke Hackett

Supervisor:
Dr. Gary Allen

Examiner:
Dr. Sotirios Batsakis

Friday, 26^{th} April 2014

Abstract

Cryptic crosswords are a unique style of crosswords in which the answer to each given clue is a word puzzle. An answer can only be obtained if the cryptic clue is read in the correct way. Often when the clue is surface read, the clue makes no sense at all. The challenge is to find a way in which the reading of the clue leads to a solution.

Cryptic crosswords are a popular type of puzzle found in many parts of the world. Most of the national UK newspapers will print cryptic crosswords of varying difficulty on a daily basis. Many users can often become frustrated when a clue appears to be unsolvable.

Clues that appear to be unsolvable are often all part of the fun of cryptic crosswords. However each clue can take some time to solve due to the fact that there are many forms of wordplay.

Contents

List of Figures

List of Tables

Chapter 1

Problem Analysis

An initial problem analysis has been conducted to ensure that the overall project remains focused upon the original problem.

This chapter will discuss key topics and will recommend that these topics are researched further to help support the project. The problem analysis will also define the problem in more detail, to help with understanding the project and its purpose.

1.1 Defining the Environment

Cryptic crosswords are a popular type of puzzles found in many parts of the world. Most commonwealth national newspapers will print cryptic crosswords of varying difficulty on a daily basis.

Cryptic crosswords are a unique style of crosswords, in which the answer to each given clue is a word puzzle. An answer can only be obtained if the cryptic clue is read in the correct way. Often when the clue is surface read, the clue makes no sense at all. The challenge is to find a way in which the reading of the clue leads to a solution.

1.2 Defining the Problem

Many users can often become frustrated when a clue appears to be unsolvable. It is the vast range of possible clues that often makes solving not only challenging but interesting as well.

Fundamentally, the overall aim of this project is to develop a piece of software that is able to solve any given type of cryptic crossword clue.

By using some form of natural language processing and one or more cryptic algorithms, it should be possible to generate an answer to a given clue.

Once a clue has been correctly "guessed" it can simply be returned to be user. It is the "guessing" of the answer that this project will primarily focus upon.

1.3 Research Areas

Before undertaking the project an initial review was conducted. The review's objective was to determine the feasibility of the project as a whole. The review also covered whether or not the project has been completed before.

From the outlined background and problem information it is clear that cryptic crosswords are a popular form of entertainment. It is also clear that some clues are particularly difficult to solve, and users may often ask other people for help in solving a given clue.

1.3.1 Cryptic Crosswords

A review of the national UK newspapers was conducted to determine whether or not there is a pattern in cryptic crosswords. Of all the newspaper's websites that were reviewed (The Guardian, The Times, The Independent and The Mirror) it was clear that the cryptic crosswords are the same style.

Each clue is categorised as being either 'across' or 'down' with its corresponding grid number. Each clue will also contain the number of letters the answer should be. An example is show below:

12. The seamstress's sensation? (4, 3, 7) =>PINS AND NEEDLES

The Guardian's website utilises web standard technologies such as HTML and CSS, and also provides an option to solve a clue. The Mirror's website follows a similar approach to the Guardians website; however solutions can only be obtained by dialing a premium telephone number.

The Times and the Independent both utilise a different approach and that is to serve a Java applet. Both Java applets allow the user to solve a clue should they get stuck. The Times provides puzzles as part of their paid subscription service.

All of the above newspapers publish cryptic crosswords upon a daily basis, with the solutions to the crosswords appearing in the next day's newspaper.

Following from the crossword review, a second review into cryptic crossword solvers was undertaken. The objective of this review was to determine whether or not computerised cryptic crossword solvers exist. The three cryptic crossword solvers that were looked at were One Across, Crossword Tools and Cryptic Solver.

Each of the solvers manages to solve some clues with the same answers, with other clues providing a range of possible answers.

Crossword Tools (Crossword Tools, 2013) is a paid subscription based service, which allows users to enter a clue and a pattern. A pattern can contain part of the answer or the number of letters the answer has. If multiple answers are available, they are displayed. An example is shown below:

Kind of dog (10) =>the answer is 10 letters long.

Kind of dog (?????????r) =>the answer is 10 letters long, final letter is 'r'.

Cryptic Solver (Cryptic Solver, 2013) is a free service that offers the same functionality as Crossword Tools. Although Cryptic Solver does provide the correct answer, it does not necessarily provide the correct answer at the top of the list.

Finally One Across (One Across, 2013) provides all the same functionality as the previous two solvers, along with a score. The score is linked to the number of people who have used the given answer (effectively it's a ratings system). One Across uniquely

highlights how it has managed to come to the answer, showing the break downs of each sentence. As with Cryptic Solver, One Across is a free service that doesn't require a subscription.

1.3.2 Natural Language Processing

In order to correctly solve a clue, some form of natural language processing will be required. It is the natural language processing that will try to deduce the meaning of a clue. It is the meaning that can then be aligned with possible answers.

An example of natural language processing can be found within the One Across application. Given a clue (and a pattern) it will try to provide an accurate solution:

> Spin broken shingle (7) =>ENGLISH

In order for the answer to be obtained, One Across will follow a natural language processing path and will provide its trace path. The trace path shows how the clue has been broken down to get to the answer. The trace path for the above clue can be found below:

> 'spin' is the definition.
>
> 'broken' means to anagram 'shingle' to get ENGLISH.
>
> ENGLISH matches 'spin' with confidence score 100%.

1.3.3 Application Platform

The existing products that have been discussed within this problem analysis have all been accessible via a browser. Although this is an acceptable platform, there could be a better platform that allows uses to utilise the technology easier.

As previously mentioned, most crosswords are designed for users who have a few minutes to spare on the move. As many people own a smartphone and/or a tablet, there may be a gap in the market for a high quality mobile cryptic crossword solver.

An in-depth review will need to be conducted in order to deduce the viability of this proposal.

Chapter 2

Research

In the previous chapter a detailed approach to what the project will cover was discussed. In order to correctly implement a working piece of a software a number of areas will need to be researched thoroughly.

Based upon the problem analysis, the the main topics that have been identified are:

- Cryptic Crosswords
- Natural Language Processing
- Application Platform
- Web Services

2.1 Crosswords

Arthur Wynne produced the first crossword puzzle which was printed on December 21st 1913. A crossword is a puzzle which involves the solver resolving the answer to a clue and placing it in the correct space within the grid.

A grid is made up of black and white squares, the black squares are blanks and the white squares are where the solver must place the answers. A crossword grid comes with a set of clues. The clues are usually arranged based upon their positing within the grid. Clues that that appear downwards in the grid are kept separate to the clues that appear across the grid.

The white squares which are used for the first letter of an answer to a clue usually have a number in the top left hand corner to indicate the clue which links to this area of the grid.

There are different types of crosswords such as quick, cryptic and double-clue. A quick crossword has clues which simply define the answer. A cryptic crossword is more complex as it has word play as well as simple definitions and many different types of clues. A double-clue crossword combines the two and allows for a simpler option for the solver when the cryptic clues become too difficult.

2.1.1 Users

Kathryn Friedlander and Philip Fine (Friedlander and Fine, 2009) carried out an investigation into whether the amount of cryptic crosswords completed by a solver determined how successful they were at them.

To complete this study they gathered data from 241 people, this data can be used as part of research to determine the typical audience cryptic crosswords have.

The following facts about the user base are taken from the results section of the paper (Friedlander and Fine, 2009):

- "209 M, 32 F"
- "mean age=53 years, range=23-83"
- "mean time spent=8 hours per week, range=1-30"

2.1.2 Cryptic Crosswords

Most cryptic clues consist of two different parts, the word play and the definition itself. The definition is like a clue found within a quick crossword and the word play is an indication to the answer.

Clue types such as double definition and purely cryptic break the usual format of cryptic clues. Double definitions miss out the word play whereas the purely cryptic clues miss out the usual simpler definition and become a fully cryptic definition. Other types of clue add to the usual format when smaller clues are embedded within the larger clue to assist with the word play.

Punctuation within clues should always be disregarded unless it is a question mark or an exclamation mark. Punctuation such as commas and hyphens are used to distract the solver from the answer usually by attempting to dictate how the clue is read.

A question mark tells the solver that the clue requires creative thinking to work out the answer which could be witty in nature. An exclamation mark can mean that the word play and the definition may intersect which is otherwise known as a clue of the type "& lit". Articles within clues can also be very important and should not be disregarded when reading the clue.

Cryptic clues which have a particular tense will always be for a clue with the same tense. Similarly a plural clue determines that the answer will also be plural.

2.1.3 Crossword Clue Types

An important skill needed to solve a cryptic crossword is to be able to spot the type of clue given. Below is a list of the most common types of cryptic clue and expected rules the clue should follow so they are identifiable.

Purely Cryptic

Although clues within a cryptic crossword usually include both a definition and word play, this type of clue is an exception because the whole clue is a definition written in an unusual way.

