Software Project Management Plan

for

Codemmando

Hazard-Removing Robot

Prepared by:
Wong Jun Xian, Eugene *(A1656759)*
Libin Babu *(A1674841)*
Oscar Garcia *(A1633661)*
Soe Htike *(A1674836)*
Daniel Ho Canyi *(A1652591)*
Lim Tze Hwa *(A1642297)*
Akhil Prakash Sincy *(A1679207)*

School of Computer Science
The University of Adelaide

7 October 2016

Contents

Version	Date	General Description
1.0	7 October 2016	First draft.

1 Introduction

1.1 Purpose and Scope

The purpose of this project is to develop a robot system to replace humans for dangerous labour work in a hazardous environment. The robot that can follow a path to a dangerous car located in the designated place in a city and remove the car out of the city. The robot shall be able to perform the task autonomously or be remotely operated by an operator while maintaining a connection with the control panel to exchange important information happening real-time on the ground.

This document will describe the project plan,processes, structure, roles, deliverables, risk analysis, management, work plan and supporting plans.

1.2 Assumptions and Constraints

For this project we are using Lego Mindstorms EV3 robot for the removal of dangerous car located in the city. LeJOS is the firmware using for this EV3 robot. The programing language will be using java.We are using a GUI to display the map which the robot travels for mission and also displays operators menu to control the robot. GUI will also be programmed using java.

1.3 Project Deliverables

This part outlines the deliverables for this project and the due dates for each deliverable

1.3.1 First Drafts

1. Software Requirements Document - due 14/09/16
2. Software Project Management Plan - due 07/10/16
3. Software Design Document - due 19/10/16

1.3.2 Final Versions

1. Software Requirements Document - due 13/11/16
2. Software Project Management Plan - due 13/11/16
3. Software Design Document - due 13/11/16

4. Individual Testing Report-13/11/16
5. User Manual - due 13/11/16

1.3.3 Other Deliverables

1. Team Poster- due 10/09/16
2. Risk management video presentation- 9/10/16
3. Milestone 1- 14/10/16
4. Milestone 1 video presentation- 16/10/16
5. Configuration Management and Quality Process Assessment video presentation- 23/10/16
6. Milestone 2- 27/10/16
7. Milestone 2 video presentation- 27/10/16
8. SRS presentation video- 27/10/16
9. SDD video presentation- 3/11/16
10. Tools presentation video- 3/11/16
11. Testing video presentation- 12/11/16
12. SPMP video presentation- 12/11/16
13. Mystery presentation- 12/11/16

1.4 Evolution of the Plan

This plan will be reviewed and is subject to change during the course of this project. The plan will be modied to keep the document updated with the progress of the project.

Any member of the project team can make changes to this document. Changes shall be made only on notifying the documentation lead during the progress meetings. Changes to the information will be lead to a new SPMP with a new version number.

2 References

1. COMP SCI 3006NA / COMP SCI 7015NA - Software Engineering & Project: Trimester 3 - 2016 Study Guide Ngee-Ann Adelaide Education Centre. School of Computer Science. The University of Adelaide.

2. Software Engineering (9th Edition) Ian Sommerville

3. What is Incremental model- advantages, disadvantages and when to use it? http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/

4. Incremental build model - https://en.wikipedia.org/wiki/Incremental_build_model

5. Git Workflows and Tutorials — Atlassian Git Tutorial - https://www.atlassian.com/git/tutoria workflows/feature-branch-workflow

6. A successful Git branching model http://nvie.com/posts/a-successful-git-branching-model/

3 Definitions

SRS Software Requirements Specification
SDD Software Design Document
SPMP Software Project Management Plan
GUI Graphical User Interface

4 Project Organisation

4.1 Roles and Responsibilities

Responsibilities and roles for this project are divided among the group members based on each persons area of proficiency. The key roles and responsibilities of the members are detailed as below.

4.1.0.0.1 Minutes Manager: Tze Hwa

4.1.0.0.2 Agenda Manager: Tze Hwa

4.1.0.0.3 Documentation Lead:

- SRS - Eugene
- SPMP - Daniel
- SDD - Oscar

4.1.0.0.4 Version Control/Repository Managers: Eugene and Daniel

4.1.0.0.5 Lead Programmer (Robot): Oscar

4.1.0.0.6 Lead Programmer (Control Panel): Soe

4.1.0.0.7 Quality Assurance Coordinator: Akhil

4.1.0.0.8 Database Development Team: Libin

4.1.0.0.9 Graphical User Interface Development Team: Akhil, Daniel, Eugene, Libin, Oscar, Soe and Tze hwa

4.1.0.0.10 Integration Team: Akhil, Daniel, Eugene, Libin and Tze Hwa

4.1.0.0.11 Testers: Akhil, Daniel, Eugene, Libin, Oscar, Soe and Tze hwa

5 Risk Management Plan

A comprehensive risk management plan is necessary so that the team can successfully carry out its mission of building the robot hardware and software system that fulfills its client's requirements completely and successfully. Even a small chance of having an unexpected incident happening will affect the development process of the robot and the normal working of the robot is considered as a risk.

This section lists out all the possible risks neatly classified under six different categories as Requirement risks, People risks, Tools Risks, Estimation Risks, Organisational risks and Technology risks.

5.1 Risk Management Process

The team will conduct risk assessments according to risk management process illustrated below. A good and comprehensive risk management process will help to identify a number of risks before it happens during the project which in turn will help to save time and money by avoiding and rectifying the risks in an early stage. However, it is not a guarantee that all the risks can be identified before it happens but conducting the risk management process help the team to have a smooth transition during the different stages of the project by identifying most of the risks. But not conducting risk management process will most probably result in the delay of project delivery, consumption of extra budget and may even result in the failure of the project in the worst case.

The team will identify each and every risk as identified in our Risk Assessment Process and will evaluate the Probibility, Impact, Prevention Strategies, Contingency plan and Monitoring Indicators so that our client feel comfortable in case any of the risks happen during the process.

The probability of the risk determines how often the particular risk is likely to happen during the project development. The likelihood is also classified as Low, Medium and High depending on the frequency of how often the risk is likely to happen.

The impact of the risk will indicate how the particular risk will affect the project. The team has categorised the Impact into Tolerable with no or less impact, Severe with a major impact on the project timelines, and Catastrophic with project failure.

Prevention strategies will help to restrict a particular risk from happening by following the prevention methods suggested by the team.

The contingeny plan will list down instructions to be carried out for resolving the particular risk.

Monitoring indicators will list down the possible signs of the particular risk. This will help to identify the risk during incidents and to isolate the risk.

