SWANSEA UNIVERSITY

MENG COMPUTING

CSM14 - INDIVIDUAL PROJECT

Methodology and Requirements

Luke Hammond - 522196

Project supervisor: Dr N A Harman

January 5, 2012

Contents

1 Introduction

The aim of this document is to provide several key elements of this project. Firstly, a brief introduction to the project will be provided which covers the client and the goal of the project. Following this, a complete and concise list of requirements for each aspect of the system will be given.

Also presented in this document is a risk analysis section which identifies any potential problems that could occur during the course of the project. The final sections of this document cover the methodologies that will be used to develop the application and a project plan.

1.1 The Client

In 2004 Symltech Ltd was formed as a joint venture company to develop a remote worker management system for construction plant operator RNA Plant Ltd. The developed system called GlimpseTM comprises an MS SQL database, a desktop application, some middleware and a Windows Mobile application. Glimpse has been successfully used by RNA since 2005 bringing numerous efficiency benefits. In order for the system to continue in use, it needs some updating.

1.2 Project Goal

The client would like to move their current system to a thin client model relying on a web service to communicate between the distributed elements of the system. The goal of this project is to do just that using WCF services. The focus of this project will not be concerned with updating the middleware or current database but will be on the application currently being used in office. However, it is important for the developed system to work with the current database schema and middleware.

2 Requirements

2.1 Diary

The current system allows the user to do multiple job manipulation tasks from the 'Diary' screen. It is important that the same functionality is available in the web application version. The table below gives a complete and concise list of requirements that relate to jobs.

ID	Requirement Description
R-DI-1	The system must allow the user to create new jobs.
R-DI-2	When creating a job it must be possible to assign a job to a customer.
R-DI-3	The system must allow a job to be allocated to a site. Only sites related to the assigned customer may be chosen.
R-DI-4	It must be possible to assign an order to a job.
R-DI 5	The system should allow a job to be set as requiring or not requiring a signature.
R-DI-6	It must be possible to assign a job booked job times and actual job times.
R-DI-7	When a job is being created, the system should allow a booked site start time to be assigned.
R-DI-8	A job must also have a booked site finish time.
R-DI-9	It must be possible to assign booked travel time for a job.
R-DI-10	The system must allow a job to be allocated booked break time. Break time can be assigned a maximum value of 60 minutes and a minimum value of 0 minutes. 15 minute intervals between these values can also be used.
R-DI-11	It must be possible to set a job rate for a job. The system must set this value using the rate from the order assigned to the job. The value can also be overwritten manually.
R-DI-12	If a job has been completed, the system must allow the actual site start and site finish times to be adjusted manually.
R-DI-13	It needs to be possible to set a job as a "one off" or as a repeating job. A job must be able to repeat daily, weekly or monthly.
R-DI-14	The system must allow an allocation status to be set for a job. A job can either be allocated to an operator or subcontractor or can be unallocated.
R-DI-15	If a job has been allocated to an operator then the system needs to allow an operator to be chosen along with the plant that is to be used for the job.
R-DI-16	If a job has been allocated to a subcontractor then the system needs to allow a subcontractor to be chosen, the plant type to be chosen, an agreed price for the job and an order number to be specified.
R-DI-17	If a job has been created with an unallocated status it should be possible to assign an operator or subcontractor at a later point.
R-DI-18	The system must also allow comments to be wrote against a job.
R-DI-19	It must be possible to delete a job.
R-DI-20	The details of a job must be editable.
R-DI-21	The system should provide a search feature that makes it possible to find specific jobs.
R-DI-22	If a job has been completed then it should be possible to print a receipt locally or send a receipt via email.

2.2 Maintenance

Before the system can be used to create jobs and assign them to the workforce, it will be necessary to create Customers, the Workforce and Equipment & Services. As well as creation it should be possible to view, edit and delete them. These tasks can be executed in the 'Maintenance' screen of the current system. It is important that these features are also present in the new system. A list of requirements related to the tasks mentioned above are listed below in several tables. The first covers equipment and services.

ID	Requirement Description
R-MAES-1	The system should allow the user to create 'Plant Types'. A 'Plant Type' should be seen as a general object, rather than an actual object, that could be instantiated to create an real object. so, a *car* could be a general object, as it could refer to any car. But, when the registration of a car is know it can be classed as a real object.
R-MAES-2	It must be possible to give a 'Plant Type' a name (mandatory) and a description (optional).
R-MAES-3	The system must also allow the details of created 'Plant Types' to be retrieved.
R-MAES-4	It must be possible to edit the details of 'Plant Types'.
R-MAES-5	The system should provide a way of deleting 'Plant Types'.
R-MAES-6	It will be necessary for the system to be able to create instantiations of the general objects. The instantiations will be the actual pieces of equipment, vehicles or plant the company has available to use. These will be referred to as 'Plant'.
R-MAES-7	A 'Plant' must be assigned a Registration/ID.
R-MAES-8	A 'Plant' must refer to a 'Plant Type'.
R-MAES-9	A 'Plant' must also have a status. The status of a 'Plant' can either be available, unavailable or retired.
R-MAES-10	It must also be possible for a 'Plant' item to have several other optional fields. These are year, make, model, VIN, colour, date purchased, purchase price and current value.
R-MAES-11	The system should also allow a 'Plant' item to be assigned to a particular operator.
R-MAES-12	It should be possible to assign a 'Plant' item a maintenance schedule. This will include dates for the next MOT, next service, next check and road tax expiry. It must also be possible to add comments to the maintenance schedule.
R-MAES-13	It must be possible to record events concerning a particular item of 'Plant' so its history can be viewed at any time.
R-MAES-14	The system must allow 'Plant' items to be edited
R-MAES-15	The system must also allow the details of a 'Plant' item to be retrieved.
R-MAES-16	When new items of 'Plant' have been created, 'Plant' details have been changed or a 'Plant' item has been removed, the system must be able to send an updated 'Plant' list to each PDA/smartphone.
R-MAES-17	It must be possible to remove a 'Plant' item.
R-MAES-18	The system must allow a 'Service' to be created. A 'Service' is something that the organisation will carry out using a 'Plant' item or multiple 'Plant' items.
R-MAES-19	It is required that a 'Service' must have a name and a brief description.
R-MAES-20	It must be possible to assign a charge to a 'Service'. Either by the hour or in units.
R-MAES-21	A 'Service' that is to be charged by the hour needs to have three different rates. Namely, a standard hourly rate, an evening hourly rate and a weekend hourly rate.

