Software Project Management Plan

for

Lunar Rover Mapping Robot Controller

Version 2.0.0

Prepared by Simon Gray
SEP UG-17

School of Computer Science,
The University of Adelaide

2017 October 30

Change History

Version	Dated	Edited By	Change Summary
0.0.1	06/08/2017	Simon Gray	Initial Draft
0.0.2	04/09/2017	Yong Yang	Grammar Fix
0.0.3	04/09/2017	Benjamin Schuh	Grammar Corrections
1.0.0	04/09/2017	Simon Gray	Draft
1.0.1	29/09/2017	Simon Gray	Added Milestone 1 Details, Developing Version
1.0.2	29/09/2017	Benjamin Schuh	Unification of document style and grammar, Added Milestone 2
1.1.0	29/09/2017	Simon Gray	Prototype Development, Added: Risk 22
1.1.1	29/10/2017	Simon Gray	Including Ben for roles and updating naming scheme
2.0.0	30/10/2017	Benjamin Schuh	Final grammar and format

Related Documentation

ID	Document Name	Version
1	Software Requirements Specifications	2.3.0
2	Software Design Documentation	2.0.0
3	Testing Report	1.0.0
4	User Manual	1.0.0

Contents

1 Introduction

1.1 Scope

This document aims to define the activities to be completed, management of activities, procedures, schedules, deliverables and resources required to deliver the Lunar Rover Mapping Robot (Rover).

1.2 Assumptions and Constraints

Assumptions are conditions assumed but not necessarily proven or confirmed at the start of the project. The list will contain all necessary information that is assumed throughout the project's development. Constraints are the limitations set during the project about potentially, required item availability or possible tool usage.
Refer to Appendix A for the list of Project Assumptions.
Refer to Appendix B for the list of Project Constraints.

1.3 Program Objectives

The main objective of this project is to create an autonomous Rover that will safely travel to a user targeted location, and mapping all objects that are encountered during the journey. After arriving at the user designated location the Rover will observe and map the location while looking for the remnants of Apollo 17. After it has found the remnants of Apollo 17 it will continue mapping all safe areas, until it has explored everything possible, it will then return to its initial deployment location. The Rover will have its safety as a priority and shall avoid the No Go Zones (NGZs), those that are found in the environment and also objects that are manually designated from the user.

1.4 Reference

The following documents are referenced by the SPMP.

Table 1: References

Ref. No.	**Reference**
Ref 01	Project Deliverables
Ref 02	Course Group Project Specifications
Ref 03	Client Group Project Specifications
Ref 04	Documentation- Software document and agenda template
Ref 05	Formal Spec
Ref 06	DTD
Ref 07	Coding Conventions Source: http://www.oracle.com/technetwork/java/code-conventions141999.html

Note: Referenced documentation 01 through to 06 was located on the University of Adelaide MyUni website page for Software Engineering & Project

1.5 Evolution of the plan

This SPMP may be updated time from time over the course of this project and changes to the plan will be made as efficiently as possible at the earliest opportunity for required project

personnel. As this project is being delivered using agile methodologies the framework of the project shall be more accommodating of changes, helping to ensure the projects success.

1.6 Definitions

Table 2: Acronyms

Acronyms	**Meaning**
GUI	Graphical User Interface
NGZ	No Go Zone
Rover	"LEGO MINDSTORMS Education EV3 Core Set" construction hardware
SDD	Software Design Document
SPMP	Software Project Management Plan
SRS	Software Requirement Specifications
UI	User Interface
DTD	Document Type Definitions

2 Project Deliverables

1. Software Requirement Specification (SRS): The report of all the functional and non-functional client requirements and use cases that elaborate on user interactions with the Rover that the software will facilitate.
   Final version is due: 30th of October 2017 at 23:59

2. Software Project Management Plan (SPMP): The software project management plan describes the project, including work plan, project management process, definitions, scope, assumptions, and constraints. It will also list all the roles and responsibilities required from everyone that is participating in the construction of the project.
   Final version is due: 30th of October 2017 at 23:59

3. Software Design Documentation (SDD): A technical report outlining the description of the software design and architecture of the product.
   Final version is due: 30th of October 2017 at 23:59

4. User Manual: The Report for how the user is to operate the deployed software. The language of the manual will allow users with minimal knowledge of the project to understand how to control the Rover. There will also be a list of the limitations of the software, stating what it is designed to do and what it isn't designed to do. This will cover both the GUI and the Rover's capabilities.
   Final version is due: 30th of October 2017 at 23:59

5. Test Report: The report will elaborate on all tests run on the software throughout its development.
   Final version is due: 30th of October 2017 at 23:59

6. Minutes: The minutes will include the items discussed in each of the meetings, all decisions made, dates confirmed and tasks that individuals or groups have taken. It may also include, but isn't limited to, completed requirements and either positive processes, negative processes or processes that could be improved for future benefit.
   Final version is due: 30th of October 2017 at 23:59

7. Time sheet: The report of all the time that the personnel of the project worked on its construction, listing the tasks that were accomplished that week, along with what were the planned tasks for that week.
   Final version is due: 30th of October 2017 at 23:59

3 Project Approach

3.1 Project Stages

The project has been broken into five stages all of which are outlined below. All stages must be completed sequentially and stage 05 must be completed prior to the final due date for the necessary project documentation to be released. All constructed software and documentations are also due by this date.

Stage 01: Preparation and Planning
Involves gathering client specifications which will be confirmed through client meetings that will happen once a week. Organisation of personnel to create required documentation will also be appointed, which will ensure all parties will know who is responsible for what parts of the project.

Stage 02: Design and Discovery
Defining how the software will be constructed and what consistent format and design will be used for all necessary documentation. The limitations of the LEGO MINDSTORMS Education EV3 Core Set hardware will also be evaluated, which will provide required knowledge of limitations of hardware. The physical design of the robot will be completed in stage 3.

