Carnegie Mellon University

Robotics Capstone Project

System Readiness Review

Friction Force Explorers:

Don Zheng
Neil Jassal
Yichu Jin
Rachel Holladay

supervised by
Dr. Cameron Riviere

Version 1.0
April 11, 2017

Contents

1 Build Progress **2**
1.1 Electromechanical Updates 2
1.1.1 Electrical Updates 2
1.1.2 Mechanical Updates 2
1.2 Software Update 2
1.2.1 Locomotion 2
1.2.2 Localization 2
1.2.3 Scheduling, Distribution and Planning (SDP) 3
1.2.4 User Interface 3
1.3 Integration Updates 3

2 Project Management **5**
2.1 Work Breakdown Schedule 5
2.2 Schedule 18

3 Requirements Tracking **18**
3.1 Objectives Tree 19
3.2 Requirements Traceability Matrix 27

4 Risk Management **28**

5 Testing and Evaluation Plan **31**

List of Figures

1 Annotated AprilTag testing setup 3
2 Drawing Interface for Inputting Requests 3
3 Full WBS for the project 5
4 Electromechanical WBS section 6
5 Software WBS section 7
6 Integration WBS section 8
7 Schedule via Gantt Chart 18
8 Full Objectives Tree 20
9 Objectives Tree: Is Safe Branch 21
10 Objectives Tree: Is Portable Branch 22
11 Objectives Tree: Drawing Tool is Easy to Operate Branch 23
12 Objectives Tree; Is Mobile Branch 24
13 Objectives Tree: User-Friendly Branch 25
14 Objectives Tree: Performance Guarantees Branch 26
15 Requirements Traceability Matrix 27
16 Risk 1: Defective Parts 28
17 Risk 2: Unavailable Member 28
18 Risk 3: Breaking Parts 29
19 Risk 4: Mecanum Drive Too Unstable 29
20 Risk 5: Localization not precise enough 30
21 Risk 6: Unexpected Budget Overruns 30

1 Build Progress

This section details system development and progress made since the last milestone presentation. Progress is split into three major sections: electromechanical, software, and integration. Electromechanical updates detail chassis build progress, as well as setup of the electronics to drive the motors for locomotion and using the writing implement. Software updates describe progress towards subsystem completion. Integration details updates made with regard to integrating the electromechanical and software systems, and testing functionality.

1.1 Electromechanical Updates

1.1.1 Electrical Updates

Electrical progress is separated into two main parts: connecting the motors to the motor controller and Raspberry Pi, and powering the motors and Raspberry Pi controller. Each of the motors was wired up to an Adafruit Motor Controller for easy use, which mounts as a shield on top of the Pi. For the time being, we have chosen not to connect the motor encoders. Given that we are using localization for motions, and vector directions for the robots are updated at every control loop iteration, we have hypothesized that localization will be enough to ensure accurate motion. The robots will never be moving more than a few inches without updated directional commands, making fine tuned encoder-based motor control unnecessary.

The battery packs for the Raspberry Pi and motors were directly connected to the respective pieces of hardware. Until the electronics mount is built, we have been placing the battery packs on the robot, or holding them during testing. We plan to attach them to the mount using velcro.

1.1.2 Mechanical Updates

YJ: Talk about webcam jig YJ: Building new robot, motor shafts, etc.

1.2 Software Update

We detail the software progress made across the following subsystems. Most subsystems have reached the point of usability, and at this point most additions enhance ease-of-use and functionality.

1.2.1 Locomotion

The locomotion subsystem has been fully implemented as a part of the onboard controller code. The subsystem is capable of determining motor commands based on a target vector direction to move the robot along the specified vector. The robots have mecanum wheels and can therefore move omnidirectionally. This fact, coupled with the fact that we plan to only have to move along fixed straight-line vectors as specified by the SDP subsystem, means the robots never have to rotate. The goal of locomotion is only to translate along vectors, and never rotate. However, implementation of the mecanum control equations includes the ability to have the robots rotate during operation. The main use of rotation will be to correct any rotational error detected by the localization subsystem.

1.2.2 Localization

The localization subsystem has been completed, and is successfully able to use a combination of the AprilTags library, and Boost Python to transmit position and orientation of each AprilTag back to the controller. The controller then computes an affine warp using the specified corner tags, and warps the coordinates into a fixed dimension space. For example, if the input space is from coordinates (0,0) to (10,10), the controller will warp the space from pixel coordinates to the (0,0), (10,10) frame. While this can potentially warp the image being drawn as it stretches to accomodate a fixed input space, the change in dimensions is small enough not to affect output quality. Orientation of the robots is also computed, and will be sent to the robots to correct any rotational error. Fig.1 shows an annotated test image of the six AprilTags to be used, with their respective labels and tags marked for visualization.

Figure 1: Annotated AprilTag testing setup

1.2.3 Scheduling, Distribution and Planning (SDP)

In advance of SDP integration we formatted our planner to take as input a standarized message type. Additionally, we added flexibility to the planning system by allowing inputs of arbitrary dimension. Given an input of size M by N the planner will re-scale the drawing to match the size of the drawing surface, X by Y. Following these updates the current planner was integrated into main code base. We look forward to testing it and adding on-board collision prevention soon.

1.2.4 User Interface

We completed development of a UI that allows users to draw the lines they would like the robots to complete. The interface is shown in Fig.2. The user drags their mouse to draw a series of lines. The user has the option to clear their current drawing, allowing them to start over. Once the user is finished they can input the filename under which they would like to save the drawing and exit the tool. The tool records the line drawing and saves it to our database. We envision using this tool to create our own drawings and to add an interactive element to our demo.

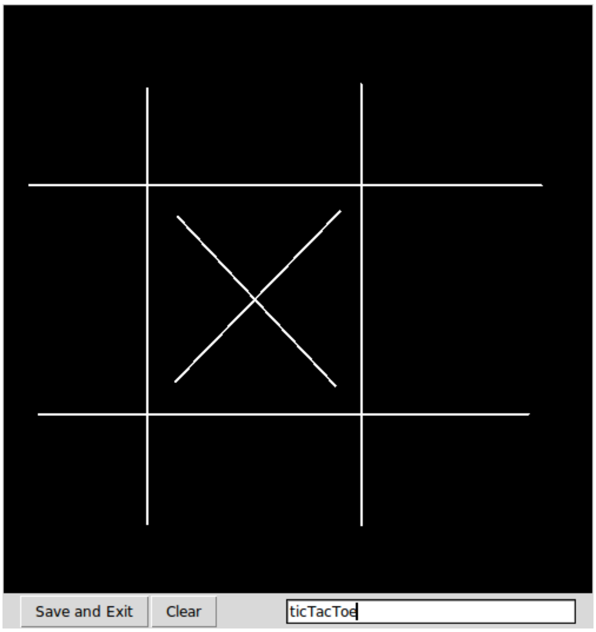

Figure 2: Drawing Interface for Inputting Requests

1.3 Integration Updates

System integration has involved assembling the electronics - including the motors and Raspberry Pi controller, and attaching them correctly to the mechanical system. For now, we were able to combine the locomotion subsystem with the onboard controller to send vector commands to the motors. The robot is able to move omnidirectionally and correct for rotation, however without localization it is impossible to detect rotational error. After running some simple motion tests without the encoder, we found that the robot was well within our positional accuracy requirements, and we do not believe the encoders will be necessary. Once localization is connected to the onboard system, we will be able to confirm that the encoders are not necessary.

Next steps involve integrating the communication and localization systems. Current tests used the onboard system only to run locomotion commands. The first task is to enable offboard communication to send locomotion commands to the robot. Once consistent and accurate communication is established, localization will be added. Using localization, we can begin running simple plans that move the robot from point to point within the designated drawing space.