R-MAES-22	If a 'Service' is to be charged per unit then it must have a standard unit rate and a weekend unit rate. It must also be possible to describe the unit type.
R-MAES-23	It must be possible to assign a minimum hire duration value for each service.
R-MAES-24	It must be possible to specify which equipment or 'Plant' are required to carry out the 'Service'. A 'Service' may require more than one piece of equipment or 'Plant'. In this case each item needs to be specified.
R-MAES-25	Each service is required to have a flag that denotes if a service requires, or does not require, multiple pieces of equipment or 'Plant' (Multi-plant).
R-MAES-26	The system should allow a 'Service' to be removed.
R-MAES-27	It must be possible to edit the details of a 'Service'.
R-MAES-28	The system must also allow the details of a 'Service' to be retrieved.

The table below covers requirements that relate to the workforce. The workforce covers users of the system, operators and subcontractors.

ID	Requirement Description
R-MAW-1	It must be possible to create users than can use the system.
R-MAW-2	A user is required to have a first name, surname, login name and a password.
R-MAW-3	Each user must be assigned one of three levels of rights to the system. These are full, read only and read and write.
R-MAW-4	It needs to be possible to assign a user a mobile number and a land line number, email address and home address. These are optional.
R-MAW-5	The system must provide a way of retrieving user details.
R-MAW-6	The details of a user must be editable.
R-MAW-7	It must be possible to remove users.
R-MAW-8	The system must provide a way creating operators.
R-MAW-9	It is required for an operator to have a first name, surname, nickname, login name and a password.
R-MAW-10	The system must also allow an operator to be assigned an employment status. The status can either be available (employed by the organisation) or left (no longer works for the organisation).
R-MAW-11	There must also be a way of setting an operator as unavailable for certain dates. Comments about why the operator is unavailable will also need to be recorded.
R-MAW-12	It will be necessary to retrieve details about an operators' unavailability.
R-MAW-13	The system must provide a way of assigning an operator a licence number, email address, land line number and mobile number and an address. These fields are optional.
R-MAW-14	It is important that the skills of each operator are recorded. A skill refers to which 'Plant Types' the operator can use.
R-MAW-15	The system must provide a way of retrieving an operators details.
R-MAW-16	It must be possible to edit the details of an operator.
R-MAW-17	It must be possible to remove an operator.
R-MAW-18	When new operators have been created, operator details have been changed or an operator has been removed, the system must be able to send an updated operator list to each PDA/smartphone.
R-MAW-19	The system must provide a way of creating subcontractors.
R-MAW-20	A subcontractor must have a name, contact name and a status (active or inactive).

R-MAW-21	It needs to be possible to assign a subcontractor an email address, mobile number, fax number and land line number.
R-MAW-22	A subcontractor must have an address associated with it.
R-MAW-23	It is important that the skills of each subcontractor are recorded.
R-MAW-24	It must be possible to retrieve the details of a subcontractor.
R-MAW-25	It must be possible to edit the details of a subcontractor.
R-MAW-26	A subcontractor needs to be deletable.

A list of requirements relating to customers are displayed in the table below. These requirements cover accounts, regional offices and sites which are all related to customers.

ID	Requirement Description
R-MAC-1	It must be possible to create a customer account.
R-MAC-2	A customer must have a name, status and an address associated with it.
R-MAC-3	The customer account status can either be OK, caution, stop or unknown.
R-MAC-4	It should also be possible to assign a customer account a phone number, fax number and an email address.
R-MAC-5	The system should provide a way of setting custom rates for the services a customer will receive. By default the standard rates will apply.
R-MAC-6	The system must allow customer account details to be edited.
R-MAC-7	The system must provide a way of retrieving customer account details.
R-MAC-8	It must be possible to delete a customer account from the system.
R-MAC-9	The system must provide a way of assigning a regional office to a customer account.
R-MAC-10	The regional office needs to have an address.
R-MAC-11	A regional office can also have an email address, office contact name, office phone number, office fax number and a contact phone number.
R-MAC-12	It needs to be possible to create sites.
R-MAC-13	A site needs to have an address associated with it.
R-MAC-14	It is also important that a site has contact details when linked to a customer. These can be an email address, contact name, site phone number, site fax number and contact phone number.
R-MAC-15	It is important to be able to link a site to a customer, but a site does not need to be linked to a customer when being created. The link between a customer and a site should be able to be broken without removing the site.
R-MAC-16	It needs to be possible to edit all parts of a site.
R-MAC-17	The system must provide a way of retrieving site details.
R-MAC-18	It must be possible to delete a site.

There are also three other sections in the ‘Maintenance’ screen of the current system. These are ‘Administration’, ‘In the Field’ and ‘Other’. The functionality provided by these sections is also required in the new system and the requirements relating to them are listed below. Like the previous three sections, the requirements will be split into three tables and will first cover ‘Administration’.

ID	Requirement Description
R-MAA-1	The system must provide a way of creating orders.

