Newspaper Delivery System
Project Management Report

Aaron Molloy, James Kenny, Taras Voloshyn

B.Sc. Software Engineering Year 4
November, 2013
Newspaper Delivery System
Project Management Report

compiled by

Aaron Molloy (A00172521)
Jimmy Kenny (A00177486)
Taras Voloshyn (A00174389)

B.Sc. Software Engineering Year 4
November, 2013
Contents

Glossary of Terms ........................................................................................................................................... 3
Introduction ....................................................................................................................................................... 4
Project Organization ........................................................................................................................................ 6
Risk Analysis .................................................................................................................................................... 7
Hardware and Software Resource Requirements ........................................................................................... 9
Work Breakdown Structure ........................................................................................................................... 10
Project Scheduling ......................................................................................................................................... 15
Monitoring and Reporting ............................................................................................................................. 22
Glossary of Terms

**AJAX** – A technique for creating fast and dynamic web pages

**Cloud Service** – Computer services provided to customers over the Internet

**Critical Path** - The sequence of dependent steps that determine the minimum time needed to carry out an operation

**Eclipse** – An Integrated Development Environment used for Software Development

**GIT** – Repository used for version control

**Javascript** - A programming language

**MySQL** – A database management system

**PHP** – A programming language

**V-model** — one of the models which defines processes and their order within software development lifecycle. This model is based around continuous testing all the way through the project.

**WBS** — Work Breakdown Structure

**Xampp** - Open source cross-platform web server solution stack package
Introduction

Purpose of the Document

The purpose of this report is to outline the various project management considerations that we will undertake in designing the Newspaper Delivery Service. The report will detail:

- **System Overview** – A short description of the project requirements
- **Project Organisation** - List of the roles required to undertake the project
- **Risk Analysis** – An analysis of the risks involved in the project
- **Resource Requirements** – Details of the Hardware and Software resources required to undertake the project.
- **Work Breakdown Structure** – A diagrammatic breakdown of the work involved in bringing the project to completion.
- **Project Schedule** – A schedule of every aspect of the project detailing the timeframes involved in Requirement Elicitation, Design, Implementation and Testing.
- **Monitoring and Reporting mechanisms** – Details of how the project will be monitored and reported.

System Overview

This Newspaper Delivery System is intended to manage the delivery of newspapers and magazines in a small town or area of a larger town. It is intended for use by newsagents who are only casual users of computer systems and should run on a PC or similar hardware.

Factors which should be taken into account in specifying and designing this system are:

- For each delivery person, the system must print, each day, the publications to be delivered to each address.
- The system should also print, for the newsagent, a summary of who received what publications each day.
- Once a month, bills are delivered to customers along with their newspapers. These bills should be computed automatically by the system.
- Customers come and go and may be away temporarily on holiday or on business.
- Not all customers necessarily have a delivery every day.
- The system should be able to manage some simple geographic information so that it prints information for the delivery person in the order in which publications are delivered.
It was decided to design the system as a web application hosted by a cloud service for the following reasons:

- Our skillsets lay mostly in the web development area.
- A requirement for the system that it would be used by newsagents who are only casual users of computers. We therefore felt that a web application would be easier for the newsagent to use and more intuitive to learn.
- A web application gives greater accessibility.
- A system hosted by a cloud service would be easier and more efficient for backup and security purposes.
Project Organization

The project will be undertaken by a three man team comprising of Aaron Molloy, Jimmy Kenny and Taras Voloshyn. Mike Russell will act as the Project Manager and the team will periodically report progress or any serious issues to him.

Each team member shall share responsibility for Requirements, Design, Implementation, Testing, Integration and Deployment of the system but will be specifically responsible for the following features of the system:

- Customer Details and Printing Functions – Aaron Molloy
- Geographical Information and Summary Functions – Jimmy Kenny
- Database and Billing Functions – Taras Voloshyn
Risk Analysis

The Risk Exposure values in the table above were calculated from two other values: Probability of Occurrence and the Consequences of Occurrence measures. The metrics for the Probability of Occurrence calculations were chosen because they increase evenly from low to high which makes it easier to assign probability factors to our risks. As you can see, we have used 20% increments for each measure. The reason we didn’t use a more detailed measurement (e.g. 1%) was due to the complexity it would introduce when deciding what the probability of each risk with consideration to the type of system we are developing.