Parallel testing will involve the writing implement. Now that the robots can move individually, testing and improvements to writing while drawing will start. Tests of writing quality during various motions and writing speed limits will be done to ensure we can meet quality and consistency requirements for the final drawing.

2 Project Management

2.1 Work Breakdown Schedule

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

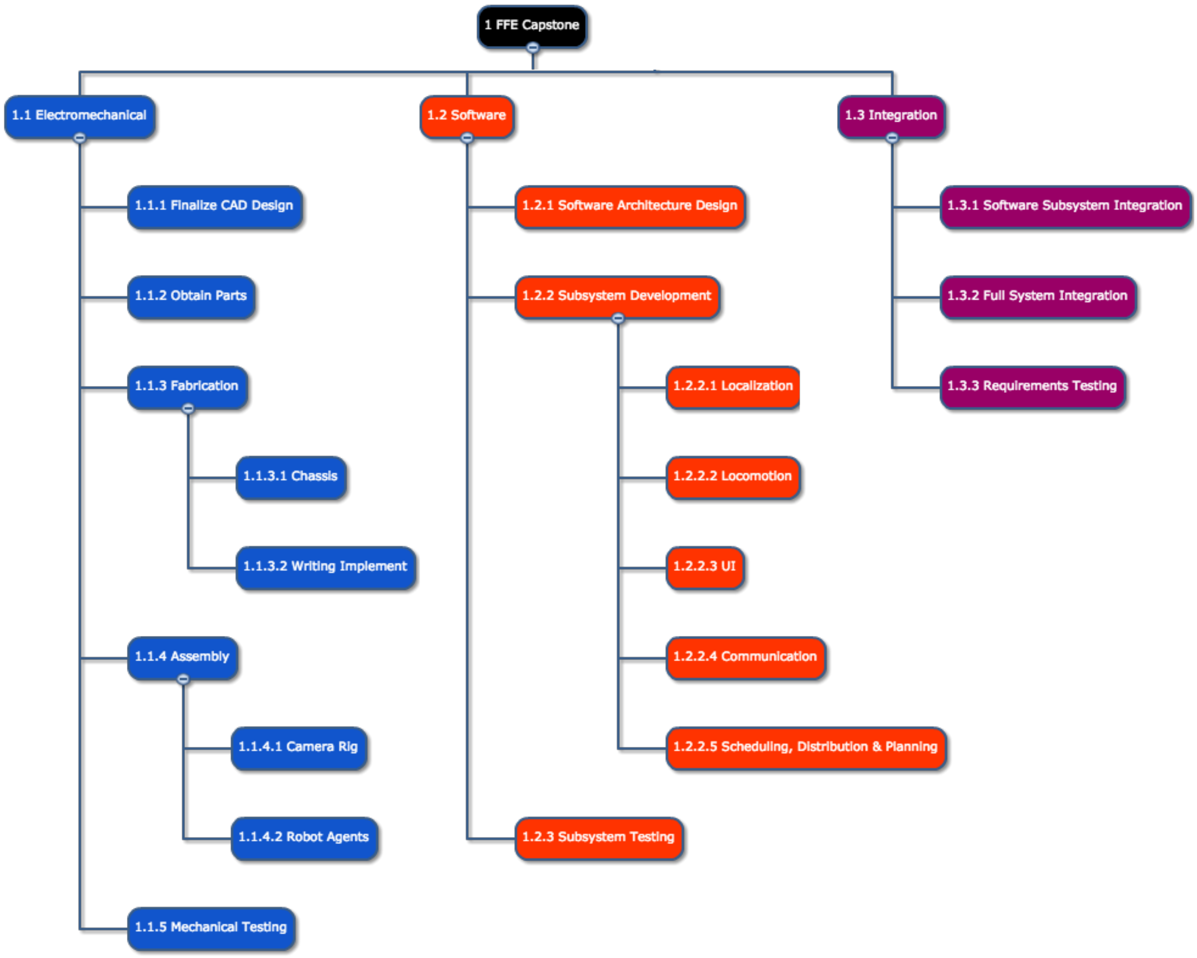

Figure 3: Full WBS for the project

