Carnegie Mellon University

Robotics Capstone Project

System Design and Development Document

Friction Force Explorers:

Don Zheng
Neil Jassal
Yichu Jin
Rachel Holladay

supervised by
Dr. Cameron Riviere

Version 1.0
February 14, 2017

Contents

List of Figures

1 Build Progress

We chronicle our build progress thus far, splitting up efforts into electromechanical updates and software updates. This corresponds to the bases of our tree, excluding the integration branch which happens as a final step.

1.1 Electromechnical Updates

As shown below in Fig.1.1, we have made three changes to the electromechanical system since the critical design review. 1) Chassis material is changed from acrylic to plywood. Since we plan to use laser cutting as main fabrication method, fabricating wood would generate less hazardous fume then fabricating acrylic does. Also, wood has higher strength to density ratio, which could make the robots more lightweight. 2) Raspberry Pi is now located above the chassis, instead of below it, so that the robot has space to stack multiple motor HATs. 3) Painting mechanism is redesigned to reduce mechanical complexity.

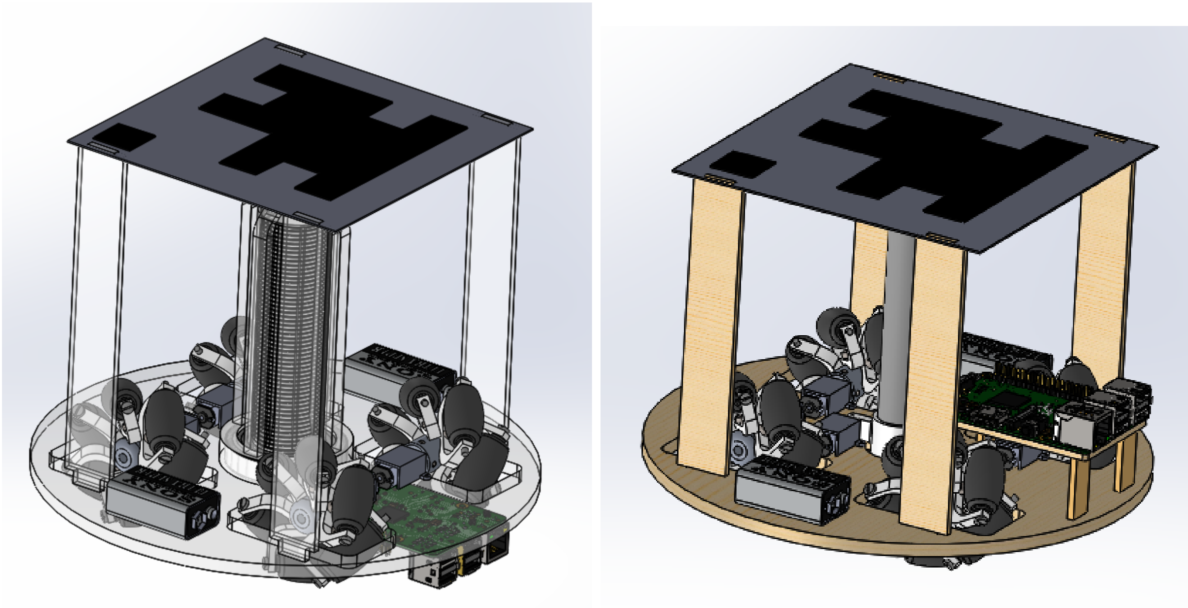


Figure 1: Full System Comparison, old (left) versus new (right)

Fig.1.1 and Fig.1.1 compare the improved painting mechanism to the old design. Instead of a screw type of actuation, the robot now uses a lever mechanism to press the chalk marker on drawing surfaces. The driving motor is fixed to the chassis via an off-the-shelf motor case. This motor then rotates a 3D printed marker holder with the chalk marker installed. By control the rotation direction and voltage input of the motor, the robot can either lift up or down the marker. This design change reduces painting mechanism's number of components from 5 to 3 and dramatically reduced the amount of material that needs to be 3D printed, which reduce fabrication cost and fabrication time. To secure the chalk marker better, we may add internal ribs in the marker holder or design it into a snap-fit component. This design decision will be made when we receive the ordered chalk markers. Almost electromechanical components are ordered. We expect to start fabrication later this week.