R-MAA-2	An order must have a date and a reference name. The reference name is mainly for the customer so that they can easily refer to an order. The same reference name can be used multiple times as each order has a unique internal reference. If a customer uses the same reference name, it allows jobs to be grouped for costing.
R-MAA-3	An order must have a customer assigned to it as well as one of their sites.
R-MAA-4	It must be possible to assign a service to an order.
R-MAA-5	Two optional fields that an order can have are a contact name and a description.
R-MAA-6	An order should be able to have limits assigned to it. The limits can be set either by a maximum value, number of jobs or an end date.
R-MAA-7	It should also be possible to create an order that is open ended so it does not have limits.
R-MAA-8	The system should provide a way of editing order details.
R-MAA-9	The system must provide a way of retrieving order details.
R-MAA-10	It should be possible to delete an order.
R-MAA-11	The system must provide a way of adding addresses.
R-MAA-12	It should be possible to edit the details of an address.
R-MAA-13	The system should provide a way of retrieving all the addresses in the system.
R-MAA-14	If an address is linked to a customer, site etc, then it should be possible to retrieve the address associations.
R-MAA-15	It should be possible to retrieve only the addresses that are not linked to a customer, site etc.
R-MAA-16	It should be possible to delete an address as long as it is un-linked.
R-MAA-17	The system should allow GPS coordinates to be associated with an address.

The next list of requirements are related to the ‘In the Field’ section. This section deals with the equipment the operators can use and have been assigned.

ID	Requirement Description
R-MAF-1	It needs to be possible to add a PDA/smartphone (device) to the system.
R-MAF-2	A device needs to have a unique reference ID and an ID code. Preferably, these should be the same.
R-MAF-3	The system must record the devices phone number, along with the date it was bought, the manufacturer and the model of the device.
R-MAF-4	A device also needs a status which can either be set to available or unavailable.
R-MAF-5	It should be possible to edit the details of a device.
R-MAF-6	A device must also be deletable from the system.
R-MAF-7	It needs to be possible to retrieve details about a device.
R-MAF-8	The system must provide a way of allocating a device to an operator.
R-MAF-9	It must be possible to change the device-to-operator allocation.
R-MAF-10	It must also be possible to unallocate a device from an operator.
R-MAF-11	The system must provide a way of allocating a ‘Plant’ item to an operator.
R-MAF-12	It must be possible to change the plant-to-operator allocation.
R-MAF-13	It must also be possible to unallocate a ’Plant‘ item from an operator.

The final section of the ‘Maintenance’ screen is titled ‘Other’. This section allows the user a way of

managing jobs that have been marked as requiring a digital signature but have not had one allocated. The requirements for this section are below.

ID	Requirement Description
R-MAO-1	It must be possible of retrieving all jobs that require a digital signature but have not had one allocated.
R-MAO-2	It should be possible to assign any of the jobs that require a digital signature to an operator. This is so that the operator can collect a signature.
R-MAO-3	The system must also allow jobs to be set as not requiring a signature. This may be done if manual paperwork has been completed.

2.3 Mapping

Mapping is another important part of the current application as it allows the application users to view a range of GPS related information. Such information can include the most recent location of an operator, address locations and the most recent location of a 'Plant' item. The GPS coordinates relating to the operators are sent from their PDA/smartphone to the database. Below is a list of mapping requirements.

ID	Requirement Description
R-MAP-1	It should be possible to retrieve the most recent GPS coordinates of an operator.
R-MAP-2	It should also be possible to retrieve an operators GPS coordinates that lay between two times.
R-MAP-3	It should be possible to retrieve the most recent GPS coordinates of a 'Plant' item.
R-MAP-4	It should also be possible to retrieve GPS coordinates of a 'Plant' item that lay between two specified dates and times.
R-MAP-5	The system should allow the GPS coordinates of the last known location of each operator to be retrieved.
R-MAP-6	The GPS coordinates of the last known location of each 'Plant' item should also be able to be retrieved.
R-MAP-7	The system should allow the current GPS coordinates of each operator to be retrieved. The term 'current' means up to 24 hours ago.
R-MAP-8	The system should allow the current GPS coordinates of each 'Plant' item to be retrieved.
R-MAP-9	It should be possible to retrieve all GPS coordinates that refer to a particular job.
R-MAP-10	The GPS coordinates of where an operator has logged in or out of the application, on their PDA/smartphone, should be retrievable. This should be between two dates.
R-MAP-11	If an address has GPS coordinates associated to it, then the system must allow the coordinates to be retrieved.

2.4 Invoicing

Another section in the current application allows the user to perform several tasks relating to invoicing. These include creating invoices and registering invoice payments. It is important for these functions

to be present in the web application version. A list of the requirements for the invoicing section are presented below.

ID	Requirement Description
R-IN-1	The system must allow an invoice to be created for a single job.
R-IN-2	It must be possible to create an invoice that contains multiple jobs. An invoice can only contain multiple jobs if the customer and the order number are the same.
R-IN-3	It must be possible to find jobs in the system that need to be invoiced.
R-IN-4	An invoice must be assigned to a customer. The customer will be the same as the one assigned to the job/s on the invoice.
R-IN-5	An invoice requires a date stamp, values for total net cost, VAT and grand total.
R-IN-6	It should be possible to apply a discount to an invoice total. This can either be as a percentage or a fixed value.
R-IN-7	The system should allow comments to be wrote against an invoice.
R-IN-8	If a signature has been captured for a job, then it should be possible to include that signature on the invoice.
R-IN-9	If GPS data has been collected for a job, then it should be possible to include a map showing the data collected.
R-IN-10	If a invoice has been prepared then it should be possible to send the invoice via email or print locally.
R-IN-11	The system must provide a way of retrieving details of an invoice.
R-IN-12	It should be possible to mark a job as invoiced in cases where a manual invoice has been completed.
R-IN-13	It should be possible to register payments against an invoice.
R-IN-14	An invoice payment must have a date stamp, payment reference and the amount paid.
R-IN-15	It should be possible to edit an invoice payment.
R-IN-16	The system must allow an invoice payment to be deleted.

2.5 Settings

There is a settings section in the current application being used. This allows the user to manage options that are used in other parts of the application. In the web application version a sub set of these settings are required. The list below covers these requirements.