5.2 Risk Categories and Identification

All the different types of risks identified during the risk management process has been separated into 6 different risk categories as below.

5.2.1 Technology Risks

Technology Risks are related to the technical and functional aspects of the robot. They typically arise from software and hardware system used for the project.

Risk No. 1	Robot Component failure right before mission
Probability	Medium
Impact	Severe, as component failure will result in a function not to be performed which can be manageable depending on the component.
Prevention Strategies	Prepare a checklist to verify each component on a weekly basis to identify possible defects of components.
Contingency Plans	Identify and replace faulty components. Avoid running the robot having a faulty component as it may cause other components to fail as well.
Monitoring Indicators	Early indicators will be related to the functions of the particular components. If a particular function is not being executed properly by the robot, a component issue could be one of the causes.

Risk No. 2	Communication problems between Control Panel and Robot
Probability	Moderate
Impact	Catastrophic, as the user loses control of the Robot.
Prevention Strategies	Ensure the antenna and network configuration of the robot are working properly before deploying the robot for the mission.
Contingency Plans	The robot stops entirely till connection is re-established. A support team would go on-site to fix the connection if necessary.
Monitoring Indicators	Every 5 seconds, the robot will send a ping request to ensure a connection with the control panel is available. If a response isnt received, the robot will retry the ping again. If the second ping fails to get an acknowledgement, communication is taken to be broken.

Risk No. 3	Robot moves out of user connection range
Probability	Low
Impact	Catastrophic, as the mission can no longer continue safely.
Prevention Strategies	The movement of the robot would be tracked constantly, ensuring the robot doesnt move out of range. The connection status of the robot will be visible from the GUI so that the user may take caution while the connection strength weakens.
Contingency Plans	The robot stops entirely till connection is re-established. A support team would go on-site to fix the connection if necessary.
Monitoring Indicators	Slowing response times for messages sent from the control panel to the robot.

Risk No. 4	Hardware failure during mission
Probability	Low
Impact	Catastrophic, as it will affect the normal working of robot and may result in uncontrolled behaviour of robot.
Prevention Strategies	To maintain a checklist to be done every time before each mission to discover possible issues.
Contingency Plans	Use the Emergency stop button to avoid damaging the robot and its surroundings, and have on-site engineers replace the sensor.
Monitoring Indicators	User must actively monitor the robot to identify any abnormal behavior in the normal function of robot and must identify the issue as soon as possible.

5.2.2 People Risks

People risks refer to risks that associated with and arises from project team members.

Risk No. 1	Medical conditions or Unplanned issues for employees
Probability	Low
Impact	Tolerable to Severe depending on the role of employee(s)
Prevention Strategies	Although the employees have their on specific role, it is required to assign each employee to have at least one cross function role so that the employees can cover each other in case an issue arises.
Contingency Plans	Quickly identify the employee in team with suitable skills to accommodate the particular task and brief the task properly for follow up.
Monitoring Indicators	Advise employees to inform the team and organization if they plan to take leave and to arrange possible coverer to carry on the task in order to avoid delay in project.

Risk No. 2	Resignation of employees
Probability	Low
Impact	Severe to Catastrophic, depending on the role of employee(s)
Prevention Strategies	Before the project starts, advise employees to avoid joining in the team if they are planning to resign from the organization before the project completion. Inform them about incentives after completion of project and possible fine if resignation happens in before project delivery. Introduce cross function role to each employee to better management.
Contingency Plans	Assign the cross function role employee to be the temporary lead for the particular role. Hire a new employee with suitable skills to follow up on the project as soon as possible.
Monitoring Indicators	Monitor the progress of each employee. Identify the potential employee to resign by checking progress and contribution of the employee towards the project. Check with other employees regarding the employee resignation status.

Risk No. 3	Poor Co-ordination, Communication and teamwork
Probability	Moderate
Impact	Severe, as this will result in delay of project delivery.
Prevention Strategies	Brief all the employees individually to make sure the project aspects have been understood properly. Advise employees to inform the management in case the team is unable to make decisions on project due to different opinion from employees.
Contingency Plans	Quickly escalate the issues to management team to sort out possible ways along with client to deliver project in time. Identify the root cause of issue and brief all employees on how it could have been resolved properly. Encourage employees to avoid further issues and to work as a team to tackle all issues.
Monitoring Indicators	Management team must always monitor the activities and progress of the project to ensure everything is running smoothly. If any discrepancy is noted, quickly arrange a meeting with the team and sort out the issues and resolve it as soon as possible.

5.2.3 Organisational Risks

Organisation risks come from the organisational environment where software requirements for hazard removing robot are first conceived and where development occurs.

Risk No. 1	Unclear designations of roles among team members.
Probability	Low
Impact	Tolerable, as unclear designations of roles will result in inefficiency.
Prevention Strategies	Management should understand the strengths and weaknesses of each employee before assigning roles and responsibilities to them.
Contingency Plans	Switch the roles of employees to ensure a better fit.
Monitoring Indicators	Monitoring the progress of project in each stage will help to determine which areas need improvement.

Risk No. 2	Reorganization of clients management team
Probability	Low
Impact	Tolerable, the requirements and milestones of the project may not see much change.
Prevention Strategies	NA
Contingency Plans	Discuss the potential changes to the SRS with the new management, and negotiate the new timelines to accommodate the changes if there is any
Monitoring Indicators	Disorganization on the clients end, bad news about clients company.

5.2.4 Tools Risks

Tools Risks are related to the project related components and tools which are being utilized in the project.

Risk No. 1	File Management System corruption
Probability	Low
Impact	Severe as all the project files will be corrupted.
Prevention Strategies	Always create a backup copy of every file and keep it in a separate drive for recovery purpose.
Contingency Plans	Recover the files from the backup files. If there are no backups, collect all the possible files from the employees local storage and re-create the repository using these files.
Monitoring Indicators	NA

5.2.5 Requirements Risks

Requirements risks arise from customers initiated changes as a result of changes in business requirements and the process of managing those changes. They are also related to the functional and nonfunctional requirements of the project.

Risk No. 1	Miscommunication of requirements between client and project team
Probability	Low
Impact	Severe
Prevention Strategies	Upon receiving an update or change in requirements, confirm with customer regarding the proposed change and relay the changes back to project team for follow up.
Contingency Plans	Identify the root cause of miscommunication and discuss suitable strategies with client to fix the issue with minimum time delay and effort.
Monitoring Indicators	Project manager should keep track of each requirement and must relay it back to client for verification and confirmation. Client will inform if there is any further changes required on the provided update.

