

DEVELOPMENT PROCESS AND IMPLEMENTATION DOCUMENT

MECHATRONICS 4TB6
GROUP 2

NATHAN FUJIMOTO
PRAKHAR GARG
JOSH GILMOUR
AARON JASS
TYLER JASS
FAUZIA KHANUM
JACK LIU
GAGAN SINGH



MODERN MOBILITY
DEVELOPMENT PROCESS AND IMPLEMENTATION
DOCUMENT

Doc: DOC0003
Rev: -

Table of Contents

Table of Tables	3
Revision History	4
Purpose	5
1 Overall Process Workflow	6
2 How Steps are Completed	8
2.1 Tools and Versions	9
2.2 Special Instructions	10
2.3 Standards Followed	10
2.4 Delegation of Steps	11
3 How Version Control is Being Used	12
4 Dealing with Changes to Artifacts	13
4.1 Bug Tracking/Change Request Tool	13
4.2 Documentation of Change Requests/Bugs	13
4.3 Classification of Changes	13
4.4 Disposition of Changes	13



MODERN MOBILITY
DEVELOPMENT PROCESS AND IMPLEMENTATION
DOCUMENT

Doc: DOC0003
Rev: -

Table of Tables

Table 1 - Overall Process Workflow	6
Table 2 - Tools and Versions	9
Table 3 - Delegation of Steps	11
Table 4 - Git Organization	12

Revision History

Rev	Author(s)	Description of Change	Peer Reviewed	Date
-	T.Jass A.Jass N.Fujimoto J.Gilmour P.Garg G.Singh F.Khanum	Original Document	T.Jass A.Jass N.Fujimoto J.Gilmour P.Garg G.Singh F.Khanum	21-Nov-2017



MODERN MOBILITY
DEVELOPMENT PROCESS AND IMPLEMENTATION
DOCUMENT

Doc: DOC0003
Rev: -

Purpose

The purpose of this document is described the development process and implementation at a high level. Ideally what is described here will indicate that Modern Mobility is working effectively to produce a system of good quality and be able to meet deadlines with reasonable effort.

1 Overall Process Workflow

Table 1 - Overall Process Workflow

Step Number	Description	Input	Output	Acceptance Criteria
1	Draft Designs			
	a) Draft Motor Drive Design <ul style="list-style-type: none">Motor SystemBraking System	Goals and Requirements Tools	2a	The design step is completed when there are clear and descriptive instructions/designs available to complete the implementation. All members have agreed upon and are up to date on the designs.
	b) Draft Navigation System Architecture	Goals and Requirements	2b	
	c) Draft User Application Interface Design	Goals and Requirements	2c	
	d) Draft System Integration Design	Goals and Requirements	2a, 2b, 2c	
2	Implementation of Designs			
	a) Implement and Test Motor Drive System <ul style="list-style-type: none">Motor SystemBraking System	1a, 1d	3a	Motors can engage and disengage. Braking works when powered & unpowered. Follows the design.
	b) Implement and Test Navigation System	1b, 1d	3b	Successfully move from start location to the desired location.
	c) Implement and Test User Application Interface	1c, 1d	3c	Clear and timely communication from the User interface to the mechanical system.

3	Final Designs (Corrected Draft Designs)			
	a) Final Motor Drive Design <ul style="list-style-type: none">Motor SystemBraking System	2a	4a	Design’s corrected to include any item changes made during first implementation. Final design steps are clearly indicated and known by all team members
	b) Final Navigation System Architecture	2b	4b	
	c) Final User Application Interface Design	2c	4c	
	d) Final System Integration Design	2a,2b,2c	4a, 4b, 4c, 4d	
4	Final Implementation			
	a) Implement and Test Final Motor Drive Design <ul style="list-style-type: none">Motor SystemBraking System	3a, 3d	N/A	We have a final working product that follows our noted requirements and goals. All systems work together seamlessly with no issues. All systems are documented correctly with detailed bug and issue tracking for future references.
	b) Implement and Test Final Navigation System Architecture	3b, 3d	N/A	
	c) Implement and Test Final User Application Interface Design	3c, 3d	N/A	
	d) Implement and Test Final System	3d	N/A	

2 How Steps are Completed

When considering the overall workflow, this section describes the process of each step while considering tools are unique to each step. Please refer to **Table 2** for a list and descriptions of all the required tools.

Draft Designs

The draft designs are all the initial engineering designs and system architecture models for the motor and braking systems, navigation library and the user interface application. The file management tools involved would be Google Drive, Google Docs and Git. For drafting designs tools that would be involved include Autodesk Inventor and Draw.io (including any handwritten designs).

Implementation of Designs

The implementation of designs is the prototyping and coding of all the necessary project components of the SmartWalker, and integrating each in order to create a fully functional implementation. The tools software tools involved are Git, Ubuntu MATE, ROS and Android Studio. All the mechanical and electrical tools will be used for the implementation of designs.

