



# Software Project Management Plan

for  
Archaeology Robot  
Group 13

Yufeng Bai 1600095	
Dawei Geng 1219181	
Jun Chen 1206265	
Quang Khoi Nguyen 1187070	
Shikai Li 1214223	
Yunyao Yao 1203525	
Yatong Zhou 1204471	

Version 0.1  
October 19, 2012

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Purpose and Scope . . . . .	5
1.2	Assumptions and constraints . . . . .	5
1.3	Project deliverables . . . . .	6
1.4	Evolution of the plan . . . . .	6
<b>2</b>	<b>References</b>	<b>7</b>
<b>3</b>	<b>Definitions</b>	<b>8</b>
<b>4</b>	<b>Project Organization</b>	<b>9</b>
4.1	Roles and responsibilities . . . . .	9
<b>5</b>	<b>Risk management plan</b>	<b>12</b>
5.0.1	Purpose . . . . .	12
5.0.2	Risk Assessment . . . . .	12
<b>6</b>	<b>Process model</b>	<b>19</b>
6.1	Waterfall Model . . . . .	19
6.1.1	Waterfall Model Life Cycle . . . . .	20
6.1.2	Advantages . . . . .	20
6.1.3	Disadvantages . . . . .	20
6.1.4	Minimize the Negative Impact . . . . .	20
<b>7</b>	<b>Work plan</b>	<b>22</b>
7.1	Work activities . . . . .	22
7.1.1	Documents . . . . .	22
7.1.2	Robot Physical . . . . .	22
7.1.3	Host Software . . . . .	23
7.1.4	Robot Software . . . . .	23
7.2	Milestones . . . . .	23
7.2.1	Week1 . . . . .	23
7.2.2	Week2 . . . . .	24
7.2.3	Week3 . . . . .	24
7.2.4	Week4 . . . . .	24
7.2.5	Week5 . . . . .	24
7.2.6	Week6 . . . . .	24
7.2.7	Week7 . . . . .	24
7.2.8	Week8 . . . . .	24
7.2.9	Week9 . . . . .	24
7.2.10	Week10 . . . . .	24
7.2.11	Week11 . . . . .	24
7.3	Schedule allocation . . . . .	24

CONTENTS	3
7.4 Resource allocation . . . . .	31
<b>8 Supporting plans</b>	<b>32</b>
8.1 Configuration management plan . . . . .	32
8.1.1 SVN Control Repository . . . . .	32
8.1.2 File Folder . . . . .	33
8.2 Documentation plan . . . . .	33
8.3 Quality assurance plan . . . . .	34
8.3.1 Verification and Validation process . . . . .	34
8.3.2 Standard . . . . .	34
8.3.3 Testing . . . . .	35
<b>A Risk Report Forms</b>	<b>37</b>
<b>B Project Gantt Chart</b>	<b>40</b>
<b>C Glossary</b>	<b>42</b>

## Revision History

Name	Date	Reason For Changes	Version
Dawei Geng	21 Aug 2012	basic framework of the SPMP and Chapter 1	0.1
Dawei Geng	23 Aug 2012	Chapter 3 & Section 4.1	0.19
Dawei Geng	24 Aug 2012	Section 4.2.1 & 4.2.2	0.2
Dawei Geng	25 Aug 2012	Chapter 5	0.3
Jun Chen	25 Aug 2012	Spelling & grammar check , error fix	0.31
Yufeng Bai	27 Aug 2012	Chapter 7	0.32
Yaong Zhou	29 Aug 2012	Error Fix and Grammer Check	0.33
Dawei Geng	01 Sep 2012	5.1.4 & Appendix edit & bug fix	0.4
Jun Chen	02 Sep 2012	Grammar Check & Fixing Layout Error	0.4
Dawei Geng	02 Sep 2012	Re-fix Layout Error &	0.45
Khoi Nguyen	04 Oct 2012	Review, Final Version	0.46
Yufeng Bai	07 October	Fixing chapter 6	0.47
Khoi Nguyen	07 Oct 2012	Testing plan	0.48
Jun Chen	09 Oct 2012	Review, fixing chapter 7-8	0.49
Jun Chen	10 Oct 2012	layout fixing	0.50

# Chapter 1

## Introduction

### 1.1 Purpose and Scope

This document aims to provide information that is required to manage the overall design, implementation and maintenance of the project of archaeology robot.

This document will be divided into seven parts, including the Introduction, Definitions, Project organization, Risk management plan, Process model, Work plan, and the Supporting plans.

1. **Project organization** describes the responsibilities for each member of the team and the reason behind this arrangement.
2. **Risk management** outlines the risks may be found during the development of the project, also includes risk analysis and the processes and strategies for the team members to control or overcome these risks.
3. **Process model** describes the process model will be used throughout the whole development. Including the critical paths and major stages taken in the development. Rational behind this choice will be provided, along with the advantages and disadvantages of this process model.
4. **Work plan** includes the tasks will be done to deliver the product, and the milestones that the team will be meeting during the development. Work plan will also include the duration of the tasks and the allocation of the resources to these tasks.
5. **Supporting plans** outlines the configuration management plan, the documentation plan, and quality assurance plan.

### 1.2 Assumptions and constraints

This project will be developed under such assumptions and constraints:

1. Every group member has a laptop/desktop with LeJos Driver installed.
2. Every member's computer needs to install JRE and SDK.
3. Every member's computer needs to have access to Internet and group's SVN repository.
4. A Bluetooth connection can be created between the robot and every team member's PC.
5. A Lego Mindstorm NXT robot will be used in this project.
6. This project has a duration of 11 weeks.

7. Every member have the knowledge or skills to carry out the tasks which are assigned to them.
8. This project requires at least 10 hours per week workload for every team member.

### 1.3 Project deliverables

The following items will be delivered to the client throughout the development of this project.

1. Team Poster
2. Software Requirement Specification (SRS)
3. Software Project Management Plan (SPMP)
4. Software Design Document (SDD)
5. Testing Report
6. User Manual
7. Project Milestone Demonstration
8. Final Project Demonstration

### 1.4 Evolution of the plan

This plan will be made by the team during the requirements elicitation and system design. The plan shall be made by gathering the feedback from the clients, and negotiation with clients about the milestones. In the software developing stage, the plan shall be changed under the circumstances:

1. Requirements changed by the clients
2. A defect of the plan was found
3. Other unexpected situations such as a team member is missing.

At the planning stage, clients shall be able to change or raise requirements at anytime. However once the plan have been made and the team enter the developing stage, if a change is needed, the clients shall negotiate with the project manager along with team members to achieve the changes.