Risk No. 2	Frequent changes of requirements by client
Probability	Low
Impact	Severe
Prevention Strategies	Team will confirm with customer on each change made on requirements to agree that there will not be further changes in the requirements. If unsure, team will advise client to finalize the requirement and relay it back to project team. Team will analyze the changes and advise on the possible issues that will happen on project delivery, time and cost in order to avoid major requirement changes in project.
Contingency Plans	Changes on requirement will be properly communicated to team and management will advise best possible strategy to meet the requirements.
Monitoring Indicators	NA

5.2.6 Estimation Risks

Estimation risk refers to underestimation of time, effort, and capability required to develop the software system that fulfills all software requirements specifications elicited from the client with no major errors and defects residing inside.

Risk No. 1	Underestimation of project delivery time
Probability	Low
Impact	Severe, as it may result in the failure of project to be delivered in time.
Prevention Strategies	Discuss the project deliverables together as a team to find the best estimate for the time required to complete the project. Relay this to customer for review and adjust the timing according the clients needs during negotiation process. Allocate more manpower to finish tasks that have been identified by management team as high-risk.
Contingency Plans	Request an extension of the deadline of project and provide reasons for its being necessary.
Monitoring Indicators	Requirements of project are not being completed within the allocated time.

Risk No. 2	Frequent changes of requirements by client
Probability	Low
Impact	Severe, as it may result in failure of project to be delivered in time.
Prevention Strategies	Discuss the required time with team members and add a buffer for tackling unexpected events that crop up during project development which may affect the project delivery timelines.
Contingency Plans	Request an extension of the deadline of project and provide reasons for its being necessary.
Monitoring Indicators	Requirements of project are not being completed within the allocated time.

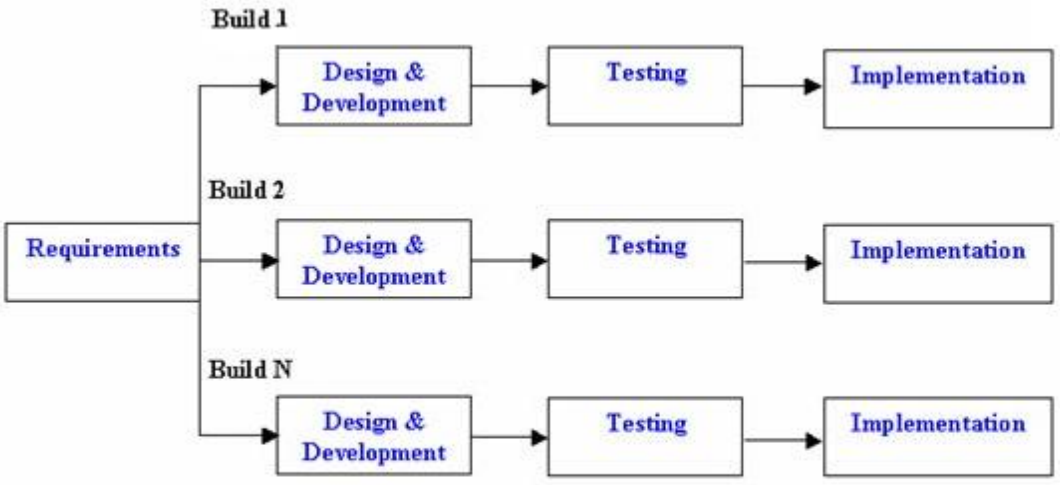
Risk No. 3	Robot behaviour did not meet expectations during Project delivery.
Probability	Low
Impact	Severe, as it depends on type and amount of deviations of robot physical behaviour from expected outcomes.
Prevention Strategies	To conduct regular testing of latest working versions of software codes to monitor if robot hardware is working as expected.
Contingency Plans	Have videographic evidence that the robot has ever been able to work as per expectations before showcasing during project delivery.
Monitoring Indicators	NA

6 Process Model

The process model that will be used for this project is based on the incremental model.

The team kick-starts the project by gathering and defining the requirements and features as detailed as possible. After which, the project is broken down into multiple smaller milestones. Each milestone is being managed based on the waterfall model, which includes the design, implementation, testing, deploying, and reviewing phases. During the review stage, the requirements and features are reviewed as well, to make sure that everything is still feasible and consistent.

This approach allows the team to produce working software quickly, that would be useful for soliciting feedback from the frequent meetings with the client. Safety features can also be given the needed attention and testing as their development would be contained in iterations.



Incremental Life Cycle Model

7 Work Plan

7.1 Work activities

The completion of the Hazard-Removing Robot project would require these main tasks to be completed:

- Control Panel program
- Robot program
- Integration
- Documentation

7.1.1 Control Panel program

The user interface of the Control Panel is what the operator will see and use during the robots mission. Actions done by the user via the controls on the interface and status displays will be powered by the Control Panel program itself.

1. Interface Design and Planning
   The robot operators requirements have to be translated into an interface design, that determines what controls and displays need to be on the user interface, which in turn determines part of the Control Panels feature set.
   Resource: Akhil, Daniel, Eugene, Libin, Tze Hwa, Oscar, Soe
   Time estimate: 3 days

2. Coding of Project Skeleton
   A continuous integration and ant build process will be put in place to ensure the Control panel project can be built in a consistent manner. The main program will be created at this stage, that will link the components of the Control Panel and have them work together.
   Resource: Soe
   Time estimate: 1 day

3. Design and building of data storage component
   Some data will have to be persistent, and will be either produced during the mission or loaded for the mission. The data storage layer will have to be built in this stage and integrated with the Control Panel.
   Resource: Akhil, Eugene, Libin, Soe
   Time estimate: 1 day

4. Java swing layout
   The Control Panel would use the Java swing toolkit to construct its user interface. During this stage, the Control Panel project would be set up, and a graphical user interface with functionality would be created, according to the design that has been planned. Graphical Elements would also have to be created. Robot responses would be simulated.
   Resource: Akhil, Eugene, Libin, Soe
   Time estimate: 3 days

5. Coding of Map component
   The Map is a major and complex component of the Control Panel. During this stage, This component would be built, tested, and integrated into the main Control Panel project.
   Resource: Akhil, Eugene, Libin, Soe
   Time estimate: 14 days

6. Integration of Control Panel Program
   Due to the distributed nature of the team and the complexity of the Control Panel, it is likely that some components would have been built in isolation. The components will have to be put together to work as a whole.
   Resource: Akhil, Eugene, Libin, Soe
   Time estimate: 3 days

7.1.2 Robot program

The robot will have to be loaded with a customized program that handles movement, sensors, and coordination with the Control Panel program via a network. This would require development work, which could happen in parallel to some of the work activities relating to the control panel.