Word play within other clue types assist the user in being able to determine their answer is correct as well as solve them, because there is no word play more than one answer could be found which are incorrect. Possible indicators that denote a purely cryptic clue are:

- Question mark
- Exclamation mark

Example: *Frames for summer's activities? (5)* - (Upadhyay, 2008d)
Answer: ABACI

- "Summer" as in a person who does mathematical sums
- An abacus is a frame which holds moving beads
- As the clue is plural so must the answer be, hence abaci

Hidden

The answer for a hidden clue is concealed within the clue itself and can be spread over more than one word as well as possibly being hidden in reverse. The clue will have a definition, an indicator that the answer is hidden within the clue and a word or set of words which have the answer in them. Possible indicators that denote a hidden clue are:

- Word/s e.g. contains, in, within, held by, from
- Large words with a hidden word indicator before it may have the answer inside them
- A clue which seems inelegantly written or a clue which contains proper nouns

Example 1: *Metal concealed by environmentalist (4)* - (Upadhyay, 2008f)
Answer 1: IRON

- "Concealed by" is a phrase indicator for hidden clues
- "Environmentalist" is a large word with an indicator in front
- "Metal" is the definition so the answer is a type of metal which can be found within the word "environmentalist", hence iron

Example 2: *Mountain range in central Taiwan (5)* - (Upadhyay, 2008f)
Answer 2: ALTAI

- "in" is a word indicator for hidden clues
- "central Taiwan" contains a proper noun which indicates the answer is hidden here
- "Mountain range" is the definition
- Without knowledge of mountain ranges the answer could be narrowed down to the following words (assuming "central" would not be within the clue without a purpose):
  - TRALT
  - RALTA
  - ALTAI
  - LTAIW
- If some of the crossword is completed within the area this clue is placed, the correct answer could be found through trial and error.

Charades

A charade clue forms its answer with the use of smaller answers to smaller clues within the main clue. Abbreviations and first/last letters of words are common within charade clues to make up the complete answer.

Two or three parts are usually within the charade clue to solve the correct answer, they may not be in the right order however, and word indicators will be used to warn the solver.

Other types of clues can also be used within charade clues for the different parts such as reversals and homophones, if this is the case there will be indicators for the specific type. Possible indicators that denote a charade clue are:

- Words e.g. with, follows, behind, after to indicate joining of answers to parts of the clue

Example: *Prior belted one that is ultimately right (7)* - (Upadhyay, 2008c)
Answer: EARLIER

- "Prior" is the definition
- "belted one" gives earl
- "that is" gives the abbreviation for i.e. or ie
- "right" gives the abbreviation for r
- All the segments put together give the word "earlier" which can also mean "prior"

Anagrams

Anagram clue types have a definition, a word or phrase to indicate the clue is of this type and an element called "fodder".

An anagram is a word whose letters can be rearranged to form another word; within an anagram clue the letters to rearrange are known as "fodder" and are placed next to the indicator. Possible indicators that denote an anagram clue are:

- Words which could mean change or shifting
- A clue which seems inelegantly written or a clue which contains proper nouns

Example: *Toy breeds trained to find out a place for pearls (6,3)* - (Upadhyay, 2008b)
Answer: OYSTER BED

- "trained" indicates the clue is an anagram
- "Toy breeds" is an abnormal phrase and is the "fodder" of the clue
- to find out a place for "pearls" is left to become the definition
- "Toy breeds" is then moved around to give oyster bed

Homophones

A homophone is a word which sounds like another word but has a separate meaning. This type of clue has a definition, a word or phrase which means the same as the homophone to find and an indicator. Possible indicators that denote a homophone clue are:

- Words which indicate hearing or sound e.g. said, heard
- Normally the indicator for a homophone is next to the word or phrase which is to be used to find a homophone

Example: *Refer to a location, reportedly (4)* - (Upadhyay, 2008g)
Answer: CITE

- "reportedly" is the homophone indicator
- "a location" is the phrase which needs to be used to find a homophone
- "Refer to" is the definition
- A location can be otherwise known as a "site", hence cite

Acrostics

An acrostic clue commonly involves picking the first letter from a group of words and putting them together to form the answer. It is possible that the clue will require the last or middle letters from words to solve them or that the letters should be put together in reverse order.

This clue type has a definition and an indicator as well as "fodder" which in this case means the group of words the necessary letters will come from. Possible indicators that denote an acrostic cryptic clue are:

- Words which could mean start or beginning
- If the clue is unusually long and so is the number indicator to determine how long the answer should be

Example: *Some URLs recommended for beginners to explore online (4)* - (Upadhyay, 2008a)
Answer: SURF

- "beginners" is the indicator
- As the clue states the answer should be of length four it is assumed "to explore online" is the definition as there are only three possible letters for an acrostic in the phrase
- "Some URLs recommended for" is the "fodder" for the clue. The indicator implies that the first letters should be taken from the first letters of the words within the "fodder", hence surf

Palindromes

A palindrome is a word which reads and looks the same when it is reversed. This type of clue has an indicator and a definition. Possible indicators that denote a palindrome clue are:

- Phrases which may mean either way or going around in circles

Example: *Unacceptable, going up or down (3,2)* - (Connor, 2012a)
Answer: NOT ON

- "going up or down" is an indicator as it could mean "in either direction" like the format of a palindrome
- "Unacceptable" is then left as the definition which gives the answer, not on

Reversals

A reversal requires the solver to reverse a number of letters to give a new word. The clue consists of a definition, an indicator that the clue is a reversal clue and some "fodder" which is a phrase or word which could contain the letters to be reversed or a smaller clue which leads to the letters which need to be reversed. Possible indicators that denote a reversal clue are:

- Words/phrases used for directional purposes e.g. left, up
- The word indicators may also be relative to the direction the clue should be placed within the crossword (down or across)

Example: *Stop the flow in crazy get-up (3)* - (Upadhyay, 2008i)
Answer: DAM

- "Stop the flow" is the definition
- "get-up" is the indicator that the clue is a reversal
- Another word for "crazy" is mad which reversed gives the answer dam

"& lit"

"& lit" clues, which means "and literally so", is a type of clue where the definition and the word play are the same and are not split out into separate phrases or words as with other clues. The definition is the whole clue and the word play can be one or more of any of the normal clue types such as anagrams and charades. Possible indicators that denote a "& lit" clue are:

- Exclamation mark

Example: *Cop in male form (9)* - (Upadhyay, 2008h)
Answer: POLICEMAN

- The whole clue is the definition
- "form" is an indicator for an anagram clue
- "Cop in male" can be rearranged to policeman which is also the answer to a "Cop in male form"

Double Definition

A double definition clue has no word play and is purely a clue with two (or possibly more) definitions which lead to the same answer. Possible indicators that denote a double definition clue are:

- Possibly shorter than most clues (2 or 3 words)
- Although it is advisable to ignore punctuation when solving cryptic clues, a double definition may have a piece of punctuation separating the definitions.

Example: *Robust author (5)* - (Upadhyay, 2008e)
Answer: HARDY

- Both words are separate definitions which could both mean hardy

Containers

A container clue includes three definitions and an indicator. One of the definitions is for the final solution whereas the other definitions describe two separate answers where one is contained within the other.

They are similar to charade clues in the way that other types of clues can be used within container clues such as anagrams and charades themselves. Possible indicators that denote a container clue are:

- Word indicators which could be used to indicate the inner word (e.g. inside, held) or the outer word (e.g. outside, external)

Example: *Building for the workers in principle (8)* - (Upadhyay, 2009a)
Answer: TENEMENT

- "principle" is the definition for the outer word which gives tenet
- "in" is the indicator that the answer for the inner word will be placed within the outer word (tenet)
- "the workers" is the definition for the inner word which is men
- 'Building' is the overall definition which could mean tenement which is also given when men is put within tenet

Deletions

Deletion clues require the solver to retrieve the answer by looking at the definition and the word play and removing the correct letters from the correct word. The clue will not usually have the word which needs letters removing from it directly within the clue. Possible indicators that denote a deletion clue are:

- Words to indicate letters should be removed from a certain place within the word such as its first or last letter
- Words to indicate certain letters should be removed from the word found. These could be words that can be abbreviated

Example: *Little shark edges away from diver's equipment (3)* - (Upadhyay, 2009b)
Answer: CUB

- "edges away from" indicates that the first and the last letter should be removed from the answer which comes from "divers equipment"
- "divers equipment" is otherwise known as scuba
- Removing the "s" and the "a" from scuba leaves the word cub which is also the answer given from the definition "Little shark"

Spoonerisms

A spoonerism clue has a definition and word play which is usually a phrase which describes another. This phrase, which is usually two words long, is taken and the first letter of each is swapped around to gain a new phrase which is in turn the answer to the clue. Possible indicators that denote a spoonerism clue are:

- The word "Spooner" or "Spooners" is placed within the clue

Example: *Spooner's cheerful enthusiast? He'll get you across (8)* - (Connor, 2012b)
Answer: FERRYMAN

- "Spooners" indicates that this clue is a spoonerism
- "Hell get you across" is used as the definition
- "cheerful enthusiast" could also be known as a merry fan
- Swapping the first letters from the words merry and fan give the answer ferryman

Pattern

This type of clue has a definition, an indicator that the clue is of the type pattern and a phrase or word which has the answer within them arranged as a pattern. These patterns can be odd or even letters joined together to make a word or possibly letters picked from regular intervals.