Stage 03: Build and Test
Construction of UI so it will be meaningful for a user without prior software experience. The software will also be tested as it gets developed through each iteration. The Rover's physical design will be complete so that it will be able to automatically update a map of the terrain it has encountered for the user's visual benefit. Testing will be done in sync with construction. Note: This stage will be done in sprints, working on a subset of the requirements each time, ensuring that any possible issues on the constructed software will be found at the earliest point possible.

Stage 04: Quality Assurance
The quality assurance testing will commence in Stage 03 and will be completed before Stage 05. The activities involved are:

- Test Planning
- System Testing
- System Integration Testing
- Performance Testing

Stage 05: Compilation and Presentation
Compilation and review of all the required documentation listed in Chapter 2: Project Deliverables.

4 Process Model

This project will use a hybrid of agile methodologies and waterfall methodologies, specifically, however, the disciplines of the agile methodology that will be used are the Scrum and kanban methodologies. The project will use the waterfall model for the stage guide, but will mainly adopt the agile Scrum method processes for stage 03 development as this will be done in sprints, updating and testing the software to correspond to the priorities of the requirements. The SRS will function as the sprint backlog because it will contain all the project requirements and the corresponding priorities of each. Members of the project will work on completing requirements of the highest priority first to ensure that the project will meet all of the minimum required specification. For this project, that is the requirements that have a high priority. At the end of each sprint that will be constructed in Stage 03, there will be a retrospective phase to reflect on the processes that were effective for that sprint, the processes that were detrimental and other processes that need to be developed but are showing future promise. Any changes that result from the retrospective phase can then be applied to subsequent sprints.

4.1 Scrum Methodology

The Scrum methodology works in sprints that aim to go through the product backlog, which is effectively the list of requirements. The sprints have four phases, the sprint planning, deciding what work from the product backlog that will be worked on, the development phase that will only last a designated amount of time (one week in our case), the review phase then follows and finally the retrospective phase, where the group members discuss what what was done well, what wasn't done well, and what can be improved upon. During this duration the Scrum Master will keep the team focussed on the goal. The sprint cycle will then continue once the prior sprint has been completed. This cycle will be repeated until all the requirements to be implemented in stage 3 are complete or we run out of development time.

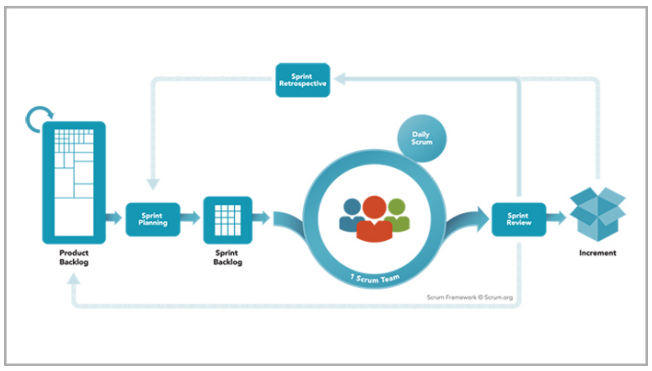

Figure 1: Scrum Example Diagram

Pros:

1. Important and required functionalities will be done first
2. Clients are heavily involved throughout the projects duration
3. Provides more flexibility than many other methodologies
4. The client has earlier visibility of the software as there are multiple sprints each including a demonstration to the client

5. At the end of a sprint there is a reflection phase, that will let you review what went well, what was disadvantageous and what could be improved upon
6. With frequent delivery at the conclusion of each sprint, the clients will be able to experience the current state of the software

Cons:

1. Correctly following the scrum methodology requires all personnel to understand how the methodology is desired to operate, and this will take time
2. Agile prioritises working software over comprehensive documentation and so the documentation may fall behind because of all the sprints, this is more common then in a methodology like waterfall
3. In a project where costs are involved, this methodology will be unable to give an actual value for "how much this will cost me", as for a real work project the cost is a very important topic
4. Heavy participation of the client is required, this may not be possible in a real world environment because there is the potential to have many other projects running at the same time or the client may have competing priorities

4.2 Kanban Methodology

The Kanban methodology is structured to achieve the single highest priorities first, and then once that requirement with the highest priority is advanced as far as possible, only then the next highest requirements development can begin. This methodology usually has 5 stages which each requirement will need to go through in order to advance to the next stage. They are, in order: Analyse, Design, Code, Test and then Deploy (Note that the stages can vary from project to project). If there are individuals that are unable to help the development for a stage then they will work on the next highest priority requirement that they are able to, ensuring that their time will not be wasted.

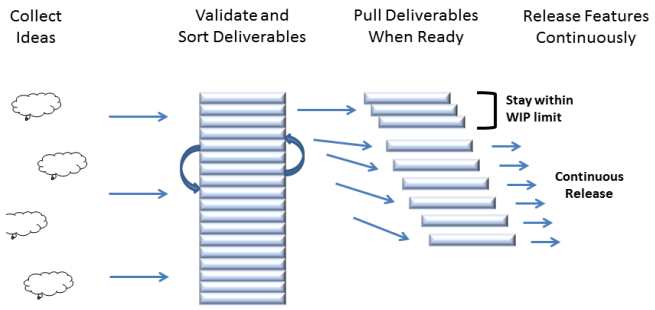

Figure 2: Kanban Iteration

Pros:

1. Important and required functionalities will be done first
2. Planning flexibility: Product owners are able to change priorities of items and this is accommodated by the product backlog
3. Fewer bottlenecks; assuming the items are chosen in the correct manner then there will not be as many situations that you have to wait for other items to be completed before you can start your own

Cons:

1. A lack of timing, as there is no time frame associated with each phase
2. It is difficult to forecast the completion of development as requirements are only reviewed at the deployment stage
3. Increased amount of overhead because there are more builds and releases because of the continuous release cycle

4.3 Waterfall Methodology

The waterfall model could also be called a linear-sequential life cycle model, because each prior phase must be completed before the next phase can begin. At the end of each phase a review takes place to determine if the project is on the right path.