If a defect of the plan was found, it will be raised in the group meetings and the whole team shall discuss and respond to the possible changes.

Other unexpected situations such as team member missing , will lead to the whole team meeting with the client to discuss how to overcome such situations.

## **Chapter 2**

## **References**

1. The project description by the client: <http://forums.cs.adelaide.edu.au/forum/SEP-S2-2012/ProjectDescription2012.pdf>
2. The SPMP template: <http://forums.cs.adelaide.edu.au/forum/SEP-S2-2012/SPMP.pdf>
3. The lecture notes: <http://forums.cs.adelaide.edu.au/course/view.php?id=452>
4. JUnit: <http://www.junit.org>
5. Minutes of client meeting
6. Minutes of Internal meeting

# **Chapter 3**

## **Definitions**

Acronym	Definition
Gantt Chart	A type of bar chart, which illustrates a project schedule.
GUI	Graphic User Interface
L <small>A</small> T <small>E</small> X	Document markup language for T <small>E</small> X typesetting program
QA	Quality assurance
SEP	Software engineering project
SPMP	Software Project Management Plan
SRS	Software Requirements Specification
SVN	Subversion repository
T <small>E</small> X	A system for computer typesetting. Used by L <small>A</small> T <small>E</small> X
XML	Extensible Markup language

# Chapter 4

## Project Organization

### 4.1 Roles and responsibilities

The team for this project is made up of seven members, which will be divided into project manager, documentation manager, QA manager, developer, and QA engineer. Each team member will have one or two specific role, however, a team member will be limited to his role. However, he or she is free to help others. In this chapter, each role's main responsibilities will be explained and the team members who are assigned to the role.

**Role:** Project Manager

**Member:** Dawei Geng

**Responsibilities:**

1. Ensuring the goal of this project will be achieved to the client's satisfaction.
2. Developing project milestones and overall project plan.
3. Organizing team members and their roles to fulfill the project needs.
4. Ensuring the project runs according to the milestones and project plan.

**Rational:**

Being a project manager requires leadership, team spirit, problem solving skills, and much more. Dawei Geng have worked in groups and stood out when a plan is needed for the project. In case of leadership, team working, and project planning, Dawei Geng did a very nice job.

**Role:** Documentation Manager

**Member:** Yufeng Bai

**Responsibilities:**

1. Organizing the project's documents and keeping documents up to date.
2. Helping project manager organize meeting agendas and minutes.
3. Keep track of the SVN repository.
4. Spelling and grammar checking.

**Rational:**

As a documentation manager Yufeng Bai must take full responsible for all the document changes and delivery, this job requires responsibility and professional English skills to manage the documents. Yufeng Bai has an excellent written skills to meet the demand of the role.

**Role:** QA Manager

**Member:** Khoi Nguyen

**Responsibilities:**

1. Planning, organizing all the test-related tasks
2. Setting up the test strategies
3. Reviewing the code and test plans with QA engineers.

**Rational:**

Besides of being a good QA engineer, the role requires additional skills such as communication, creativity and team management. Khoi can be a qualified QA manager.

**Role:** Main Developer

**Member:** Khoi Nguyen, Yatong Zhou, and Yaoyun Yao

**Responsibilities:**

1. Developing the software of the robot and the host machine based on the requirements.
2. Following the milestones and project plan.
3. Delivering high quality code and software.

**Rational:**

**Role:** GUI Developer

**Member:** Yatong Zhou and Shikai Li

**Responsibilities:**

1. Developing the graphical user interface.
2. Providing test GUI to the main developers.

**Rational:**

Yatong Zhou and Shikai Li have high level of developing graphic user interface using Java and understand the concept of presenting the useful information to the user in an elegant and efficient way.

**Role:** QA Engineer

**Member:** Shikai Li and Jun Chen

**Responsibilities:**

1. Being responsible for all the test-related tasks.
2. Following the testing plans.
3. Code reviewing.
4. Writing test report.

**Rational:**

Sikhai Li and Jun Chen are excelled at problem solving and trouble shooting. At the same time, both of them are patient, which is important in this kind of role.

# Chapter 5

## Risk management plan

### 5.0.1 Purpose

Risk is defined as the events which occurred during the development of the project, and could have positive or negative impact to the development. A certain risk may be caused by one or more causes and may have one or more impacts. This plan document's processes, tools, and procedures will be taken to manage or control the events which could have negative effect to the project. It will include:

1. Foreseeable Risks Identification.
2. Risk's Impact Prediction.
3. Risk's Likelihood Prediction.
4. Risk Response Plan.

As summary, this plan will list each identified risk based on the priority: the likelihood of certain risk and the impact of it. A plan in which to reduce the risk from occurring will be suggested.

### 5.0.2 Risk Assessment

This section will list each identified risk and give it a ID initialed with letter "R". The probability and the impact of each risk will be represented as the category of "Low", "Medium", and "High".

**R001: Team member unable to work**      **Probability:** High      **Impact:** Medium

**Description** As a three months project, team members who get sick or have personal affair will be unable to perform enough in the project. Therefore, someone else in the team will have to double the workload.

**Indicator** A team member does not attend lectures and meetings with no work submitted for over one week or call for absence.

**Mitigation** All the work shall be evenly assigned. Every group member needs to pay attention to his/her health condition.

**Response Plan** Divide missing team member's workload into parts and assign them to other group members who are available to do it.

**Category** Project risk

**R002: Conflict between group members**      **Probability:** Medium      **Impact:** Low

**Description** Team members have arguments or conflicts about the development or design.

**Indicator** Team members are arguing or refusing to complete their tasks.

**Mitigation** Every issue regarding the design or implementation of this project will be discussed in the internal meeting or client meeting. The solution of this issue shall satisfy every team member.

**Response Plan** An emergency meeting will be held within the group to discuss the controversial issue. A voting process may be held.

**Category** Project risk

**R003: Fail or late delivery**      **Probability:** Medium      **Impact:** Medium

**Description** Team member's deliverable is not on time.

**Indicator** Late commits or no commit over the preset deliverable deadline.

**Mitigation** Ensure team members have proper workload, understanding of the importance of delivery on time.

**Response Plan** Other team members shall help the team member who fail or late of delivery to make the project follow the project plan.

**Category** Project risk

**R004: Lack of contribution**      **Probability:** Medium      **Impact:** Medium

**Description** Team member fail to contribute as much as others in the team.

**Indicator** Team member has no or lack of submission to the SVN repository.

**Mitigation** Project manager shall separate the workload properly.

**Response Plan** When this problem is found, team leader or project manager shall "rebalance" the workload.

**Category** Project risk

**R005: Team member leave the project permanently**      **Probability:** Low      **Impact:** High

**Description** Due to possible reasons, one team member may withdraw from the course. He/she leaves the project permanently.