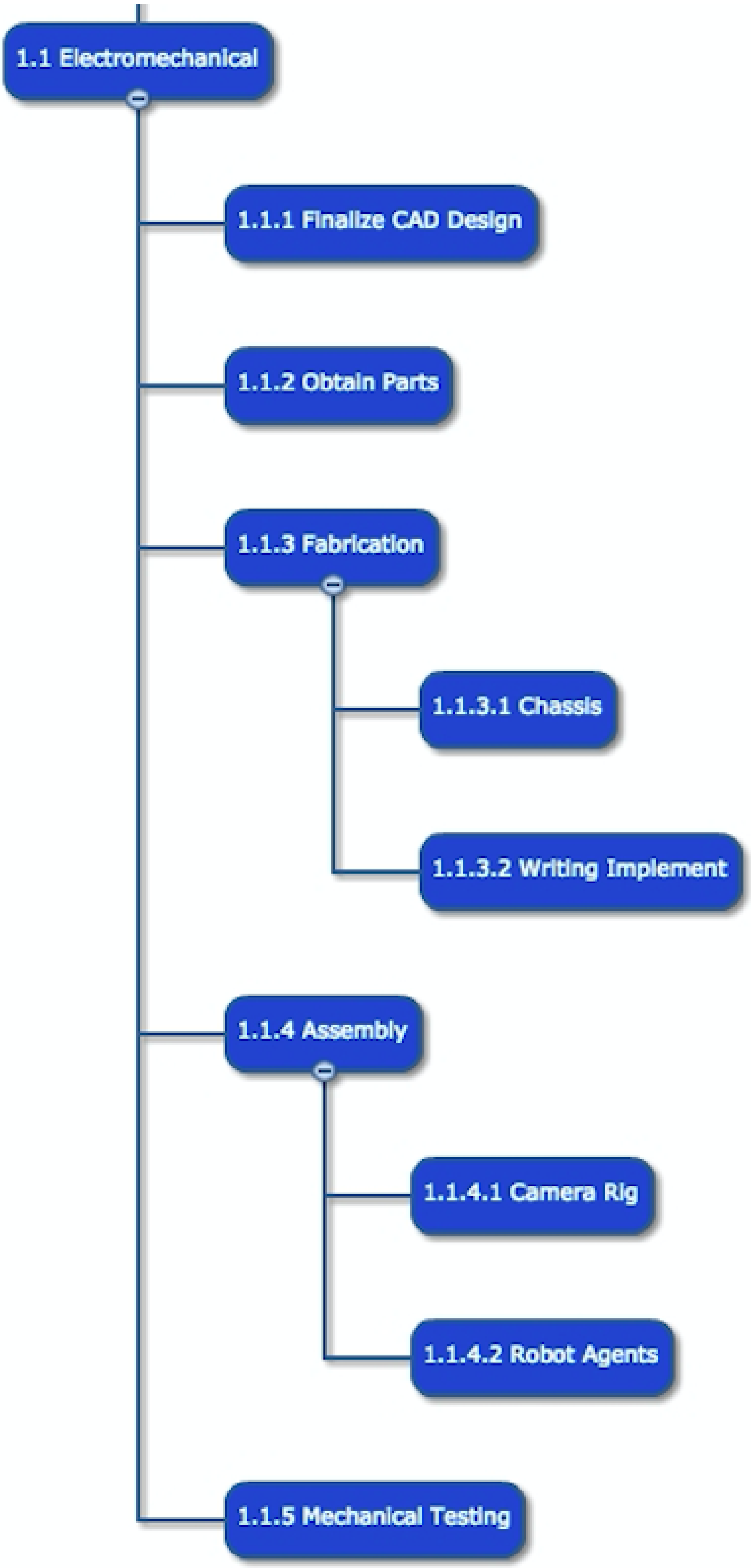

Figure 4: Electromechanical WBS section

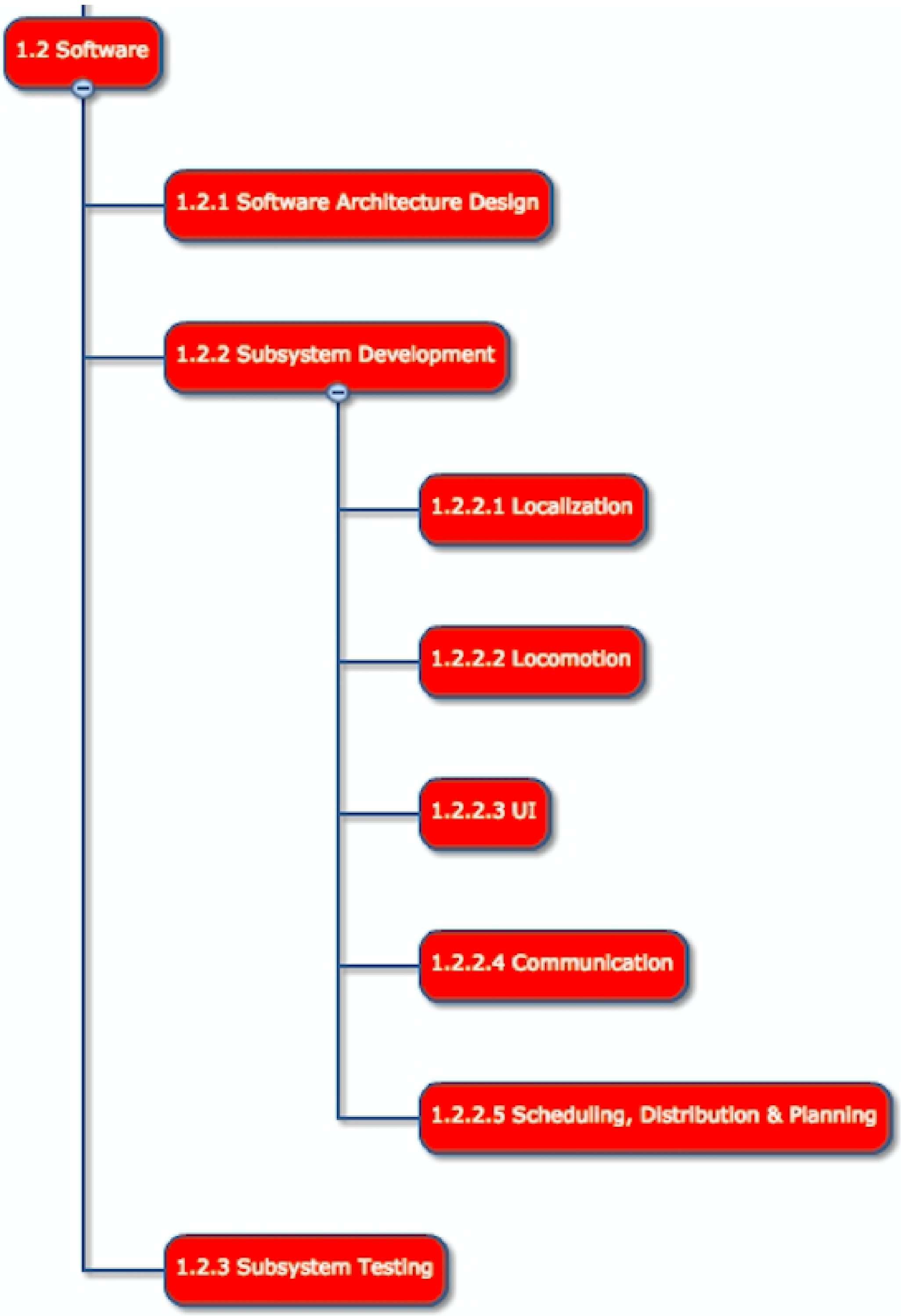

Figure 5: Software WBS section

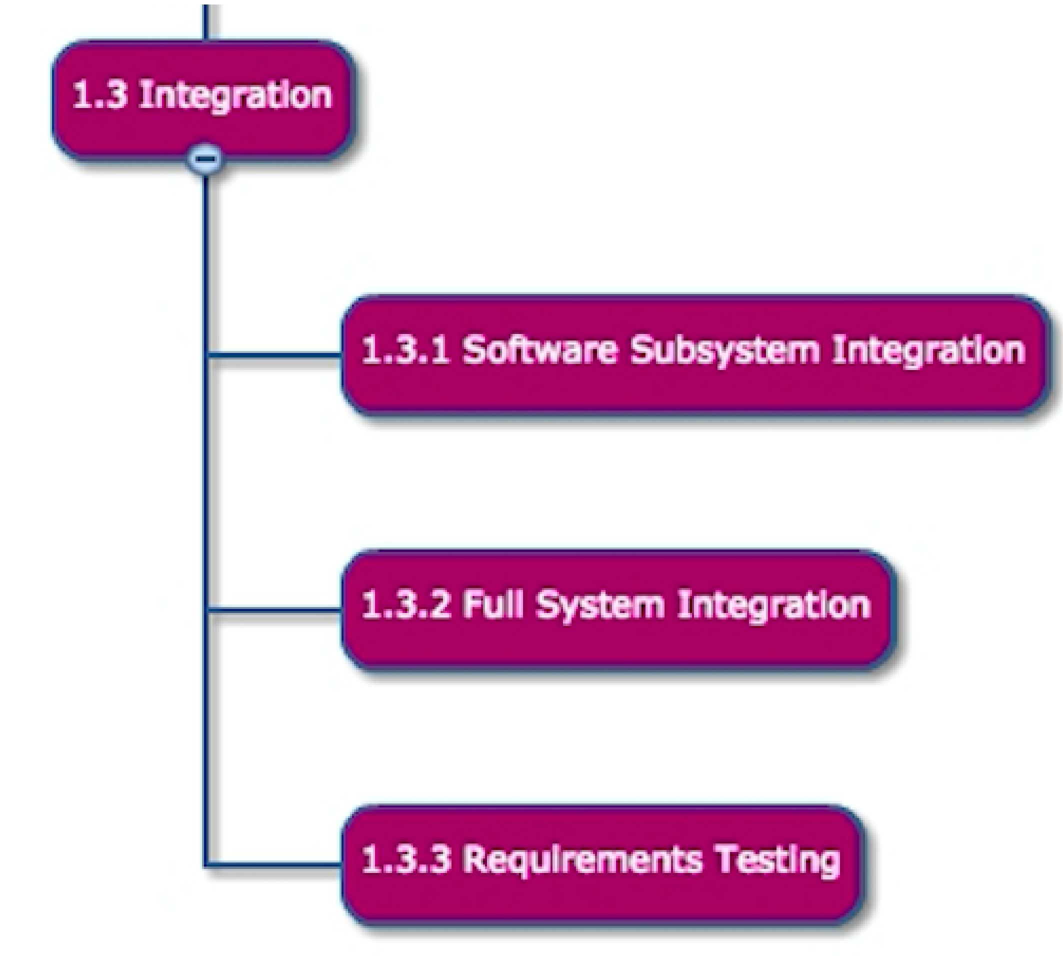

Figure 6: Integration WBS section

The WBS dictionary entries include more information on each of the work elements of the project. Information such as estimates for the amount of time each task will take and their dependencies will help us adhere to our schedule, while determining the owner of each task will improve tractability of the workflow.