ID	Requirement Description
R-ST-1	The system must allow the company details to be changed. Here, 'the company' refers to the company using the application e.g. RNA Plant.
R-ST-2	It should be possible to change the company name, phone and fax number, email address, mobile number, website URL and address.
R-ST-3	It should also be possible to change the value used for VAT.
R-ST-4	The system must allow minimum and maximum job duration values to be set.
R-ST-5	It must be possible to set time boundaries for evening and weekend work.
R-ST-6	The system must be able to retrieve a list of 'Plant' items which require maintenance to be carried out on them within a set number of weeks.
R-ST-7	The system must be able to communicate with the current database and middleware application that is being used.

2.6 Messaging

It is possible to send messages to and receive messages from operatives in the current desktop application. This functionality should also be present in the web application version. The requirements relating to messaging are listed below.

ID	Requirement Description
R-MES-1	The system must allow messages to be sent to operatives.
R-MES-2	It should be possible to send a new message to one, many or all operatives.
R-MES-3	Each message must have a date-time stamp.
R-MES-4	The system should provide a way of retrieving all messages that have been sent to operatives.
R-MES-5	It should also be possible to retrieve all messages that have been sent from operatives to the office.
R-MES-6	The system must allow messages to be deleted.

2.7 User Interface

Below is a list of other requirements that the system must meet.

ID	Requirement Description
R-UI-1	A user must be able to interact with the system through a web browser.
R-UI-2	It is necessary for a user to login before they can use the system.
R-UI-3	A user must be able to exit the system.
R-UI-4	Certain functionality must only be available through the UI to users who have the required permissions level.
R-UI-5	The user interface must allow a user to perform any of the functions mentioned in the previous sections.

2.8 Non-functional Requirements

Below are a list of non-functional requirements for the web application.

ID	Requirement Description
R-NF-1	Any code that is written needs to be easy to understand.
R-NF-2	The system needs to be maintainable.
R-NF-3	Where possible, parts of the system should be reusable.
R-NF-4	The system should be stable.
R-NF-5	The usability factor of the system must be good.
R-NF-6	The system must be able to handle several parallel connections.
R-NF-7	It is important for the system to be reliable.
R-NF-8	Security needs to be considered and implemented where possible/necessary.
R-NF-9	The user interface must be simple and intuitive.
R-NF-10	The UI should be kept consistent in terms of design style.

3 Risk Management

The process of risk management should be present in the life-cycle of any software project. By identifying risks that could potentially occur in a project and addressing or eliminating them, the project is less likely to fail. It is the aim of this section of the document to do just that.

Potential risks that could occur during this project are listed in the table below. Each risk has been assigned a probability and an impact score. The probability score states how likely the risk is to occur and the impact score specifies how damaging the risk would be if it were to occur. Each score can be between 1 and 5 inclusively. A probability score of 1 means that the risk is not likely to occur, but a score of 5 means that the risk is very likely to occur. Similarly with the impact score, an impact score of 1 means that should the risk progress into an incident then the effect on the project will be minimal, but an impact score of 5 means that the risk will have extremely damaging consequences. To get an overall risk score, the impact and probability scores get multiplied together. As well as a risk score, each risk has an associated explanation stating how the impact of the risk will be reduced.

ID	Risk	Probability	Impact	Total
1	**Impractical time plan** Due to being relatively inexperienced in undertaking large software development projects, there is a risk that the time plan is unrealistic. If this is the case then there is potential for not all of the requirements to be met. **Contingency plan** To reduce this risk, a lot of time and careful thought went into creating the time plan. However, if certain parts of the project overrun their allocated time slots then it will be necessary to look over the remaining work and re-engineer the time plan. As the project progresses this risk should be reduced as my ability to estimate time requirements will be improved.	3	4	12
2	**Underestimating total workload** During the duration of this project, there will also be other work that will need to be completed. It is possible that the workload will be underestimated and therefore reduce the amount of time available for working on this project. **Contingency plan** This risk will be reduced by my experience gained during my Level 3 project.	3	3	9
3	**Short term illness** It is possible that during the duration of the project I pick up some form of short term illness. This may prevent me from working on the project as much as I would like to for a short period of time. If this is the case, time may be lost and I will have to find some time to catch up. **Contingency plan** To reduce the impact short term illness has on the project, I will not let the project drop behind schedule. Therefore, when I am ill the backlog of work will only be minimal and catching up should not be a problem.	4	2	8
4	**Long term illness**	1	5	5

	It is possible that during the duration of the project I pick up some form of long term illness. This may prevent me from working on the project for a long period of time. If this is the case, time will be lost and may prevent the project from being completed. **Contingency plan** The probability of long term illness affecting the project is considerably low. There is not much that can be done to reduce the impact of long term illness should it occur. The best that can be done is to hope that it doesn't occur.			
5	**Chosen technologies may be insufficient** It is possible that certain functionality cannot be implemented using the chosen technologies. This is due to being relatively unfamiliar with the technologies being used in this project and not completely finding out their deficiencies. **Contingency plan** A lot of time has been spent researching the technologies that will be used for the project to ensure that they can provide the necessary functionality that is required. If a part of the project cannot be implemented using the chosen technologies then it may be necessary to re-engineer the project using different technologies or to remove the feature if it isn't a major part of the system.	3	5	15
6	**Feature creep** At some stages during project development, I may see the possibility to add new features to the application that weren't originally specified. If this happens it will reduce the amount time available to work on the specified functionality. Potentially causing some specified features to not get implemented. **Contingency plan** To prevent feature creep, the specification will be followed strictly and only when the specification has been met will extra features be added.	2	4	8
7	**Complex features** It is possible that some of the functionality that is required may be complex to implement. If this has not been taking into consideration in the time plan then extra time will have to be spent on the complex feature to get it implemented or it will have to be left out. Spending more time on the feature may cause other features to go unimplemented. **Contingency plan** Depending on the importance of the feature, will depend whether or not extra time is spent implementing it. If it is not an important feature then time will be focused on other parts of the system first to ensure other features are not compromised.	2	4	8
8	**Loss of work** Throughout the development of the project it is possible that some or all of the work gets lost. This may be because of deleting files by accident or file corruption. **Contingency plan** To reduce this risk I will be taking regular backups of all my work. I will keep a copy on an external hard drive, a copy on Dropbox and a working copy on my computer.	3	5	15
9	**Computer failure**	2	5	10