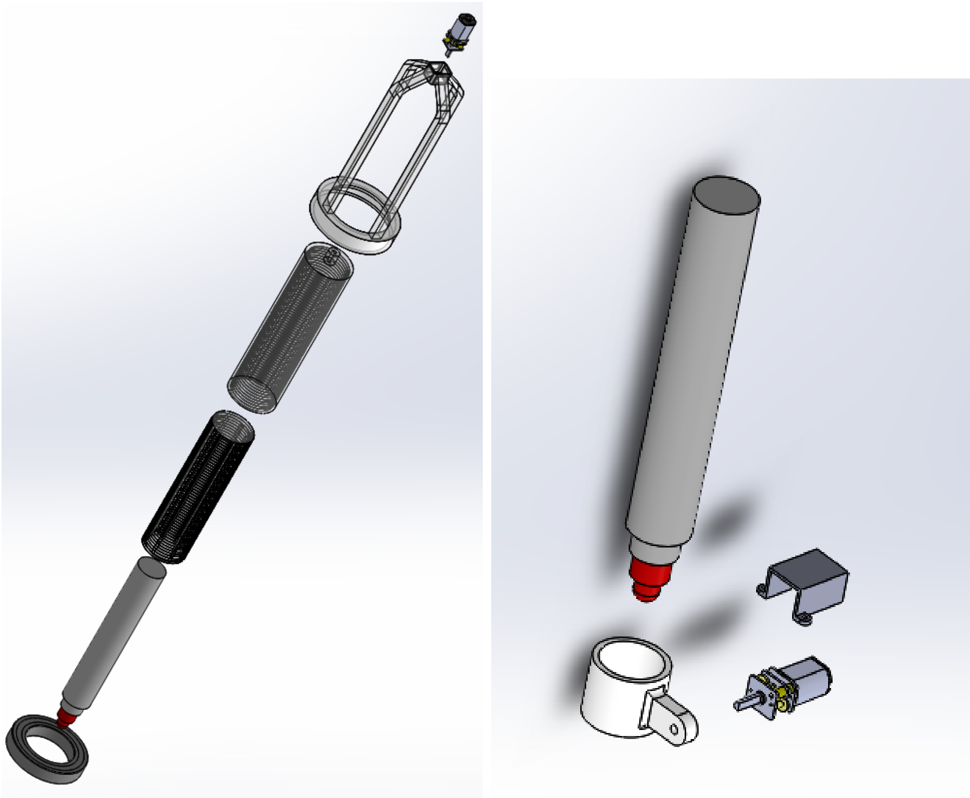


Figure 2: Painting Mechanism Changes, old (left) versus new (right)

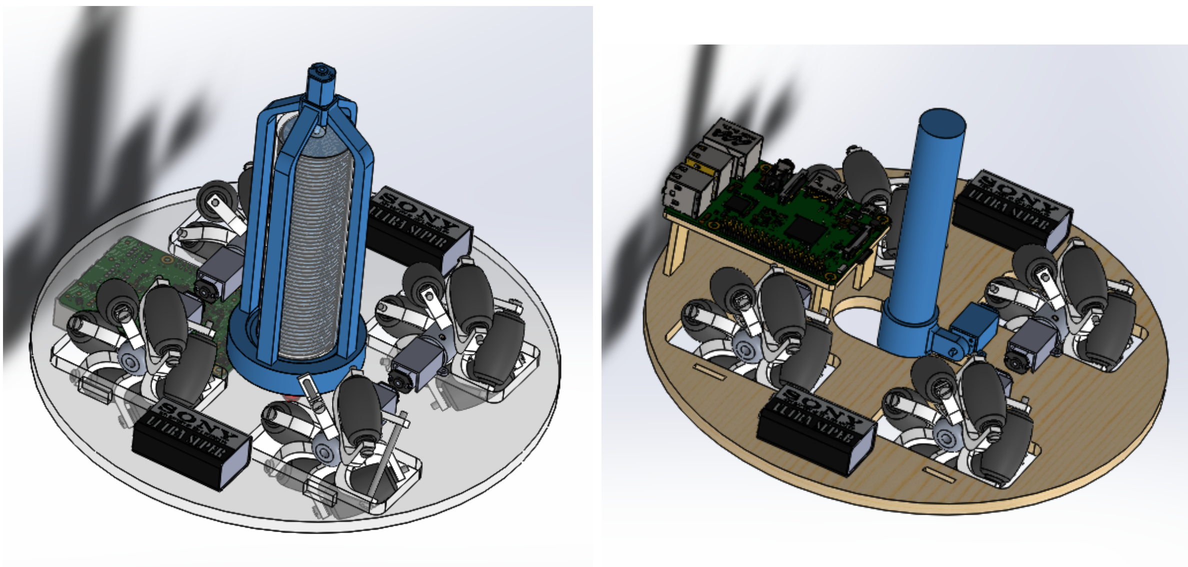


Figure 3: A comparison of the old (left) versus new (right) chassis with the painting mechanism exposed

1.2 Software Update

In our software development process, our first step was to design a software architecture. Like most robotic systems, our robots involve several interlocking pieces of software that need to be well organized in order to function. We took our psuedo-code developed last semester and converted it into interlocking code skeletons that serves as the groundwork for the rest of our software development. Having determined how to separate the work between modules we can now develop each one independently.