Pattern word play can also be joined together with other types of clues such as charades to form a more complex answer. Possible indicators that denote a pattern clue are:

- Words which could mean even, odd or routine

Example: *Beasts, free, ginned, we hear regular losses there! (8)* - (Upadhyay, 2009d)
Answer: REINDEER

- "regular" indicates it is a pattern clue as well as "losses" to indicate the dropping of letters.
- "Beasts" is the definition.
- "free, ginned, we hear" holds the answer reindeer by picking out the first "r" within free and each letter alternately from then on.

Substitutions

A substitution clue involves removing letters from a word and replacing it with another to retrieve the answer. There are two definitions within the clue, one definition to retrieve the word to substitute letters from and another to define the final answer.

The letter or letters to substitute are usually an abbreviation which can be found within the clue itself. Possible indicators that denote a substitution clue are:

- Words which mean substitution e.g. replace, switch, exchange

Example: *Unexciting story gets mark for length (4)* - (Upadhyay, 2008j)
Answer: TAME

- "mark for length" is an indicator for a substitution clue
- "mark" can be abbreviated to "m" and "length" can be abbreviated to "l", therefore replace "l" with "m"
- "story" is the definition for the word which needs the substitution and could be defined as a tale
- Replacing the "l" in tale with "m" gives the word tame which can also mean unexciting

Shifting

A shifting clue has an indicator, a definition of the final answer and another definition for the word which needs to be used to shift a letter to a different position within the word to find the final answer.

The shifting of a letter could be moving the first letter to the last position in the word or in a more complex clue letters could be shifted within the middle of the word. Possible indicators that denote a shifting clue are:

- Words e.g. shift, change, move
- Phrases e.g. head to foot

Example: *Character needs help, head to foot (4)* - (Upadhyay, 2009e)
Answer: BETA

- "head to foot" indicates moving a letter from the front of a word to the end
- "help" is the definition for the word which requires letter shifting and can also be defined as abet
- Moving the first letter of "abet" to the end gives the word beta which is a "Character"

Exchange

An exchange clue is similar to a shifting clue however instead of only one letter shifting positions within a word, two letters within a word exchange places to form a new word.

Typically the letters to exchange will be the first and last letters of a word or two letters next to each other, however it is possible more than one letter on each side will need to be swapped.

For example, the word "rage" can be split into two sections "ra" and "ge" which can then be exchanged to make the word "gear". Possible indicators that denote an exchange clue are:

- Words e.g. swap, exchange, change

Example: *Doomed king switching sides? True (4)* - (Upadhyay, 2009c)
Answer: REAL

- "switching sides" indicates that this clue is an exchange clue
- A "Doomed king" can also be known as a "Lear"
- "True" is the definition which can also be defined as "real" which can be gained by exchanging the first letter of "Lear" ("L") with the last letter ("r")

2.2 Natural Language Processing

An inherent disparity between the languages used and understood by human beings and those which are interpreted by machines introduces the requirement for a system which is able to provide a form of compatibility between the two. Natural language can be described as a system of communication that has not consciously been invented (Collins, 2004) and typical examples of this are the languages used by human beings to communicate with each other. English and Korean are two such examples of natural languages and these evolve over time as words are added and removed and the rules that define the language are refined or adapted, meaning there is no immutable grammar for the language.

Conversely, formal (or artificial) languages are those which have been fabricated for a specific purpose, and can be described using precise and unambiguous mathematical rules (Jiang et al., 2010). An example of such a mathematical rule is the BackusNaur Form (BNF) grammar. Programming languages such as Java or Python are examples of artificial languages, as these have been created with a specific purpose in mind. The vocabularies of these languages are well defined and free from ambiguities, making interpretation by the languages corresponding compiler a black or white task where the given language to compile is either valid or invalid.

Natural language processing (NLP) allows for the manipulation of natural languages by a computational device (Bird, 2009) and there are a number of identified steps of the NLP process which have been developed, from scanning and parsing, to meaning extraction and integration. In the context of solving cryptic crosswords, a given clue would need to be broken down into meaningful components which would allow, as an example, the word play and the definition components to be identified and passed forward to the appropriate algorithms to identify the correct answer.

```
Reversible fasteners pinning baker, European
(TOP (NP (NP (NNP Reversible) (NNS fasteners)) (VP (VBG pinning) (NP (NN baker,)
)) (. European)))
```

Figure 2.1: Creating a parse tree for a given sentence

(Halpern, 2013)

2.2.1 The Steps of Natural Language Processing

Scanning

A number of steps are typically involved in the processing of a natural language. Assuming the input format of the natural language is in a textual form as sentences (rather than as speech or handwriting); these chains of text may be broken down into their separate components - the words which make up the sentences. This process is known as scanning, and may itself comprise of several sub-steps. The first step could likely involve breaking down the given textual input into separate sentences (Apache, 2013).

```
C:\Users\u0955187\Downloads\apache-opennlp-1.5.3-bin\apache-opennlp-1.5.3\bin>op
ennlp SentenceDetector en-sent.bin < input.txt
Loading Sentence Detector model ... done (0.068s)
Reversible fasteners pinning baker, European
```

Figure 2.2: A single sentence is already in the desired state

Once the separation of any sentences present in the input has occurred, these can be further divided into the separate words to achieve an array-like data structure, such as [*Reversible*, *fasteners*, *pinning*, *baker*, *European*], a process which is referred to as tokenising. It is important to consider the inclusion and placement of any punctuation provided in the input text, as this may provide some indication as to the separation of the cryptic crossword components and increase the chances of successfully calculating the answer. In the example provided, European corresponds to the definition component of the clue, where Reversible fasteners pinning baker relates to the word play component. Both components have been clearly separated by a comma, though not all cryptic clues are presented in this fashion.

Parsing

Parsing is a process used to ensure that a given sentence conforms to a grammar and thus showing that it is syntactically correct (McCluskey, 1999), and this process can be demonstrated by creating a parse tree to represent the structure of a given sentence. It may become apparent when attempting to solve the clues of a cryptic crossword that certain elements of the provided clue sentences bear some relation to the correct answer. To give an example, it may prove more likely that the adjectives or verbs present in the input are synonyms of the correct answer, where there may be little probability that the answer stems from a synonym of any present determiners (the, this, a, some) or prepositions (on, beneath, over, under). Likewise, adjectives or adverbs that hint at a particular action such as *reversible* or *backwards* could reveal themselves as more likely to be indicators that describe in what way the clue should be manipulated.

Referencing

The ambiguous nature of cryptic clues may not allow for a straightforward parse of an already ambiguous (natural) language, possibly contributing to the complexity of the process of solving. It may be the case that multiple parses of the same clue have to be taken forward to be processed before the solution can be found. Once a parse tree has been selected for the given input, a number of steps can be performed, which attempt to reduce the level of ambiguity. Lee McCluskey outlines four methods which can be applied to achieve this:

- **Contextual Disambiguation:** Such as referring to segments of a previous piece of text. Fred Bloggs may have been declared in a previous sentence, and from that point referred to as he.
- **Physical Constraints:** This refers to the literal meaning of a sentence of phrase, and uses known facts to assist in the reduction of ambiguity. The phrase I answered the door in my pyjamas could be taken to mean that the subject was either wearing their pyjamas when they answered the door, or that the door was wearing the subjects pyjamas when they answered it. As it is unlikely that the door was physically wearing pyjamas, the parse suggesting this could be disregarded.
- **Default Roles in Known Verb Structures:** Some verbs may only be applied to a subset of subjects in order to make reasonable sense. In the example sentence fruit flies like a banana, one parse may take fruit as the noun, and flies as the verb. In this scenario, fruit is physically incapable of flying, and so an alternative parse which taken fruit flies as the noun-phrase is more likely to be the correct parse of the sentence.
- **General Defaults:** To say that one is going to visit an old friend may imply that the friend has been known for a long time, or that the friend is of old age. While the correct interpretation may be context specific, the likelihood is that one particular interpretation will hold precedence over another. In this case, the friend is likely to have been known for a long time.