**Indicator** The team member notifies the team with certain intention.

**Mitigation** The manager should ensure that there is no critical important job that can be done by just one person.

**Response Plan** An emergency meeting will be held within the group immediately. The leaving member's original unfinished task will be reallocated to the remaining members.

**Category** Project risk

**R006: Project file lost**      **Probability:** Low      **Impact:** High

**Description** Data or files may lose caused by computer damage.

**Indicator** Team member's hardware has problems.

**Mitigation** Every team member shall keep committing to the SVN repository regularly.

**Response Plan** Data rescue methods will be carried out in order to retrieve the missing data or files.

**Category** Product risk

**R007: SVN failure**      **Probability:** Low      **Impact:** High

**Description** The team's SVN repository is "offline" or does not work correctly.

**Indicator** Grope member continuously fail to commit or update changes from the SVN repository. In addition, system failed or other error messages are received continuously.

**Mitigation** A backup repository shall be set up by the documentation manager to ensure the failure of the main repository won't affect the current workflow.

**Response Plan** Try to contact course coordinator and get the approximate date of the system recovery. In the meantime, try to retrieve files from the main repository and backup repository.

**Category** Product risk

**R008: Damage to the robot**      **Probability:** Low      **Impact:** High

**Description** One component of the robot or the entire robot was damaged by the team member.

**Indicator** Missing or broken parts of the robot or the robot won't work correctly.

**Mitigation** Team member shall take care of the robot parts and the intelligent brick.

**Response Plan** Report to project manager and course coordinator for the broken of the robot.

**Category** Project and product risk

**R009: Losing locker key**      **Probability:** Low      **Impact:** Medium

**Description** The key was lost by team member.

**Indicator** The misplacing of the key was reported.

**Mitigation** Once the team gets the locker key, a backup copy of the locker key shall be made and kept in safe place. The person in charge of the key will ensure key is kept safe at all times.

**Response Plan** Report to the course coordinator, and move the robot to a safe place using the copy of the key.

**R010: Milestone failure**      **Probability:** Low      **Impact:** High

**Description** Team is unable to meet preset milestone or major milestone which agreed with clients.

**Indicator** Team fail to implement certain the functionality by due date of milestone.

**Mitigation** Project manager shall check the progress of the project every day, and ensure the milestone will be finished on schedule.

**Response Plan** Hosting a Renegotiation with client to set new milestones. Re-schedule the tasks to team members. Team member will put more workload into this project to ensure the new milestone will be finished on time.

**Category** Project and product risk

**R011: Requirement changes**      **Probability:** Low      **Impact:** Low

**Description** Requirements pulled from client or internal meetings.

**Indicator** Client asks for changes of requirements or team decides to change requirements.

**Mitigation** In the requirements elicitation and milestone phase, the team shall make sure all the requirements are recorded and both sides(the team and the client) are satisfied.

**Response Plan** Based on new requirements' importance, the project plan shall be re-arranged. QA team shall make sure the change won't have negative impact on the project's quality.

**Category** Project and product risk

**R012: Bluetooth connection error**      **Probability:** High      **Impact:** Medium

**Description** The team member's PC will not connect to the robot's intelligent brick.

**Indicator** Team member's PC is not able to pair with robot through Bluetooth, or could not find the device over the NXJ Browse and NXJ Control.

**Mitigation** 2011 or later model Macintosh Computer(Mac) has known issue of connecting with the robot, team members who are facing such issue shall try to working with the Bluetooth-enabled Windows PC. On the other hand, team members who own a Windows PC shall check their Bluetooth before the project starts.

**Response Plan** Team member who is working with a Windows PC, if it has no Bluetooth component, shall purchase or borrow a Bluetooth connector in order to connect to the robot. Other team members who only have Macintosh computer can work on the school computer to get access to the Bluetooth connection.

**Category** Product risk

**R013: Undefined risks**      **Probability:** Medium      **Impact:** Medium

**Description** Undefined issues which have negative impact to the project.

**Indicator** Situations defined as a negative impact happened which could not find in the risk management plan and have no response plan recorded.

**Mitigation** The initial risk management plan shall be as comprehensive as possible.

**Response Plan** The one who find this issue report to the project manager. A emergency meeting shall be held to discuss such issue and overcome the risk by implementing the required response.

**Category** Project and product risk

**R014: SVN commit conflict risk**      **Probability:** Medium      **Impact:** Medium

**Description** Two or more group members are editing a same document. One of the members finishes his part and commit to the SVN repository. Others will not able to commit since their edition is not the newest version.

**Indicator** One group member find that he cannot commit a file to the SVN repository, meanwhile, error message comes out continuously.

**Mitigation** The group member should backup the parts he contributed immediately. The group member should inform other members the conflict exists, and check out the newest version, add his back up part and commit repository.

**Response Plan** Tasks allocation needs to be reviewed in the next group meeting. It is necessary to make sure that no more than one group members working on the same document in different working spaces.

**Category** Product Risk

**R015: Bluetooth Connection Risk**      **Probability:** Medium      **Impact:** Medium

**Description** The bluetooth connection is not working properly in some system, which leads to the PC can not connect with the robot.

**Indicator** At the beginning of the project, some team members found that the Mac with Mountain Lion system is not able to use the bluetooth to connect with the robot. There are also some team members' PCs do not have the bluetooth.

### Mitigation

1. If the Mac with Mountain Lion system can not connect with the robot by bluetooth, we decide to use Lion system instead of the Mountain Lion system.
2. If some PCs do not have the bluetooth, we decide to use the external bluetooth connection for these PCs

**Response Plan** After testing, the Lion system is able to use the bluetooth to connect with the robot correctly. The external bluetooth connection is also working in all PCs which do not have the bluetooth.

**Category** Product Risk

**R016: Lacking of the JAR package**      **Probability:** Medium      **Impact:** Medium

**Description** The JAR packages are used to make sure the whole system working correctly, which include the leJOS JAR, Java JAR the makefile JAR .etc.

**Indicator** Almost every developer's PC is lack of the JAR package, which has a bad influence on the project process.

**Mitigation** After negotiation, we decide to use one internal meeting to install all JAR packages which are used for the project together. It is also an opportunity to discuss all system problems and exchange experience.

**Response Plan** After this internal meeting, every developer's PC installs all needed JAR package and is able to work properly.