Final Designs (Corrected Draft Designs)

The final designs are the corrected versions of the draft designs after evaluating the acceptance criteria from the implementation of the draft designs. It involves updating the designs for the subsystems, and involves the same tools as the draft designs stage.

Final Implementation

The final implementation is the final prototyping and coding necessary to create and implement the final designs for the SmartWalker. It involves creating new mechanical and electrical components if necessary and re-coding any applications if necessary. It involves the same tools as the implementation of draft designs stage.

2.1 Tools and Versions

Table 2 - Tools and Versions

Tool	Version	Purpose/Function
Software Tools:		
Google Drive	N/A	Storage accessible by entire group for deliverable documents.
Google Docs	N/A	Real-time collaboration.
Draw.io	N/A	Online web tool for model mapping and flow chart development.
Git	2.15	Software Version control, deliverable document, and internal document storage.
Ubuntu MATE	3.0	OS being used on our systems controller (Raspberry Pi 3B).
ROS (Robotics Operating System)	Kinetic	Robotics operating system library being used to develop project's autonomous navigation capabilities.
Android Studio	3.0	Environment for Android App Development.
Autodesk Inventor	2017	Model all physical Robotics systems.
Mechanical Tools:		
3D Printer	N/A	Prototype any mechanical parts for motor and braking systems.
Hand Drill	N/A	Used for drilling holes
Saw	N/A	Used for cutting material
CNC Mill	N/A	Produce finished mechanical parts for motor and braking systems.
Electrical Tools:		
Soldering Iron	N/A	Soldering wires/ circuit boards



MODERN MOBILITY DEVELOPMENT PROCESS AND IMPLEMENTATION DOCUMENT

Doc: DOC0003
Rev: -

Multimeter	N/A	Measure voltage and current
Power Supply	N/A	Powering prototype circuits/Motors
Oscilloscope	N/A	Measuring electrical signals

2.2 Special Instructions

All document resources will be backed up to the corresponding version control systems.

2.3 Standards Followed

- All code should be commented to allow for quick and painless troubleshooting, our motto is ***“Comment as if the next person who reads your code is a serial killer”***
- When coding, no function should have more lines than what can be seen on your computer screen at one time
- No software or hardware tested by the person who implemented it can be deemed “Working” (basically do not test your own work, get someone else to test it)
- Metric Units will be used for all modeled components

2.4 Delegation of Steps

Legend

1 - Primary Responsibility

2 - Secondary Responsibility

3 - Tertiary Responsibility

Highlight - Persons Responsible for section of project

Table 3 - Delegation of Steps

	Josh	Aaron	Jack	Nathan	Fauzia	Prakhar	Tyler	Gagan
Phone application development		1/2	1		3	2	3	
Autonomous navigation software	1		2	2				
Low-level software	2	3		1	1		1	
System integration	3		3		2	1*		3
Wheels and brakes modifications		1/2		3		3	2	1
Handle Bar (brake interference)						1		2

3 How Version Control is Being Used

Version control will be used to manage all source code, and as well as store documents. The source code will be organized into different directories based on the application. Documents including will be both deliverables and internal documents used for team knowledge sharing.

Our Git is organized into folders that relate to different aspects of the project. These folders are described in the table below.

Table 4 - Git Organization

Folder	Purpose
CADs	Store the CAD files created of the walker and other components
Low level	Store documentation and files related to the software built for the Arduino and Raspberry Pi
Phone application code	Store documentation and files related to the phone application that is used to control and setup the walker.
ROS	Store documentation and files related to the ROS software on the walker

4 Dealing with Changes to Artifacts

4.1 Bug Tracking/Change Request Tool

If we find that we are running into a lot of bugs and are spending quite a bit of time resolving them we will look into using Bugzilla. It is free for any number of users, compatible with Linux, MacOS, and Windows, and contain the following:

- Time tracking
- Reporting
- Duplicate bug detection
- Customization: custom fields to suit application

4.2 Documentation of Change Requests/Bugs

All of documentation is first drafted in Google Docs. Google Docs allows real time collaboration, commenting, and revision control. We can use this to document and track changes during the creation and drafting process. Once complete, we can export to word and finalize the document.

4.3 Classification of Changes

Changes will have different statuses such as:

- In queue (Waiting to be dealt with)
- Working (Currently being investigated)
- Resolved (Issue/Bug successfully dealt with)
- Unresolved (Unable to solve this issue/bug)

They should also have different levels of urgency based on how core the issue/bug is to the particular systems functionality.

4.4 Disposition of Changes

Change statuses will be verified and changed to different status by the appropriate party. For example, anyone can raise an issue and add them to “In Queue”. Depending on the given issue, a particular team may be responsible to fix that issue. Only that team can change that issue’s status to “Working”. Since it would bad practice to have the same person who fixed the issue, also verify the issue, someone else would be responsible for moving the issue to “Resolved”. If the person fixing the issue cannot fix it, they can move it to “Unresolved” so someone else can try it.