1. Program structure and class design
   During this stage, research is done on the Robots API and a class structure is planned for the Robot program. A project skeleton is also set up.
   Resource: Oscar
   Time estimate: 3 days

2. Movement component
   The robot would need to be able to execute a set of movements on cue, either from the robots AI or from a transmitted message from the Control Panel.
   Resource: Daniel, Tze Hwa, Oscar
   Time estimate: 5 days

3. Sensors component
   Information picked up via the Robots sensors will have to be interpreted and sent to both the robots AI and to the network socket connecting to the Control Panel.
   Resource: Daniel, Tze Hwa, Oscar
   Time estimate: 4 days

4. Artificial Intelligence
   During this stage, the artificial intelligence algorithm would be written and combined with sensory input and simulated map data, to produce commands for the movement and communication (for feeding back to the Control Panel) modules.
   Resource: Daniel, Tze Hwa, Oscar
   Time estimate: 5 days

5. Integration of Robot Program
   Although it is likely that the Robot program may be continuously integrated, some time has to be allocated for resolving issues that may arise from different parts of the program working together.
   Resource: Daniel, Tze Hwa, Oscar
   Time estimate: 3 days

7.1.3 Integration

The Robot and Control Panel have to work as 1 system, although their programs are hosted in different locations. Development and testing have to be done in order to ensure that the performance and functionality of the communication system sitting between the 2 programs can meet the mission's needs.

1. Build configuration
   The continuous integration system in use will be made to always create stable and compatible versions of the Control Panel and Robot programs.
   Resource: Akhil, Daniel, Eugene, Libin, Tze Hwa, Oscar, Soe
   Time estimate: 4 days

2. Inter-program communication
   Protocols and interfaces for communications between the robot and the control panel will be planned, confirmed and built so that the two programs can exchange information needed for the mission
   Resource: Akhil, Daniel, Eugene, Libin, Tze Hwa, Oscar, Soe
   Time estimate: 8 days

7.1.4 Documentation

The systems documentation will describe the systems capabilities, how it evolves through the efforts of the project team, and its architecture. There would also be a user manual that provides direction in operating the system. The entire project team would work on the documentation.

1. SRS
   The requirements of the project are described in detail in this document

2. SPMP
   This document outlines the plan taken by the team to meet project goals, and also the organization of the team.

3. SDD
   This document contains the technical design of the system, that should meet every requirement and thus serve as the only reference for the building of the system.

4. User Manual
   The manual will include installation, pre-install and operation instructions, to help the client complete the mission.

7.2 Milestones

7.2.1 GUI Mockup

Diagrams of a proposed layout for the Control Panels user interface will be shown to the client, to solicit feedback.
Completion Criteria: The milestone is achieved if a mockup with all the user requirements is produced and feedback on it is received from the client
Due: 29/9/2016

7.2.2 Basic Robot Movement

The Robot will be programmed to execute the set of moves it needs for the mission
Completion Criteria: The milestone is achieved if the robot makes all maneuvers that have been planned while being presented to the client.
Due: 14/10/2016

7.2.3 Map Editor

The Map Editor milestone consists of demonstrating the ability to load an XML map, display a map on the GUI, edit no-go zones, and zoom in and out.
Completion Criteria: The milestone is achieved if the functionalities of the Map Editor is presented to the client, and feedback is received.
Due: 27/10/2016

7.2.4 Communication

The Communication milestone consists of getting the host to obtain robot sensor readings and invoke robot movements on demand. The emergency stop and the handling of disconnection are also included in this milestone.
Completion Criteria: The milestone is achieved if the client is shown that the Control Panel has control and visibility of the robot and that the robot behaves safely when the connection is lost.
Due: 21/10/2016

7.2.5 Prototype 1

Prototype 1 will have the basic requirements for the safe completion of the hazard-removing robot mission.
Completion Criteria: The milestone is achieved if the client is shown the setup and progression of a mission, demonstrating the safe and successful handling of scenarios such as obstacle detection, no-go zone appearance and disappearances, manual control of the robot, and variance in lighting and network environments.
Due: 27/10/2016

7.2.6 Prototype 2

Prototype 2 will have enhancements and bug fixes to prototype 1
Completion Criteria: The milestone is achieved if the client is shown that the performance of the system is improved, and problems have been rectified and feedback has been translated into modifications of the system.
Due: 4/11/2016