	It is possible that at any stage during the project, the computer that I am developing the application on becomes faulty. This will cause development work to stop until a solution is found. **Contingency plan** As well as my main computer I also have an old laptop. As I will be taking regular backups of my work computer failure should not be a major problem as I will be able to transfer my work onto my laptop and carry on working.			
10	**Final product doesn't meet the specification** Once the application has been developed it may not meet the specification. **Contingency plan** This risk should not be a major problem as I will always be working against the specification. I will also try to provide the client with the ability to test the application as it is being developed so they can provide me with feedback.	2	5	10
11	**Code may be inadequately tested** It is possible that, inadequate test cases are created. If this is the case then the results of the tests may be misleading and provide false positives. **Contingency plan** I will create a detailed test plan that involves unit tests and functional tests. This will ensure that the application is thoroughly tested and it meets the clients requirements.	2	4	8
12	**Social events** During the duration of the project it is very likely that I will be dedicating some of my time to social events. If I give too much of my time for these occasions the project may fall behind schedule. **Contingency plan** To prevent this, I will be very strict about the time I spend on social events.	2	3	6
13	**Failure to correct known problems** During testing, problems with the system may be brought to light. If problems that are found are not corrected, this may cause problems with the system and may prevent it from working as desired. **Contingency plan** No extra functionality will be added to the application until the problems are corrected. This means fixing them as soon as they are found.	1	4	4
14	**Bad naming conventions used** It is very likely that during development bad names will be chosen for variables, classes and methods. This is likely to be down to laziness. If this happens then the code will become harder to comprehend. **Contingency plan** During development I will constantly look over code to ensure no bad naming conventions are used. I will not deem a class to be complete until I'm satisfied that good naming conventions have been used throughout the class.	3	3	9
15	**Failure to document methods and classes**	3	2	6

It is possible that I forget to document some classes and methods as I am developing the application. If this happens then, when looking back at written code I may not be able to understand the code straight away.

Contingency plan

To make sure methods and classes are documented, I will not deem a class to be complete until I'm satisfied that the class and its contents are well documented.

---

16 **Being unfamiliar with the programming languages** 3 2 6

I have had limited experience with the programming languages being used to develop this project. Therefore, coding may take longer than expected at the start of the project as I am getting used to the languages.

Contingency plan

The effect of this risk should be minimal as I have spent time learning the programming languages. Also, the languages I will be using are similar to languages that I have used in the past.

4 Methodologies

4.1 Software Engineering

When developing any major piece of software, it is a good idea for the software developers to adopt a software engineering methodology. Currently, there are many different software engineering methodologies which can be followed. Each with their own advantages and disadvantages. It is the goal of this section to provide descriptions of three possible methodologies that could be used for this project. As well as this, the methodology that is going to be used for this projected will be stated giving reasons as to why the methodology has been chosen.

4.1.1 Waterfall Model

The waterfall model is the classic life cycle model. It has many different variations, each containing a different number of phases. The version that will be discussed here can be seen as being made up of 6 distinct phases. The phases that make up the waterfall model are as follows:

1. Requirements
2. Design
3. Implementation
4. Testing
5. Installation
6. Maintenance

During software development when following this model, the phases listed above are followed sequentially in ascending order. This is done in such a way that, the developers may only progress to the next phase when the preceding phase has been completed. This means that phases should not be visited again once they have been completed. A diagrammatic view of the waterfall model is shown in Figure 1. It shows the phases of the waterfall model and the direction of flow between each phase.

The main advantage of using the waterfall model arises from the strict nature in which a phase must be complete before progressing to the next phase. This ensures that bugs will be detected and corrected during the early phases like the system design phase. By doing so, bugs should not go undetected until the testing phase which then become more costly to correct, in terms of time and money. Another advantage of the model is that it is easy to understand.

A major disadvantage of the waterfall model also arises due to its strict phase completion nature. As a stage should not be revisited once it has been deemed complete, it is essential that the client knows exactly what they want during the initial phases of the project. Following the model strictly, any functionality alterations requested by the client during later phases (e.g. implementation) of the project may not be made as it is not possible to revisit earlier phases (e.g. requirements). To include these alterations it would be necessary to restart the entire process, wasting a considerable amount of time and money.

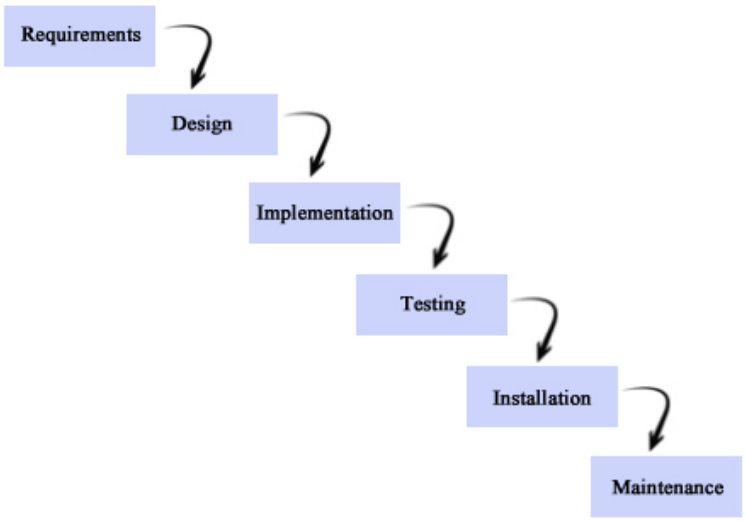


Figure 1: Waterfall Model.