WBS#:	1.1.1	Task:	Finalize CAD Design
Est. Effort (hrs):	3	Owner:	Eric
Resources:	CAD software	Work products:	CAD files
Description:	Update the CAD design to fit the final design of the robot		
Input:	Previous designs, new design ideas and requirements		
Dependencies:	Complete design review		
Risks:	Designs cannot be completed on time		

WBS#:	1.1.2	Task:	Obtain Parts
Est. Effort (hrs):	20	Owner:	Don
Resources:	Parts list	Work products:	Parts order receipt
Description:	Finalize the parts list, and contact the necessary people to ensure that parts are ordered		
Input:	CAD designs, electronics designs		
Dependencies:	Finalized CAD designs		
Risks:	Parts ordering procedure is more time consuming than expected		

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

Fig.7 shows our current progress towrads meeting our schedule. At this point, the electromechanical design is slightly behind - the camera rig assembly is still being built, and the second robot has not been constructed yet. The camera rig has been recently updated in design, and parts have been ordered. We do not expect assembly to take long, or hinder progress in integration. The second robot parts have already been ordered. We delayed building it to ensure we could finalize the robot design before building a second one.

Software implementation at this point is complete. All of the major subsystems are functional, and the only additions are to enhance or add additional features. Other software changes are being done for integration, to allow the various subsystems to work with the hardware, or with each other. The software design and implementation has reached usability and is therefore on schedule.

Integration is will underway, which is the majority of the work we have left. Some subsystems have been integrated, and others are actively being worked on. Motion onboard the robot is completed, and there are plans to finish integrating the communication and localization subsystems within the next week.

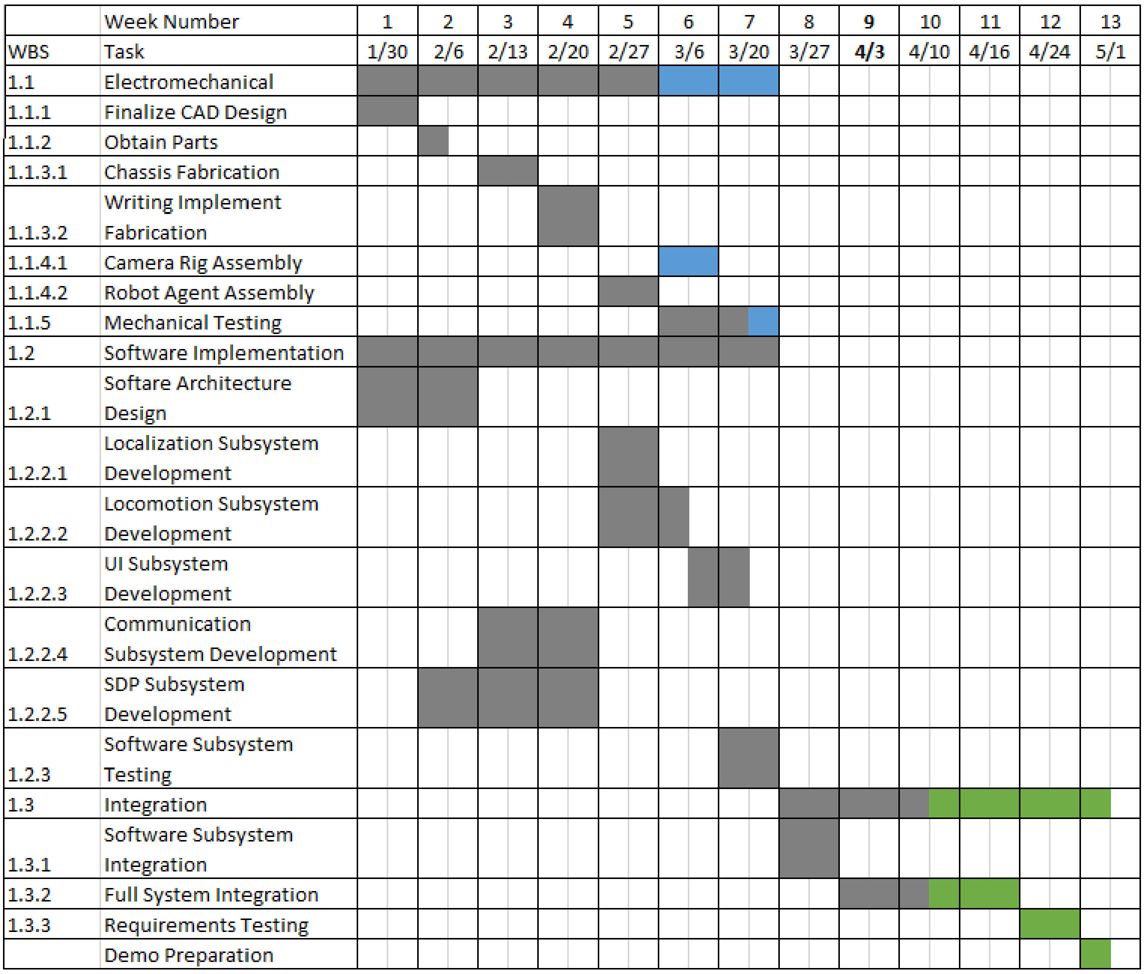

Figure 7: Schedule via Gantt Chart

3 Requirements Tracking

For readbility we provide a summary of our requirements below.

Requirement	Title
FR1	Omnidirectional Movement
FR2	Autonomous
FR3	Robots Localize Globally and Locally
FR4	Within Bounds
FR5	Insert Writing Tools
FR6	Remove Writing Tools
FR7	Replace Writing Tools
FR8	Coordination
FR9	Drive Control System
F10	Turn on or off writing tool
F11	Input Drawing Plan
F12	Robots Know Progress
F13	Kill Switch
F14	User Interface to Robot
NFR1	Docmentation
NFR2	Error Handling
NFR3	Weight Restriction
NFR4	Size Restriction
NFR5	Efficiency
NFR6	Quality
NFR7	Battery Power
NFR8	Reliability
NFR9	Reliable Communication
NFR10	Budget
NFR11	Safe
NFR12	Positional Accuracy
NFR13	Rotation Accuracy
NFR14	Tool Switching Duration

3.1 Objectives Tree

We created an objectives tree to better organize our requirements, and prioritize them based on system purposes and goals. After analysis of our requirements, we formed the objectives tree in Fig.8 with the following categories:

1. Is Safe (Fig.9)
2. Is Portable (Fig.10)
3. Drawing Tool is Easy to Operate (Fig.11)
4. Is Mobile (Fig.12)
5. User-Friendly (Fig.13)
6. Performance Guarantees (Fig.14)

The category for 'Is Safe' (Fig.9) encompasses requirements for the robot staying within bounds, maintaining reliable communication, existence of a kill switch, and overall safe operation. These requirements breakdown how safe usage of the robot can be achieved, through both system design and user operation.

'Is Portable' (Fig.10) specifies system constraints that enable the robots to be able to be transported easily. The battery-powered requirement ensures the robots do not need external power during operation. Weight and size requirements were further categorized into physical constraints, to emphasize the importance of those requirements on portability outside of system operation.

Subtree 'Drawing Tool is Easy to Operate' (Fig.11) ensures the writing tool is easy to maintain and use both during and before or after system operation. The main subtree describes tool maintenance. Requirements under maintenance include inserting, removing, and replacing the tool, as well as duration requirements for replacing the writing tool. Other requirements in this category relate to having the ability to enage or disengage the writing tool. This requirement is involved with system operation, and ensures the robot can change the tool status so the robots can move regardless of whether it is drawing.

The ‘Is Mobile’ (Fig.12) tree categorizes mobility requirements and constraints for the robot agents. Both positional and rotational accuracy are categorized under their own Accuracy subtree. Other leaves in this subtree ensure the robot agents have their own drive control systems, can localize, and are able to move autonomously in any direction on a 2D plane.

We also chose to separate out requirements that relate to engaging the user and enable a user-friendly experience. These fall under the ‘User-Friendly’ subtree (Fig.13). Both documentation and budget requirements were categorized here - these requirements are more likely to be for users interested in adapting or recreating our system. As a result, other requirements were further categorized into a user-interaction subtree. These constraints denote existence of a UI, error handling, and the ability for users to input their drawing plan.

