Version 4.0

Life Cycle Plan (LCP)

Mission Science iRobots

Team 07

Jiashuo Li	**Project Manager, Life Cycle Planner, Developer**
Chen Li	**Requirements Engineer, Software Architect, Developer**
Farica Mascarenhas	**Operational Concept Engineer, IV&V, Quality Analyst**
Hanadi Mardah	**Tester**
Sergey Mukhin	**Prototyper, Developer**
Yun Shao	**Feasibility Analyst, Developer**

Version History

Date	Author	Version	Changes made	Rationale
08/20/12	SK	1.0	Original for CSCI 577; Tailored from ICSM OCD Template	To fit CS577 Course content
09/28/14	SR	1.1	To evaluate skills and responsibilities of the team members	To understand team skills and assign them various roles
10/10/14	SR	2.0	Sections 2, 3.1, 3.2, 4, 5 are added.	To modify version 1.1 according to the requirements of the FC package
10/18/14	SR	2.1	Sections 3.1, 5 were updated.	To modify version 2.0 according to the changes mentioned by the ARB.
11/27/14	SR	3.0	Sections 1.2, 2, 3, 5 were updated. Section 6.1 was added	To modify version 2.1 according to the requirements of the DC package.
12/08/14	SR	3.1	Section 5 were updated.	To modify version 3.0 according to the changes mentioned by the ARB.
02/09/15	Jiashuo Li	3.2	Chapter 5.2 COCOMO estimation Chapter 2 Milestones and Products	Re-estimate for re-based DCR. Add description for following development.
04/01/15	Jiashuo Li	4.0	Chapter 5. 2 COCOMO estimation Section 6.2.2 Core Capabilities Drive-Through Results Section 6.3 Adherence to Plan	Re-estimate after CCD. Add client feedback for CCD. Add description for following development.

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the LCP

The purpose of the life cycle plan is to streamline the project into various phases so that the entire development team and client can achieve improved development speed, improved quality, improved project tracking and control, improved relations and minimal exposure to risks.

1.2 Status of the LCP

The status of the LCP is currently at Development Commitment Package version number 3.2.

The major changes from Foundation Commitment Package version number 3.1 are:

1. Update milestone ,products and responsibilities for this semester
2. Update approaches
3. Update of development iteration plan.

The list of deliverables and the overall strategy to develop the project are also added. The resources of the project required for the project are also estimated to analyze the project's feasibility within 24 weeks.

1.3 Assumptions

- The duration of the project is 24 weeks, which are 12 weeks in Fall 2014 and 12 weeks in Spring 2015.
- The team will get a licensed version of all the software to be used for the project.
- Each team member will stick to his responsibilities mentioned in section 2 during each phase and will perform them accordingly.
- The elementary schools will like the GUI which has been developed for the iRobot.
- There will be regular meetings with the client, to discuss the progress, issues and other concerns.
- 5 on-campus students and 1 DEN student is an optimum number of staff required to do this project in the given schedule.

2. Milestones and Products

2.1 Overall Strategy

The development of the GUI for Mission Science iRobot is going to be from scratch. The project will use the ARCHITECTED AGILE process of the Incremental Commitment Spiral Model as all the components are going to be custom made.

Exploration Phase
Duration: 09/12/14 – 09/26/14
Concept: The team should focus on understanding the current system and design the business work flow in the Exploration phase and would conduct regular weekly meetings with the client to discuss and understand current system, requirements, concerns and progress.
Deliverables: Valuation Commitment Package
Milestone: Valuation Commitment Review
Strategy: One Incremental Commitment Cycle

Valuation Phase
Duration: 09/29/14 – 10/24/14
Concept: To evaluate the risks, the SCSs (Success Critical Stakeholders) including the students and course instructors, and the developers will have win-win negotiations. The team will gather requirements and then along with the stakeholders, they will prioritize the requirements and a proposed system will be defined by these win-win negotiations. Based on this definition the team prepares initial prototypes of the high risk win conditions.
Deliverables: Core Foundations Commitment Package, Draft Foundations Commitment Package, Project Effort Reports, Project Plan, Progress Reports, Prototype Report, System and Software Architecture Description, Supporting Information Document.
Milestone: Foundation Commitment Review
Strategy: One Incremental Commitment Cycle

Foundations Phase
Duration: 10/20/2014– 12/1/2014
Concept: In this phase, the feasibility of each requirement (Win condition) is determined and development starts with, usually, the most feasible and required features. Continue risk assessment process, regular stakeholder meetings are to be taken every week, regular progress reports and effort reports to be submitted every alternate Monday, project plans are to be prepared and released on project web-page, risk resolution, assessing project status, sharing implementation jobs.
Deliverables: Development Commitment Package
Milestone: Development Commitment Review
Strategy: One Incremental Commitment Cycle