<table>
<thead>
<tr>
<th>Risk No.</th>
<th>Risk Description</th>
<th>Probability (P)</th>
<th>Consequence (C)</th>
<th>Risk Exposure (RE)</th>
<th>Plan (Aviation &amp; Mitigate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Development time underestimated</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>Allow for contingency</td>
</tr>
<tr>
<td>2</td>
<td>Requirements change</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>Analyze the impact of the new requirements on entire project and discuss with customer of necessity.</td>
</tr>
<tr>
<td>3</td>
<td>Hardware failure</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>Have access to spare parts.</td>
</tr>
<tr>
<td>4</td>
<td>Database failure</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td>Do regular backups.</td>
</tr>
<tr>
<td>5</td>
<td>Staff illness</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Ensure that staff understand each other's tasks. Reassign tasks between remaining members to a team.</td>
</tr>
<tr>
<td>6</td>
<td>Budget cuts</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>Understand hard caps outs in costing would have a detrimental effect on development process.</td>
</tr>
<tr>
<td>7</td>
<td>Underestimation of resources</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>Understand capabilities for given requirements.</td>
</tr>
<tr>
<td>8</td>
<td>Misunderstanding of the requirements</td>
<td>3</td>
<td>4</td>
<td>12</td>
<td>Plan interview with all the requirements are understood correctly.</td>
</tr>
<tr>
<td>9</td>
<td>Poor project planning</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>Review project plans at regular intervals.</td>
</tr>
<tr>
<td>10</td>
<td>Lack of skills/knowledge</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>Ensure adequate training of team members.</td>
</tr>
<tr>
<td>11</td>
<td>Incomplete hardware requirements</td>
<td>3</td>
<td>4</td>
<td>12</td>
<td>Test hardware requirements e.g. Test server under high load.</td>
</tr>
<tr>
<td>12</td>
<td>Change in client management during the project</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>Make sure that contract fees are clear.</td>
</tr>
<tr>
<td>13</td>
<td>Deadlock</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>Ensure adequate security measures are taken e.g. strong firewalls, virus protection etc.</td>
</tr>
<tr>
<td>14</td>
<td>System doesn’t perform to expectations</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>Understand requirements fully. Ensure hardware and tools are of the same specification.</td>
</tr>
<tr>
<td>15</td>
<td>Budget overrun</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>On-going review of project spend. Interact with customer to ensure requirements are sound</td>
</tr>
<tr>
<td>16</td>
<td>Requirements inadequate</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>Read and understand requirements fully. Make sure designer has appropriate skill level.</td>
</tr>
<tr>
<td>17</td>
<td>Design of system inadequate</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>Read and understand requirements fully. Make sure designer has appropriate skill level.</td>
</tr>
</tbody>
</table>

Probability of Occurrence

The Risk Exposure values in the table above were calculated from two other values: Probability of Occurrence and the Consequences of Occurrence measures. The metrics for the Probability of Occurrence calculations were chosen because they increase evenly from low to high which makes it easier to assign probability factors to our risks. As you can see, we have used 20% increments for each measure. The reason we didn’t use a more detailed measurement (e.g. 1%) was due to the complexity it would introduce when deciding what the probability of each risk with consideration to the type of system we are developing.

<table>
<thead>
<tr>
<th>Category</th>
<th>Probability Measure</th>
<th>Probability Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>1</td>
<td>0% to 20% chance of occurrence</td>
</tr>
<tr>
<td>Low</td>
<td>2</td>
<td>21% to 40% chance of occurrence</td>
</tr>
<tr>
<td>Average</td>
<td>3</td>
<td>41% to 60% chance of occurrence</td>
</tr>
<tr>
<td>High</td>
<td>4</td>
<td>61% to 80% chance of occurrence</td>
</tr>
<tr>
<td>Very High</td>
<td>5</td>
<td>81% to 100% chance of occurrence</td>
</tr>
</tbody>
</table>
Consequences of Occurrence

The table below details the metrics used for the Consequences of Occurrence evaluations.

