CS-M14 Industrial Project
Methodology & Requirements Document

Matthew James Lewis
523015@swansea.ac.uk

December 28, 2011

1 Introduction

1.1 Project Overview

The aim of this project is to develop an interactive web application to be used by the National Mass Spectrometry Centre, NMSSC, at Swansea University and it's customers. The centre houses a number of mass spectrometry machines and provides a service which allows customers to send samples to the centre for analysis. The centre then sends the analysis results back the the customer. The main purpose of the application is to provide an online service to customers of the centre that allows them to submit and track the progress of any samples which have been sent to centre for analysis. This system can be likened to that of an order tracking service, offered by some e-commerce websites, in that it allows the customer to see when their order has been received, processed, picked and dispatched for delivery.

The application will also provide management functionality to the staff at the centre. This part of the system will provide the details of all the samples submitted by the customers and then provide the utilities for them to move samples through each processing stage and provide the results back the the customer once their sample has been analysed.

Currently an existing system already exists at the centre but contains limited functionality and lacks a modern look and feel. It was also developed using technologies which are no longer used and proves to be difficult to maintain. Together with the current online system, the centre also uses other methods of managing samples. Therefore, the ultimate underlying aim of this project, is to produce a new modern unified system that can be easily maintained and contains all the functionality that centre would like to provide to its customers and the functionality that is needed to easily manage all the samples.

This document will outline the methodologies which I have chosen to use during the development of the application together with an analysis of the potential risks involved while undertaking this project. Finally, this document contains the list of requirements, in a structured format, set by the centre. Using these requirements I will produce a formal specification of the features of the application which will be used during development.

1.2 Current NMSSC System

Each year, various institutions from all over the country to the centre for use of the provided service. If successful in their application, the institutions are given unique Swansea Codes which is, in essence, their username for the system which they then use to log on to the web application. This code is also used to identify all samples submitted by that institution.

In some cases an institution can have many different people, such as post-graduates, researchers etc, submitting different samples all using the same 'Swansea code'. For this reason, the system allows a number of submitters to be created on the system. These can then be used by each person in the institution to easily submit and track their own samples.

To submit samples, the submitter must first log on to the system using their institutions Swansea Code and password. They then create a new sample submission by entering various details such as molecular weight, chemical formula, COSSH level and any other additional comments and also choosing which name they would like to submit under. This is then attached to the sample and displayed with all the other details. To help identify samples within an institution each sample must be given a unique sample reference by the submitter. It is also possible for the submitter to upload a chemical structure file in a specific format.

After submitting the online sample submission they then post the physical sample vial to the centre for analysis. When the centre receives the sample in the post it is marked as received on the system. The sample is then booked in by linking the physical sample to the online submission. This is achieved by applying a sticker containing an unique code to the physical sample vial and then adding the unique code to the electronic submission.

The sample then moves through the various stages of analysis in the lab until the results are produced. These results are then emailed back to the customer using the the email provided during the application process.

1.3 Potential Improvements

After being guided through both the interface and the source code of the existing system, a number of potential improvements were discussed with the centre.

Modern look & feel

The application currently has a very simple and plain interface using simple HTML links and table with little or no styling. This is one area where the centre would like to improve upon by introducing a more modern interface while keeping the same or increased level of functionality.

Current development technologies

As previously mentioned the application was developed using ASP, a language which has been succeeded by new and more advanced web technologies. The centre has identified that the system needs to be re-implemented using these new languages.

Cleaner architecture

After viewing the current architecture of the application it is clear that a re-write is required to aid any future maintenance and extensions of the applications. To achieve this a new simple architecture must be implemented with full commenting and documentation.

Drawing tool for chemical structures

Another potential feature identified by the centre is to provide an online drawing tool to the customers which would allow them to draw the chemical structures of their samples while submitting their online sample submissions. It is already possible for customers upload a file produced by third party software and a drawing tool could be introduced as an alternative option.