**Category** Product Risk

**R017: JRE 32bits and 64bits**      **Probability:** Medium      **Impact:** Medium

**Description** Since every developer uses different PC, the JRE of different PC is different sometimes. It is necessary to unify the JRE

**Indicator** The program of the system is different in 32 bits JRE and 64 bits JRE. Sometimes, the PC with 64 bits JRE can not implement the function which is made by 32 bits JRE.

**Mitigation** After negotiation, we decide every developer to use the 32 bits JRE.

**Response Plan** After unifying the JRE, the program from different developer is able to work correctly.

**Category** Product Risk

# Chapter 6

## Process model

The Waterfall Model has been chosen for this project for various reason. The first one is that it has strict deadline which help the team focus on the software delivery on time. On the other hand, the feature of having stable requirements can be very helpful to the team's development.

### 6.1 Waterfall Model

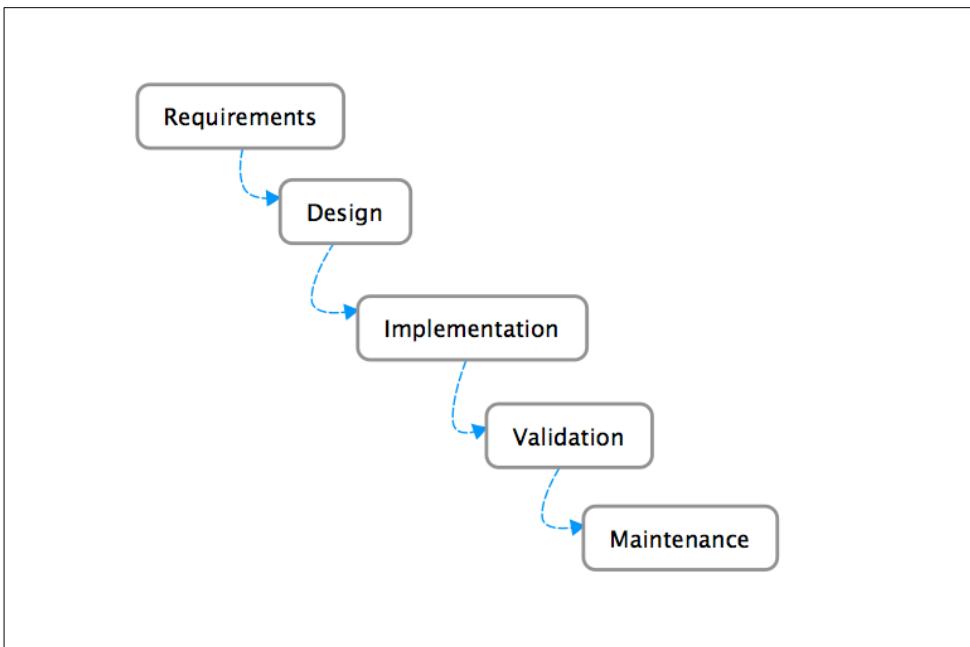


Figure 6.1: The Waterfall Model

The waterfall model is a sequential design model , which has a more simple and disciplined approach. The waterfall model provides a structured development approach. It includes:

1. Requirements
2. Design
3. Implementation (Coding or construction of the project)
4. Validation (Testing or debugging)
5. Maintenance

### 6.1.1 Waterfall Model Life Cycle

**Requirements** The requirement phase is the initial stage of the whole life cycle. At the Requirements specification phase, team will perform requirements elicitation based on the project description and the client meetings. All the selected requirements will be listed in the SRS document and implemented during the development phase.

**Design:** The Design phase is the phase that team members to design and establish a project plan. It is essential for the team to design all the aspects of the project carefully, since any changes made in later step will be very difficult to implement.

**Implementation:** Once the entire system is designed properly and completely, the main developers will start to implement the software of the project at this phase. Proper SRS documents and plans at earlier phases will be helpful.

**Validation:** After the implementation is done, the QA engineers will start the Validation phase, involving testing the overall system and ensuring all requirements have been achieved. The tests include to detect whether the system is working as the previous plan.

**Maintenance:** After the first release of the software, the team will need to maintain the system and fix possible bugs which would be found by users.

### 6.1.2 Advantages

1. Simple and disciplined linear software process flow.
2. Well written documentation will be produced earlier in the software life cycle.
3. Software release to the client on deadline.

### 6.1.3 Disadvantages

1. Unable to adopt change during development
2. Heavy documents
3. Late discovery of technical problems.
4. Can not have working software until the end of the development.
5. Testing is only completed at later stages of the project .

### 6.1.4 Minimize the Negative Impact

The waterfall model is a one way model, which is not allowed to make much change on the later phase. In order to overcome some disadvantages and minimise the potential risk of the Waterfall Model, developers need to apply the following improvements to the Waterfall Model.

1. In order to avoid changes after the Requirements phase, the requirements stage will require well documented questions and answers from the client. Any unclear features of the project must be dealt with immediately by appropriate confirmation from the client during the client meetings.
2. During the design phase, all components of the system need to be properly managed and designed with the correct requirements. In this stage, developers use one Gantt chart to ensure against any setbacks to the project. It is essential to use the risk plan to avoid and fix any issues.

3. Documentation manager need to keep all the documents up to date, and adapt to the requirements changes.
4. The implementation stage should not be started until all features are well designed and negotiated by all team member. It is necessary to record all potential issues to prevent any change of from previous phase. It is also essential to release the working prototypes to the client frequently in order to gather more feedbacks from the client.
5. During project validation, the system needs to be tested thoroughly. It is required to test the system by all team members, more test will be helpful to fix more bugs.

# Chapter 7

## Work plan

### 7.1 Work activities

#### 7.1.1 Documents

SerialNumber	Name	TimeRequired	Description
W000	Poster	2 Days	Produce a poster that show the pictures and the important information(such as name, degree and specialty) of the all the group members.
W001	SRS	1 week	A Software Requirements Specification document (SRS) is produced. It defines all the details as to the requirements of the robot software and the requirements of the host software.
W002	SPMP	1 week	Produce a Software Project Management Plan (SPMP), which provides all the information about how and who works on which aspect of the project and time arrangement.

#### 7.1.2 Robot Physical

SerialNumber	Name	TimeRequired	Description
W003	Robot Build	3 Days	The robot must be built to complete all the tasks that are required. That is including detect all the elements, mark them on the map and auto-pilot.
W004	Robot Testing	1 Day	Once the robot building is completed, it should be tested properly. These must include movement tests which tests that if the robot can moves by any directions correctly and smoothly on the map; accuracy tests which tests that if the robot can move accurately(for example if the program let the robot rotate 90 degrees, the robot should rotate exactly 90 degrees. Tests can be repeated to find small error); stability tests which aim to test that if the robot is likely to fall apart.