Foundation Phase – Rebaselining:
Duration: 01/12/2015 – 02/10/2015

Concept: Since some teammates might leave and new teammates might come, the roles & responsibilities should be reallocated. In addition, after a winter break, teammates should spend time on accessing project status, preparing for development phase, and testing.
Deliverables: Rebaselined Development Commitment Package
Milestone: Rebaselined Development Commitment Review
Strategy: One Incremental Commitment Cycle

Rebaselining – Code Review:
Duration: 02/11/2015 – 03/04/2015
Concept:
The development formally starts now and all team members devote himself/herself into the intensive development process.
At the same time, testing plans is implemented. Unit testing test-cases are designed and performed on code under development. Module/functionality testing plans are also released with the client in order to guarantee the quality of the product.
Deliverables: Core Capabilities Drive-through
Milestone: Design Code Review
Strategy: Several incremental commitment cycles in order to refine the product again and again according to the feedback of clients

Code Review – TRR:
Duration: 03/04/2015 – 04/08/2015
Concept:
The basic modules of the system have been designed and implemented. Core capability modules are also implemented.
At the same time, unit testing for basic modules are designed and implemented.
Deliverables: Transition Readiness Package
Milestone: Transition Readiness Review
Strategy: Continue the previous commitment cycles to add more functions and features to the system. Make code peer review within the team to ensure the quality of the system.

2.2 Project Deliverables

2.2.1 Exploration Phase

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	09/22/2014	.doc, .pdf	Soft copy
Valuation Commitment Package • Operational Concept Description (OCD) Early Section • Life Cycle Plan (LCP) Early Section • Feasibility Evidence Description (FED) Early Section	09/29/2014	.doc, .pdf	Soft copy
Evaluation of Valuation Commitment Package	10/11/2014	.xls	Soft copy
Project Plan	Every alternate Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every alternate Wednesday	.xls	Soft copy

2.2.2 Valuation Phase

Table 2: Artifact deliverable in Valuation Phase

Artifact	Due date	Format	Medium
Draft Foundations Commitment Package: • Operational Concept Description (OCD) • Feasibility Evidence Description (FED) • Life Cycle Plan (LCP) • System and Software Architecture Description (SSAD) • Prototype report (PRO)	10/13/2014	.doc, .pdf	Soft copy
Evaluation of Draft Foundations Commitment Package	10/13/2014	.doc, .pdf, Bugzilla	Soft copy, Bugzilla
Response to Evaluation of Draft Foundations Commitment Package	10/15/2014	.doc, .pdf, Bugzilla	Soft copy, Bugzilla
Foundations Commitment Package:	10/20/2014	.doc, .pdf	Soft copy

• Operational Concept Description (OCD) • Feasibility Evidence Description (FED) • Life Cycle Plan (LCP) • System and Software Architecture Description (SSAD) • Prototype report (PRO) • System and Software Requirements Definition			
Bugzilla report	Every Monday	Text	Bugzilla Website
Project Plan	Every alternate Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every alternate Wednesday	.xls	Soft copy

2.2.3 Foundation Phase

Table 3: Artifacts Deliverables in Foundation Phase

Artifact	Due date	Format	Medium
Draft Development Commitment Package: • Operational Concept Description (OCD) • Feasibility Evidence Description (FED) • Life Cycle Plan (LCP) • System and Software Architecture Description (SSAD) • Prototype report (PRO) • Test Plan and Cases • Transition Plan	12/01/2014	.doc, .pdf	Soft copy
Evaluation of Draft Development Commitment Package	12/05/2014	.doc, .pdf, Bugzilla	Soft copy, Bugzilla
Response to Evaluation of Draft Foundations Commitment Package	12/08/2014	.doc, .pdf, Bugzilla	Soft copy, Bugzilla
Development Commitment Package: • Operational Concept Description (OCD) • Feasibility Evidence Description (FED)	12/08/2014	.doc, .pdf	Soft copy

• Life Cycle Plan (LCP) • System and Software Architecture Description (SSAD) • Prototype report (PRO) • System and Software Requirements Definition(SSRD) • Test Plan and Cases (TCP)			
Bugzilla report	Every Monday	Text	Bugzilla Website
Project Plan	Every alternate Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every alternate Wednesday	.xls	Soft copy

2.2.4 Re-baselined Foundation Phase

Table 4: Artifacts Deliverables in Re-baselined Foundation Phase

Artifact	Due date	Format	Medium
Microsoft Project Plan	Bi-weekly Monday	.mpp, .pdf	Soft copy
Progress Report	Bi-weekly Monday	.xls	Soft copy
Risk Analysis	Bi-weekly Monday	Text	Part of Progress Report
Development progress report	Weekly	Text	Google Group
Development feedback	As needed	Text	GitHub
Re-baselined Development Commitment Package: • Operational Concept Description (OCD) • Feasibility Evidence Description (FED) • Life Cycle Plan (LCP) • System and Software Architecture Description (SSAD) • Prototype report (PRO) • System and Software Requirements Definition(SSRD) • Test Plan and Cases (TCP)	02/11/2015	.doc, .pdf	Soft copy