Providing analysis results to customers

A major improvement identified by the centre is to make the results produced by the analysis of samples available online for the customers to download directly or by viewing them in an in-built viewer. However, for the viewer to work, it would require a standard file type for the results to be agreed upon.

2 Methodology

2.1 Development Model

While developing the project I have chosen to use the iterative development model as my software development methodology. The primary reason for choosing this model was because it allows me to work closely with the staff at the centre by gradually producing partial prototypes versions of the applications which can be reviewed by myself and the staff of the centre. After the prototype has been reviewed and the centre are happy with the functionality I can then progress onto developing the next set of features/functionality another prototype version. This process will repeat until all functionality has been introduced and reviewed successfully. This approach of producing prototypes and receiving feedback from the centre will ultimately result in an application which is as close to the requirements as possible while still allowing myself to produce the application without constant supervision by the centre.

To begin with, initial planning on the whole project is undertaken. The model then moves into prototype cycles which consist of four stages:

- Planning
- Design
- Implementation
- Evaluation

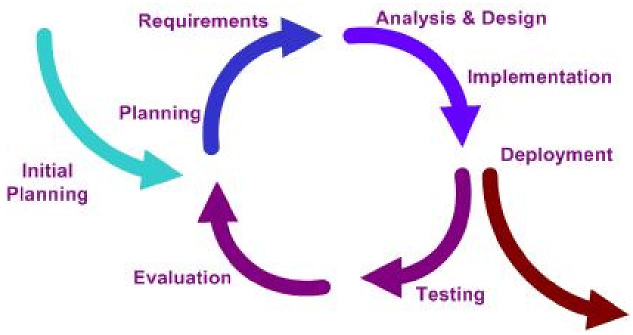

Figure 1: Iterative development cycle.

After each cycle is complete, a new cycle begins on the next feature of the system. This process is shown in Figure 1.

At each prototype cycle, as the features are planned, designed, implemented and evaluated with the centre's staff any modifications that are required can be undertaken with minimum effect on the rest of the application. On the other hand, if I had chosen to use the **waterfall model** for example, the whole system would developed before evaluation is undertaken. This would require more detailed and thorough initial planning by myself with the centre in order to develop an application that meets their requirements and expectations and any modifications could potentially cause major problems when altering some functionality which is relied upon by another part.

Like the Waterfall model, I also decided against the **V Model** as it is essentially an extension of the Waterfall method in that it requires for a whole system to be developed. After the functionality has been implemented it is tested and evaluated. If any modifications need to be made the model moves back to the planning and design stages before implementing the change. Using this model for the project would still require thorough planning and design with the centre before work on the application is started and could still potentially cause complications while changing or modifying the application.

2.2 Development Tools & Techniques

While developing the system I have also chosen to follow an existing software architecture with which I have had extensive experience with. By choosing to follow this architecture it allows me to produce simple, clean and high-quality code which, in the long run, will prove beneficial for any future maintenance or extensions to the application.

To aid with the development of the application I have also chosen to use the Netbeans Integrated Development Environment (IDE) as it will allow me to easily debug and perform tests on the source code that I produce resulting in higher quality and consistent code.

3 Risk Analysis

3.1 Non-technical Risks

The non-technical risks I have considered along with the steps taken to prevent them or action to be taken should they occur are outlined as follows:

Failure to adequately judge the time and resources required.

This issue will be minimised using careful consideration of the project deadlines and ensuring that I adhere strictly to any I set myself. The project is also based on re-implementing and improving upon an already existing system and, therefore, I have all the required knowledge about how the system should work before starting. This should prevent any problems when developing the system to the centre's requirements.

Underestimation of non-project-related workload

This project will not be the sole focus over the coming months as I am currently completing my Masters degree in Computer Science. However, after completing a large project for my dissertation, I feel I have gathered enough experience to be able to balance other work with the requirements of this project.