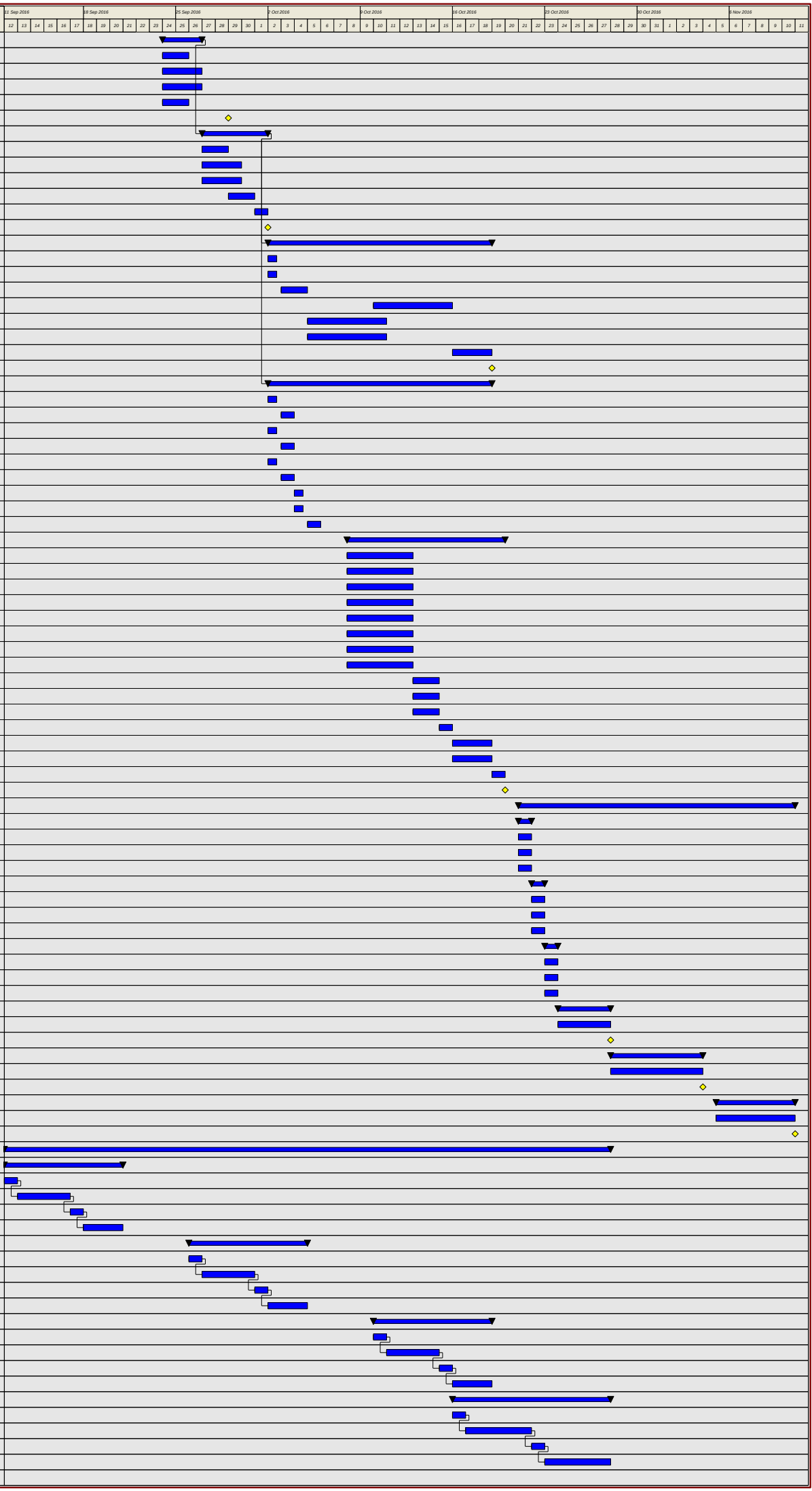
7.2.7 Release

The release will have enhancements and bug fixes to prototype 2
Completion Criteria: The milestone is achieved if the client is shown the successful completion of the hazard removal mission using our system.
Due: 11/11/2016

7.3 Schedule Allocation

The work activities are broken down into small individual tasks and shown in the diagram below, with their estimated start and end times.

ID	Task Name	Start	Finish	Duration
1	**GUI mockups & Framework**	**24-Sep-16**	**26-Sep-16**	**3d**
2	Technology Stack Confirmation	24-Sep-16	25-Sep-16	2d
3	Dtermine Map Display & icons	24-Sep-16	26-Sep-16	3d
4	Determine Allowed Manual Movement Controls	24-Sep-16	26-Sep-16	3d
5	Determine Log Data & Audit List	24-Sep-16	25-Sep-16	2d
6	Milestone 1 : GUI mockup Deliverable	29-Sep-16	29-Sep-16	0d
7	**Basic Robot Movement & Control**	**27-Sep-16**	**02-Oct-16**	**5d**
8	Project Skeleton Setup	27-Sep-16	28-Sep-16	2d
9	Motor Control & Movements	27-Sep-16	29-Sep-16	3d
10	EV3 Remote & Local Controls	27-Sep-16	29-Sep-16	3d
11	Ant Script & Automation	29-Sep-16	30-Sep-16	2d
12	Junit Test Cases on Basic Robot Movements	01-Oct-16	01-Oct-16	1d
13	Milestone 2 : Basic Robot Movement & Control	02-Oct-16	02-Oct-16	0d
14	**Integrated Command & Control**	**02-Oct-16**	**19-Oct-16**	**17d**
15	Define XML Data Structure	02-Oct-16	02-Oct-16	8h
16	Setup Code Skeleton Structure	02-Oct-16	02-Oct-16	8h
17	Map Display & Grid Layout	03-Oct-16	04-Oct-16	2d
18	Zoom Functionality	10-Oct-16	15-Oct-16	6d
19	Manage No-Go Zone	05-Oct-16	10-Oct-16	6d
20	Manage Obstacles	05-Oct-16	10-Oct-16	6d
21	Unit Testing on GUI Components	16-Oct-16	18-Oct-16	3d
22	Milestone 3 : Control Panel GUI	19-Oct-16	19-Oct-16	0d
23	**Robot Sensors & Movements**	**02-Oct-16**	**18-Oct-16**	**17d**
24	Detects obstacles from distance	02-Oct-16	02-Oct-16	8h
25	Unit Test on Obstacle Detection	03-Oct-16	03-Oct-16	1d
26	Detects beacons	02-Oct-16	02-Oct-16	8h
27	Unit Test on Beacons & Turning	03-Oct-16	03-Oct-16	1d
28	Detects & Follow Black Line (Street)	02-Oct-16	02-Oct-16	8h
29	Unit Test on Path Follower	03-Oct-16	03-Oct-16	1d
30	Detects the target car	04-Oct-16	04-Oct-16	8h
31	Lift up/Put down the car	04-Oct-16	04-Oct-16	8h
32	Unit Test on Obstacle Removal	05-Oct-16	05-Oct-16	1d
33	**Robot & Control Panel Communications**	**08-Oct-16**	**20-Oct-16**	**12d**
34	Connect Robot & Control Panel	08-Oct-16	12-Oct-16	5d
35	Exchange Movement Info	08-Oct-16	12-Oct-16	5d
36	Exchange Robot's Speed, Location & Bearing Angle	08-Oct-16	12-Oct-16	5d
37	Exchange Obstacle Sensing Info	08-Oct-16	12-Oct-16	5d
38	Exchange Color Reading Info	08-Oct-16	12-Oct-16	5d
39	Exchange Battery Level Info	08-Oct-16	12-Oct-16	5d
40	Enforce Emergency Stop & Manual Control	08-Oct-16	12-Oct-16	5d
41	Unit Test on Data Exchange & Remote Commands	08-Oct-16	12-Oct-16	5d
42	Connection Status & Heartbeat	13-Oct-16	14-Oct-16	2d
43	Maintain Live Connection & Reconnection	13-Oct-16	14-Oct-16	2d
44	Event Logging & Audit Trail	13-Oct-16	14-Oct-16	2d
45	Unit Test on Connectivity & Exception Handling	15-Oct-16	15-Oct-16	1d
46	Shortest Path Route Calculation by Control Panel	16-Oct-16	18-Oct-16	3d
47	Shortest Path Route Recalculation by Robot	16-Oct-16	18-Oct-16	3d
48	Unit Test on AI & Performance	19-Oct-16	19-Oct-16	1d
49	Milestone 4 : Robot-Host Communication Protocol	20-Oct-16	20-Oct-16	0d
50	**Integration, Compilation, Prototype, Testing & Release**	**21-Oct-16**	**11-Nov-16**	**21d**
51	**GUI Components**	**21-Oct-16**	**21-Oct-16**	**1d**
52	Combine all features of GUI Host Component	21-Oct-16	21-Oct-16	1d
53	Resolve Conflicts	21-Oct-16	21-Oct-16	1d
54	Testing all GUI functionalities	21-Oct-16	21-Oct-16	1d
55	**Robot Components**	**22-Oct-16**	**22-Oct-16**	**1d**
56	Combine all features of Robot Functions	22-Oct-16	22-Oct-16	1d
57	Resolve Conflicts	22-Oct-16	22-Oct-16	1d
58	Test all Robot's Functionalities	22-Oct-16	22-Oct-16	1d
59	**GUI + Robot Communication**	**23-Oct-16**	**23-Oct-16**	**1d**
60	Test Communication between Host and Robot	23-Oct-16	23-Oct-16	1d
61	Test Exception Handling & unexpected scenarios	23-Oct-16	23-Oct-16	1d
62	Debugging & Troubleshooting	23-Oct-16	23-Oct-16	1d
63	**Finalize First Prototype**	**24-Oct-16**	**27-Oct-16**	**4d**
64	Debugging & Troubleshooting	24-Oct-16	27-Oct-16	4d
65	Milestone 5 : First Working Functional Prototype	28-Oct-16	28-Oct-16	0d
66	**Review & Enhance Second Prototype**	**28-Oct-16**	**03-Nov-16**	**7d**
67	Debugging & Troubleshooting	28-Oct-16	03-Nov-16	7d
68	Milestone 6 : Second Working Improved Prototype	04-Nov-16	04-Nov-16	0d
69	**Final Release version of the software**	**05-Nov-16**	**10-Nov-16**	**6d**
70	Compile, Debug & Troubleshoot	05-Nov-16	10-Nov-16	6d
71	Milestone 7 : Final Release Version of the Software	11-Nov-16	11-Nov-16	0d
72	**Documentations**	**12-Sep-16**	**27-Oct-16**	**46d**
73	**SRS**	**12-Sep-16**	**20-Sep-16**	**9d**
74	Setup SRS Google Doc for Collaboration	12-Sep-16	12-Sep-16	1d
75	Write out Draft & Individual Inputs on SRS	13-Sep-16	16-Sep-16	4d
76	Group Discussion & Review SRS Draft	17-Sep-16	17-Sep-16	1d
77	Consolidate, Finalize First Version of SRS	18-Sep-16	20-Sep-16	3d
78	**SPMP**	**26-Sep-16**	**04-Oct-16**	**9d**
79	Setup SPMP Google Doc for Collaboration	26-Sep-16	26-Sep-16	1d
80	Write out Draft & Individual Inputs on SPMP	27-Sep-16	30-Sep-16	4d
81	Group Discussion & Review SPMP Draft	01-Oct-16	01-Oct-16	1d
82	Consolidate, Finalize First Version of SPMP	02-Oct-16	04-Oct-16	3d
83	**SDD**	**10-Oct-16**	**18-Oct-16**	**9d**
84	Setup SDD Google Doc for Collaboration	10-Oct-16	10-Oct-16	1d
85	Write out Draft & Individual Inputs on SPMP	11-Oct-16	14-Oct-16	4d
86	Group Discussion & Review SDD Draft	15-Oct-16	15-Oct-16	1d
87	Consolidate, Finalize First Version of SDD	16-Oct-16	18-Oct-16	3d
88	**User Manual**	**16-Oct-16**	**27-Oct-16**	**12d**
89	Setup User Manual Google Doc for Collaboration	16-Oct-16	16-Oct-16	1d
90	Individual Contributions & Idea Draft Input	17-Oct-16	21-Oct-16	5d
91	Group Discussion & Review inputs from first Draft	22-Oct-16	22-Oct-16	1d
92	Consolidate, Finalize First Version of User Manual	23-Oct-16	27-Oct-16	5d
93				