4.1.2 Spiral Model

Whereas the waterfall model can be seen as a document-driven approach to software development, the spiral model is a risk-driven approach. This is because a major part of the spiral model focuses on risk assessment and management. The diagram in Figure 2 is a diagrammatic representation of the spiral model. It shows the spiral model uses the idea of iterative development through the generation of multiple prototypes.

It is clear from Figure 2 that one spiral (phase) on the diagram is made up of four distinct stages. The four stages that make up a single phase are:

1. **Determine objectives** - The objectives of the project are determined. Along with objectives, alternatives and constraints are also established.
2. **Identify and resolve risks** - During this stage, each of the alternatives are evaluated. It is part of this stage that associated risks must be identified and resolutions for each risk must be found. A rough software prototype is also built in this stage.
3. **Develop and test** - The third stage of the model requires the prototype to be tested and risk resolution strategies carried out.
4. **Plan the next iteration** - The next iteration is planned based on the outcome from step 3.

However, it is possible to iterate through these stages as many times as is necessary to reach a satisfactory detailed design. Each iteration adds a new spiral to the diagram and generates another prototype which is a more complete version of the software in comparison to its predecessor. The prototype then gets evaluated by the client. During this evaluation, the client can request any modifications that they would like to see be made in the next iteration. As mentioned previously, a major part of this model is concerned with risk analysis which in turn hopes to minimize the risk of failure among other things. Risk analysis is performed during each iteration of the four stages which determines whether certain choices are made or not. As soon as a satisfactory design has been obtained

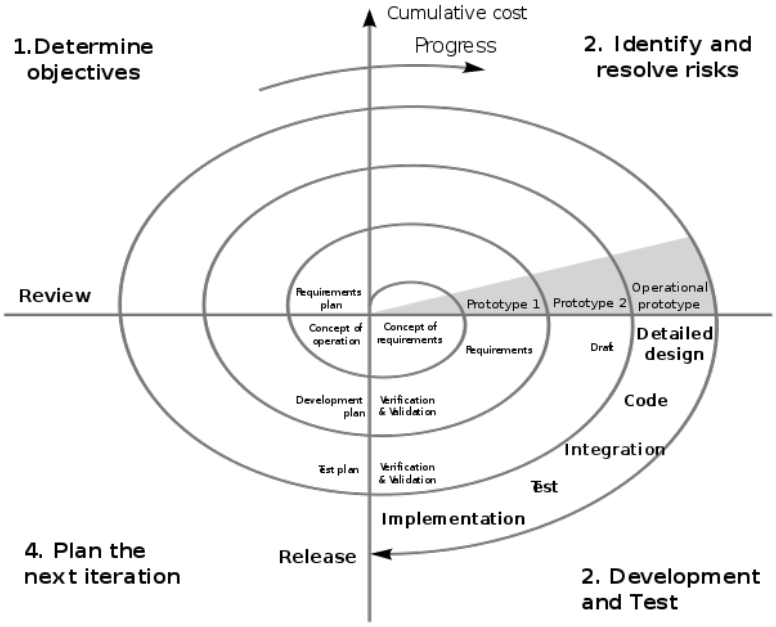


Figure 2: Spiral Model.

from prototype development, the software is developed by essentially following the waterfall model. However, the first two stages of the waterfall model are omitted.

The spiral model has several advantages. These are stated below.

- A major benefit of using the spiral model in comparison to the waterfall model is that it introduces risk management.
- The client is able to refine their requirements as the project progresses ensuring that the final product is exactly what they require. This is because of the models incremental nature.
- Development work is able to start early on as they do not have to wait for the requirements and design processes to be 100% complete.
- Although cost is not a factor for this project, the spiral model makes cost estimation easier.

Some of the disadvantages of the spiral model are given below.

- It is important that risk analysis is carried out properly. This means that risk assessment expertise is required. Otherwise, problems could arise from risks going unnoticed.
- The amount of risk analysis that needs to be performed may be seen as excessive, taking up valuable time.
- In general, this model works best for large projects.
- The success of a project relies on the risk analysis phase being carried out properly.

4.1.3 Scrum

The final methodology that will be considered for this project is called *Scrum*. The Scrum methodology is an agile development process whereby software gets developed a bit at a time through repeated cycles. Each cycle is called a *sprint* and it is assigned a fixed period of time. During a sprint, the development team are required to add functionality to the software and produce working and tested software. The functionality that is to be added during a given sprint gets decided during a sprint planning meeting. During this meeting the product backlog is examined and the items for the next sprint are chosen. The product backlog is a priority ranked list of features that the client requires to be present in the final product. This list can grow as times goes by. Once the items are chosen from the product backlog they are put onto a sprint backlog. The sprint backlog contains a list of what needs to be done during the next sprint and a date specifying when it must be completed by. Once the date is reached a sprint review meeting must be held to evaluate what was achieved during the sprint. It is not uncommon for items on the current sprint backlog to not have been implemented during the sprint. If this is the case then they are put back onto the product backlog ready for the next sprint. The Scrum process is depicted in Figure 3.