Illness

Unfortunately it cannot be ruled out that I may suffer from illness during the project which may cause the work rate to decrease. For this reason I will set the deadline to finish the project by early enough so that any possible time lost due to illness will not cause the project to overrun the academic deadline set by the University.

Failure to meet requirements

This project has a clear set of requirements set by the centre and it is imperative that all requirements set out are met. This risk is reduced by carefully choosing which requirements are absolutely necessary and which are possible future enhancements. This prevents the centre from setting out an extensive list of requirements, some of which might be extremely complicated given the timeframe to complete the project.

Failure to match customers vision of the project

As this project is being developed for a company not in the software business it is unlikely that they know exactly what they would like the application to look like and how it should behave. This could potentially result in the customer not being happy with the finished product. To combat this issue I have chosen to produce a number of partial prototypes of the application which will be evaluated by the centre. This will provide me with feedback on what has been produced so far and any modifications that are required can be made during development and not after the project has been fully coded.

3.2 Technical Risks

I have also identified a number of technical risks for the project and these are also outlined as follows:

Feature creap.

As with any major project it is possible for the developers to continually add features that are not in the original requirements of the system. To prevent this issue it is important that I produce an accurate and complete requirements list so and ensure that all required features are implemented before adding any additional functionality if time permits.

Compatibility with all major web browsers

A major issue when developing web applications is the wide range of web browsers used by people to access the world wide web. Each browser's engine works differently and can mean that one layout that works in Firefox, for example, may not work in Microsoft's Internet Explorer. This risk can be avoided by thoroughly testing all page layouts on a multitude of browsers to ensure that the page is displayed the same in each one and all functionality works the same. It is also possible to use an online service that produces screenshots of a specified URL in a greater range of web-browsers allowing the layout to be checked even more thoroughly.

Failure to adequately learn a required new technology.

With web applications there are many different popular modern technologies used and it is important that I have experience and knowledge with the languages that I have chosen. To avoid this risk, I have chosen two technologies with which I have extensive experience with both academic and personal projects. I have also chosen technology that have are the most popular and have an extensive support base if any problems arise that I may not have experience with.

4 Requirements

4.1 Funtional Requirements

Customer Web Application

ID	Requirement Description
REQ1	The application must allow customers to log in to the system using their account details.
REQ2	The application must provide the ability to add or remove submitters from the customers account.
REQ3	The application must allow customers to submit new samples for analysis.
REQ4	The application must allow customers to submit samples under different submitter identities.
REQ5	When submitting a new sample, the application must ensure customers enter various required information about the sample.
REQ6	The application must allow customers to track the progress of any samples they have submitted.
REQ7	The application must allow samples to be created and saved so that they can be edited and submitted at a later date.
REQ8	The application must provide functionality for drawing or uploading a chemical structure file for a submitted sample.
REQ9	The application must provide a list of all the customers samples in an organised manner.
REQ10	The application must allow the sample list to be sorted or filtered by various parameters.
REQ11	The application must allow for the customer to retrieve the results of any completed samples they have submitted either by downloading them or displaying them in a on-line viewer.

Sample Administration Web Application

ID	Requirement Description
REQ12	The application must allow for the lab staff to login to the system.
REQ13	The application must allow the user to view all the samples submitted by a customer.
REQ14	The application must allow the sample list to be filtered by a time period.
REQ15	The application must allow the user to unsubmit samples.
REQ16	The application must allow the user to view the details of submitted samples together with the chemical structure that the customer provided when submitted.
REQ17	The application must provide functionality that allows the user to attach result files to a sample after it has been analysed by the lab
REQ18	The application must allow the user to move samples through the various stages of processing.
REQ19	The application must provide the ability for a new year codes to be started.
REQ20	The application must allow the user to view samples for previous year codes.
REQ21	The application must allow the user to delete samples.
REQ22	The application must follow the labs current sample identification system when booking in and processing samples.