We have begun by focusing in on the communication and SDP (scheduling, distribution and planning) modules.

The communication module has taken the roles of establishing connections, sending and receiving messages, and generating messages. The subsystem will keep track of TCP connections to each robot, and monitor them for any changes that could signal loss of connection. It will also manage receiving and parsing data from the onboard controllers. The onboard controllers send any motor encoder and error data, which the communication subsystem parses into data usable by other subsystems. Motor encoder data is passed into the localization subsystem, and error data is processed to determine if the system should be paused or shut down. Finally, the communication subsystem will take data from the locomotion and writing modules, and parse them into proto3 messages to be sent to the onboard controllers via TCP.

In our SDP module, we have edited our work distribution method to take advantage of a more greedy approach. We define the cost of the work for each robot, with the goal of keeping these costs as equal as possible. When iterating through our set of lines, we add the next new line to the robot with the smaller total cost work thus far. We then update that robot's work to account for the cost to get from it's current position to the start of the line and the cost of drawing the line. As we progress through, we eagerly reorder and reorient the lines to optimize cost. This is in contrast to our previous method, which reodered only at the end and separated the lines greedily with respect to spatial dimension.

In developing the SDP module, we have begun a framework for the UI module, adding the capability to read in assignments. We expect to continue to develop much of the UI module in tandem with the other pieces, as UI visualization serves as a vital tool in development.

2 Project Management

We present our project management plan for the project, include a Work Breakdown Schedule of our tasks and Gantt chart or scheduling.

2.1 Work Breakdown Schedule

In this section, we present the Work Breakdown Schedule for the project.