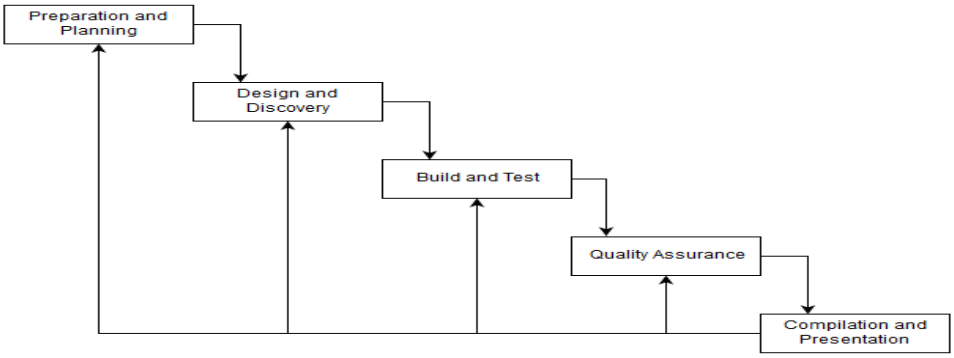

Figure 3: Waterfall Diagram

Pros:

1. Clearly defined stages, easy to tell where the project is at, along with what it has done and what it needs to do
2. Time for expected completion of a task is clearly defined, shows if the project is deviating from schedule
3. What you plan is what you get, there won't be any surprises on what you will need to do
4. The model is really easy to understand

Cons:

1. By not gathering all requirements or incorrectly identifying all the needed components the project could be required to start at the first phase again and then continue from that point potentially getting the project behind schedule if identified too late.

2. Assumes you know all requirements at the start, meaning risk isn't mitigated until late in the project
3. Its needs a large portion of the project to be completed before you have a project to deliver; deliverables are only at the end of the project, not at any earlier point
4. Changes can have a snowball effect, a small change to the requirements late in the project for example could need many changes for the final product

4.4 Process Model Choice

A hybrid approach using select features of each methodology was chosen.
Parts of all three methodologies were chosen because each had certain characteristics that were desirable for the project.
The waterfall model's project stage was considered beneficial because it gave a clear definition of what each stage of the project was, along with showing that the project is unable to advance to the next stage without finishing the current one.

The agile methodology provided the requirement prioritisation, and the two subsections of the agile methodology used where:
The Scrum methodology, as the sprints are beneficial because of the weekly client meeting that we have, this means that it was easy to have a meeting before the client meeting to discuss what we have done, what needs to be presented and what we plan to do before the meeting next week.
The Kanban methodology is useful because it ensures the highest priority requirements are progressed as far as possible before work commences on the next highest priority requirement.

The project has used parts of all three of the previously mentioned methodologies to give the project the process model that will work best with this project.

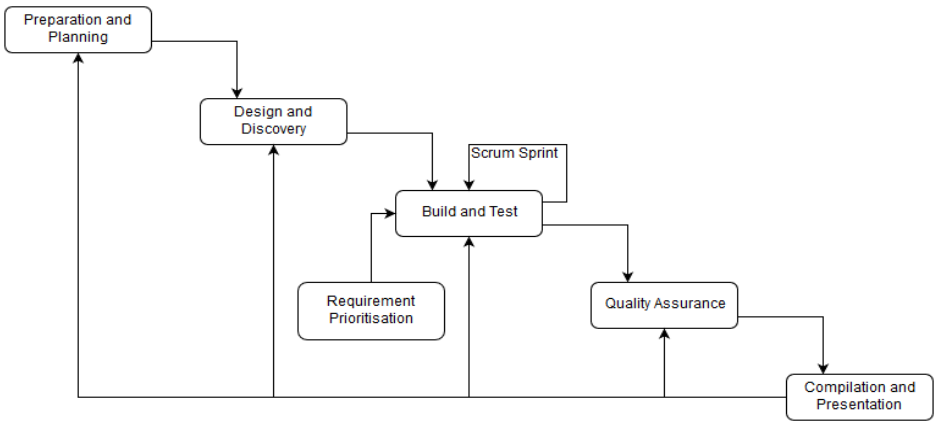

Figure 4: Combined Process Model Choice

5 Project Organisation

5.1 Roles and Responsibilities

Table 3: Group Role Allocation

Role	**Job Description**	**Name**	**Reason for Job**
Project Manager, Developer	Make sure all of the tasks necessary are allocated along with planning, design and execution of the project	Gray, Simon James	Appointed Project Manager because of past experiences with time management and sporting experiences.
Business Analyst, Developer	Responsible for gathering all of the requirements and the interaction with the clients	Yang, Yong	Appointed because interest was shown in desire to do this job, also found he was the best suited personnel for requirement gathering.
Technical Analyst, Developer	Lead for the technical development of the project, assigning what tasks need to be done and who is responsible for the completion of them	Kozirev, Matthew	Appointed as interest was show in leading the technical team because of a desire to be in a leadership position.
Lead Tester, Developer	Responsible for making sure that the software has been thoroughly tested	Pham, Hoang Long	Experience coding and was the first person to put their hand up to be the leading tester.
Document Manager, Developer	Ensure that documents are consistent among all necessary documentation	Schuh, Benjamin Thomas	Experience writing documents because of studying an engineering degree which requires multiple documents to be produced, meaning he was the best fit for the job.
Software Developer	Developing the software required and assistance in documentation development	Khuc, Ngoc Chau	Appointed software developer because of readiness to do any required task for the benefit of the project
Software Developer	Developing the software required and assistance in documentation development	Zaky, Katon Akhmad	Appointed software developer because of readiness to do any required task for the benefit of the project

6 Conventions

6.1 Naming Conventions

All project documentation shall be named in the form:

`UG-17_[TYPE]_[DESCRIPTION]_[VERSION]`

The type will display what the documentation is, for example, SRS, SDD, SPMP, etc. All of the type definitions will be between three and seven characters long. The description will state what the document is. For example, this could be a draft or the minutes taken with the date. The version will then reflect the current version in the schema labelled in the section 6.2 Version Number Schema.