7.4 Resource Allocation

Tasks will be allocated to our team members in the following way:

- Control Panel Program: Akhil, Eugene, Libin, Soe
- Robot Program: Daniel, Tze Hwa, Oscar
- Integration: Akhil, Daniel, Eugene, Libin, Oscar, Soe, Tze Hwa
- Documentation: Akhil, Daniel, Eugene, Libin, Oscar, Soe, Tze Hwa

8 Supporting plans

8.1 Configuration management plan

8.1.1 Repository folder structure

The root level of the repository contains the following folders:

- /docs/
  This folder stores all the important documents of the project, which includes the SRS, SPMP, SDD, minutes and agendas of meetings, etc

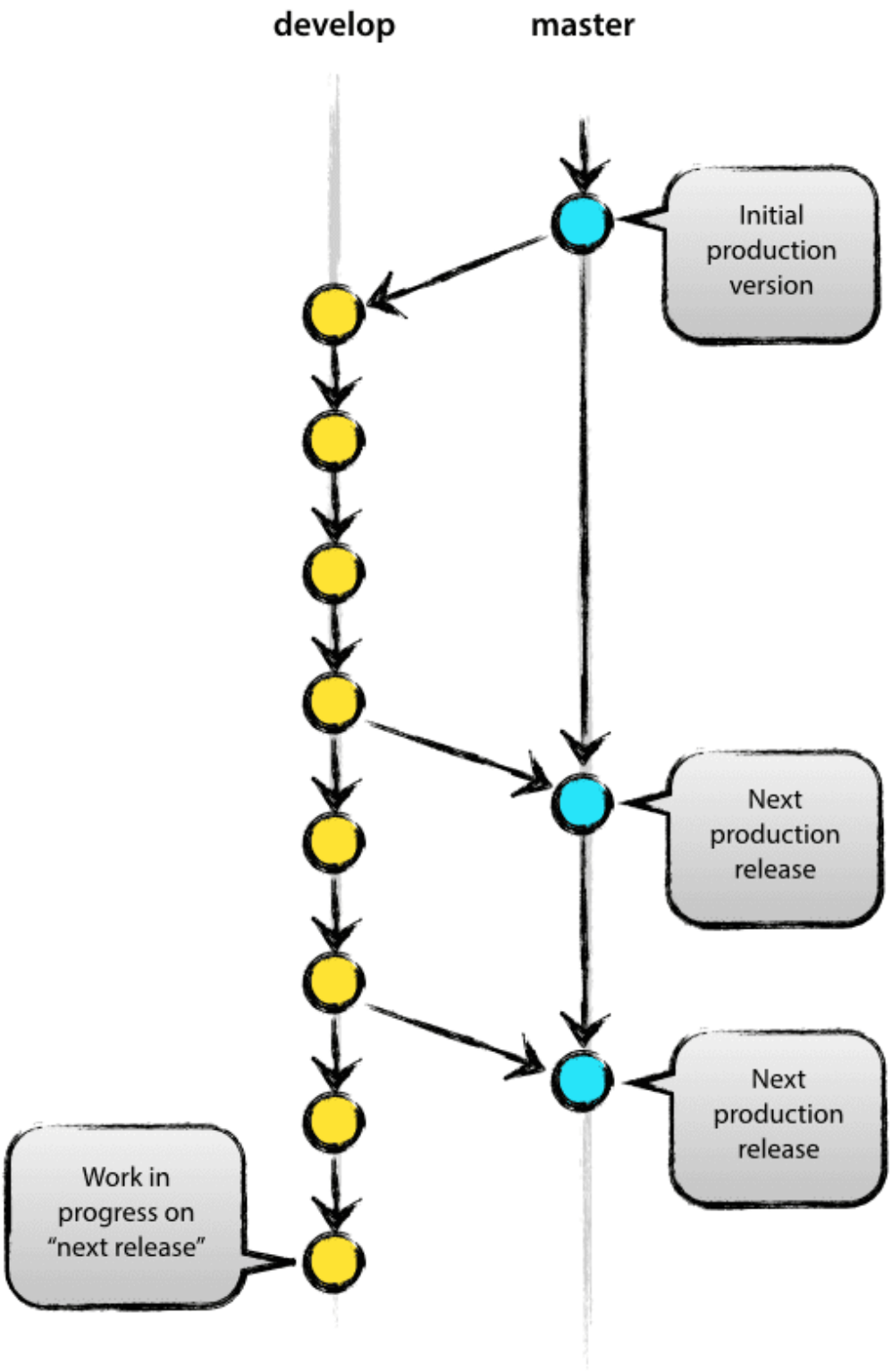
- /src/
  This folder contains the source codes of the project. It is separated into 2 folders, namely, control panel and robot. As their names suggest, the control panel folder contains source codes for the control panel software while the robot folder contains source codes to be run on the robot.