<table>
<thead>
<tr>
<th>Consequence Measure</th>
<th>Consequence Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Insignificant</td>
</tr>
<tr>
<td>2</td>
<td>Tolerable</td>
</tr>
<tr>
<td>3</td>
<td>Serious</td>
</tr>
<tr>
<td>4</td>
<td>Critical</td>
</tr>
<tr>
<td>5</td>
<td>Catastrophic</td>
</tr>
</tbody>
</table>

- **Insignificant** — something that will not affect project development.
- **Tolerable** — something that affects development of a project at some level (e.g. Requirements Change, Staff getting sick).
- **Serious** — something that might significantly influence scheduling and project development (e.g. Lack of knowledge).
- **Critical** — something that might cause project development to freeze (which also affects scheduling) for example misunderstanding of the requirements or hardware failure.
- **Catastrophic** — project might not be completed at all (e.g. Data loss, budget cuts).
Hardware and Software Resource Requirements

**Hardware Requirements:**

- PC x 3
- Printer
- Internet Connection

**Software Requirements:**

- Eclipse with JUnit plugin shall be used to develop and test the project.
- MySQL shall be used to develop the database required for the project.
- Google Maps integration shall be used to find out where customers are located.
- PHP shall be used in the development of the project.
- Xammp
- JavaScript
- AJAX/JSON
- Microsoft Office Excel shall be used to organize and input users billing information.
- Microsoft Office Word shall be used to write up the requirements document.
- HTML
- CSS shall be used to design the look of the application.
- GIT shall be used to manage the changes to the project.
Work Breakdown Structure

In order to do a WBS, we had to choose a software development model for our project. Due to amount of sensitive data (Customer details etc.) that would be handled by the system, we decided that we should devote as much time as possible to the testing phases of the development particularly with regard to security of data. We decided to use the V-model as we believed that this would be more suitable in addressing those concerns.

So our WBS was based around phases of the V-model such as the different levels for Requirements, Design, Implementation and Testing. We also found it necessary to have some preparation and post analysis within the company so we can split the roles and gather necessary metrics afterwards. For the actual WBS, a top-down approach was used which was more logical to us, as we didn’t know at the time all the possible underlying tasks we might have.
Requirements Work Breakdown Structure

- Interviews
  - Prepare questions
  - Organize time
  - Interview
- User Req.
  - Req. Analysis
  - Design of UAT cases
  - System req. meeting
- System Req.
  - Req. Analysis
  - Design of system test cases

In-house Preparation Work Breakdown Structure

- Role splitting
  - Pick testing strategy
  - Set up required tools
- In-house Preparation

High Level Design Work Breakdown Structure

- Design of system architecture
- Integration test cases design
- High Level Design
Unit/Module Design

- Design printing module and test cases
- Design summary module and test cases
- Design billing module and test cases
- Design customer settings module and test cases
- Design geo info module and test cases
- Design of Database

Unit/Module Work Breakdown Structure

Implementing

- Implementing Database module
- Implementing printing module
- Implementing summary module
- Implementing billing module
- Implementing customer settings module
- Implementing geo info module

Implementation Work Breakdown Structure
Unit Testing Work Breakdown Structure

Integration Testing Work Breakdown Structure
System Testing Work Breakdown Structure

- Execution of test cases
- Analysis of test results
- Error fixing
- Final test execution

User Acceptance Testing Work Breakdown Structure

- Full system installation and configuration
- System demonstration
- Meeting with customers
- Beta system testing with real users
- Final Acceptance Meeting

In-house Work Breakdown Structure

- Define/Update processes employed
- Gather necessary metrics (i.e. test cases failure rate)
Project Scheduling

Overall structure
We used Microsoft Project to schedule our project. The structure was dictated by the Software Development Model we chose – the V model. From the figure below it can be seen that the project is scheduled to take 130 days but we have also built in 10 days for Contingency. Over these 130 days the project will be overseen by Mike Russell, the Project Manager.

Requirements
Requirements is scheduled to take 16 man days and 15 actual working days. Two tasks run concurrently with the rest running consecutively.
In-house Preparation

This phase will take 3 working days with all tasks running consecutively.