6.2 Version Number Schema

All documents that require a version number schema shall be defined by three numbers. The first digit specifies the major version and is increased by 1 when major changes are made to the documentation. The second digit is for the addition of new content, and the final digit is for minor fixes.

6.3 Code Commenting Scheme

We used the standards from reference 07 as our starting point for commenting code however we made small modifications as the group saw fit.
The documentation standards agreed by the group were as follows:

1. At the start of all included code file there will be a comment explaining what the code does, who the author/s is/are, the current version of the code and the date that the code was last modified. An example of this would be:

```
/**
 * The GridMap class contains functions relating to map operations. It draws
 * the map as a grid with each cell representing a small area. Functions relating
 * to the modifications of these cells such at the lining functions are contained
 * within this class
 * @author Hoang Long Ngoc Pham, Yong Yang (Maximilian), Katon Akhmad Zaky
 * @version 0.2.0
 * @since 20-9-17
 */
```

2. All functions will have a comment at the beginning, giving a brief summary of what the goal of the function is, the author of the function, what input variables it takes if any and what the function is expected to return, if any. An example of this would be:

```
/**
 * Converts a double to a float
 * @author Simon Gray
 *
 * @param input(double) value to be converted
 * @return = float value
 */
public float convertToFloat(double input)
{
    return (float)input;
}
```

3. All variables will have a short description explaining their purpose. An example of this would be:

```
//The map offset to reposition the grid map to the center of the map display
private int mx, my;
```

4. Comments may also be added to sections of a function to explain what the purpose of certain lines may be, as this may not be intuitive.
   An example of this would be:

```
// If it's a click, a signal mouse is pressed and a new NGZ is being drawn
if ( clicked == false )
{
    clicked = true;
    shapeToFill = true;
    newNGZ.clear();
}
```

7 Work Plan

7.1 Work Activities

For efficiency, the project will be divided into two sections; the GUI section, and the Rover section. In each of these groups, members will be divided into smaller subgroups to complete tasks of the highest priority first. All members of both groups will be interchangeable and personnel will be changing tasks to ensure that the highest priority requirements will be completed first. The Rover group and the GUI group will also be broken into smaller subgroups as required for tasks like map representation along with buttons, layout, localisation, movement, mapping and path finding, communications, and safety.
Refer to Figure 5 for a visual display of the work breakdown structure.
Everyone will work on documentation when necessary, however, the Product Manager will be accountable for working on the SRS, the Project Manager will be accountable for the SPMP and the Chief Engineer will be accountable for the SDD. Once a task has been completed to an acceptable standard then the personnel will be interchangeable between the two groups of the GUI and the Rover.

7.1.1 Group Breakdown

Rover Group
The rover group will involve four members, Matthew Kozirev, Benjamin Thomas Schuh, Ngoc Chau Khuc and Simon James Gray. The tasks of this group revolve around the Rover, with the task of producing software to safely move to a targetted location avoiding NGZ's, this will involve producing an algorithm to calculate the shortest safe path to the targetted location. The group will also need to accurately keep track of the rovers coordinates and heading while the rover is in deployment. The mapping will be done by Ben and Simon. The communication between the Rover and GUI will be done by Matthew. The general movement of the Rover will be done by Chau. The safety of the Rover while in movement will be addressed by all individuals of the Rover group.

GUI Group
The GUI group will involve three members, Yong Yang, Katon Akhmad Zaky and Hoang Long Pham. The task of the GUI Group is the interface that will be displayed to the user. The GUI will be split into multiple sections:

1. Main Rover Map: Displaying the map of the "Moon" in the accepted range
2. Mini Map: Displays a smaller map around the rover
3. Tool bar: Rover options and settings
4. Control Panel: User Manual
5. Legend: Visual guide for corresponding meaning of all symbols on the map

Katon, Hoang Long - Rover maps
Katon, Hoang Long - Colour Scheme
All - Map Functionality
All - Tool bar
More information regarding the tasks of each of these groups is outlined in greater detail in the second related document (ID no.2), which is the SDD, along with the following figure.