The final categorization, ‘Performance Guarantees’ (Fig.14) denotes overall system requirements to ensure the final drawing meets specifications. These requirements include ensuring the robots know their own progress, and coordinate with each other. In addition, requirements specifying system efficiency, reliability, and overall drawing quality fell into this category.

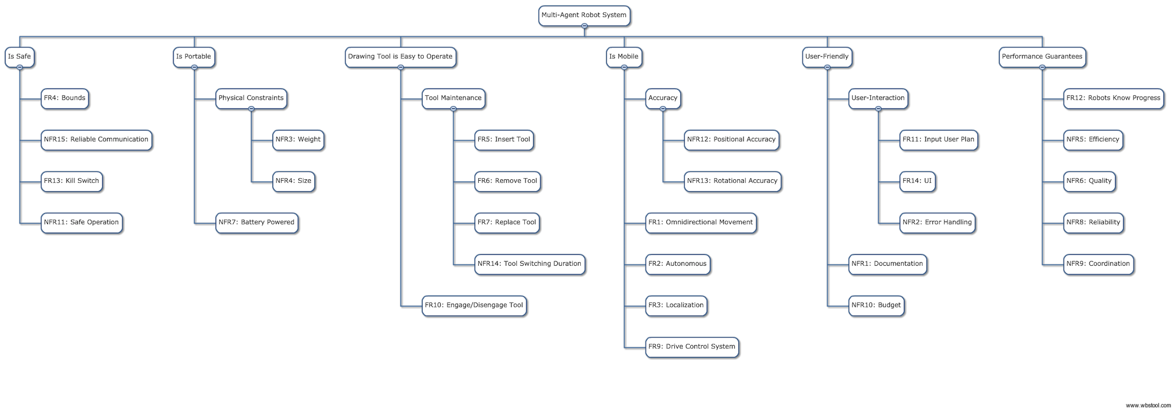

Figure 8: Full Objectives Tree

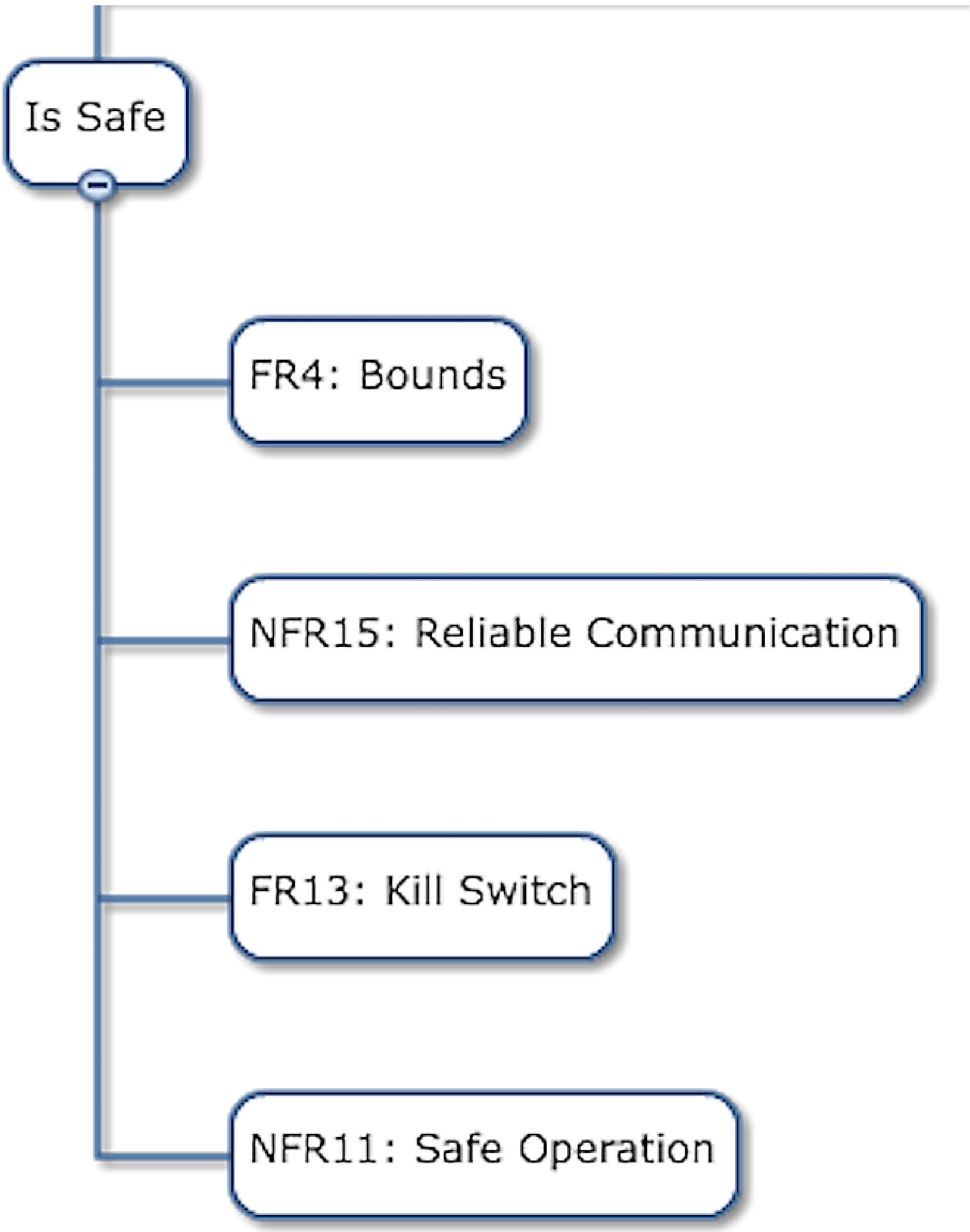

Figure 9: Objectives Tree: Is Safe Branch

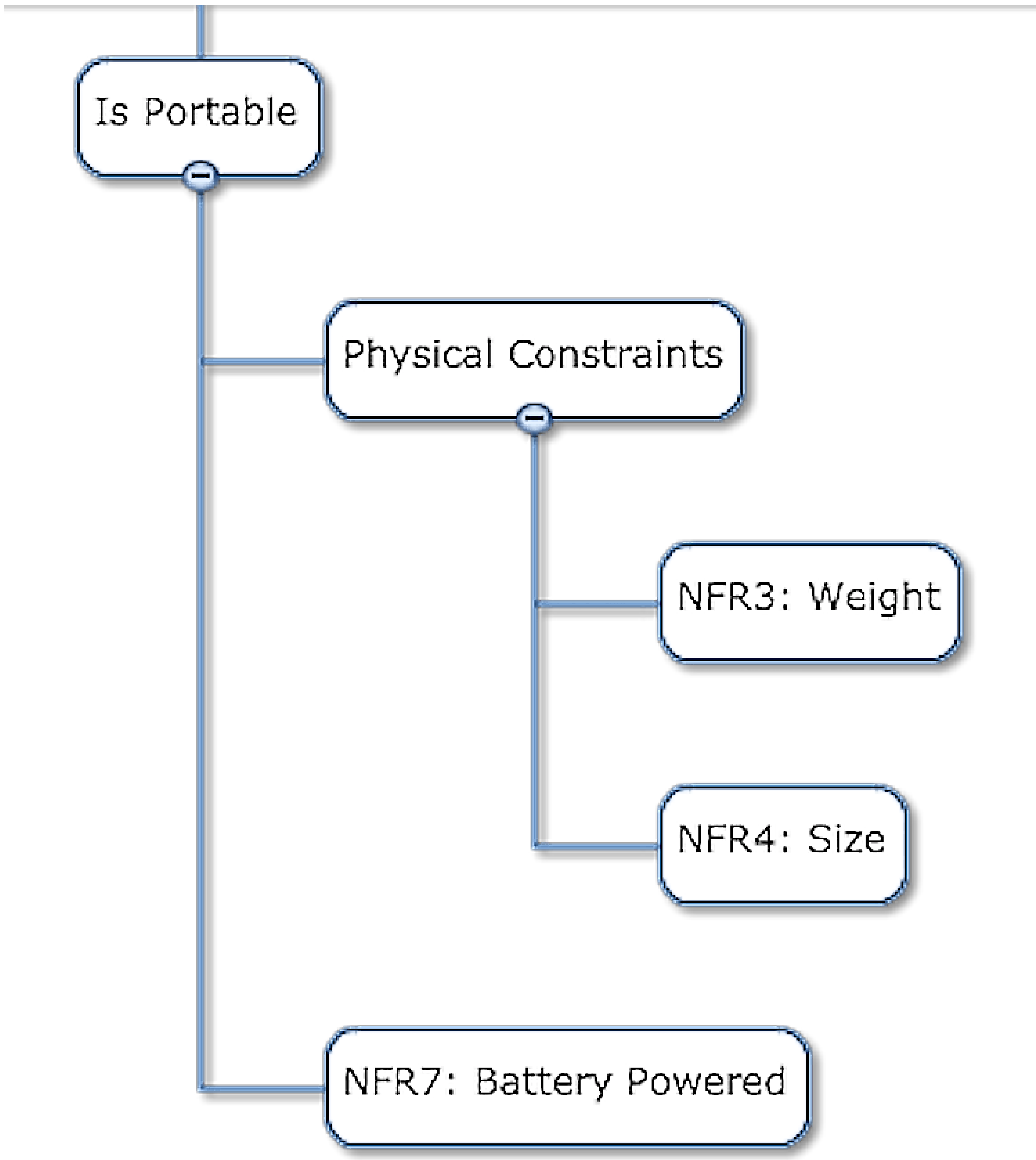

Figure 10: Objectives Tree: Is Portable Branch

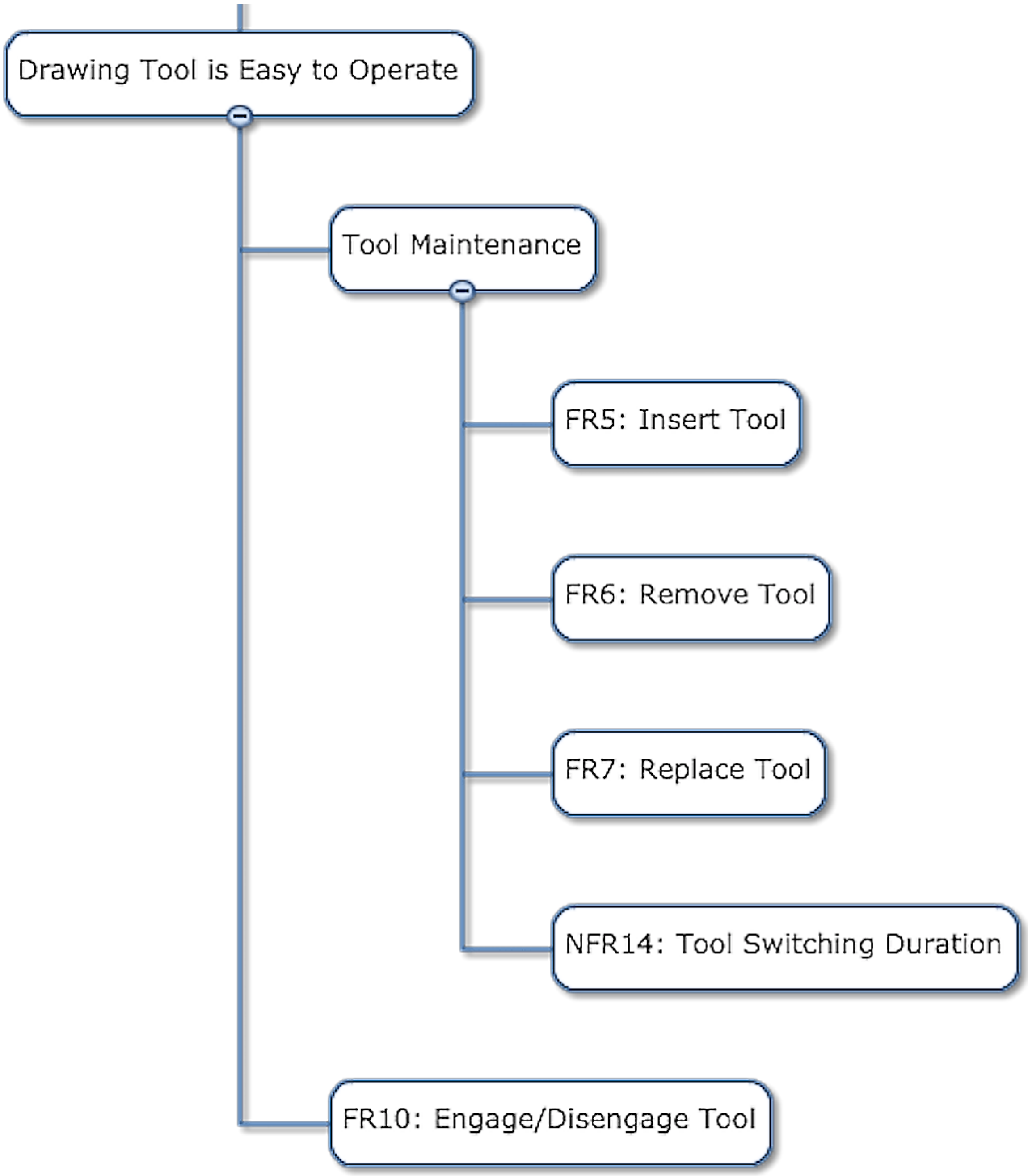

Figure 11: Objectives Tree: Drawing Tool is Easy to Operate Branch

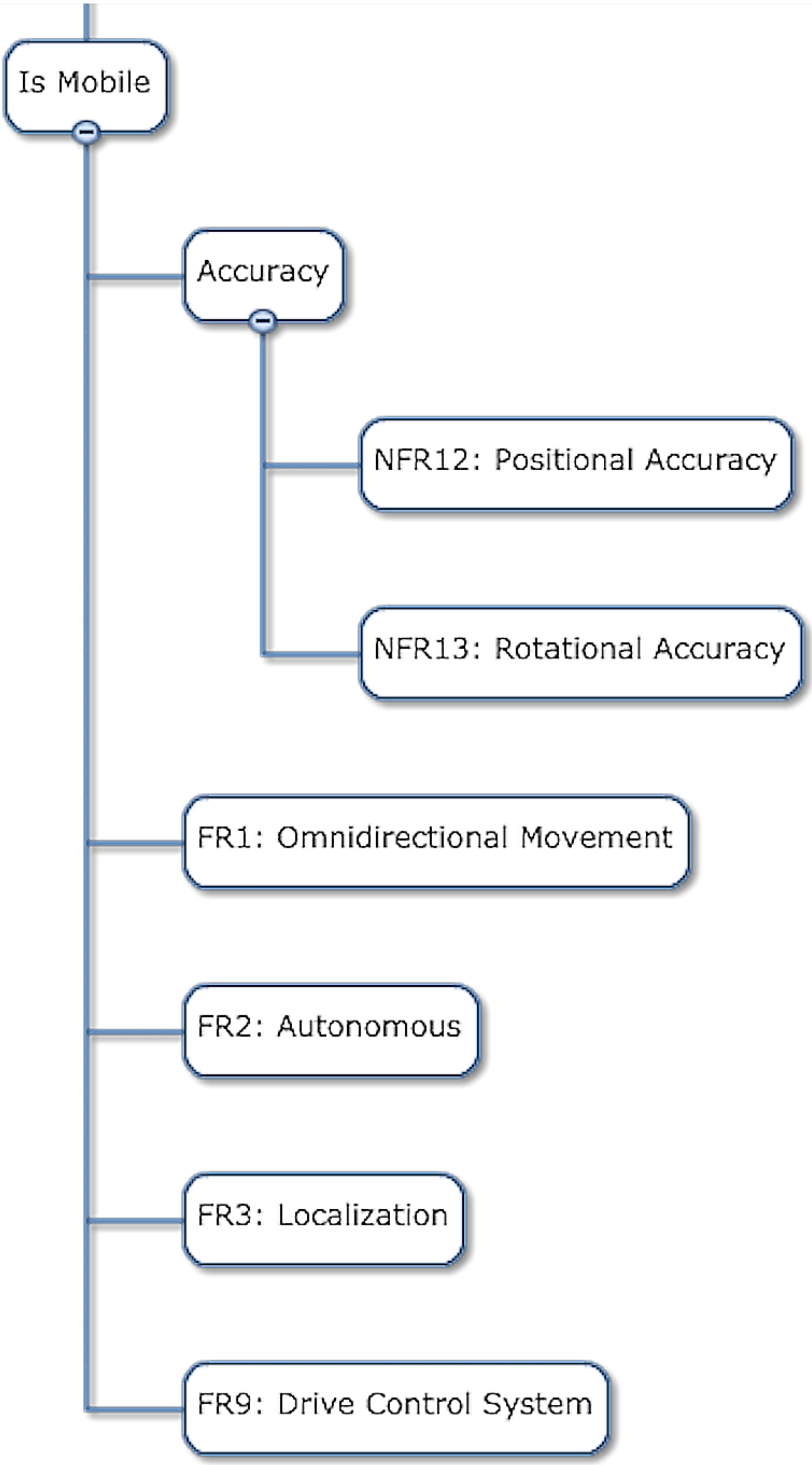

Figure 12: Objectives Tree; Is Mobile Branch

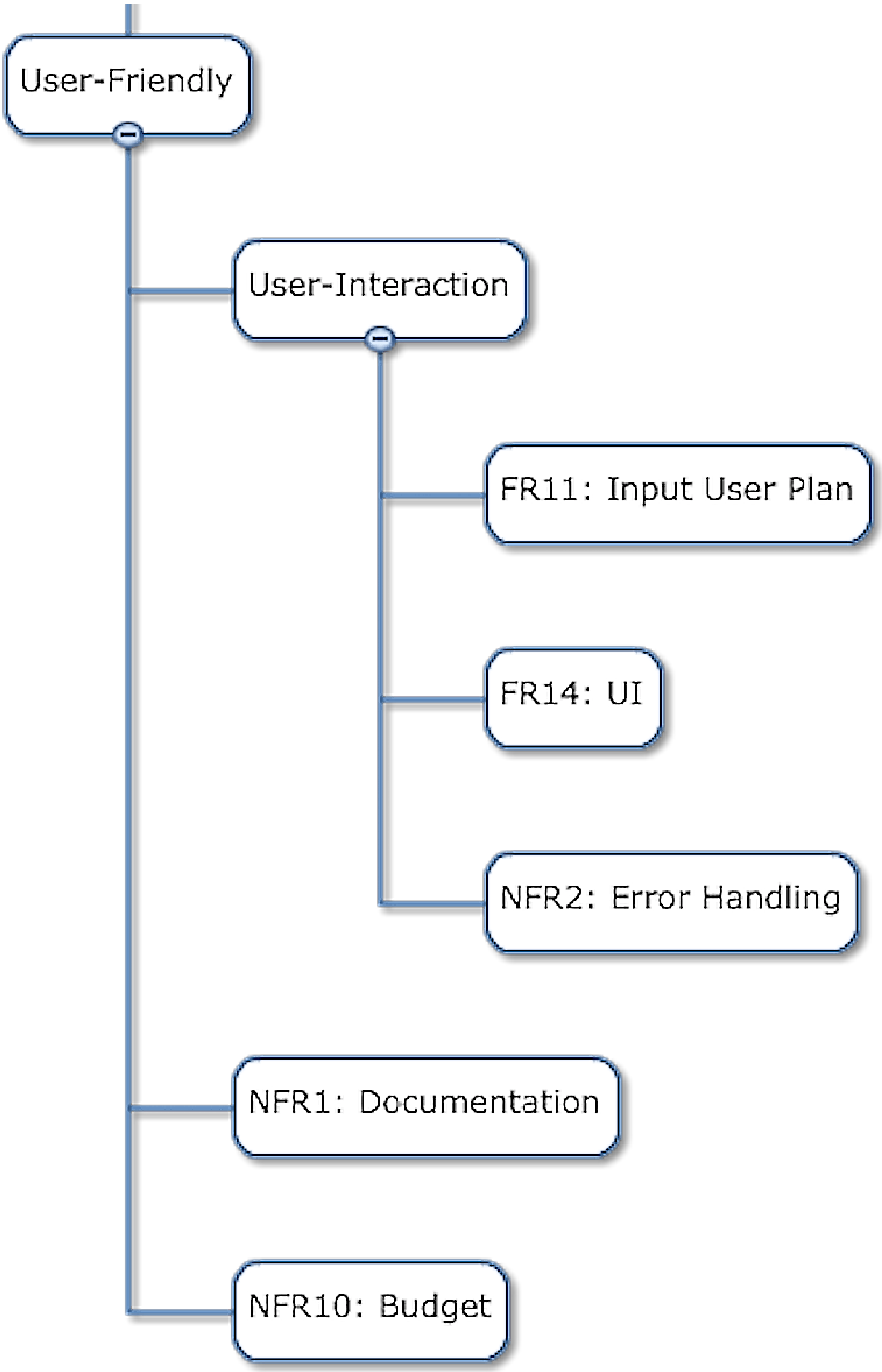

Figure 13: Objectives Tree: User-Friendly Branch

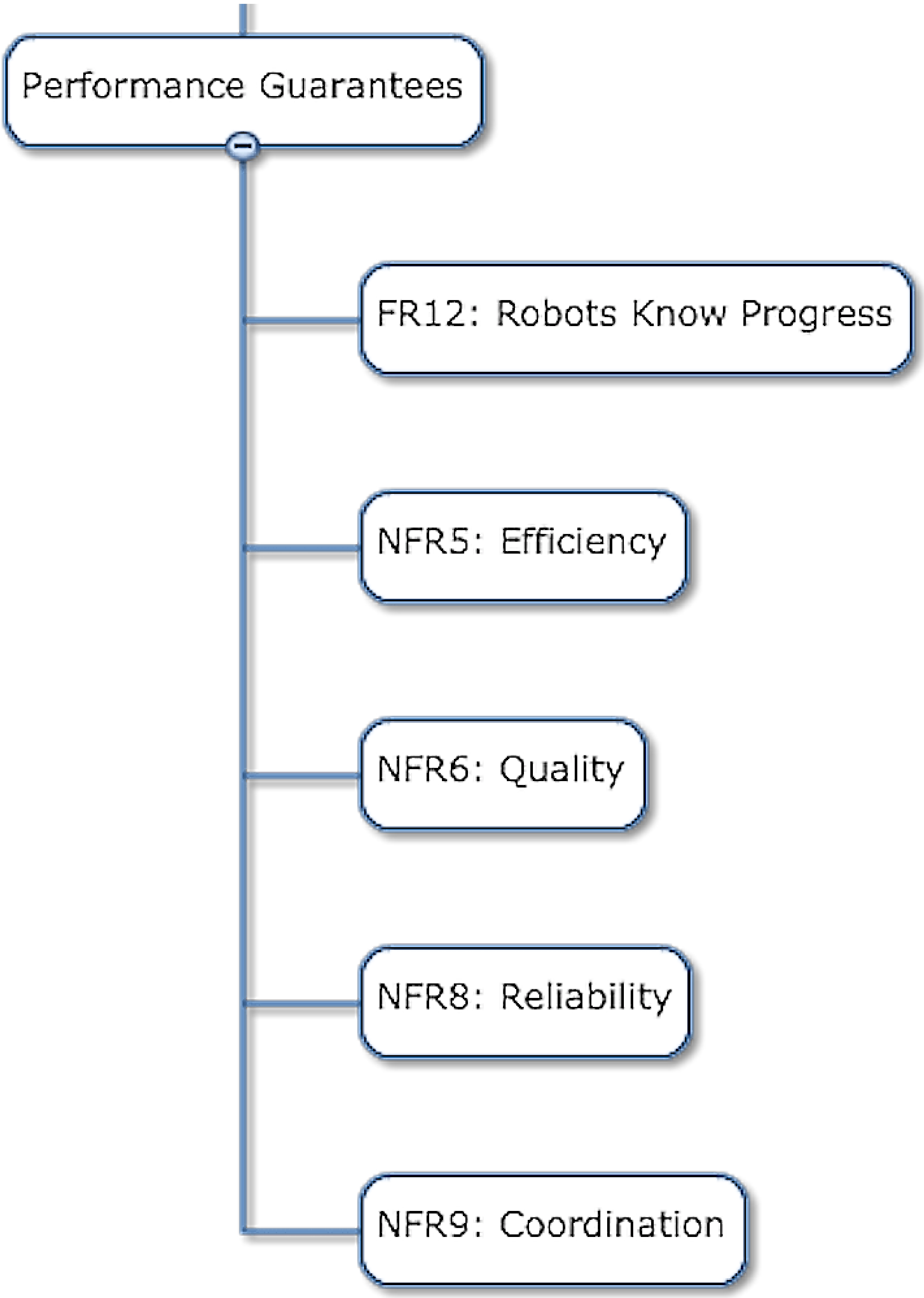

Figure 14: Objectives Tree: Performance Guarantees Branch

3.2 Requirements Traceability Matrix

We present our requirements traceability matrix in Fig.15. We categorize our requirements into functional and nonfunctional requirements. For each requirement, we classify which subsystem it relates to: writing, communication, locomotion, localization, SDP (scheduling, distribution and planning) and mechanical structure. Each requirement is coloered green if initial testing has shown that we achieve the requirement.

	Subsystems	Writing	Communication	Locomotion	Localization	SDP	Mechanical Structure
Functional Requirements	FR1: Omnidirectional Motion			X			