2.2.5 Development Phase

Table 5: Artifacts Deliverables in Re-baselined Foundation Phase

Artifact	Due date	Format	Medium

Microsoft Project Plan	Bi-weekly Monday	.mpp, .pdf	Soft copy
Progress Report	Bi-weekly Monday	.xls	Soft copy
Risk Analysis	Bi-weekly Monday	Text	Part of Progress Report
Development progress report	Weekly	Text	Google Group
Development feedback	As needed	Text	GitHub
Re-baselined Development Commitment Package: • Operational Concept Description (OCD) • Feasibility Evidence Description (FED) • Life Cycle Plan (LCP) • System and Software Architecture Description (SSAD) • Prototype report (PRO) • System and Software Requirements Definition(SSRD) • Test Plan and Cases (TCP) • Transition Plan (TP) • Regression Test Package (RCP) • Support Plan (SP) • Release Description (RD)	04/08/2015	.doc, .pdf	Soft copy

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities

Except for the client and developer team, the Mission Science iRobot project has two other success critical stakeholders:

- Elementary Students: These students will use the GUI to generate instructions to control the iRobot.
- Undergraduate Students: The undergraduate students will continuously monitor the GUI during development and train the elementary school students and teachers about how to operate the GUI.
- Elementary School Teachers: The teachers will help the elementary school students to develop logical statements using the GUI.

3.2 Responsibilities by Phase

The following table is a template for stakeholder's responsibilities in each phase.

Table 6: Stakeholder's Responsibilities in each phase

Team Member / Role	Primary / Secondary Responsibility				
	Exploration	**Valuation**	**Foundations**	**Development-** Construction Iteration	**Development-** Transition Iteration
Prof. Darin Gray Client	**Primary Responsibility** - Explain scope and primary requirement - Contribute to the win conditions - Clarify the problems from development team	**Primary Responsibility** - Assess work artifacts and provide feedback - Identify shared vision, goal, and concepts	**Primary Responsibility** - Provide feedback for prototypes	**Primary Responsibility** - Test system development modules - Provide feedback of system features	**Primary Responsibility** - Accept the training - Prepare for system transition
Jiashuo Li Project Manager, Life Cycle Planner, Developer	NA	NA	NA	**Primary Responsibility** - Record Project progress - Create detailed project plan - Create life cycle plan - Manage client interaction - Develop GUI instruction representation module **Secondary Responsibility** - Bugzilla management - GitHub repositories management	TBD

				- Monitor the development of each modules - Integrate the system	
Chen Li Requirements Engineer, Software Architect, Developer	NA	NA	NA	**Primary Responsibility** - Identify system and software requirements definition - Develop Microcontroller emulator - Keep track of system architecture, detail will architecture degradation and complete SSAD	TBD
Farica Mascrenhas Operational Concept Engineer, IV&V, Quality Analyst	NA	NA	NA	**Primary Responsibility** - Create operational concept description - Assess operational concept - Analyze and prioritize capabilities to prototype - Verify system development process and product quality - Review the project artifacts. **Secondary Responsibility** - Keep track of the win Conditions being the shaper of the project using Winbook.	TBD
Hanadi Mardah Tester	NA	NA	NA	**Primary Responsibility** - Design and implement unit testing cases - Perform testing on product modules and generate testing report - Design functionality testing cases - Perform testing on hardware devices to give feedback to the developers and client	TBD
Sergey Mukhin Prototyper, Developer	NA	NA	NA	**Primary Responsibility** - Develop Drag-and –drop GUI and parameter panels - Develop other add-on GUI modules in conjunction to the feedback received from peers and client	TBD
Yun Shao Feasibility Analyst, Developer	NA	NA	NA	**Primary Responsibility** - Develop Translator module - Document feasibility evidence description - Assess feasibility evidence - Risk analysis, mitigation and documentation	TBD

3.3 Skills

Table 7: Skills

Team members	Role	Required Skills

Common		C# Git Visual Studio 2013
Chen Li	Requirements Engineer, Software Architect, Developer	UML C++ Serial Port programming Website development Analytical skill
Farica Mascarenhas	Operational Concept Engineer, IV&V, Quality Analyst	Analytical Skill Quality Assurance
Hanadi Mardah	Tester	Unit testing Quality Assurance
Jiashuo Li	Project Manager, Life Cycle Planner, Developer	Project management .NET WPF Communication skill
Sergey Mukhin	Prototyper, Developer	.NET WPF
Yun Shao	Feasibility Analyst, Developer	Analytical skill