7.1.2 Work Interaction Chart

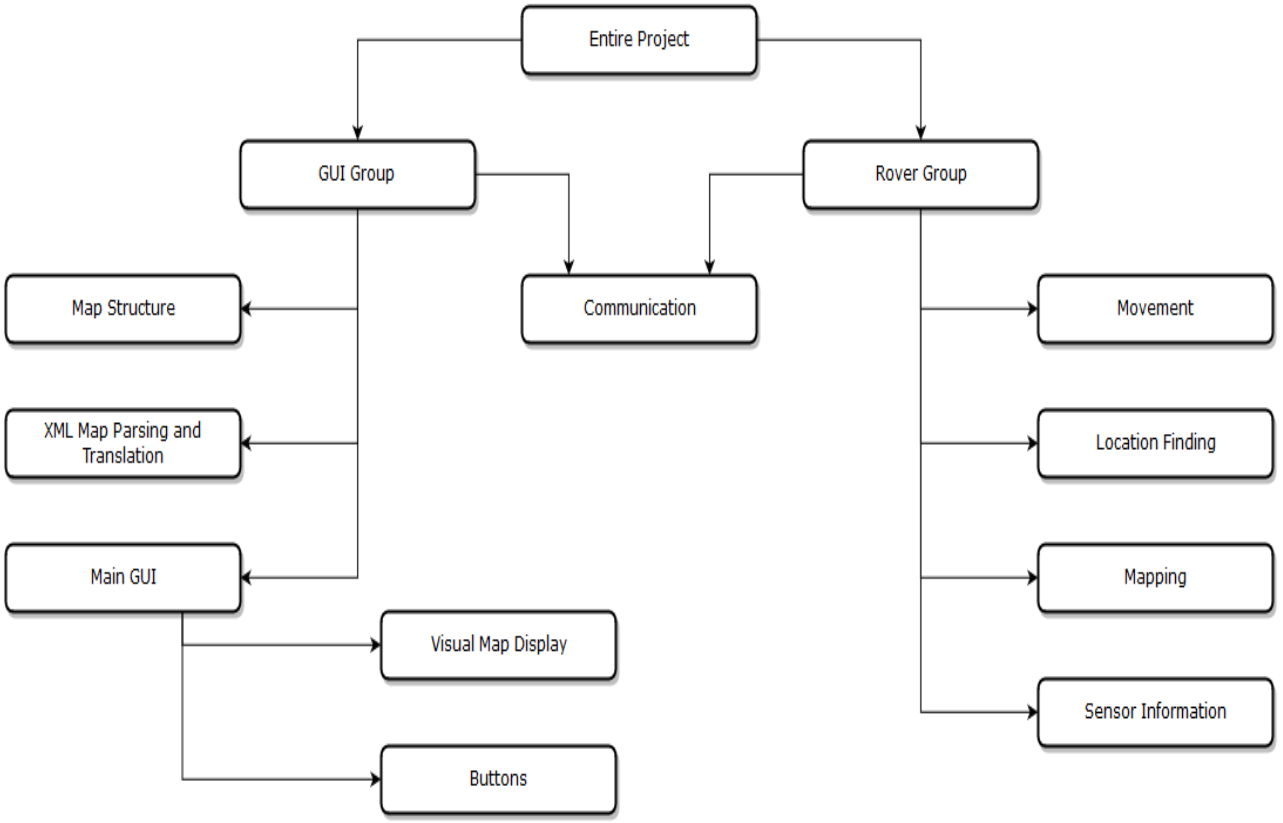

Figure 5: Work Breakdown Structure

7.2 Milestones

The project will have two milestones that will be set by the project team on the meeting of week 7 (05/09/2017) and that of week 8 (12/09/2017) respectively. Both milestones will be due on the next client meeting, that is, on (12/09/2017) and (03/10/2017)

Table 4: Project Milestone Due Dates

Ref.	**Due Date**	**Time Due**	**Milestone**
01	22/08/2017	09:00	Software Requirement Specifications Draft
02	29/08/2017	09:00	Software Requirement Specifications Review
03	05/09/2017	09:00	Software Project Management Plan Draft
04	12/09/2017	09:00	Software Project Management Plan Review
05	12/09/2017	09:00	Risk Management Plan Review
06	12/09/2017	09:00	Configuration Management Plan Review
07	03/10/2017	09:00	Software Design Document Draft
08	10/10/2017	09:00	Software Design Document Review
09	10/10/2017	09:00	Code Review
10	10/10/2017	09:00	Testing Review
11	27/10/2017	23:59	Software Final Version
12	28/10/2017	23:59	Testing Report Final Version
13	28/10/2017	23:59	Software Requirement Specification Final Version
14	28/10/2017	23:59	Software Project Management Plan Final Version
15	28/10/2017	23:59	Software Design Document Final Version

7.2.1 Milestone 01

The items that were selected for milestone 1 were as follows:
Requirement 0003 - Self Protection(Partial: One of the three sensor input)
Requirement 0005 - Moving to designated location(Partial: By input of distance and degree
Requirement 0006 - Rover Map
Requirement 0007 - Options to import and export Rover Maps(Partial: With mock up data)
Requirement 0009 - Option to mark NGZs on the Rover Map
Requirement 0010 - Option to designate a point to move to
Requirement 0011 - Four arrow keys to control the Rover's movement
Requirement 0012 - Option to switch between manual mode and automatic mode

7.2.2 Milestone 02

The items that were selected for milestone 2 were the remaining high priority requirements to ensure that the product would meet the lowest level requirements:
Requirement 0001 - Target Locating
Requirement 0002 - Area Survey
Requirement 0003 - Self Protection(Full implementation)
Requirement 0004 - Automated Return
Requirement 0005 - Moving to a designated location(Full implementation)
Requirement 0007 - Option to import and export Rover Maps(Full implementation)

Requirement 0008 - Options to choose different colour schemes
Requirement 0021 - Option to remove user drawn NGZs
Please refer to the Document "Software Requirement Specifications" (ID 01), for a detailed list of what is included in each of the requirements.