	FR2: Autonomous			X	X	X	
	FR3: Localize Globally and Locally				X		
	FR4: Within Bounds			X	X		
	FR5: Insert Tool	X					
	FR6: Remove Tool	X					
	FR7: Replace Tool	X					
	FR8: Coordination					X	
	FR9: Drive Control			X			
	FR10: Turn on/off Tool	X					
	FR11: Input Plan		X			X	
	FR12: Know Progress		X		X	X	
	FR13: Kill Switch	X	X	X			
	FR14: User Interface		X			X	
Nonfunctional Requirements	NFR1: Documentation	X	X	X	X	X	X
	NFR2: Error Handling	X	X	X	X	X	X
	NFR3: Weight Restriction						X
	NFR4: Size Restriction						X
	NFR5: Efficiency					X	
	NFR6: Quality	X		X	X		
	NFR7: Battery Power	X		X			
	NFR8: Reliability	X	X	X	X	X	X
	NFR9: Reliable Communication		X				
	NFR10: Budget						X
	NFR11: Safe			X	X	X	X
	NFR12: Positional Accuracy			X	X		
	NFR13: Rotational Accuracy			X	X		
	NFR14: Tool Switching Duration	X					X

Figure 15: Requirements Traceability Matrix

4 Risk Management

In this section, we revisit the risks defined by our previous document. The risk tables have been updated with the actions we've taken to address the risks.

Risk ID:	Risk Title:	Risk Owner:
1	Defective Parts	Don
Description:		
Parts that we ordered arrived defective or do not perform to specifications		
Consequences:	Risk Type:	
We need to reorder parts, expending extra time and budget	Parts	
Risk Reduction Plan:	Expected Outcome	
We will order only parts that have been extensively reviewed, or we have experience with, and order extra parts	We will be able to properly deal with any parts that break during the development process	

Consequence					
1	2	3	4	5	
					5
					4
					3
	X				2
					1 Likelihood