### 7.1.3 Host Software

SerialNumber	Name	TimeRequired	Description
W005	GUI	5 Days	Produce a GUI that has map display, manual control button, battery capacity, Bluetooth status. It should be clear and easy to use. It allow user for full control over the robot.
W006	Manual Control	5 Days	After the robot is built correctly and fully tested, the manual control software should be developed to achieve the basic movement: move forward, move backward, turn left, turn right. The robot can be controlled with PC keyboard or GUI.
W007	Speed Control	2 Days	Add speed Control function into the GUI to allow users adjust speed while the robot is moving.

### 7.1.4 Robot Software

SerialNumber	Name	TimeRequired	Description
W008	Obstacle Detection And Avoidance	2 Weeks	Robot can detect a obstacle in front and automatically find a path to avoid that obstacle. The robot should also mark that obstacle on the map. If the position of a obstacle is already known, the robot must avoid it and correct the position if there are errors.
W009	Hidden Wall Detection	1 Week	Implement the function that let the robot can detect hidden walls by light sensor and mark them on map. The hidden wall is defined by different Grayscale.
W010	No-go Zone Detection And Avoidance	1 Week	Be able to find the No-go zones and avoid them. The robot should also mark them on the map. If the position of a No-go zone is already known, the robot must avoid it and also correct the position or complete the border of that No-go zone.
W011	Path Finding	1 Week	Add a path finding algorithm to the robot, let it automatically drive itself to the destination and avoid all the obstrucer. The map is assumed to be correct, but when the robot finds an element that is not correctly marked on map it should correct it and recalculate path according to the new map.

## 7.2 Milestones

### 7.2.1 Week1

**Introduction** All the group members will be introduced and will begin reading the requirements of project.

### 7.2.2 Week2

**Poster** A poster will be made to introduce all the team members to client.

### 7.2.3 Week3

**Client Meeting** The first meeting with client will be held this week. During the meeting everyone will be introduced to the client and may ask questions about the requirement. We should start to assemble our robot now.

### 7.2.4 Week4

**Client Meeting** The second client meeting will be held, and further requirements will be discussed.

### 7.2.5 Week5

**SRS** The first draft of SRS is completed this week as well as a basic version of GUI.

### 7.2.6 Week6

**Robot Movement and Bluetooth Connection** Some basic movement function should be now achieved, such as move forward, backward and 90 degree turn. The robot should also be able connected with computer via Bluetooth.

### 7.2.7 Week7

**Manual Control** Robot movement can be fully controlled by computer

### 7.2.8 Week8

**Obstacles Location** Robot will be able to locate Obstacles and avoid them, then mark them on the map.

### 7.2.9 Week9

**Map Editor** GUI should now be able to load map file and edit map. It can add or delete elements on the map.

### 7.2.10 Week10

**Map Drawing and Auto-pilot** Robot will fully pilot and discover the map by itself.

### 7.2.11 Week11

**Error Correction and Improvement** Tests will be run on the robot to find the error and correct them. Also if there are any improvement can be made, it should all be done at the end of this week.

## 7.3 Schedule allocation

**ID:** P

**Name:** Poster

**Start:** 31/07/12

**End:** 3/08/12

**Duration:** 4 days

**Task:** Design a poster for our group, with every group member's information. Print out the poster before the due date.

**ID:** SRS1D

**Name:** Software Requirements Specification 1st Draft

**Start:** 8/08/12

**End:** 20/08/12

**Duration:** 9 days

**Task:** Divide the SRS document into parts in the group meeting. Every group member completes the parts he assigned to.

**ID:** SPMP1D

**Name:** Software Project Management Plan 1st Draft

**Start:** 20/08/12

**End:** 3/09/12

**Duration:** 10 days

**Task:** Divide the first SPMP draft document task into parts in the group meeting. Every group member completes the parts he assigned to.

**ID:** SDD1D

**Name:** Software Design Document 1st Draft

**Start:** 27/08/12

**End:** 14/09/12

**Duration:** 15 days

**Task:** Divide the first SDD draft document task into parts in the group meeting. Every group member completes the pieces he assigned to.

**ID:** TR

**Name:** Test Report

**Start:** 10/09/12

**End:** 05/10/12

**Duration:** 20 days

**Task:** Test every implemented function during these four weeks, add the test result to the test report after every testing.

**ID:** UM

**Name:** User Manual

**Start:** 28/09/12

**End:** 19/10/12

**Duration:** 15 days

**Task:** User Manual is divided into different parts according to the robot's functions. Every group member is required to write the part of the function that he developed.

**ID:** MN1

**Name:** Milestone Negotiation 1

**Start:** 13/08/12

**End:** 20/08/12

**Duration:** 5 days

**Task:** None

**ID:** MN2

**Name:** Milestone Negotiation 2

**Start:** 7/09/12

**End:** 14/09/12

**Duration:** 5 days

**Task:** None

**ID:** GUIF

**Name:** GUI Framework

**Start:** 2/08/12

**End:** 7/08/12

**Duration:** 3 days

**Task:** None

**ID:** GUICP

**Name:** GUI Control Panel

**Start:** 2/08/12

**End:** 6/08/12

**Duration:** 3 days

**Task:** None

**ID:** GUIMDP

**Name:** GUI Map Display Panel

**Start:** 6/08/12

**End:** 7/08/12

**Duration:** 1 days

**Task:** None

**ID:** GUI RIP

**Name:** GUI Information Panel

**Start:** 8/08/12

**End:** 10/08/12

**Duration:** 2 days

**Task:** None

**ID:** GUI SLP

**Name:** GUI System Log Panel

**Start:** 13/08/12

**End:** 14/08/12

**Duration:** 1 days

**Task:** None

**ID:** GUISM

**Name:** GUI Settings and Mode Change  
**Start:** 14/08/12  
**End:** 15/08/12  
**Duration:** 1 days  
**Task:** None

**ID:** GUIM

**Name:** GUI Settings and Mode Change  
**Start:** 16/08/12  
**End:** 16/08/12  
**Duration:** 1 days  
**Task:** None

**ID:** BR

**Name:** Build Robot  
**Start:** 1/08/12  
**End:** 14/08/12  
**Duration:** 11 days  
**Task:** None

**ID:** RM

**Name:** Robot Movement  
**Start:** 2/08/12  
**End:** 17/08/12  
**Duration:** 11 days  
**Task:** None

**ID:** BC

**Name:** Blue-tooth Connection  
**Start:** 16/08/12  
**End:** 16/08/12  
**Duration:** 1 days  
**Task:** None

**ID:** MCon

**Name:** Manual Control  
**Start:** 17/08/12  
**End:** 20/08/12  
**Duration:** 1 days  
**Task:** None

**ID:** RMM

**Name:** Robot Movement Milestone  
**Start:** 20/08/12

**End:** 20/08/12  
**Duration:** 1 days  
**Task:** None

**ID:** MCre  
**Name:** Manual Create  
**Start:** 20/08/12  
**End:** 21/08/12  
**Duration:** 1 days  
**Task:** None

**ID:** MDR  
**Name:** Map Display of Robot, Walls and Hidden Objects  
**Start:** 24/08/12  
**End:** 27/08/12  
**Duration:** 1 days  
**Task:** None