7.3 Schedule Allocation

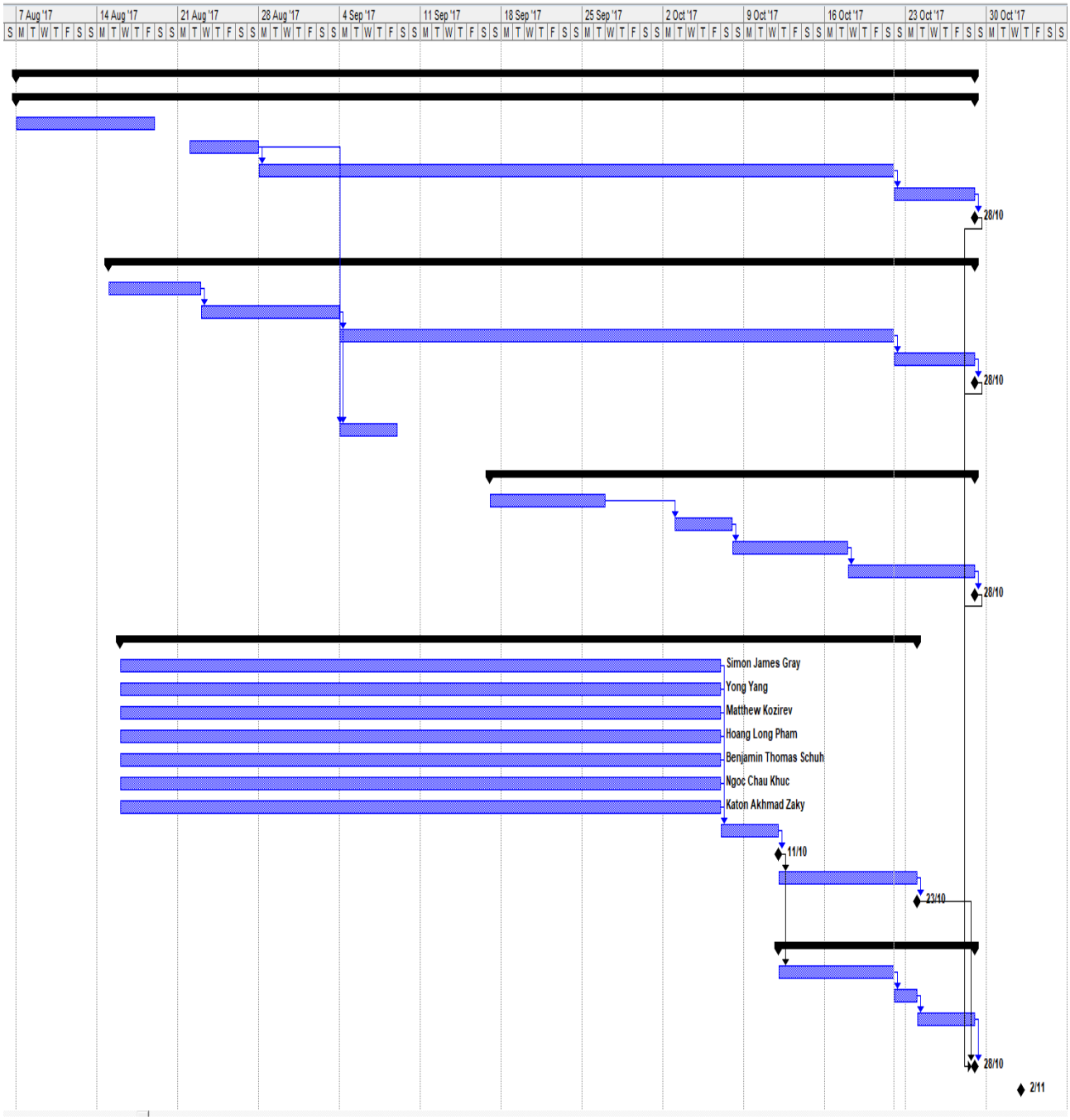

Figure 6: Gnatt Chart

Task Name	Duration	Start	Finish	Predecessors	Resource Names
Project Starts	0 days	Mon 24/07/17	Mon 24/07/17		
Lunar Rover Mapping Robot Project	**83 days**	**Mon 7/08/17**	**Sat 28/10/17**		
Software Requirement Specifications	**83 days**	**Mon 7/08/17**	**Sat 28/10/17**		
Software Requirement Specifications Draft	12 days	Mon 7/08/17	Fri 18/08/17		
Software Requirement Specification Review	6 days	Tue 22/08/17	Sun 27/08/17		
Software Requirement Specification Final Vers	55 days	Mon 28/08/17	Sat 21/10/17	6	
Contingency	7 days	Sun 22/10/17	Sat 28/10/17	7	
Software Requirement Specification Due Date	0 days	Sat 28/10/17	Sat 28/10/17	8	
Software Management Plan	**75 days**	**Tue 15/08/17**	**Sat 28/10/17**		
Software Project Management Plan Draft	8 days	Tue 15/08/17	Tue 22/08/17		
Software Project Management Plan Review	12 days	Wed 23/08/17	Sun 3/09/17	12	
Software Project Management Plan Final Versi	48 days	Mon 4/09/17	Sat 21/10/17	13	
Contingency	7 days	Sun 22/10/17	Sat 28/10/17	14	
Software Project Management Plan Due Date	0 days	Sat 28/10/17	Sat 28/10/17	15	
Risk Management Plan	5 days	Mon 4/09/17	Fri 8/09/17	13,6	
Software Design Document	**42 days**	**Sun 17/09/17**	**Sat 28/10/17**		
Software Design Document Draft	10 days	Sun 17/09/17	Tue 26/09/17		
Software Design Document Review	5 days	Tue 3/10/17	Sat 7/10/17	21	
Software Design Document Final Version	10 days	Sun 8/10/17	Tue 17/10/17	22	
Contingency	11 days	Wed 18/10/17	Sat 28/10/17	23	
Software Design Document Due Date	0 days	Sat 28/10/17	Sat 28/10/17	24	
Construction	**69 days**	**Wed 16/08/17**	**Mon 23/10/17**		
Software Development	52 days	Wed 16/08/17	Fri 6/10/17		Simon James Gray
Software Development	52 days	Wed 16/08/17	Fri 6/10/17		Yong Yang
Software Development	52 days	Wed 16/08/17	Fri 6/10/17		Matthew Kozirev
Software Development	52 days	Wed 16/08/17	Fri 6/10/17		Hoang Long Pham
Software Development	52 days	Wed 16/08/17	Fri 6/10/17		Benjamin Thomas Schuh
Software Development	52 days	Wed 16/08/17	Fri 6/10/17		Ngoc Chau Khuc
Software Development	52 days	Wed 16/08/17	Fri 6/10/17		Katon Akhmad Zaky
Code Review	5 days	Sat 7/10/17	Wed 11/10/17	28,29,30,31,32,3	
Software Ready For Test	0 days	Wed 11/10/17	Wed 11/10/17	35	
Bugfix and Rework	12 days	Thu 12/10/17	Mon 23/10/17	36	
Software Finalised	0 days	Mon 23/10/17	Mon 23/10/17	37	
Verification and Validation	**17 days**	**Thu 12/10/17**	**Sat 28/10/17**		
Software Testing	10 days	Thu 12/10/17	Sat 21/10/17	36	
Testing Review	2 days	Sun 22/10/17	Mon 23/10/17	41	
Testing Report Final Version	5 days	Tue 24/10/17	Sat 28/10/17	42	
Project End Date	0 days	Sat 28/10/17	Sat 28/10/17	43,38,25,16,9	
Project Presentation	0 days	Thu 2/11/17	Thu 2/11/17		