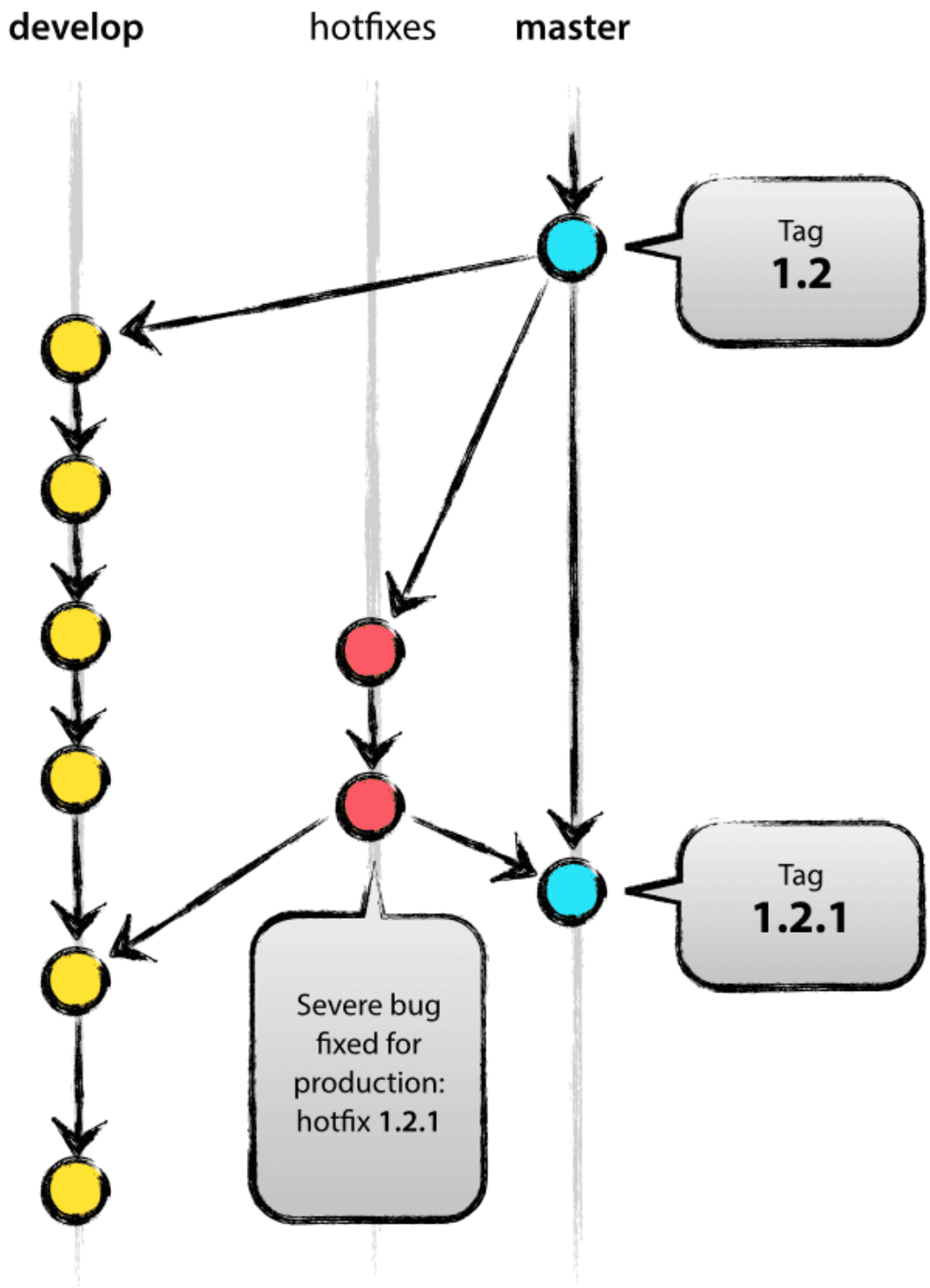
8.1.2 Versions management

Tags will be used to label the stable versions and milestones achieved during the development process. Naming conventions of vX.X.X will be used. There is a changelog.md on the root of the repository to briefly describe the updates of that particular version.