**ID:** Map Display of Zone (Unexplored, Explored and No-go Zone)  
**Name:** Manual Control  
**Start:** 27/08/12  
**End:** 29/08/12  
**Duration:** 2 days  
**Task:** None

**ID:** DSL  
**Name:** Display System Log  
**Start:** 22/08/12  
**End:** 24/08/12  
**Duration:** 2 days  
**Task:** None

**ID:** HCL  
**Name:** Handle Communications Loss  
**Start:** 28/08/12  
**End:** 31/08/12  
**Duration:** 3 days  
**Task:** None

**ID:** GUIM2  
**Name:** GUI Milestone 2  
**Start:** 4/09/12  
**End:** 4/09/12  
**Duration:** 1 days  
**Task:** None

**ID:** AW**Name:** Avoid Walls**Start:** 23/08/12**End:** 7/09/12**Duration:** 11 days**Task:** None**ID:** SRM1**Name:** Smart Robot Milestone 1**Start:** 7/09/12**End:** 7/09/12**Duration:** 1 days**Task:** None**ID:** FHO**Name:** Find Hidden Objects**Start:** 7/09/12**End:** 14/09/12**Duration:** 5 days**Task:** None**ID:** ASUA**Name:** Auto Survey Unexplored Area**Start:** 7/09/12**End:** 19/09/12**Duration:** 1 days**Task:** None**ID:** SRM2**Name:** Smart Robot Milestone 2**Start:** 20/09/12**End:** 20/09/12**Duration:** 1 days**Task:** None**ID:** DTMH**Name:** Data Transfer Between Machine and Host**Start:** 21/09/12**End:** 28/09/12**Duration:** 1 days**Task:** None**ID:** XFLS**Name:** XML File Load and Save**Start:** 1/10/12

**End:** 4/10/12  
**Duration:** 1 days  
**Task:** None

**ID:** FO  
**Name:** Full Operation  
**Start:** 5/10/12  
**End:** 19/10/12  
**Duration:** 10 days  
**Task:** None

**ID:** PM  
**Name:** Project Milestone  
**Start:** 22/10/12  
**End:** 22/10/12  
**Duration:** 1 days  
**Task:** None

## 7.4 Resource allocation

Task ID	Name
P	Dawei Geng
SRSD1	Dawei Geng, Yufengbai, Jun Chen, Yatong Zhou, Shikai Li and Quang Khoi Nguyen
SPMPD1	Dawei Geng, Yufengbai, Jun Chen, Yatong Zhou, Yunyao Yao
SDDD1	Dawei Geng, Yufengbai, Yunyao Yao
TR	Yufengbai, Yatong Zhou, Yunyao Yao
UM	All Members
MN1	All Members
MN2	All Members
GUIF	Yatong Zhou
GUICP	Dawei Geng, Yatong Zhou and Shikai Li
GUIMDP	Yatong Zhou and Shikai Li
GUI RIP	Dawei Geng, Yatong Zhou and Shikai Li
GUISLP	Shikai Li
GUISMC	Shikai Li
GUIM	Shikai Li
BR	Yunyao Yao
RM	Dawei Geng and Yufengbai
BC	Dawei Geng and Yufengbai
MCtrl	Dawei Geng and Shikai Li
RMM	Dawei Geng and Yufengbai
MCre	Yatong Zhou, Shikai Li
ME	Yatong Zhou, Shikai Li and Quang Khoi Nguyen
MF	All Members
MDR	Dawei Geng, Yufengbai, Yatong Zhou, Shikai Li and Quang Khoi Nguyen
MDZ	Dawei Geng, Yufengbai, Yatong Zhou, Shikai Li and Quang Khoi Nguyen
DSL	Dawei Geng, Yatong Zhou, Shikai Li and Quang Khoi Nguyen
HCL	Dawei Geng, Yufengbai, Jun Chen, Yatong Zhou, Shikai Li
GUIM2	All Members
SRM1	All Members
FHO	Dawei Geng, Yufengbai and Quang Khoi Nguyen
ASUA	Dawei Geng, Yufengbai, Yunyao Yao, Yatong Zhou, Shikai Li and Quang Khoi Nguyen
SRM2	All Members
XFLS	Dawei Geng, Yufengbai, Jun Chen and Quang Khoi Nguyen
FO	All Members
PM	All members

# Chapter 8

## Supporting plans

### 8.1 Configuration management plan

The configuration management plan demonstrates the basic plan about how we manage our project properly. It is important and necessary to make sure all items related to client's requirement are under control of the developers. The configuration management plan is also an approach to track the development and history of each developer's task. In our project, we mainly use the SVN control repository and File Folder for configuration management. The following is the specific explanation and structure of SVN and File Folder.

#### 8.1.1 SVN Control Repository

Since every developer's documents and codes are saved in SVN control repository, it is convenient to check everyone's contribution to the project by client. SVN is also working as a database which is used by developers to look up the process and modification of each one.

Then team members are able to understand the newest version of the documents and codes as soon as possible. Each group member has right to upload, download, check and modify(include delete) the files in SVN repository.

Here is the structure of SVN Repository:

**1. SEP-Group13-Code:** This folder is used to save all code related to control the robot and PC in this project.

**2. SPMP draft:** This folder is used to store all versions(except final version) of Software Project Management Plan report.