(McCluskey, 1999)

A parsing algorithm is used to obtain a complete parse tree for a given sentence, providing insight into the complete structure of the sentences. It is possible that this may not be required for the task at hand; a full parse may provide more information than what is required to solve a cryptic clue. An alternative mechanism exists, chunking, which may also be referred to as *light parsing*. This technique divides a sentence into a series of non- overlapping chunks of text, which provide an overview of the input sentence (Litman, 2003). One of the major benefits of this technique is the removal of the need to resolve ambiguity, as chunks of the sentence are non-recursive, unlike parsing which produces components of a sentence which may be nested in each other.

```
Reversible_NNP fasteners_NNS pinning_VBG baker,_NN European._.
```

Figure 2.3: Assigning *Parts of Speech* tokens to an input sentence

Meaning Extraction

Meaning extraction refers to the process of taking the parse and storing it in a way which accurately represents the information it models (McCluskey, 1999). For example, the following cryptic crossword clues all map to the same answer, and whilst they are written differently, they each imply the same meaning of *DRILL*:

- *Doctor gives patients exercise*
- *Doctor needing a doctor for practice*
- *GP not fit for practice*

(Gordius, 2003)

Effective meaning extraction will ensure that each distinct clue may help to ensure that variations of the same will be mapped to a single, internal representation and this will allow for the answer to be retrieved from a database without having to recalculate what has already been processed once.

2.2.2 Natural Language Processing Libraries

There exists a range of NLP libraries that offer many natural language processing functions for a variety of programming languages. Development on a number of these libraries has become stale, and others are available only for use after purchasing a commercial licence. Below are a selection of available libraries which have recently been updated and are available under some form of free software licence.

Library	**Licence**	**Last Updated**	**Supported Languages**
OpenNLP	Apache 2.0	April 2013	Java
NLTK	Apache 2.0	November 2012	Python
Stanford CoreNLP	GNU General Public Licence	June 2013	Java

Table 2.1: A comparison of available NLP libraries

OpenNLP

OpenNLP provides the ability to carry out a range of natural language processing functions, including parsing, chunking and name finding, where the latter aims to recognise proper nouns in specified text such as person names. The features of the library utilise a maximum entropy model, and this allows the performance of the software to be enhanced as it uses training data to learn (Apache, 2013). This allows for each component of the toolkit to refine as the subset of training data increases over time, with the aim of increasing the accuracy of the corresponding NLP component. The library is available for use in a Java environment or can be accessed directly through a command line interface, but the general usage of the library requires providing an applicable model and language data as input, for which the model will be utilised for the language processing.

Models A number of models are provided for use with the NLP library. These include, but are not limited to, models to assist in the process of parsing, chunking and detecting sentences. A number of libraries also exist which allow for the identification of person names, company names, times, dates or location in input text. These existing models may be further trained in a bid to increase their effectiveness.

DocumentCategoriser Another prominent feature of the library is the ability to classify input text into a range of predetermined categories using the aforementioned maximum entropy model. Once a model has been created, which contains a series of example inputs along with their corresponding categories, further inputs to the system will be paired with a *best outcome* classification.

```
<terminated> NLP [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (Nov 1, 2013 3:18:08 PM)
Category for: Posy says no, according to Spooner is: SPOONERISM (0.18832631809541703)
Category for: Preparation for documents once Spooner's transporting luggage is: SPOONERISM (0.005477603261500726)
Category for: Preserve holy one appointed by Spooner is: SPOONERISM (0.0034704375965808894)
Category for: Theme''s proper order reversed is: REVERSAL (0.016981012268904402)
Category for: This compiler going to the stake? Quite the reverse! is: REVERSAL (3.198100077871112E-4)
Category for: This vehicle reverses into its club is: REVERSAL (0.01006333130189762)
```

Figure 2.4: A demonstration of the OpenNLP DocumentCategorizer

NLTK

The Natural Language Toolkit is an alternative library which exists with an open-source licence variant. NLTK exists for use with the Python programming language and possesses a similar infrastructure to OpenNLP, where trained models are used to allow each component of the software package to function correctly (NLTK Project, 2012). The package is supplemented by a comprehensive e-book, detailing and providing examples of the usage of each component available.

2.3 Mobile Platforms and the Market

The demand for applications to be portable on mobile phones and tablets has become a fashion as well as accessibility on the go. In the current market Cryptic crosswords are available in a variety of newspapers, magazines and online websites. These are everyday media in which a person can access their favourite Cryptic Crosswords and participate in. With the expansion of mobile phone and tablet applications research has shown that a unique Cryptic Crossword solver has not been incorporated in any of the mobile operating systems. There are applications, which can be downloaded, that are pre-solved crosswords but there is no real solver, which solves Cryptic Crosswords in real time. In fact research has shown that there is only a very small handful of Cryptic Crossword applications across all mobile operating systems.

2.3.1 Mobile Operating Systems

The Oxford English dictionary states an "Operating System" as 'the low-level software that supports a computers basic functions, such as scheduling tasks and controlling peripherals' (Dictionaries, 2011).

A mobile operating system has the same definition but supports the basic functions for handheld devices such as mobile phones and tablets. In the current market the term mobile operating system is associated with smartphones rather than mobile phones due to the powerful processors, which are embedded in the mobile phone devices.

The official first mobile operating system for smartphones was by IBM introduced in 2000 as the 'Simon' but it was Ericson who created the first all in one smartphone with the Symbian OS, which incorporated a keyboard hence, the term smartphone was introduced. This ran on various mobile phones by companies such as Ericcson, Samsung but predominantly on smartphones by Nokia. This created a path for a market, which was to be dominated by others in the coming future (Akman, 2013).

By 2007 from the back of a successful campaign selling music devices known as the iPod, Apple introduced the Apple IPhone which came with their fist mobile operating system the IOS which was a full operating system used from the Mac OS X 10 (Apple, 2007).

By July 2008 Apple released IOS 2.0 and with this came the app store. This was a revolution to the mobile market allowing a platform for third party developers to sell and market their own applications for the mobile operating system. On the 7th January 2013 Apple announced that they have had more than 40 Billion downloads of apps through their app store (Apple, 2013).

In September 2008 Google released their own version of a mobile operating system Android which had its similarities of marketing with its very own app store known as

the android market which since has been rebranded to the Google play store.

In April 2009 Blackberry also launched its own application store called the Blackberry World, which works with their mobile operating system the Blackberry 10 (Cha, 2009).

Finally windows released their version of an app store for distribution of applications on October 26th 2012 (Businessline, 2012).

Although there are now several platforms that run on a mobile device, Android and iOS combine for 91.1% of the Worldwide Smartphone OS Market (IDC, 2013).

This shows that although blackberry has been in the market since 2009 there isnt much of a rise to interests in their apps and while Windows is fairly new it has a lot of ground to cover to catch up with their main competitors.

For the purpose of this project it is pretty clear to what is demanded from consumers in the real world with Apple and Google being the two main competitors for mobile applications. In order to decide on what platform the project will be suitable for to design, maintain and deploy a good working product below is the following page contains a table which covers some of the reasons which could be possible to allow the team members to come to a decision to what pathway the project will be going in.

Platform	**Programming Language**	**Open Source**	**Open API/SDK**	**License**
Android	Java	Yes	Yes	Apache
IOS	Objective C	N/A	Yes	Proprietary
Blackberry	Java, C++	N/A	Yes	Proprietary
Windows	C, C++, C#	N/A	Yes	Microsoft

Table 2.2: A comparison of mobile platforms

Platform	**Latest Version**	**Debugging**	**Hardware / Software Requirements**	**Emulator**
Android	4.3 Jelly Bean	Yes	Windows / Mac	Yes
IOS	IOS 7	Yes	Intel Based Mac	Yes
Blackberry	Blackberry 10	Yes	Windows / Mac	Yes
Windows	Windows Phone 8	Yes	Windows	Yes

Table 2.3: A further comparison of mobile platforms

Platform	Underlying OS	Development Environment	Submission To App Store	Development Cost
Android	Linux	Eclipse	Unlimited	$25 One Off cost
IOS	Darwin	XCode	Unlimited	£60 Yearly / $99
Blackberry	Blackberry OS	Eclipse	100 Per Year	$100 One off cost
Windows	Windows	Visual Studio	Unlimited	$19 Yearly / Free for Dreamspark Students

Table 2.4: More comparisons of mobile platforms

Although these are the four main types of platforms available in the market for mobile development there is a growing amount of organisations, which are developing tools to allow developers to create cross platform application with ease. Two of these are Appcelerator and Adobe AIR.

Platform	Programming Language	Open Source	Open API/SDK	Underlying OS and License	Development Environment
Appcelerator	JavaScript	Yes	Yes	Linux Apache 2.0	Eclipse Based IDE / Titanium Studio
Adobe AIR	ActionScript, HTML, CSS, JavaScript	No	Yes	Darwin Proprietary	Adobe AIR

Table 2.5: A comparison of cross platform development tools

2.3.2 Appcelerator

Appcelerator is a platform created by Appcelerator Inc to allow developers to create cross platform native applications for Android and IOS. They later introduced compatibility for Blackberry 10 and are in the process of developing for Windows Phone. The main development environment used to create applications is an Eclipse based IDE known as Titanium Studio. Developers can create great looking native apps using JavaScript. The use and license of Appcelerator falls under Apache 2.0.