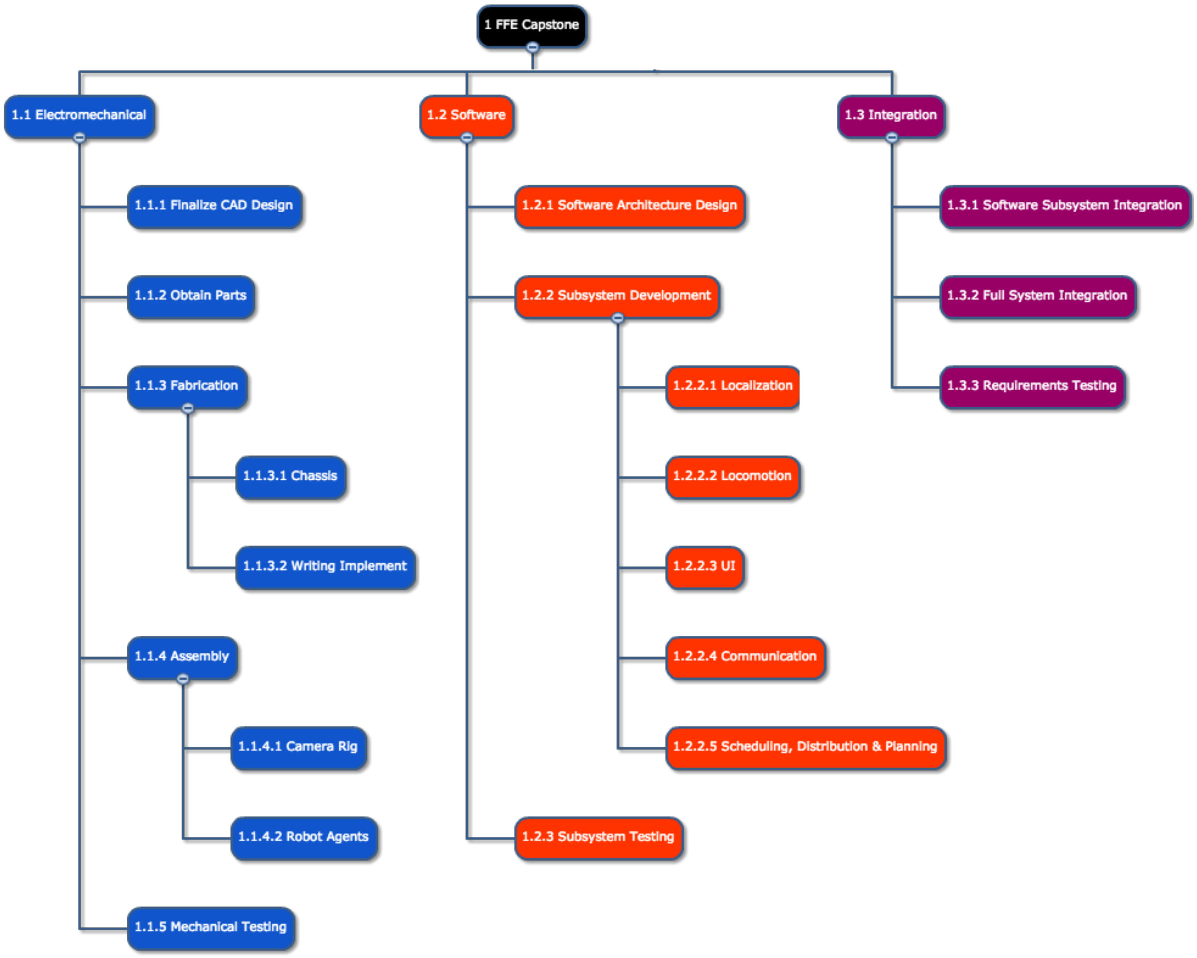


Figure 4: Full WBS for the project

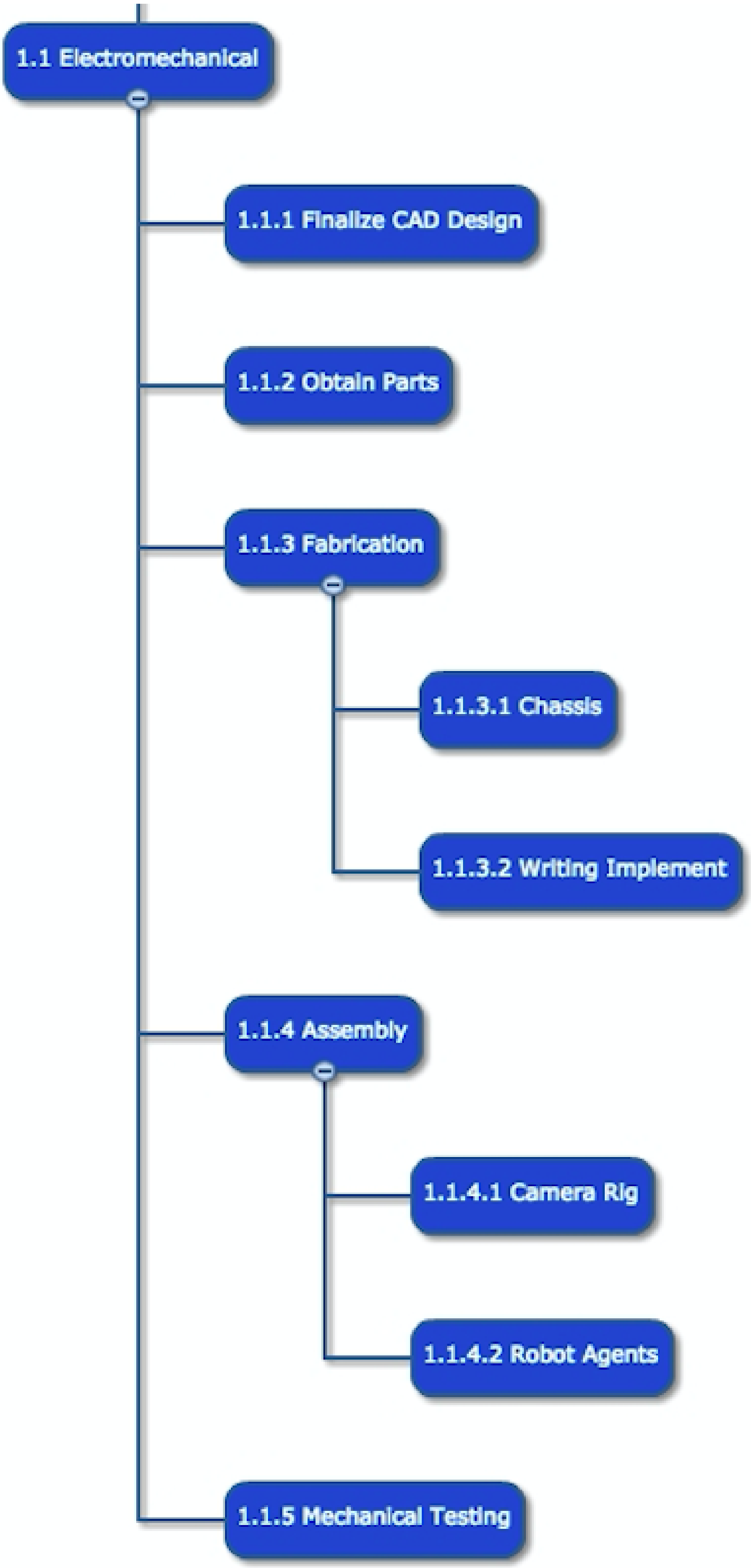


Figure 5: Electromechanical WBS section

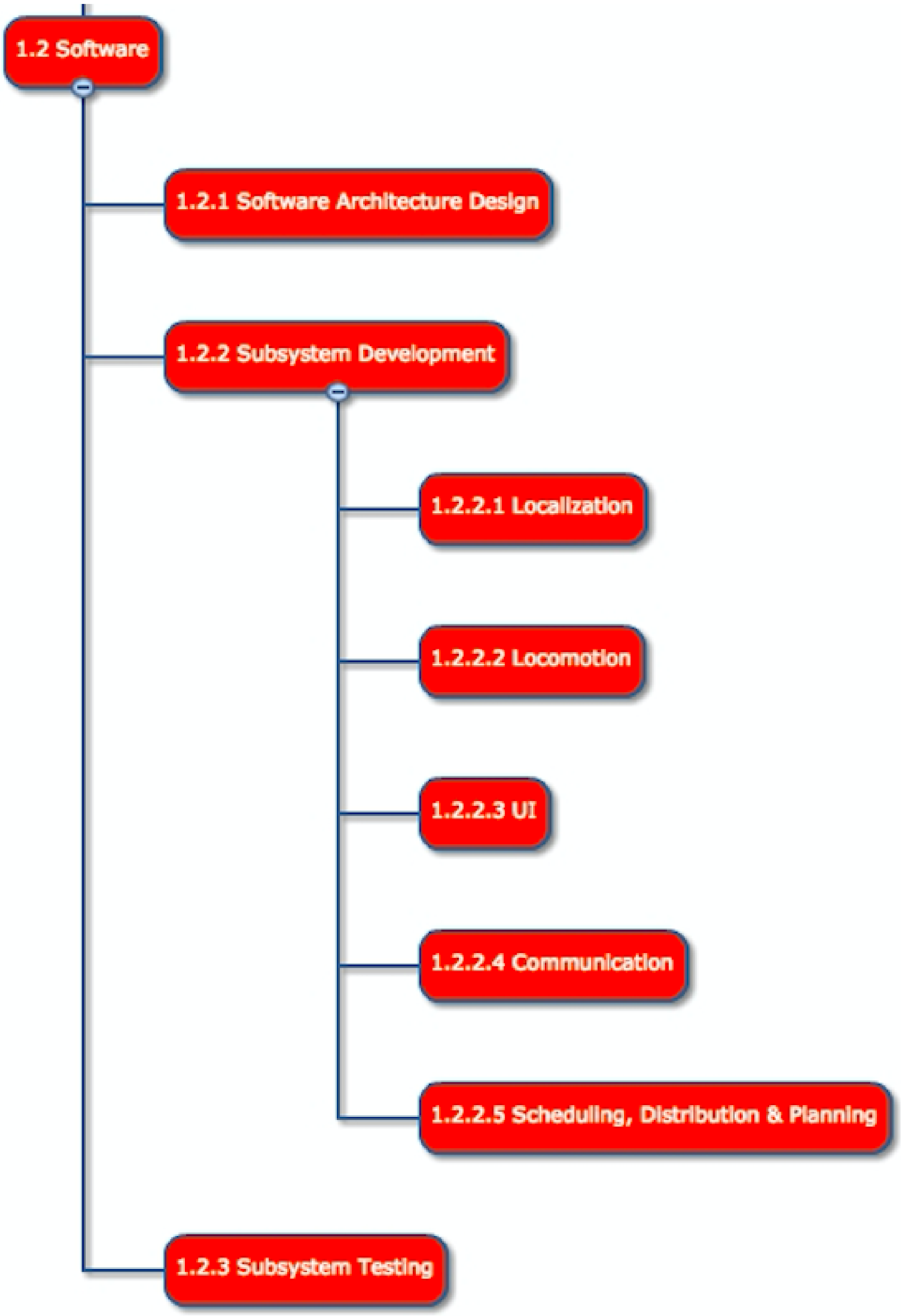


Figure 6: Software WBS section

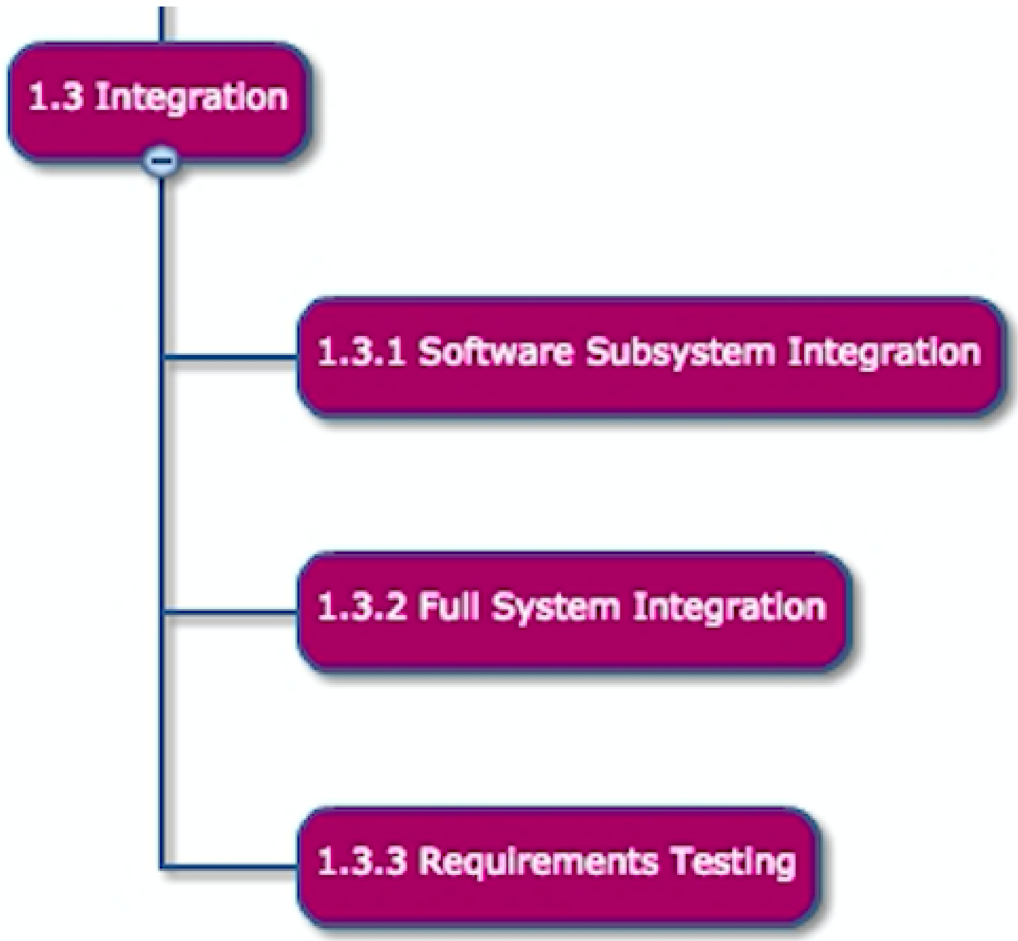


Figure 7: Integration WBS section

The WBS dictionary entries include more information on each of the work elements of the project.

WBS#:	1.1.3.1	Task:	Fabricate Chassis
Est. Effort (hrs):	4	Owner:	Eric
Resources:	CAD designs, MechE shop	Work products:	Chassis components
Description:	Use the Mechanical Engineering machine shop to fabricate components necessary to build the chassis		
Input:	CAD designs, parts		
Dependencies:	Obtain parts		
Risks:	Machine shop is not available, injury from operating machines		

WBS#:	1.1.3.2	Task:	Fabricate Writing Tool
Est. Effort (hrs):	4	Owner:	Eric
Resources:	CAD designs, MechE shop	Work products:	Writing tool components