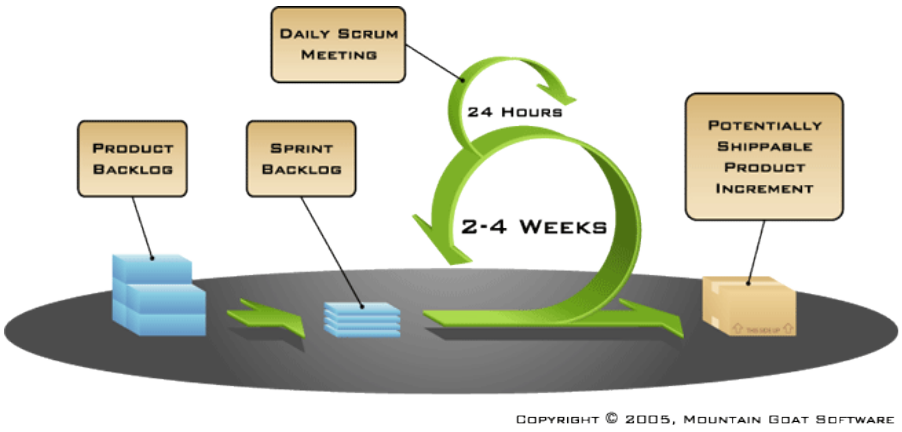


Figure 3: Scrum process.

This methodology is very different to the other methodologies already discussed as it does not contain explicit phases (e.g. requirements analysis, design, implementation...) that must be executed in a specific order. Instead, each sprint will contain a combination of these phases. Typically, each sprint will include phases for analysis, design, implementation and testing. Unlike traditional methodologies, it is possible to start with any phase and switch between them during a sprint.

The scrum methodology relies on three predefined roles being followed. These roles are the *Product Owner*, the *ScrumMaster* and the *Development Team*. The roles are briefly explained below.

- **Product Owner** - It is the job of the product owner to manage the product backlog.
- **ScrumMaster** - The ScrumMaster must facilitate the Scrum process and resolve any impediments. They are also given the task of enforcing the sprint time limits.
- **Development Team** - The main responsibility of the development team is to deliver the end product by completing the sprints on time. The development team should include members such that there exists a wide variety of skills amongst them and they are preferably self-organising.

Meetings are also an important part of the Scrum process. Some of these meetings have already been mentioned which are the sprint planning meeting and the sprint review meeting. In addition to these two meetings there is a daily scrum meeting and a sprint retrospective meeting. The daily scrum meeting is a short meeting that is timeboxed to around 15 minutes and occurs the same time every day. During this meeting, each member of the development team states what they did yesterday, what they are going to do today and impediments they face. The sprint retrospective meeting is held at the end of a sprint. During this meeting the team will discuss what went well and what could be improved based on the last sprint.

Some of the advantages of the Scrum methodology are:

- It makes coping with change easy. So, the client is able to change their requirements without the process being affected.
- At the end of each sprint there should be a working piece of software.
- It is possible to provide a good schedule for delivery if tasks have been well defined.
- It works best for projects that are fast moving with small teams.

The disadvantages of Scrum are listed below.

- It is hard to estimate how long a task will take if it has not been well defined.
- It does not work well for projects with big teams.
- Project development can be affected greatly if a team member were to leave during development.

4.1.4 Conclusion

After carefully analysing each of the methodologies discussed above, it has been decided that the Scrum methodology will be used for this project. The flexible nature of the Scrum methodology makes it perfect for this project. Also, as working software will be produced frequently, it will allow the client to provide feedback stating whether what is being produced is correct or if changes need to be made.

As the Scrum methodology has been designed with the assumption that a team will be present, it is necessary for slight modifications to be made in how the process is followed. The three roles will essentially be reduced to two. The role of product owner will be assigned to Nick Beckett of RNA Plant but he will not be able to add any major requirements to the product backlog once work has commenced, only slight modifications. The roles of ScrumMaster and development team will be undertaken by myself. The other part of the process that needs modification is in relation with the meetings. As I will be developing the software on my own, daily scrum meetings become impossible as there are no team members to inform about the progress that has been. Instead of this I will make personal notes of what I have done and what I am going to do. The sprint planning meeting will be modified in such a way that it involves me choosing what will be added during the next sprint. On some occasions I may contact Nick Beckett to ask if he has a preference of what gets added next. At the end of each sprint, there will be no need for a sprint review meeting but I will need to put back into the product backlog any items that were not included during the sprint. The sprint retrospective

meeting is held to discuss what went well during the last sprint and what could be improved. Instead of a discussion, I will think about what went well and what could be improved.

The reason the waterfall model will not be used for this project is because of its inflexible nature. Also, the spiral model was not chosen because of the need for intensive risk analysis to be carried out throughout the development. This seems unnecessary for this project and would use up valuable development time.

4.2 Testing

Testing is an important part of any software development life cycle. Without it there is a good chance that the quality of software being developed will be substandard and incorrect. Therefore, throughout the development of this system, various testing techniques will be employed. By doing this, it will ensure several things. Firstly, that the quality of the software will be of a high standard. Secondly, the number of bugs in the system will be low. Finally, it will ensure that the software does what the customer requires and meets the specification.

During the development of this system, two main testing methods will be used. These are unit testing and functional testing. The unit testing will be achieved using white-box testing and the functional testing will be accomplished using black-box testing.

4.2.1 Unit Testing

As mentioned above, the unit tests will be performed using white-box testing. When performing white-box testing, the system under test is observable by the tester. This means that as well as the input and output, the tester is able to see the internal workings of the system. This is depicted in Figure 4. By observing the structure of the code, the test engineer is able to create test cases to run against the system. It is common in white-box testing to formulate test cases that cover branches, statements and paths in the code.

Figure 4: White-box testing.

Unit testing is performed, as the name suggests, on the individual units that make up a system. For this project a unit will refer to a method in a class. So, unit testing will require tests to be created for each method of every class. By doing this, it is possible to make sure that every method produces the required output and is therefore correct. One benefit that comes with unit testing is, if such a test fails then it is relatively easy to pinpoint the error in the code.

As the system is being developed, unit testing will be applied when a class is completed. Each method of the completed class will be tested and the aim will be to obtain 100% code coverage. By

testing in this way, we can be sure that the contents of a class are correct. Therefore, we can be confident when making calls to the tested class from other classes.

4.2.2 Functional Testing

The unit tests that are performed will ensure that the code that has been developed is correct. However, the correctness is not in relation to the specification. So, by just performing unit tests the code may be correct but the overall functionality provided by the software may not be correct or insufficient. To overcome this, it will be necessary to perform functional testing.