2.3.3 Adobe AIR

Adobe Integrated Runtime (Adobe AIR) uses adobe tools such as flash to allow developers to create platform independent web apps. Unlike Appcelerator this means that applications created can only be web based and not native. For this reason a lot of developers avoid using Adobe AIR. It supports all the major vendors for mobile applications but apps created in Adobe AIR are a lot slower.

2.3.4 Native Apps

A native app is a platform independent application designed to work with a particular mobile OS. The app is installed on the device whether its a smartphone or a tablet.

Advantages

- Faster at accessing device features such as camera and accelerometer
- Easy to find in app stores such as Google Play and Apple app store
- Secure as they go through an approval phase from vendors

Disadvantages

- Expensive to develop
- Expensive to maintain
- Approval process can take from days to months
- Support of the app can be hard to maintain due to different people using different versions of operating system installed.

2.3.5 Web Apps

A web application is really a website designed to look and feel like an app. This is wrapped by the web browser, which means an Internet connection is required to use the app. Google Chrome and Safari are examples of Web apps.

Advantages

- Easy to maintain
- Can be compatible with any device
- Approval is not required
- Easy to maintain and update without affecting the user to update the software

Disadvantages

- Limited to what can be accessed on the devices
- Hard to provide support over various Web browsers
- Not easy to find and promote, users will have to browse websites
- Can be insecure due to the application being web based

There are various platforms available for mobile devices. The most popular platforms are the as previously mentioned. Although they are in contest the use and popularity of platforms vary from region to region. An article published by Sabri (2013) states that windows phone is the most popular platform in Latin America. Although the platform was produced later than the other platforms this has become popular due to the lower prices of devices. Devices such as the IPhone and the IPad are a lot dearer and can cost a user a couple of hundred pounds. In September Forbes reported that the most popular platform on mobile devices is Android.

Germany	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	78.7	78.7	0.0
BlackBerry	0.6	0.5	-0.1
iOS	11.1	9.5	-1.6
Windows	3.8	8.8	5.0
Other	5.7	2.6	-3.1

GB	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	62.7	56.3	-6.4
BlackBerry	10.1	3.7	-6.4
iOS	21.4	27.5	6.1
Windows	4.5	12.0	7.5
Other	1.3	0.5	-0.8

France	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	62.7	63.3	0.6
BlackBerry	8.2	4.2	-4.0
iOS	13.5	17.5	4.0
Windows	5.6	10.8	5.2
Other	10.2	4.1	-6.1

Italy	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	58.6	71.6	13.0
BlackBerry	3.9	2.1	-1.8
iOS	15.0	14.4	-0.6
Windows	10.3	9.5	-0.8
Other	12.2	2.4	-9.8

Spain	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	84.9	90.8	5.9
BlackBerry	5.6	0.7	-4.9
iOS	2.8	5.3	2.5
Windows	2.1	2.2	0.1
Other	4.7	1.0	-3.7

USA	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	60.7	55.1	-5.6
BlackBerry	2.1	1.8	-0.3
iOS	33.9	39.3	5.4
Windows	2.6	3.0	0.4
Other	0.8	0.8	0.0

China	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	63.2	72.4	9.2
BlackBerry	0.2	0.0	-0.2
iOS	23.5	20.8	-2.7
Windows	4.7	2.1	-2.6
Other	8.5	4.7	-3.8

Japan	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android		48.6	
BlackBerry		0.3	
iOS		47.4	
Windows		0.8	
Other		2.9	

Australia	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	65.9	62.1	-3.8
BlackBerry	1.6	0.5	-1.1
iOS	25.8	28.7	2.9
Windows	3.7	6.5	2.8
Other	3.1	2.2	-0.9

EU5	3 m/e Aug 2012	3 m/e Aug 2013	% pt. Change
Android	68.8	70.1	1.3
BlackBerry	5.8	2.4	-3.4
iOS	14.1	16.1	2.0
Windows	5.1	9.2	4.2
Other	6.1	2.1	-4.0

Figure 2.5: Mobile phone market share September 2013

(Jones, 2013)

2.3.6 Currently available Cryptic Crossword apps

After performing various research on the apple app store, the Google play store and blackberry world, it was discovered that there is not many Cryptic Crosswords available

to download. There were two, which could be clearly defined as Cryptic Crossword applications, and these have been analysed in the following sections.

2.3.7 Puzzler Super Cryptic Crosswords

Platform: Apple iOS
Price: £3.99
Compatibility: iOS 4.3 or Later
Website: https://itunes.apple.com/gb/app/puzzler-super-cryptic- crosswords/id616060420?mt=8

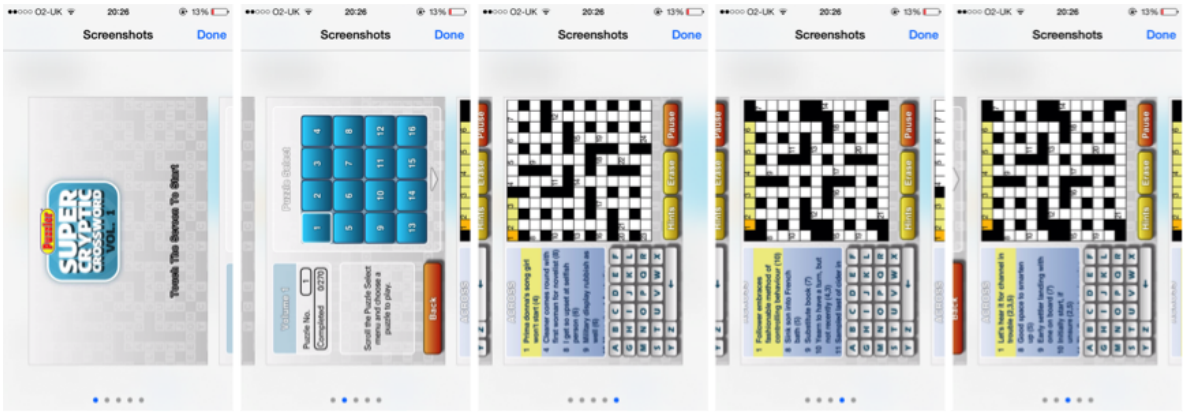

Figure 2.6: Screenshots of Puzzler Super Cryptic Crosswords on IPhone

This application contains 270 pre built crosswords. The interface is really easy to use and it clearly shows what has been completed and what needs to be completed. Whats nice about this application is that the crosswords are in various levels and as you complete one crossword you can move on to the next. What was noticed is that the application already stores the answers and also the application allows hints, which makes it a little easier to use. The other noticeable thing about this application is that it didnt fully use the native features of the mobile phone. Like the keyboard is a custom keyboard.

2.3.8 Cryptic Crossword

Platform: Apple iOS, Android
Price: Free with 2 puzzles on Apple devices- In app purchase avalable - £1.99 on Android Devices
Compatibility: iOS 6.0 or Later
Website: https://itunes.apple.com/gb/app/cryptic- crossword/id661608021?mt=8

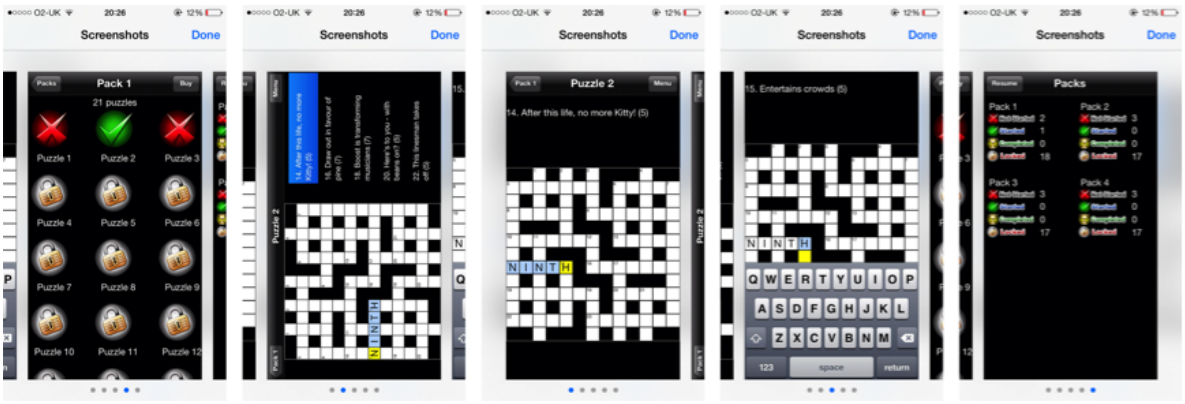

Figure 2.7: Screenshots of Cryptic Crossword on IPhone

The Cryptic Crosswords app comes with 4 different packages and a total of 81 puzzles. Each pack can be bought for £0.69 or all 4 for a discounted price of £1.99. Each package contains various cryptic crossword puzzles. After having a play around with this application at first point of contact it is noticed that look and feel of the application is not that great. The use of colours and the layout styles of the application can be better. The app consists of 81 puzzles, which can be played only after each crossword has been completed. Some of the features of the app are:

- Checking answears
- Revealing a letter, a word or the whole puzzle
- Clearing the puzzle
- Moving the character bar to next box
- Greying out completed clues

These are some of the features the application uses but there are plenty more.