Description:	Use the Mechanical Engineering machine shop to fabricate components necessary to build the writing implement		
Input:	CAD designs, parts		
Dependencies:	Obtain parts		
Risks:	Machine shop is not available, injury from operating machines		

WBS#:	1.1.4.1	Task:	Assemble Camera Rig
Est. Effort (hrs):	3	Owner:	Don
Resources:	Scrap wood	Work products:	Camera rig
Description:	Build the rig used to hold the camera for the vision system above the drawing space		
Input:	Measurements from demo space		
Dependencies:	Confirmation of demo space location		
Risks:	No extra wood is available, demo space does not have adequate room for the camera rig		

WBS#:	1.1.4.2	Task:	Assemble Robot Agents
Est. Effort (hrs):	5	Owner:	Eric
Resources:	Tools, fasteners	Work products:	Two robot agents
Description:	Use fabricated components to build the two robot agents in the system		
Input:	Fabricated components		
Dependencies:	Fabricate chassis and fabricate writing tool		
Risks:	Parts are broken during assembly, extra parts or fasteners are needed		

WBS#:	1.1.5	Task:	Mechanical Testing
Est. Effort (hrs):	3	Owner:	All
Resources:	Tools, fasteners	Work products:	Two robot agents
Description:	Perform mechanical testing on the robots in accordance with our testing guidelines		
Input:	Mechanically complete robots		
Dependencies:	Assemble robot agents		
Risks:	Tests are failed, and significant time or extra resources are needed to correct the tests		

WBS#:	1.2.1	Task:	Software Arch. Design
Est. Effort (hrs):	3	Owner:	All
Resources:	None	Work products:	Function headers
Description:	Design function I/O, and create function headers for all files we will use in the robot		
Input:	Software flowchart, decisions on software libraries		
Dependencies:	Complete design review		
Risks:	Selected software libraries have compatability issues		