1. different versions of PDF draft
2. different versions of TEX draft

**3. agendas** This folder contains the agendas of the Client Meeting and Internal Meeting.

1. Team-13-InternalMeeting-”times of the meeting”-Agenda.pdf
2. Team-13-InternalMeeting-”times of the meeting”-Agenda.tex
3. Team-13-Meeting-”times of the meeting”-Agenda.pdf
4. Team-13-Meeting-”times of the meeting”-Agenda.tex

**4. code samples** This folder is used to collect all test codes for this project.

1. GUI
2. BTRobotServer
3. Robot Movement
4. test-codes

**5. Documents** This folder is used to store the final version of documents.

**6. SRS draft** This folder is used to save all versions(except final version) of Software Requirement Specification report

1. different versions of PDF draft
2. different versions of TEX draft

**7. minutes** This folder contains the minutes of the Client Meeting and Internal Meeting.

1. Team-13-InternalMeeting-”times of the meeting”-Minutes.tex
2. Team-13-Meeting-”times of the meeting”-Minutes.pdf
3. Team-13-Meeting-”times of the meeting” -Minutes.tex

### 8.1.2 File Folder

File Folder is used to collect all materials which is printed out(including the minutes and the agendas). The feedback and the draft documents are also saved in the file folder. By using file folder, the client is easy to understand the progress of the project and is convenient to mention more requirements for us in every meeting.

The file folder is also used as a summary for every week in internal meeting. We use folder to record the complete part and incomplete part of the project. By this way, it is convenient for us to change the time schedule for every week.

## 8.2 Documentation plan

The project requires the group to produce professional quality documentation by using LATEX. All documents should use this format and it is necessary to store all documents in pdf and tex format in SVN repository. The structure of each document is required to base on the template which is post on the CS forum. For every document, we will use "divide and merge" method, which means every document is divided to different components, each group member takes responsibility to their own part. Then each group member gives a short presentation about what they write and how they plan in every week's internal meeting.

The Required documents include:

1. **Poster** The poster is used to demonstrate the personal information of each group member including the position and the main job of each one in this project. The client is able to use poster to understand each group member's strengths and weakness.

2. **Software Requirements Specification** The purpose of these documents is to describe the tasks what need to do to reach the client's requirements. In these documents, it is also necessary to demonstrate the allocation of tasks to each group member. The client also requires analysing the potential risks and demonstrating how to avoid and solve all these risks.
3. **Software Design Document** The purpose of this report is to provide an overall architecture model of the project. In addition, we are also required to give a brief introduction on individual component task. In this document, it is necessary to use some diagrams to demonstrate the design of the whole project.
4. **User Manual** The purpose of this document is to provide a guidance and instruction sheet to the client to show them how to control this whole system. In this document, the developers need to consider that some users do not have professional knowledge related to this project.
5. **Group Milestones** The milestone is used to demonstrate the progress of the project to client. For writing this document, the group members need to negotiate the working process in the internal meeting. The milestone is able to be renegotiated by discussion.
6. **Minutes** The purpose of the minutes is to record the working progress and client's requirement in each meeting, The minutes are divide as client meeting minutes and internal meeting minutes. The client meeting minutes are used to record the new requirement from the client, the internal meeting minutes are used to record the content of group members' discussion.
7. **Agendas** The agenda is used to schedule the process of each meeting(client meeting and internal meeting). The client is convenient to understand what hope to be demonstrated by developer in every meeting.
8. **Testing report** The purpose of this document is to ensure that all students participate the testing process for the software development. Every group member is required to write a testing report to demonstrate some test case.

## 8.3 Quality assurance plan

### 8.3.1 Verification and Validation process

Each group member is required to review and check the documents and codes which are written by other group members. It is necessary to make a standard including how to write the documents and the codes. In addition, we offer the position as spell checker and code checker to check and fix every document and code to make sure the documents and codes are written as the standard. If the documents or codes is too hard to understand or do not follow the standard, the checks is able to ask the developers to rewrite the documents or codes. In internal meeting, the checkers have the responsibility to demonstrate the final version to the rest developers to do the double checking together.

### 8.3.2 Standard

#### Coding standard

1. When the developers build the Java Folder, the developers are required to use the function of this Java program as the folder name. For example, GUI developers are required to name the folder as GUI, the robot developers are required to name the folder as Movement or Sensor detection. In this way, other developers are convenient to find the needed Java program.

2. The developers are required to write the developer's name, writing date and the brief explanation of this file's function on the top of each Java file. This standard is used to make the Java file clearer to be read. Meanwhile, code checkers is more convenient to check the author and the fix date.
3. The developers are required to write the specific comments for each method, which is convenient for others to read and save the reading time.
4. The group members who modify others' program are required to write the name and the fix date on the bottom of the Java file. The modify part is also needed to be labeled.
5. The standards java code and Javadoc are mainly following the oracle standards. For the specific explanation, please look up the oracle official website.

## Documents standard

1. The file name of each document(except minutes and agendas) is required to write in format like this: Team-13-document's name-version. The client minutes is required to write in format like this: Team-13-Meeting-"times of the meeting"-Minutes, the internal minutes is required to write in format like this: Team-13-InternalMeeting-"times of the meeting"-Minutes. The internal agenda is required to write in format like this: Team-13-InternalMeeting-"times of the meeting"-Agenda. The client agenda is required to write in format like this: Team-13-Meeting-"times of the meeting"-Agenda.
2. Each group member is required to understand how to use Latex to write the file. When developers submit the documents, it is necessary to upload both the tex and pdf file.
3. The documents must be submitted in the correct folder. For example, the minutes have to submit in the minutes folder. It is not allowed to submit the files in the wrong folder. The developer has to make sure the documents submit in right place.
4. If one developer wants to delete documents, it is required to negotiate with the other group members.

### 8.3.3 Testing

#### Types of testing

1. Development Testing: where the system is tested during development to discover bugs and defects. Consisted of unit testing, component testing and system testing
2. Release Testing: where a separate testing team test a complete version of the system before it is released to users.
3. User testing: where the client test the system in their own environment. Occurs in the presentation

**How the test is run** The code is tested bit by bit by JUnit to ensure that it works according to the requirement. As the software is developed, JUnit allows regression testing as the test cases is kept in an organized way. The interface is then tested by using Abbot. The working version is expected to be available at the beginning of week 11. From that point, the build will include the tests and the automated test will be scheduled everyday. As the member develops the code, he or she has the incentive of not to break the build.

**Responsibilities** Each member is responsible for unit testing his own code. As a component is usually done by more than one member, the component testing, which focus on interfaces is done by that team. Once different versions are integrated, the system testing will be done by the main developers, who include almost all the team members to detect bugs. After that, the QA engineers, which includes Khoi Nguyen, Shikai Li and Jun Chen will do the release testing and ensures that the product meets its requirement.

**Schedule** The testing is supposed to be completed before Monday the 22nd.

**Test data** The test data is generated manually by developers who try normal and exceptional scenarios. The test data is also generated by users who interact with the system.