4.2 Non-Functional Requirements

ID	Requirement Description
NFREQ1	The application must be accessible on the world wide web.
NFREQ2	The application must be compatible with all major web browsers.
NFREQ3	All code should be fully commented and documented.
NFREQ4	The application must provide a simple user interface with a modern look and feel.
NFREQ5	The application must be easily maintainable.
NFREQ6	The application must use modern web technologies

5 Project Plan

The project plan outlined below is based on the Iterative development model, discussed in Section 2.1, where partial prototypes of a system are designed, implemented and tested before moving onto the next feature(s) of the system.

5.1 Milestones

Milestone	Deliverable	Date
Milestone 1	Methodology & Requirements Document Specification Document	9th January 2012
Milestone 2	Interim Report	Start of Term 3
Milestone 3	Poster Presentation User Manual Design Document Testing Document Narrative and Reflective Account	Semester 2 Week 10

5.2 Detailed Breakdown

I have divided the project into 6 prototypes, each one adding additional functionality until the project is completed and all requirements are met. The details of each prototype are outlined below. The deadlines fore each prototype have also been shown.

Prototype 1 The first prototype will concentrate on designing and producing a new modern layout that the application will use.

Prototype 2 This prototype will be used to introduce the authentication and session functionality. It will also be used to add the 'submitters' functionality.

Prototype 3 The next prototype contains two major sections of the application. First it will add the ability for customers to submit new samples and then it will introduce the sample lists allowing the customers to view all of their samples.

Prototype 4 The fourth prototype will add all the management functionality required by the staff at the centre to move the samples through each stage of processing.

Prototype 5 The fifth prototype will be used to introduce the ability to attach result files to a sample allowing the customer to download or view them.

Prototype 6 The final prototype will add the drawing tool to the sample submission page.

Prototype	Task	Date
Prototype 1	New Layout	6th January 2012
Prototype 2	Authentication & Sessions Submitters functionality	3rd February 2012
Prototype 3	Submit Samples Sample Lists	2nd March 2012
Prototype 4	Management Functionality	23rd March 2012
Prototype 5	Result files (download & viewer)	3rd April 2012
Prototype 6	Drawing tool	20th April 2012

5.3 Gantt Chart

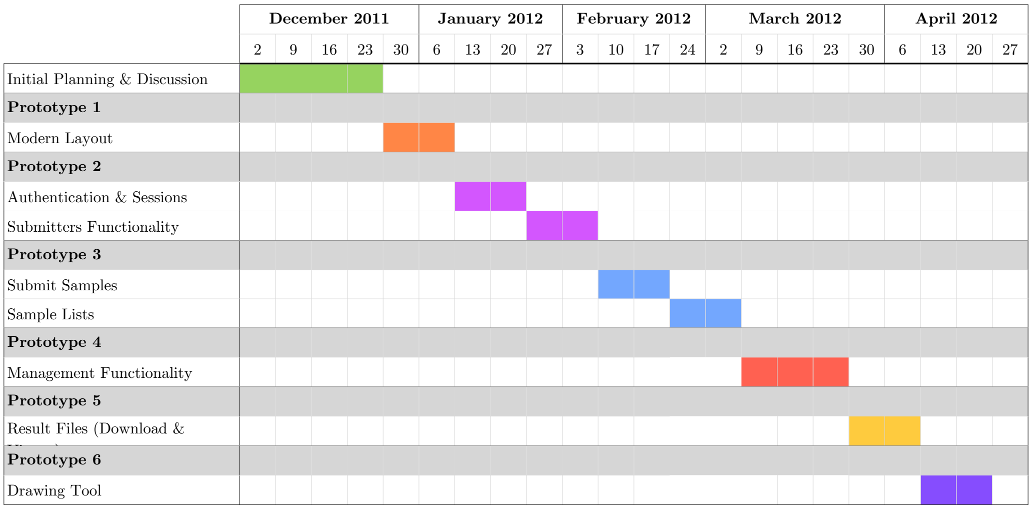

Figure 2: Project Gantt Chart