WBS#:	1.2.2.1	Task:	Localization Subsystem
Est. Effort (hrs):	6	Owner:	Neil
Resources:	AprilTag library	Work products:	Working localization
Description:	Fill in the function headers for the localization system to develop an end-to-end localization solution for the robots		
Input:	Function headers and design for localization system		
Dependencies:	Software architecture design		
Risks:	Localization system or library is unable to perform to expectations		

WBS#:	1.2.1	Task:	Locomotion Subsystem
Est. Effort (hrs):	5	Owner:	Don
Resources:	Adafruit Motor controller library	Work products:	Control system for motors, robust motion model
Description:	Create a complete set of functions that can be used to direct the robots around the workspace		
Input:	Software flowchart, decisions on software libraries		
Dependencies:	Software architecture design		
Risks:	Interfacing issues with motors, damanged electronics hardware, unreliable motion models		

WBS#:	1.2.1	Task:	User Interface Subsystem
Est. Effort (hrs):	4	Owner:	Rachel
Resources:	Various UI libraries	Work products:	User interface including calls to other subsystems
Description:	Create a visually appealing and intuitive user interface for the robot system		
Input:	Software flowchart, decisions on software libraries		
Dependencies:	Software architecture design		
Risks:	Libraries are not available		

WBS#:	1.2.2.4	Task:	Comunication
Est. Effort (hrs):	8	Owner:	Neil
Resources:	Wireless comm. libraries	Work products:	Functions for sending info. back and forth from robots
Description:	Create a reliable communication system between the robots and the central data processing unit		
Input:	Software flowchart, decisions on software libraries		
Dependencies:	Software architecture design		
Risks:	Wireless hardware is unreliable or interfaces poorly with other software or hardware		

WBS#:	1.2.2.5	Task:	SDP Subsystem
Est. Effort (hrs):	15	Owner:	Rachel
Resources:	SDP research, implementations	Work products:	Complete SDP functions
Description:	Create a flexible scheduling, distribution, and planning subsystem that efficiently assigns work to robots		
Input:	Software flowchart, decisions on software libraries		
Dependencies:	Software architecture design		
Risks:	SDP algorithms are not efficient enough to meet requirements		

WBS#:	1.2.3	Task:	Subsystem Testing
Est. Effort (hrs):	4	Owner:	All
Resources:	Software subsystems	Work products:	Complete software subsystems
Description:	Test all software subsystems to ensure that they give the expected output when provided with testing inputs		
Input:	Completed software subsystems		
Dependencies:	All software subsystem tasks		
Risks:	Software subsystems were implemented incorrectly and do not perform to expectations		

WBS#:	1.3.1	Task:	Software Integration
Est. Effort (hrs):	3	Owner:	All
Resources:	Software subsystems	Work products:	Complete software pipeline
Description:	Test integration of all software components by creating an end to end pipeline consisting of all software subsystems		
Input:	Completed and individually verified software subsystems		
Dependencies:	Subsystem testing		
Risks:	Subsystems cannot integrate with each other		