2.4 Web Services

Over the past decade web services have exploded into the computing space. However the concepts that underlie web services are not new. Web services originally evolved from the Remote Procedure Call mechanism that was found in a software development framework used in the 1990s (Distributed Computing Environment) (Kalin, 2013).

During the late 1990s, XML-RPC was developed, which was a stripped down, light weight version of the Remote Procedure Call mechanism. The XML-RPC system only supported a small number of data types along with a number of simple commands. XML-RPC contained two key features, which are use of XML serialise/deserialise for data types and the reliance on HTTP for transport XML-RPC (Kalin, 2013).

XML-RPC is designed to be as lightweight as possible, and thus can be supported on a wide range of devices. XML-RPC was ultimately implemented fully and became known as SOAP. As well as SOAP, another implementation of XML-RPC occurred, which was entitled REST. Both of these technologies fall under to the term web services.

Since 2001 a vast range of companies have adopted the web services movement including (but not limited to) IBM, Oracle, Hewlett-Packard, Amazon, Google, Facebook and Twitter (Sullivan, 2001; Kalin, 2013).

Web services generally tend to reside upon public networks such as the Internet. However it is possible for a web service to run upon a private network, such as a companys internal Intranet.

2.4.1 What are Web Services?

Although there are many companies adopting web services, the term web service has a diverse and loose definition (Kalin, 2013). During the initial explosion, many providers created heavily detailed plans upon the direction of their web service, but failed to exactly define what a web service is.

It was only until the explosion subsided that authors were able to define what a web service is (Kalin, 2013). Kalin (2013) Kalin highlighted three common characteristics between web service providers:

1. Can be thought of as a 'webified application'
2. Typically delivered over Hyper Text Transport Protocol (HTTP)
3. Typically has some form of distributed nature allowing for components to be deployed and executed across multiple devices.

For the purposes of this project a web service will be defined as:

> A service that contains one or more software components that are designed to allow machine-to-machine interaction over a network using standard protocols.

Web services follow the client-server model, which is the standard architecture for accessing a website. However unlike the traditional approaches to client/server models (such as a web server/web page setup), web services do not provide the end user with a Graphical User Interface (GUI).

The web service will provide the end user with machine readable data — i.e. the data must be put into a pre-defined GUI. This architectural design concept is not new and has been around for a number of years. Web services often can be thought of as imitating mainframes — i.e. a 'dumb terminal' sends a request to a service hosted upon a central computer system.

Web services can be broadly categorised into the distributed software systems category (Kalin, 2013). Broadly speaking a distributed software system is a system that is often split up into various components. Each component can run upon a separate physical machine, and is able to communicate with other parts of the system by passing 'messages' around. Although a web service does fit into that broad definition, there are several features that are unique to a web service.

Firstly web services heavily depend upon open, industry-standard, vendor-independent protocols such as HTTP, JSON and XML. By adding networking, data formatting and security features, web services can effectively lower start-up costs and promote interoperability between new and existing services (Kalin, 2013).

It is the interoperability that allows web services to promote language transparency.

This means that web services and client programs do not need to be programmed in the same language. Many of the popular languages (e.g. C/C++, Java, and Python) provide inbuilt libraries or frameworks in support of web services (Kalin, 2013).

Finally web services are designed to be modular in design. This allows new services to be brought online in staggered stages, as well as allowing for laying of existing services. Again as previously mentioned each new service, can be written in the same language as the last service, or use a completely new language (Kalin, 2013).

2.4.2 Web Service Categories

Web services can be divided into two distinct groups — SOAP based and REST-style (Kalin, 2013). Interestingly the distinction could be described as being little at most, but they are not necessarily directly compatible with each other.

SOAP

SOAP originally stood for Simple Object Access Protocol, but is often referred to as Service Oriented Architecture (SOA) Protocol (Kalin, 2013). At a glance the name change doesn't appear to be too trivial, but it is acutely an example of the technology becoming better defined (Kalin, 2013).

SOAP utilises concepts that can be seen throughout the industry, but none more so than the use of XML. One of the major advantages of XML, is that it is able to provide flexible, self-describing data structures that can easily be produced and read.

SOAP tries to imitate the postal system — i.e. allowing two machines to send and receive letters. In this analogy, the letter is the raw XML data, and the envelope is an additional data layer that wraps around the letter. The envelope adds additional information to the request, such as which operation is being requested, and may also include authentication and session information in envelope headers (Gershon, 2004).

In order to ensure one client or service can 'talk' to another service, SOAP responses must use a Web Services Definition Language (WSDL). The WSDL defines the inputs (e.g. parameters), the outputs, the operations, the protocols and the network addresses that are required and used by the service (Gershon, 2004).

The underlying implementation is loosely coupled with WSDL, which means the provider is able to change the implementation, without negatively impacting the end service users. It is the configurable services aspect that is the central concept behind all service oriented architectures (Gershon, 2004).

REST

REST stands for Representational State Transfer, and is a relatively new architecture for creating web services. Despite its relatively new architecture it is actively used by some of the larger vendors e.g. Google and Amazon (DOSPINESCU and PERCA, 2013).

REST relies upon the emerging architecture known as resource-oriented architecture. Essentially, these resources are a number software components that can be combined together to create reusable functionality.

As well as using a resource-oriented architecture, REST makes clever and effective use of open standard web technologies, such as the Hypertext Transfer Protocol (HTTP), the Uniform Resource Identifier (URL) and the Extensible Mark-up Language (XML) (DOSPINESCU and PERCA, 2013).

Although not all of the features have been implemented, (mostly because they are layout properties rather than data properties) the major concepts found in web technologies have been implemented and the most notable of these features are:

1. Data from the client is transmitted to the server via the URI
2. The server will perform the operation described by the HTTP method (such as GET, DELETE)
3. The URI for each resource will contain the server name and address

As previously mentioned, HTTP methods are widely used within REST. A HTTP method will describe the necessary action (Create, Read, Update and Delete — CRUD) that is required to be performed by the server (DOSPINESCU and PERCA, 2013).

The HTTP methods follow another standard in terms of the basic functions of a database management system. It must be said that REST and the HTTP protocol are mutually exclusive — REST doesn't require HTTP (DOSPINESCU and PERCA, 2013).

Table 2.6 describes common HTTP verbs and the associated CRUD operation.

HTTP verb	**CRUD operation**
POST	Create a new resource
GET	Read a resource
PUT	Update an existing resource
DELETE	Delete a given resource

Table 2.6: HTTP verbs mapped to the associated CRUD operation.

There are additional verbs, such as HEAD, TRACE, CONNECT, OPTIONS and INFO. Some of the additional verbs may not be implemented by the server and/or service for

security reasons. Every HTTP request will include a verb to indicate which CRUD operation should be performed upon the resource (DOSPINESCU and PERCA, 2013).

SOAP vs REST

Both REST and SOAP utilise standard protocols when communicating, and also originate from the same/similar specification. The real difference between the two technologies is that SOAP utilises it's own application protocol by extending current protocols — namely HTTP.

This causes a number of issues, such as protocol standardisation. Although SOAP is based upon the HTTP protocol, each client will have to correctly understand the new extended protocol — via an additional layer of software or libraries. This adds weight to the overall technology.

SOAP describes functions, and the types of data, which requires large amount of documentation in order to use the service. As well as this there are several protocols and technologies that directly relate to it, such as Web Services Description Language, Web Servicing Addressing, XML Schema Definitions.

All binary data that is to be transmitted must be first encoded in a supported format (e.g. base64), which increases processing power at both the client and server ends. All requests are transmitted via XML, which is much slower to parse and interpret than other text-based human readable data, such as JavaScript Object Notation (JSON).

REST on the other hand is based upon uniform interfaces. This means the various clients will have a small understanding of the web service, but not necessarily how it operates or what it will return.

REST doesn't need to operate over HTTP, and doesn't contain the complexity that SOAP provides. Rather than utilising XML, REST uses the standard HTTP methods to describe what a service should do. For example obtaining a resource would use GET, and for creating a resource PUT would be used.

Clients do not require additional REST supporting libraries. As long as the language supports HTTP, the client will be able to consume a REST HTTP service easily.

Unlike SOAP, REST can deliver binary data without having to encode, and responses can be formatted to either XML or the more popular JSON (due to speed increases).