<table>
<thead>
<tr>
<th>15</th>
<th>1.2 In-house Preparation</th>
<th>3 days</th>
<th>Fri 08/11/13</th>
<th>Tue 12/11/13</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1.2.1 Meeting about role splitting</td>
<td>1 day</td>
<td>Fri 08/11/13</td>
<td>Mon 11/11/13</td>
</tr>
<tr>
<td>17</td>
<td>1.2.2 Discuss test strategy/approach</td>
<td>1 day</td>
<td>Mon 11/11/13</td>
<td>Tue 12/11/13</td>
</tr>
<tr>
<td>18</td>
<td>1.2.3 Setting up of required tools</td>
<td>1 day</td>
<td>Tue 12/11/13</td>
<td></td>
</tr>
</tbody>
</table>

High-Level Design

The High-Level Design phase will take 5 days with both tasks running consecutively.

<table>
<thead>
<tr>
<th>18</th>
<th>1.3 High-Level Design</th>
<th>5 days</th>
<th>Wed 13/11/13</th>
<th>Tue 19/11/13</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>1.3.1 Design discussion about overall architecture of a system</td>
<td>3 days</td>
<td>Wed 13/11/13</td>
<td>Fri 15/11/13</td>
</tr>
<tr>
<td>20</td>
<td>1.3.2 Design of integration test cases</td>
<td>2 days</td>
<td>Mon 18/11/13</td>
<td>Tue 19/11/13</td>
</tr>
</tbody>
</table>
Unit/Module Design

This phase will take a total of 6 actual working days and 18 man days because there are two occasions when the three team members are working concurrently on tasks which take 3 days to complete. The start and end date of the concurrent tasks are the same (this type of scheduling will have an effect on the determination of the critical path of the project).

Implementation

This Implementation is an important phase and takes 28 working days and 73 man days because of tasks being done concurrently.
Unit Testing

Unit Testing is allocated 48 days because as previously stated rigorous testing is a priority when designing a system that will deal with sensitive personal data. The actual number of man days slotted is 90.
Concurrency (Implementation and Testing)

The diagram below shows how the Implementation and Testing phases run concurrently.

Integration Testing

The Integration Testing phase will take 8 working days. The tasks in this phase are all consecutive with each task being dependant on the previous.
System Test

System testing will take 12 consecutive working days with each task having a dependency on the previous one.

Use Acceptance Test

This phase of testing takes place over 20 consecutive working days with each task having a dependency on the previous one.
In-House Analysis

The In-House Analysis phase will take 3 working days or 4 man days as the two tasks run concurrently.

<table>
<thead>
<tr>
<th></th>
<th>1.10 In-House Analysis</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.10.1 Define/Update processes employed</td>
<td>3 days</td>
<td>Tue 01/04/14</td>
<td>Thu 03/04/14</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>1.10.2 Gather necessary metrics about test cases</td>
<td>1 day</td>
<td>Tue 01/04/14</td>
<td>Thu 03/04/14</td>
<td>54</td>
</tr>
</tbody>
</table>

The Critical Path

When drawing up a schedule for the management of the project we decided to run concurrent tasks over the same timeframes. As a result there are no divergent paths through the project and consequently the Critical Path is the only path and runs for 130 days.

Milestones and Deliverables

We set Milestones to correspond to the end of each phase of the project based on the V-model. When each milestone is reached we will assess the progress of the project up to that point. The milestones are as follows:

- Requirements Elicitation and User Acceptance Test
- Requirements Analysis and System Testing
- High Level Design and Integration Testing
- Unit/Module Design and Unit/Module Testing
- Implementation

Deliverables are the work products we will give the client for evaluation. The deliverables are as follows:

- Project Plan
- Requirements Document
- Design Document
- Executable Code Modules
- Integrated Modules for testing
- User Manual
Monitoring and Reporting

Throughout the course of the project we will implement Monitoring and Reporting mechanisms as follows:

- Each week there will be informal team meetings to discuss the status of the project and any issues that need addressing.
- When each Milestone is reached a formal report will be written and submitted to the Project Manager. This report will detail what was done during that particular phase and highlight any actual or potential problems (e.g. budgetary or project implementation). A formal meeting will be held with the Project Manager and the team to discuss the report. Follow-up meetings may be required.
- When a Deliverable is ready a report will be written and both will be handed to the Client. A formal meeting will then be arranged to discuss with the Client any issues arising. Follow-up meetings may be required.