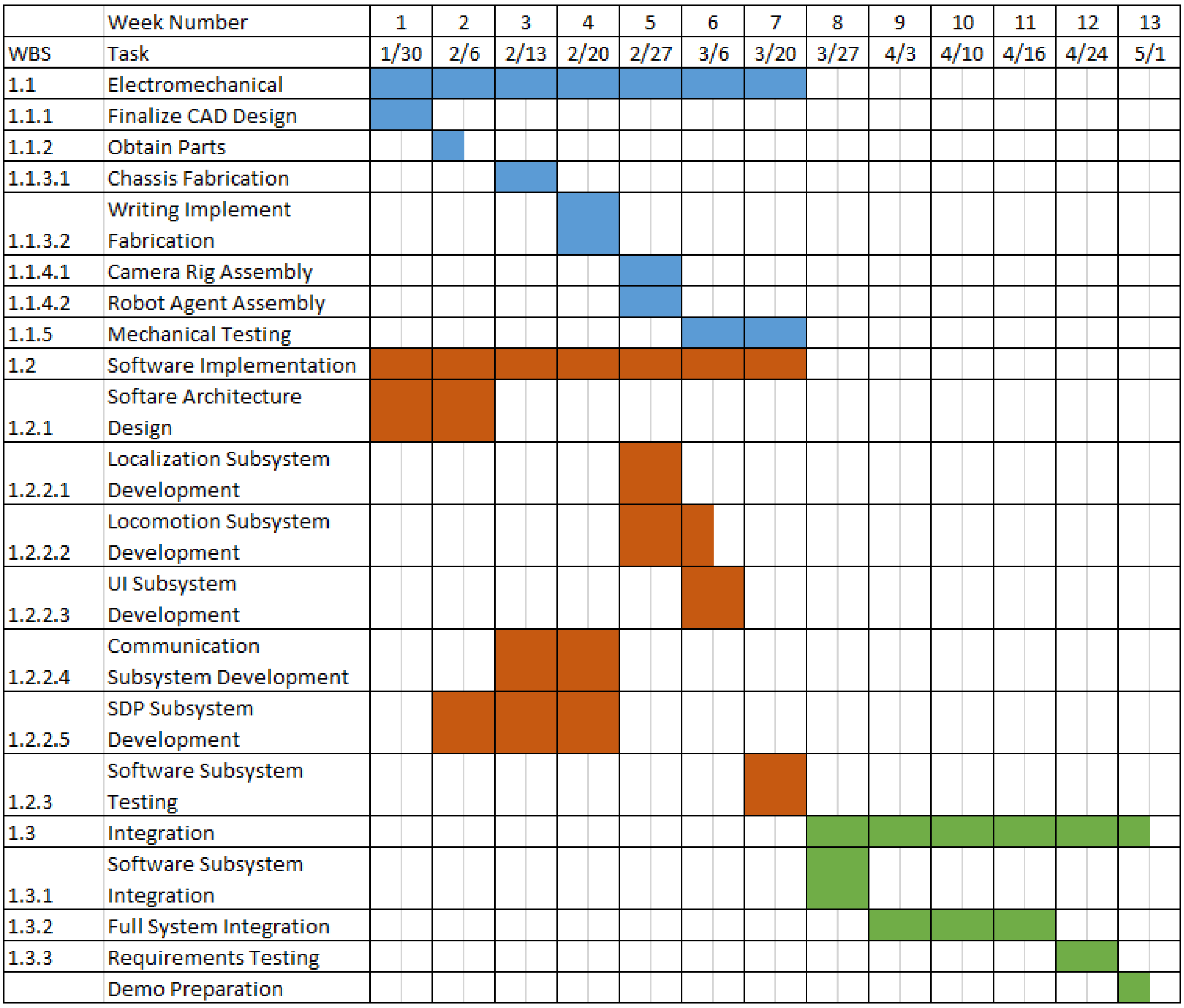
WBS#:	1.3.2	Task:	Full System Integration
Est. Effort (hrs):	3	Owner:	All
Resources:	S.W. and H.W. subsystems	Work products:	Working robot system
Description:	Complete integration of software components with hardware components		
Input:	Completed and individually verified software and hardware subsystems		
Dependencies:	Software integration		
Risks:	Software and hardware cannot interface with one another, models do not work in in practice		

WBS#:	1.3.3	Task:	Requirements Testing
Est. Effort (hrs):	5	Owner:	All
Resources:	Working robot	Work products:	Complete, working robot system
Description:	Verify the reliability and effectiveness of the robot by conducting our full testing suite		
Input:	Unverified but working robot system		
Dependencies:	Full system integration		
Risks:	Robot fails tests, need to rework some subsystems		

2.2 Schedule

Scheduling for the semester has been split into three main sections, as outlined in the wbs (Sec. 2.1). These sections were determined into electromechanical, software, and integration. Both electromechanical and software development can be implemented and built simultaneously, with integration following once both pieces are complete. By developing hardware and software at the same time, the team can make adjustments to both systems based on changes to the other. We planned the schedule to allow the last month for integration and testing, which will help us to ensure the full system works for the final demo.

We chose to represent the schedule as a Google Calendar, which allows us to integrate it with our schedules for other classes, as well as giving us convenient access and use. This can also be represented as a Gantt Chart, as in Fig.8.



	Week Number	1	2	3	4	5	6	7	8	9	10	11	12	13
WBS	Task	1/30	2/6	2/13	2/20	2/27	3/6	3/20	3/27	4/3	4/10	4/16	4/24	5/1
1.1	Electromechanical													
1.1.1	Finalize CAD Design													
1.1.2	Obtain Parts													
1.1.3.1	Chassis Fabrication													
1.1.3.2	Writing Implement Fabrication													
1.1.4.1	Camera Rig Assembly													
1.1.4.2	Robot Agent Assembly													
1.1.5	Mechanical Testing													
1.2	Software Implementation													
1.2.1	Softare Architecture Design													
1.2.2.1	Localization Subsystem Development													
1.2.2.2	Locomotion Subsystem Development													
1.2.2.3	UI Subsystem Development													
1.2.2.4	Communication Subsystem Development													
1.2.2.5	SDP Subsystem Development													
1.2.3	Software Subsystem Testing													
1.3	Integration													
1.3.1	Software Subsystem Integration													
1.3.2	Full System Integration													
1.3.3	Requirements Testing													
	Demo Preparation													

Figure 8: Gantt Chart of the semester schedule

3 Work Division

We outline the individual contributions of each team member with respect to technical and nontechnical work.

3.1 Technical Division

NJ: Fill in.

3.2 Non-Technical Division

NJ: Fill in.