Figure 7: Gnatt Chart Information

7.4 Resource Allocation

Table 5: Resource Allocation

Task	**Start Date**	**End Date**	**Responsible Individuals**
Gather Requirements	07/08/2017	05/09/2017	All Project Personnel
SRS	07/08/2017	18/10/2017	Yong Yang
SPMP	06/08/2017	21/10/2017	Simon James Gray
System Testing	09/10/2017	20/09/2017	All Project Personnel
SDD	18/08/2017	20/10/2017	Matthew Kozirev
Release	09/08/2017	28/10/2017	All Project Personnel
GUI Development	16/08/2017	27/10/2017	Yong Yang, Katon Akhmad Zaky and Hoang Long Pham
Rover Hardware Design Development	18/08/2017	18/08/2017	Matthew Kozirev and Benjamin Thomas Schuh
Rover Software Development	16/08/2017	27/10/2017	Matthew Kozirev, Benjamin Thomas Schuh, Simon James Gray and Ngoc Chau Khuc
Testing Report	09/09/2017	20/10/2017	Simon James Gray, Hoang Long Pham
User Manual	09/09/2017	20/10/2017	Yong Yang, Hoang Long Pham and Katon Akhmad Zaky

7.5 Quality Management

To ensure quality is maintained, the clients feedback will be an important aspect. To ensure the project meets its goals regular client input will ensure that the software and hardware developed will be focussed on the clients desires. Goals and objects will be clearly defined, giving the client and personnel working on the project a clear understanding of what is required of them, along with what specifications the software and hardware will need to meet. There will be certain standards for the project including the documentation and other aspects of the project, including but not limited to:

1. Adaptability: The code created will be designed in a manner that objects are configurable and hard coding is to be minimised.
2. Refactoring: The code will be modified to improve readability and usability or its structure when necessary to conform to the coding standards set by the projects team.
3. Maintainability: The code shall meet standards set by the group so that if a future project team were to take the task of maintaining the code it would be clear how it is implemented to allow for easy maintenance.

Developed code will be analysed and subjected to peer reviews by multiple personnel of the group to ensure that the coding standards are met and applied uniformly.

7.5.1 Testing

Where possible the project components will be tested by another colleague on the team. While it isn't good practice, to save time, the designer who created the code may also be assigned to test their own code. The Lead Tester is accountable for making sure that all tests are adequate and the entire system is tested.

7.5.2 Documentation Review

All required project documentation will be peer reviewed by other members of the team to ensure document quality is maintained and standards are upheld. It will also hold all involved group members accountable, as they are required to approve a document before it is able to be submitted.

8 Management

8.1 Communication and Project Artefacts

The Project Manager will keep the team informed of any changes to the project schedule. Issues and risks will be explored within the group to ensure they're dealt with efficiently.
The project artefacts shall be stored in a software development folder located on the project's GitHub repository.

8.2 Risk Management

The Risk Register in Appendix C will be constantly monitored and updated throughout the life of the project. The main strategies that we will be using for identifying risks shall be:

1. Brainstorming: Gather required project personnel to talk about what possible risks there are to the project, this will get ideas to effectively "bounce off" each other to attempt to identify possible risks to the projects development
2. Scenario Analysis: Analysing different scenarios that the project will be in throughout the projects duration, and then identifying the possible risks that could appear
3. Direct Observation: Thinking about what aspects the project will encapsulate, and what issue could happen in those environments, for example in this project personnel will be working with small Lego blocks, which could get lost resulting in a delay to the project

Risks may be identified by all personnel working on the project. A risk is something that has the potential to hinder or interrupt software or hardware development and project progression. Risks that have been discovered so far will be listed along with three other categories; the chance of it happening, the impact it will have on the project if it were to happen, and the method that will be taken to reduce the impact it would have if it occurred, or a method to reduce the probability of it occurring.

Table 6: Risk Occurrence Chance

Occurrence Chance	**Reason for Rating**
High	There is a strong possibility the event will occur
Med	The event might occur, however it isn't expected to
Low	Chance of happening during the projects duration is minimal
Rare	Manifestation of hazard is very unlikely

Table 7: Risk Occurrence Impact

Occurrence Impact	**Reason for Rating**
Critical	May cause loss of developed software and/or render past work irrelevant, along with completely stopping all project development until the issue has been dealt with
High	Occurrences that will have a severe impact of project progress or which will significantly affects the potential of the project to continue until the item is addressed
Med	Occurrences that will impact the development of the project in subsections, they are unable to completely stop progression, and normally will only effect a single aspect of the project
Low	Occurrences that will have small impacts on the project to certain subsections; single individuals will be able to assert and fix these risks

8.2.1 Risk Severity Matrix

Risk Severity Matrix					
	Impact if Occurs				
Chance of Occurrence		Low	Med	High	Critical
	Rare	Tolerable	Toerable	Undesirable	Undesirable
	Low	Tolerable	Undesirable	Undesirable	Substantial
	Med	Tolerable	Undesirable	Substantial	Intolerable
	High	Undesirable	Substantial	Intolerable	Intolerable

Figure 8: Risk Severity Matrix

8.3 Issues Management

When a risk materialises or eventuates an issue will be raised. Any issues identified that could adversely affect the performance or timely completion of any aspects of the work will be considered to have a potential impact, impediment or conflict to the project. Should an impact be identified, the course of action is dependent on the nature and origin of the occurrence. Early recognition, identification and classification of potential impediments are a key factor to ensure that proper corrective action is applied at the right time in the schedule for an optimal recovery

effect. Some potential issues that could occur include, but are not limited to:

1. Undocumented changes
2. Delayed submissions

To minimise any potential issues, close communication must be held between all working members on the project.

8.4 Scope Changes

All scope changes will be documented and addressed, and the implications of the scope changes will be clearly identified along with every possible effect of this change. Any changes that will be made shall be discussed with all related members of the project to ensure that the scope change in question is handled in an effective manner and all individuals concerned shall be aware of the effects that it will make on the existing software.