2.4.3 Clients

As previously mentioned, the broader web server architecture follows the client-server application model. When designing a client-server application, a decision has to be

made as to which operations (or parts of operations) should be performed upon the client and the server.

This decision is vitally important as it can affect the speed to which a system can be brought to market. It might also affect any additional extensions or updates that the system might receive in the future, as well as affecting the design flexibility.

In order to simplify the design the client will need to fall into one of the two categories — 'thin' client or a 'thick' client.

Thin

A thin client is a computer system that depends largely upon a main server, or a number of servers in order to complete any computation tasks. The client has no knowledge of how to process data, it simply knows how to pass data to another entity, and receive data from another entity.

A recent example of a thin client is Google's Chromebook. Unlike typical computers where by the applications are installed locally upon the computer, the Chromebook allows for applications to be installed within the cloud — upon an external server.

The thin client design presents a number of advantages and disadvantages. Firstly an application that is hosted upon a central server can be easily updated — as there is only one code base. Once the application has been updated, this will be pushed immediately to all thin clients.

This obviously provides an advantage in some use cases such as trying to sell goods over the Internet. For example if a product's price changes, the update will only need to be applied once to the central server, rather than having to update all clients wishing to purchase the product.

Thin clients will utilise powerful servers to do the majority of the processing. This allows for the thin clients to be less powerful, and hence the overall costing to reduce.

However this will mean that thin clients will have poorer response times. The main reason for this is the fact that the majority of the operations are being complete upon another machine (potentially many miles away). Simple operations such as populating a menu, might require a request to the main server, thus increasing the overall time to achieve something.

Resources within a thin client network will need to be managed more effectively. Thin clients will use more bandwidth upon the network, and will make more connections to the server. This would require the server to be able to handle lots of potentially fast and slow connections, with each connection using a wide range of internal server resources (CPU, Memory etc).

Thick

A thick client is a computer system that has little dependency upon a main server, or a number of servers in order to complete computational tasks. The client will still require a limited connection to a server, but will not use the connection as often in comparison to a thin client. A thick click will often be able to perform many operations without a connection to a network.

An example of a thick client would be a standard desktop installation. The desktop installation might provide various pieces of software that are installed locally upon the computer. For example the computer would be able to produce various documents regardless of the state of the network connection.

The thick client design presents a number of advantages and disadvantages. Firstly due to the fact that clients are able to do more of the computational work, server specifications do not need to be as high. This allows for cheaper servers to be purchased, and few overheads in terms of running and maintenance costs.

This will also lead to an increase in server capacity, again due to the fact that the client is carrying out more work. This ultimately means that the server is required to do less work, and can hence support a larger number of users.

Thick clients have an increased advantage over thin clients in terms of network connectivity. Thick clients do not require a constant connection to a server. This in turn frees up bandwidth that is being used upon the network, as well as reducing server loads.

Finally the end user is able to store files and applications locally upon the machine. This in turn allows for a faster application start up time, and a reduced file access time. Hence increasing the speed of operations, as well as reducing bandwidth upon the network.

However thick clients are more expensive to purchase, deploy and maintain. The reason being is that there will be more computers with higher specifications. This can lead to more expensive repair bills, should systems fail.

Fixing and troubleshooting become more difficult, simply because there are more machines to troubleshoot and fix should problems occur. This is obviously not a problem if there was a central server, such as found within the thin client model.

Chapter 3

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).

3.1 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).

3.1.1 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.

3.1.2 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

3.2 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).

3.2.1 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

3.2.2 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

3.3 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).

3.3.1 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)

3.3.2 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

3.4 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).

3.4.1 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

3.4.2 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

3.5 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 4

Software Specification

Cadle et al. (2010) states that there is a standard hierarchical approach in structuring requirements. Table 4 outlines the main four categories:

General	**Technical**	**Functional**	**Non-Functional**
Business constraints	Hardware	Data entry	Performance
Business policies	Software	Data maintenance	Security
Legal	Interoperability	Procedure	Legal and Access
Branding	Internet	Retrieval	Backup and Recovery
Cultural			Archiving and Retention
Language			Maintainability
			Business Continuity
			Availability
			Usability
			Capacity

Although these are some of the sections usually found in the structure of requirements there are several books which state that each section depends on the type of project being undertaken. Robertson and Robertson (2013) mention that 'a requirement is something the product must do to support its owners business, or a quality it must have to make it acceptable and attractive to the owner.' The aim of gathering the requirements has always been to ensure that all ambiguity are ironed out before a product is developed. The following sections have been derived from the book by (Robertson and Robertson, 2013). The book contains a very usefull template called the Volere template which aims to recognize as much information required when engineering requirements.

4.1 Purpose

4.1.1 Background of the Project

The purpose of this project is to produce an app that will be compatible with the three main mobile operating systems, iOS, Android and Blackberry OS. The app is to be able to solve given clues from cryptic crosswords which are widely available on publications such as the Burgundian newspaper. There is no current form mobile application in the current market which solves cryptic clues. The produced product is to allow the end user to solve some if not all types of clues which have been discussed in section 2.1.3.

4.1.2 Project Goals

The main goals of the project are:

1. Be able to solve a given clue.
2. Identify the type of clue it is.
3. Show a stack trace of how the clue was deduced.
4. Be able to store previous solved clues.

4.2 The Client, the Customer and Other Stakeholders

4.2.1 The Client

Dr Hugh Osborne
Senior Lecturer
University Of Huddersfield
h.r.osborne@hud.ac.uk

4.2.2 The Customer

The intended customer of the product are users of smartphone and tablets whom are looking to solve all those unsolvable Cryptic Crosswords. The applications will be deployed on the app market for the three listed mobile operating systems which means that the app will be available to anyone who has a compatible device with the required software. The physical deployment of the application is out of the project scope so a price for the deployment will not be discussed.

4.2.3 Other Stakeholders

For the purpose of the project the other stakeholders are as follows:

Project Supervisor
Dr. Gary Allen
Senior Lecturer
University Of Huddersfield
g.allen@hud.ac.uk

Project Examiner:
Sotirios Batsakis
University Of Huddersfield
s.batsakis-STA@unimail.hud.ac.uk

Internal Moderator
Collin Venters
Senior Lecturer
University Of Huddersfield
c.venters@hud.ac.uk

4.2.4 The Hands-On Users of the Product

To determine the users of the product it was decided yo carry out a survey in order to identify the potential users that actually will be using the product.

The following questions were asked:

- Do you play Cryptic Crosswords?
- How Often do you play?
- Where do you play?
- Do you often finish them?
- If no to the previous question, what reason don't you finish them?
- What is your age group?
- When do you play Cryptic Crosswords?
- What gender are you?
- What is the Highest qualification you have?
- What platform is your mobile phone on?

The survey was conducted between (date) and (date). It was distributed across the Department of Computing And Engineering at the University Of Huddersfield, Facebook and Twitter. The results of this survey are shown in Figure 4.1

From these results it can be deduced that .....

Figure 4.1: Cryptic Crosswords Survey Results

4.3 Mandated 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.

4.4 Solution Constraints

[R1/1]The product shall be built for the following platforms:

1. Blackberry
2. iOS
3. Android

Rationale: The product it to be able to be used on the go.
Volatility: High

[**R2/1**]The product shall require an Internet connection.

Rationale: The product cannot work without an Internet connection.
Volatility: High

4.5 Naming Conventions and Terminology

4.6 Relevant Facts and Assumptions

4.7 The Scope of the Work

4.8 Business Data Model & Data Dictionary

4.9 The Scope of the Product

4.10 Functional Requirements

These are things the product must do.

e.g The product shall produce a result upon it given a clue within 30 seconds.

4.10.1 Data entry

4.10.2 Data maintenance

4.10.3 Procedure

4.10.4 Retrieval

4.11 Look and Feel Requirements

4.12 Usability and Humanity Requirements

4.13 Performance Requirements

4.14 Operational and Environmental Requirements

4.15 Maintainability and Support Requirements

4.16 Security Requirements

4.17 Cultural Requirements

4.18 Legal Requirements

4.19 Open Issues

4.20 Off-the-Shelf Solutions

4.21 New Problems

4.22 Tasks

4.23 Migration to the New Product

4.24 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.