Figure 16: Risk 1: Defective Parts

Risk ID:	Risk Title:	Risk Owner:
2	Unvavailable Group Member	All
Description:		
A group member becomes unavailable for work due to travel, sickness, or other emergencies		
Consequences:	Risk Type:	
Work that would have been distributed to that group member needs to be reassigned	Logistical	
Risk Reduction Plan:	Expected Outcome	
We will ensure that every group member is always on the same page about progress so we don't lose too much progress	If a member becomes unavailable, it will only be for a short time and can be easily dealt with	

Consequence					
1	2	3	4	5	
					5
					4
			X		3
					2
					1 Likelihood

Figure 17: Risk 2: Unavailable Member

Risk ID:	Risk Title:	Risk Owner:
3	Breaking parts	Eric
Description:		
Parts unexpectedly break as a result of accidents or improper use		
Consequences:	**Risk Type:**	
We need to reorder parts, expending extra time and budget	Parts	
Risk Reduction Plan:	**Expected Outcome**	
We will practice safe procedures when working with parts and order extras in case	Few parts will break, and even if they do we will have extras on hand	

Consequence

1	2	3	4	5		
					5	
					4	
					3	
			X		2	
					1	Likelihood

Figure 18: Risk 3: Breaking Parts

Risk ID:	Risk Title:	Risk Owner:
4	Mecanum Drive Too Unstable	Eric
Description:		
The drive mechanism for the robot proves too be too unstable or unreliable for our purposes		
Consequences:	**Risk Type:**	
We will need to redesign the drive mechanism, expending considerable time and effort	Design flaw	
Risk Reduction Plan:	**Expected Outcome**	