8.5 Document Plan

The produced documentation shall all confer to the same specification and be produced in LaTeX, which will ensure that everything will follow the same format and design. The SRS will be mainly developed by the Product Manager, the SPMP will be mainly developed by the Project Manager, and the SDD will be mainly developed by the Chief Engineer. However, all necessary personnel working on the project will, at times, give constructive input and feedback on the produced work to ensure the best possible final product is produced. All documentation shall be reviewed by all members of the project before submission to en-sure all participants are content with the final documentation and to coincide with the project quality management.

APPENDIX

A Project Assumptions

Assumption Number	Assumption Description
A01	Meetings with the client will happen once a week
A02	All documents are required to be submitted on, or before 28/10/2017 at 23:59
A03	Meeting with client will be held on Tuesdays at 15:10
A04	Access to the LEGO MINDSTORM EV3 will be possible at all times during the project duration
A05	The software will support leJOS NXJ with version 1.7 of the Jave Development Kit
A06	All documentation needs to be held to the same standard
A07	The software language will be written in JAVA
A08	The LEGO MINDSTORM EV3 kit will contain all listed contents when first received
A09	Safe storage for the LEGO MINDSTORM EV3 kit will be possible at The University of Adelaide
A10	Standard units of metres and centimetres will be used when measuring distances
A11	All documentation will be in English of the Australian / British type
A12	The customer will alert the project personnel of any changes they desire to the requirements

B Project Constraints

Constraint Number	Constraint Description
C01	All documents need to use LaTeX in their construction
C02	The final due date is not negotiable
C03	We are unable to get additional personnel if our project is running late
C04	Code must be produced to run on the EV3 robot
C05	Clients are unable to answer highly technical questions
C06	All code must be stored on the Github portal supplied by the course coordinator
C07	Code must be documented using a tool like JavaDoc, Doxygen or something similar
C08	Testing must be done using JUnit or something equivalent
C09	A code coverage tool like, but not limited to EMMA or COBERATURA must be used
C10	Testing infrastructure must be in the GitHub repository
C11	Reused code (excluding code from the platform/framework) cannot be more than 10% of your total project.

C Risk Register

Risk ID	Risk Description	Occurrence Probability	Occurrence Impact	Method to limit impact
Risk 01	Inaccurate estimations	High	Low-High	Update estimates when inaccuracy of current estimate is observed
Risk 02	Requirements are misunderstood or misinterpreted	High	High	If anything is ambiguous, clarify the subject with the client
Risk 03	Project personnel get sick or are unable to work which will reduce project development	Med	High	Clearly define what tasks everyone is undertaking when they start working on it, so in this event it wasn't only them that were aware of their task
Risk 04	Requirements are incomplete giving the possibility to miss all client requirements	High	High	Communication with client to ensure that the project is aligned with their expectation
Risk 05	Conflicting client Specifications leading to possibility of developing incorrect software	Med	Low	Communication with client to get their desired specification
Risk 06	Requirements change possibly negating developed software	High	Low-High	Software will be constructed to accommodate changes to the best of the codes ability within the given time frame. We have used the agile Scrum methodology to help us accommodate change during coding
Risk 07	Personnel Leaves Course reducing amount of personnel available to work on project removing their domain knowledge	Rare	High	In agreement with Risk 03 there will be another person who has understanding of the personnels current tasks
Risk 08	Tasks are forgotten leading to tasks not being completed	Med	Low-High	The project manager will be accountable for communicating task allocation
Risk 09	Damaging the Rover potentially stopping ability to test software	Low	High	Keeping the Rover in the box when it isn't in use and being aware of the situations that we put it in, two hands will always be on the Rover box when it is getting transported
Risk 10	Other courses scheduled classes reducing amount time available to work on the project	High	Low	Assuming the personnel effected by this clash are organised then plans will be made to accommodate these issues

Risk ID	Risk Description	Occurrence Probability	Occurrence Impact	Method to limit impact
Risk 11	Key to Rover is misplaced giving the inability to work on the Rover	Low	Med	The personnel that access the Rover will be responsible for the key while they still have the Rover
Risk 12	Pieces of the Rover's kit are lost or the entire Rover removing the ability to do any tests with it	Low	Low-High	The personnel that are working with the Rover will not leave it unattended and shall keep track of where the pieces are distributed
Risk 13	Hardware fails so there is no ability to do hardware testing until it is replaced or fixed	Low	High	The pieces that are currently used shall be kept in the best condition possible while in use, if a single item fails work will be continued on the available components
Risk 14	Overwriting GitHub repository files with dysfunctional code negating developed items	Med	Med	All features must be implemented and tested in a separate branch then a pull request is made to facilitate code review.
Risk 15	Deleting GitHub repository removing all developed software	Rare	High	A backup of the repository will be made every release to ensure all important features are not lost
Risk 16	Rover connection is intercepted by another group causing inability to test code	Low	Med	The bluetooth pin to connect to the rover has been changed to ensure only team members can connect to the rover
Risk 17	Product does not conform to the requirements wasting	Low	Med	Features will be reviewed and compared with the Software Requirements Specification before integration and continual agreement with the client shall be sought
Risk 18	Software does not work as intended	Med	High	Design documents will be made before implementation and all features will be component tested then tested within the system
Risk 19	Project progress is not according to schedule causing deadline to be missed	Med	High	The project manager shall check the timesheet and GitHub contributions every week so that the project runs according to the schedule created and tasks can be re-allocated to ensure deadlines are met
Risk 20	There is insufficient time to implement all requirements leading to lower priority requirements to not get finished	Med	Med	All requirements are sorted by priority and the team shall work on the features of highest priority first before working on requirements of lower priority

Risk ID	**Risk Description**	**Occurrence Probability**	**Occurrence Impact**	**Method to limit impact**
Risk 21	Clients and development team don't agree on specifications of an established requirement	Med	Med	Ensure adequate communication with clients at all stages of the project to ensure requirements are correctly identified from the start
Risk 22	If we don't prepare for a client meeting before the due date then we are likely to get reduced value out of the meeting	High	Med	Project manager to make sure that meeting are prepared for in advance