To perform functional testing on a system, it is necessary to use black-box testing techniques. Black-box testing restricts the tester from seeing implementation details of the system under test. The functionality provided by the system is solely of interest during functional testing. It is because of this that, this method of testing can be seen as testing from a users perspective. A user of the system is only interested in the functionality provided (input/output behaviour) and not how the functionality was implemented (application code). This is shown in Figure 5.

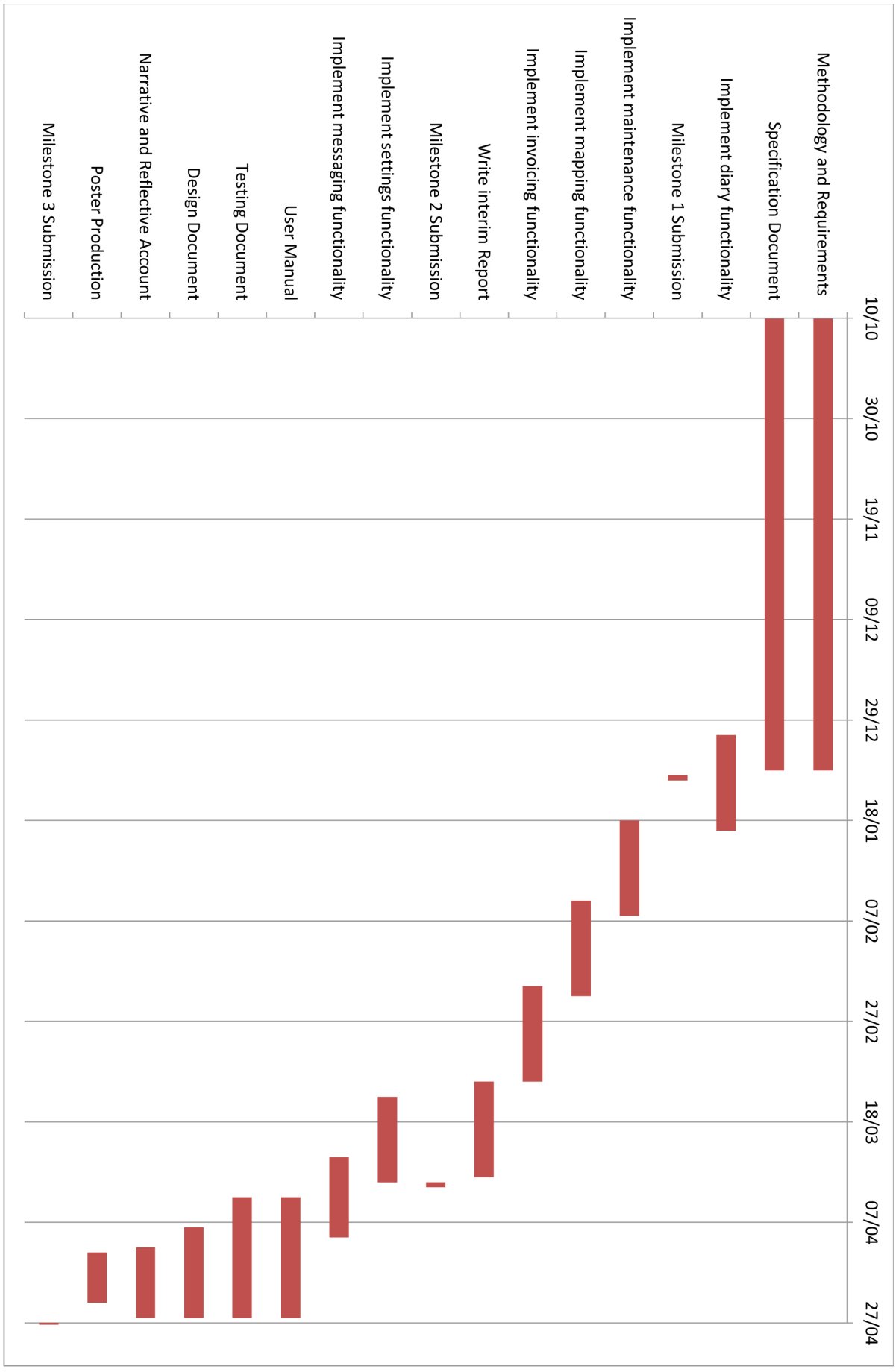
Figure 5: Black-box testing.

As the test engineer is not able to see any code, then they must create their test cases based on the system functional requirements. Whereas the test cases produced when unit testing will test the individual parts of the system, test cases produced when functional testing will test if they work together correctly and provide the required functionality. By doing this we see one of the major advantages that comes with functional testing compared to unit testing. This is why it is essential that functional testing is applied during this project. It will make sure the system does what it is meant to do.

As soon as a major section of the system has been developed, functional testing will be applied. This will check that the required functionality is correct in relation to the functional system requirements. Functional testing will also be applied when the entire system has been developed. This is to check the system works correctly as a whole and that no bugs have crept in during integration of all the sections of the system. One drawback with functional testing is that when a test case fails it is significantly harder to pin down what caused the error. Hopefully, this problem will be eliminated by performing the unit tests during throughout development.

5 Project Plan

To ensure that the project stays on track and will be completed on time, a project plan has been devised. The plan is depicted on the following page in the form of a Gantt chart.


Methodology and Requirements
Specification Document
Implement diary functionality
Milestone 1 Submission
Implement maintenance functionality
Implement mapping functionality
Implement invoicing functionality
Write interim Report
Milestone 2 Submission
Implement settings functionality
Implement messaging functionality
User Manual
Testing Document
Design Document
Narrative and Reflective Account
Poster Production
Milestone 3 Submission
10/10
30/10
19/11
09/12
29/12
18/01
07/02
27/02
18/03
07/04
27/04