We will build enough time in our schedule to deal with it if necessary, and will use suspension	The instability resulting from the wheels will be manageable	

Consequence

1	2	3	4	5		
					5	
					4	
			X		3	
					2	
					1	Likelihood

Figure 19: Risk 4: Mecanum Drive Too Unstable

Risk ID:	Risk Title:	Risk Owner:
5	Localization not precise enough	Neil

Description:
Our localization system is not precise enough to ensure that the drawings are accurate representations of input

Consequences:	Risk Type:
We will need to redesign the localization system or redefine drawing requirements	Design flaw
Risk Reduction Plan:	**Expected Outcome**
We will test the localization system early on in order to catch any design flaws within the system	Localization will work well enough for our purposes

Consequence	1	2	3	4	5	
						5
						4
						3
			X			2
						1 Likelihood

Figure 20: Risk 5: Localization not precise enough

Risk ID:	Risk Title:	Risk Owner:
6	Unexpected Budget Overruns	Rachel

Description:
We unexpectedly run out of budget, because parts cost more than expected or other parties reduce our budget

Consequences:	Risk Type:
We need to scale down our project, or possibly even acquire funds through other means	Logicstical
Risk Reduction Plan:	**Expected Outcome**
We will leave a significant buffer in our budget in case unexpected situations occur	We will have a large enough buffer that essential components will be acquired

Consequence	1	2	3	4	5	
						5
						4
						3
		X				2
						1 Likelihood

Figure 21: Risk 6: Unexpected Budget Overruns

5 Testing and Evaluation Plan

YJ: Eric update this with new information