### Testing techniques

1. Black-box testing (Partition Testing): Input data and output results often fall into different classes where all members of a class are related. Each of these classes is an equivalence partition or domain where the program behaves in an equivalent way for each class member. Test cases should be chosen from each partition
2. White-box testing (Structural Testing): use knowledge of the code to construct test cases. It has different levels of coverage, which is statement coverage (node coverage), branch coverage (edge coverage) and path coverage. White-box testing also includes mutation testing, which effectively tests the tests.

# **Appendix A**

## **Risk Report Forms**

# Risk Report

Date:	Name:	Included:
Risk Name:		
Risk Type:		
Description:		
Impact:		
Mitigation Strategies:		
Recommended Response:		
Note: Report to be evaluated at next team meeting on ___/___/___		
Submit To Project Manager		

# Risk Event Report

Date:		Risk ID:	Solved(Y/N):
Threat Type	Tick all that Apply	Current Impact (Low/Medium/High):	
Robot Safety			
Time Deadline			
Communications			
Resources			
Operator Dispaly		Possible Future Impact (Low/Medium/High):	
Robot Functionality			
Other (specify)			

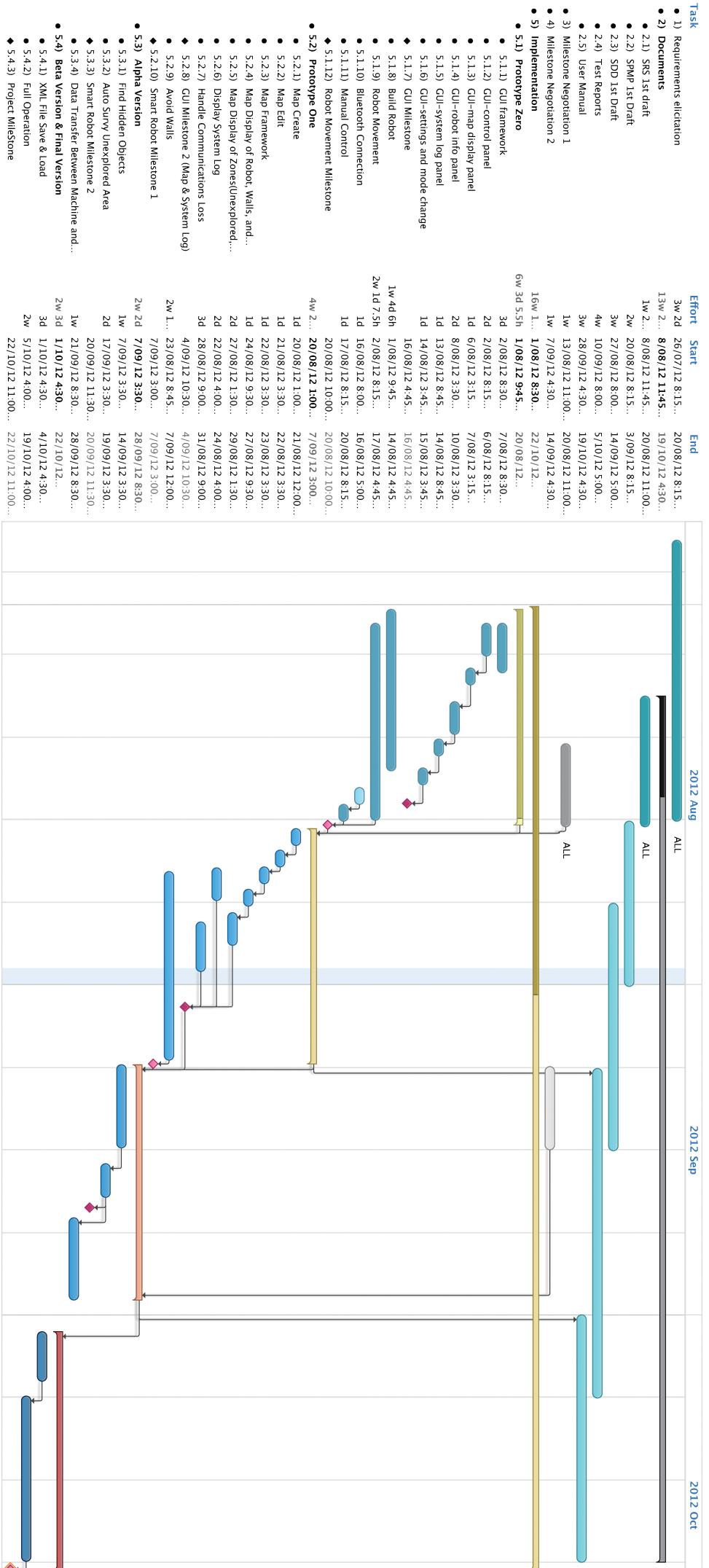
Details of Risk:

submitted to Risk Manager

Signed: \_\_\_\_\_ Date: \_\_\_\_\_

# **Appendix B**

## **Project Gantt Chart**



# Appendix C

## Glossary

**Automatic Mode:** Mode of operation, completely controlled by robot and host machine, without human input of movements.

**Bluetooth:** Wireless connection used to communicate between robot and host machine.

**Explored Area:** The area that robot have inspected, which allows robot to travel safely.

**Gird:** A size of 50mm\*50mm area in map, which contains 4 pixels. Each gird can be labeled as explored, unexplored, or "no go".

**GUI:** Graphical User Interface. It displays the map panel, control panel and status panel.

**JAR:** Java ARchive, an archive file format which is widely used by java class files.

**JRE:** Java Runtime Environment.

**Light Sensor:** Sensor used to detect hidden walls within the site, located at the front of the robot. The sensor returns the distance of the object.

**Manual Mode:** Mode of operation. Movements controlled directly through the user.

**Milestone:** A schedule to demonstrate the functions which need to implement before the due day

**"no go" zone:** A zone predefined is forbidden for the robot to enter.

**Pixel:** A size of 25mm\*25mm area in map, defined as the smallest resolution, which is also the smallest area of hidden walls that robot can detect.

**Project Risk:** The risk list include the happened and potential risks.

**QA:** Quality Assurance.

**SDD:** Software Design Document.

**SEP:** Software Engineering and Project.

**SPMP:** Software Project Management Plan.

**Ultrasonic Sensor:** Sensor used to detect walls and obstacles within the site, located at the front of the robot. The sensor returns the distance of the object.

**Unexplored Area:** The area that robot have not inspected, which can be considered dangerous.

**Waterfall Model:** A sequential design model made up of five stages.

**XML:** Extensible Markup Language. The map is saved as an XML file.