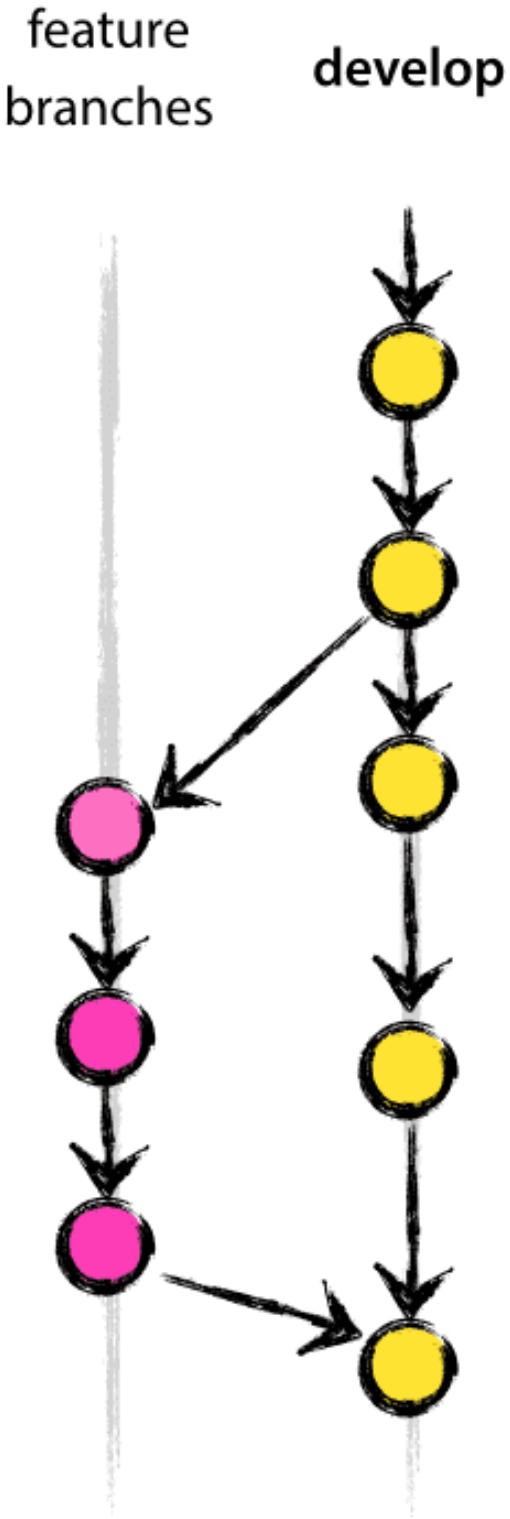
8.1.3 Git branching model

There will be 2 main branches, namely master and develop. The master branch contains the stable versions of the source codes which the tagging will be done on, while the develop branch holds the current state of the development process. When a milestone is achieved or a stable version is ready for testing, the source codes will be merged from the develop into the master.


develop
master
Initial production version
Next production release
Next production release
Work in progress on "next release"


develop
hotfixes
master
Tag
1.2
Tag
1.2.1
Severe bug
fixed for
production:
hotfix 1.2.1

Developers will be adapting the Feature Branching Workflow in the development process. The core concept is to ensure that feature development takes place in a separate branch from the develop branch. This allows multiple developers to work on the project without disturbing the main codebase. When working on a bugfix or new feature, a branch will be created from the develop branch, with naming conventions bugfix/<short description of bug¿ and feature/<user requirement ID or system feature ID >/<short description of the feature<. When the features or bugfixes are ready, they will be merged back into the develop branch.


feature
branches
develop

8.1.4 Build systemsl

Jenkins is used to automate the compiling of the software. It will pull the latest codes from GitHub and build the Ant project based on the configuration and dependencies which are defined in the *build.xml*. Jenkins will automatically execute this task once daily to make sure that there is a backup of the source codes. With Jenkins hosting a copy of the source codes, GitHub won't be a single point of failure. It will also ensure that the software is always in compilable condition.

8.1.5 Project management tools

- **Asana**
  Asana is the main platform that the team is using to track and manage the project. It is used to track the progress of each task and milestones, and to delegate them to the team members accordingly. Asana is also used to hold all the important deadlines in the project. With this tool, all the team members are able to have a clear overview of the status of the project and to be on the same page. Team members are also able to communicate effectively based on the particular task.

- **Slack**
  Slack is the main messaging platform that the team is using for online discussions. It is able to separate topics into different channels for more effective communications.

- **Google Docs**
  Google Docs is used while the team is drafting the formal documents like SRS, SPMP, and SDD. This tool allows real-time collaboration between team members and tickets to be raised to a certain portion of the documents that requires changes or clarifications. This has efficiently improved the productivity of the team in working on the documents.

8.2 Documentation plan

All the formal documents will follow a standard structure and format to ensure consistent look and ease of collaboration when writing the documents. They are to be written in Tex and typeset using LaTex. PDF versions of the documents are also kept for ease of reading, printing and distributing. All documents are committed to the repository for version controlling, so as to be able to revert back to previous versions.

8.2.1 Folder structure

All documents are stored in the /docs/ folder in the repository, separating into multiple different folders:

- */SRS/*
  SRS document will be stored in this folder, in both .tex and .pdf formats
- */SPMP/*
  SPMP document will be stored in this folder, in both .tex and .pdf formats
- */SDD/*
  SDD document will be stored in this folder, in both .tex and .pdf formats
- */agendas/*
  Agendas for all the meetings with client and internal team meetings are stored in this folder. A subfolder will be created for each agenda. Naming convention agenda- <client or team >- <date >. <tex and pdf>will be used.
- */minutes/*
  Minutes from all the meetings with clients and internal team meetings are stored in this folder. A subfolder will be created for each agenda.Naming convention minutes- <client or team>- <date>.<tex and pdf>will be used.

8.2.2 Document structure

All documents first page will be a cover page mentioning the name of the document, project name (Codemmando), names and student IDs of the team members, and the date of the current version of the document. The second page will be a table of contents, and the third with a version history stating the changes between the versions.

8.2.3 Document preparation

Each document will have a member in charge to oversee the whole process of writing of the development. Portions of the document will be delegated to team members, and this lead shall compile the efforts of everyone, and make sure the final document, in Tex/LaTex format, is properly done.

8.2.4 Document review

The same lead in 8.2.3 will also need to vet through the whole document after compiling the work from all the team members. All team members are also required to read through the whole document to feedback and contribute on portions which

they are not assigned to. When the team approves the document, it will then be sent to the client.

8.3 Quality assurance plan

8.3.1 Testing

8.3.1.1 Unit Testing

Every component of the software will be tested individually, to make sure that they are working properly independently.

8.3.1.2 Acceptance Testing

Once a requirement feature is completed, tests will be conducted to make sure that the acceptance criteria are fulfilled.

8.3.1.3 Integration Testing

After each component of the software is completed, tests will be done to make sure that all components are implemented to interface each other correctly.

8.3.2 Document validation

The lead mentioned in 8.2.3 will review the document regularly, to make sure that the software and the documents are consistent with one another.