4.24.1 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 4.1: Probability-impact table for the project's risks

4.24.2 Risk Classification

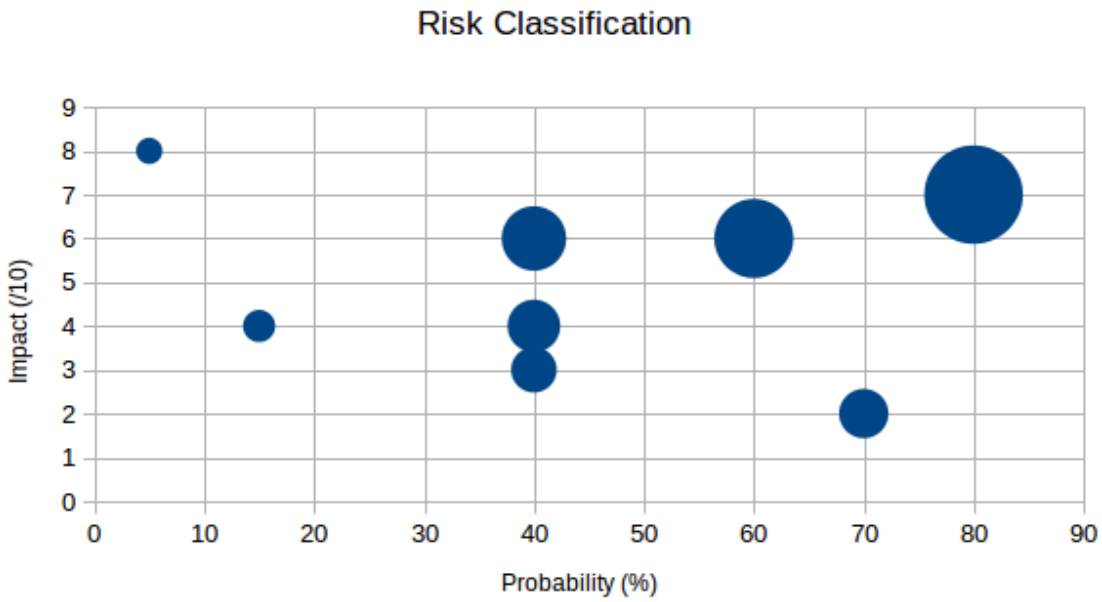

Figure 4.2: A visualisation of the identified risks of the project

Figure 4.2 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.

4.24.3 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.

4.24.4 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.

4.25 Costs

4.26 User Documentation and Training

4.27 Waiting Room

4.28 Ideas for Solutions

Bibliography

Osgur Akman. The smartphone revolution, October 2013. URL http://mail2web.com/blog/2011/05/smartphone-revolution-growth-smartphones-exchange-activesync/.

Apache. Apache opennlp developer documentation, October 2013. URL http://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html#tools.sentdetect.

Apple. Macworld san francisco 2007 keynote address, January 2007. URL https://itunes.apple.com/gb/podcast/apple-keynotes/id275834665.

Apple. App store tops 40 billion downloads with almost half in 2012, January 2013. URL http://www.apple.com/pr/library/2013/01/07App-Store-Tops-40-Billion-Downloads-with-Almost-Half-in-2012.html.

Steven Bird. *Natural Language Processing with Python*. O'Reilley Media, Inc., 2009.

Businessline. Microsoft to release windows 8 on oct 26, July 2012. URL http://search.proquest.com/docview/1321630361?accountid=11526.

James Cadle, Malcolm Eva, Keith Hindle, Debra Paul, Craig Rollaston, Dot Tudor, and Donald Yeates. *Business Analysis*. British Computer Society, second edition, 2010.

Bonnie Cha. Rim store crowned blackberry app world, March 2009. URL http://news.cnet.com/8301-17938_105-10188400-1.html?tag=mncol;txt.

John Collins. Languages, natural and formal, September 2004. URL http://www.uea.ac.uk/ j108/languages.htm.

Alan Connor. Cryptic crosswords for beginners: palindromes, November 2012a. URL http://www.theguardian.com/crosswords/crossword-blog/2012/nov/01/cryptic-crosswords-

Alan Connor. Cryptic crosswords for beginners: spoonerisms, March 2012b. URL http://www.theguardian.com/crosswords/crossword-blog/2012/mar/01/cryptic-crosswords-

Crossword Tools. Clue solver, October 2013. URL http://www.crosswordtools.com/cm/.

Cryptic Solver. Solver, October 2013. URL http://cryptic-solver.appspot.com/.

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

Oxford Dictionaries. *Concise Oxford English Dictionary*. Oxford University Press, twelth edition, 2011.

Octavian DOSPINESCU and Marian PERCA. Web services in mobile applications. *Informatica Economica*, 17(2):17 – 26, 2013. ISSN 14531305.

Prof Geoffrey Elliott. *Global Business Information Technology: An Integrated Systems Approach*. Addison Wesley, first edition, 2004.

Kathryn Friedlander and Philip Fine. Expertise in cryptic crossword performance. In *An exploratory survey*, Auckland, New Zealand, 2009. International Symposium on Performance Science.

Gary Gershon. Web services. *e-Doc Magazine*, 2004.

Gordius. Cryptic crossword no 22,717, January 2003. URL http://www.theguardian.com/crosswords/cryptic/22717.

John Halpern. Cryptic crossword no 26,091, October 2013. URL http://www.theguardian.com/crosswords/cryptic/26091.

IDC. Android and ios combine for 91.1% of the worldwide smartphone os market in 4q12 and 87.6% for the year, according to idc, February 2013. URL http://www.idc.com/getdoc.jsp?containerId=prUS23946013.

ISTQB Exam Certification. What is rad model- advantages, disadvantages and when to use it?, October 2013. URL http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/.

Tao Jiang, Ming Li, Bala Ravikumar, and Kenneth W. Regan. *Algorithms and theory of computation handbook*. Chapman & Hall, 2010.

Chuck Jones. Idc: Windows phone is the second most popular mobile platform in latin america, September 2013. URL http://www.forbes.com/sites/chuckjones/2013/09/30/iphones-market-share-down-prior-to-5s-5c-launch-with-windows-almost-double-digits-in-europe/.

Martin Kalin. *Java Web Services: Up and Running*. O'Reilly Media, second edition, 2013.

R.P. Knott and R.J. Dawson. *Software Project Management*. Group D Publications Ltd, 1999.

Diane Litman. Cs 1573: Artificial intelligence application development (spring 2003), February 2003. URL http://people.cs.pitt.edu/ litman/courses/cs1573s03/lec/chunking.ppt.

Thomas Leo McCluskey. Cha 2555 - artificial intelligence, January 1999. URL http://helios.hud.ac.uk/scomtlm/cha2555/lecture13_18.ppt.

John McManus. *Risk Management in Software Development Projects (Computer Weekly Professional)*. Routledge, 2003. ISBN 0750658673.

NLTK Project. Nltk 2.0 documentation, November 2012. URL http://nltk.org/data.html.

One Across. Crossword puzzle help, October 2013. URL http://www.oneacross.com/.

James Robertson and Suzanne Robertson. *Mastering the requirements process*. Addison-Wesley, third edition, 2013.

Sam Sabri. Idc: Windows phone is the second most popular mobile platform in latin america, August 2013. URL http://www.wpcentral.com/windows-phone-second-most-popular-mobile-platform-latin-america.

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

Tom Sullivan. Web services. *InfoWorld*, 23(11):38, 2001. ISSN 01996649.

Shuchismita Upadhyay. Acrostics, September 2008a. URL http://www.crosswordunclued.com/2008/09/first-and-last-letters.html.

Shuchismita Upadhyay. How to spot anagrams, August 2008b. URL http://www.crosswordunclued.com/2008/08/how-to-spot-anagram.html.

Shuchismita Upadhyay. Charades, November 2008c. URL http://www.crosswordunclued.com/2008/11/charades.html.

Shuchismita Upadhyay. Cryptic definitions, December 2008d. URL http://www.crosswordunclued.com/2008/12/cryptic-definitions.html.

Shuchismita Upadhyay. Decoding double definitions, October 2008e. URL http://www.crosswordunclued.com/2008/10/decoding-double-definitions.html.

Shuchismita Upadhyay. Digging out hidden words, August 2008f. URL http://www.crosswordunclued.com/2008/08/digging-out-hidden-words.html.

Shuchismita Upadhyay. Tune in to homophones, October 2008g. URL http://www.crosswordunclued.com/2008/10/homophones.html.

Shuchismita Upadhyay. &lit, literally so, August 2008h. URL http://www.crosswordunclued.com/2008/08/and-literally-so.html.

Shuchismita Upadhyay. Reversals, November 2008i. URL http://www.crosswordunclued.com/2008/11/reversals.html.

Shuchismita Upadhyay. Substitutions simplified, December 2008j. URL http://www.crosswordunclued.com/2008/12/substitutions.html.

Shuchismita Upadhyay. Containers, February 2009a. URL http://www.crosswordunclued.com/2009/02/containers.html.

Shuchismita Upadhyay. Deletions, March 2009b. URL http://www.crosswordunclued.com/2009/03/deletions.html.

Shuchismita Upadhyay. Letter exchange, December 2009c. URL http://www.crosswordunclued.com/2009/12/letter-exchange.html.

Shuchismita Upadhyay. Letter picking, April 2009d. URL http://www.crosswordunclued.com/2009/04/letter-sequences.html.

Shuchismita Upadhyay. Letter shifting, December 2009e. URL http://www.crosswordunclued.com/2009/12/letter-shifting.html.