4. Approach

4.1 Monitoring and Control

4.1.1 Closed Loop Feedback Control

- The weekly progress reports identify the activities undertaken and completed in the week. Project Plan provides the baseline for the activities. Any deviation from this baseline is identified, its severity is analyzed and action, as appropriate, is taken
- The progress reports records planned and actual efforts and tasks spent. If the difference is outside acceptable limits, then the Project Manager can initiate action by calling in team meeting and discuss this issue and follow up measures and tasks for next week
- The weekly risk analysis ensures that all risks identified have a mitigation plan. This will help to stay on track with the project schedule.
- The Winbook lists all the requirements and the risks and also prioritizes the requirements.
- The Team uses Google drive to communicate all the matters within the members and to keep all the artifacts organized. The team also uses the DEN discussion forum and blog for sharing files and for communicating.
- **Code is put on GitHub so that the development can be monitored and easily managed**

4.1.2 Reviews

- Weekly group review: This review is made every week so that each team member can contribute their work.
- IIV&V: By the den student, all artifacts are reviewed and bugs are released for each one of them.
- Win-Win Negotiations: Negotiations and review in which all values from the SCS are noted. Also help to estimate and, prioritize and order the requirements
- Review by Undergraduate Students: The undergraduate students incrementally monitor the GUI of the system and give their feedback which can be used by the developer team for further improvements.
- **Every developer must submit Development Progress Report every week so that their work can be tracked**

4.2 Methods, Tools and Facilities

Table 8: Tools and Usage

Tools	Usage	Provider
Microsoft Word 2013	Constructs documents for all artifacts. Also used for developing client interaction reports.	USC
Microsoft Project 2013	Creates Gantt charts for Progress reports and LCP	USC
Microsoft PowerPoint 2013	Creates presentation for client meeting and ARB reviews	USC
Mozilla Firefox and Google Chrome	Downloads course material and medium for communication among stakeholders	USC
Winbook V2	Facilitates and supports Win-Win negotiation	USC
Subversion	Implements version control system to maintain artifact integrity and traceability	Open Source
Microsoft Visual Studio 2013	Build .NET framework	USC
Visual Paradigm	Creates UML diagrams that are used for documenting the use cases and sequence diagrams	USC
WinAVR	Compiler for C programs after creating set of instructions to be implemented in Visual Studio.	Open Source
Google Drive	Upload Documents on Google Drive for Peer Review	Google
GitHub	**Web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM)**	**GitHub**
SmartGit	**A GUI Git repository management tool**	**Syntevo**

5. Resources

- Estimated CSCI577b Effort: 6 team members at 40 hrs/week for 12 weeks
- Total estimated effort: 74.25
- Budget information: This project has no budget for our development efforts, while the software is provided and tools are free.
- Project duration: 12 weeks (Spring 2015)
- Component modules in your development project:
  - GUI
  - Translator
  - Emulator
  - Unit Test
- Programming language used: C# .NET

Table 9: Module Lists and their SLOC

No.	Module Name	Brief Description	SLOC	REVL
1	GUI	A drag-and-drop GUI which allows users to "write" program in a graphical way	6000	40%
2	Translator	Translate high-level instructions to C code	440	10%
3	Emulator	Emulate Microcontroller	300	10%
4	HLProgram	Control the workflow of the total program	1200	5%

Table 10: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	NOM	There is a considerable understanding of the product's objectives. There are also nominal experience in developing such GUI and some need for innovative data processing and architectures.
FLEX	VHI	The choices for designing the GUI are many. The developers have the ability to choose the design whatever they deem right.
RESL	NOM	All major risks are documented with mitigation plans for each module; however there is a considerable amount of uncertainty pertaining to the success of User Interface and COTS software.
TEAM	HI	No extra efforts are needed to synchronize stakeholders and they are ready to accommodate other stakeholder's objectives.

PMAT	NOM	ICSM Principles and Guidelines are followed strictly by the developer team. Besides, WinAVR workflow is off-the-shelf which makes the workflow mature.

Table 11: COCOMOII Cost Driver for GUI

Cost Driver	Value	Rationale
RELY	NOM	If the system fails it will deter the elementary students to use the iRobot. However, it will not have a catastrophic effect on the use of iRobot by the students.
DATA	LO	There will not be a lot of data to be processed.
DOCU	NOM	Nominal Documentation is required for future maintenance.
CPLX	NOM	Highly structured programming with nested controls and extensions will be required for this module.
RUSE	LO	This module doesn't need to be reused.
TIME	HI	The product must be effective and operative, otherwise kids will lose their interest.
STOR	NOM	The high-level program will only consume thousands of bytes.
PVOL	LO	The platforms are stable and do not require frequent upgrades.
ACAP	NOM	Analysts have basic experience
PCAP	HI	Developer of GUI are familiar with C# and WPF development
PCON	HI	There will be no personnel change in this semester.
APEX	HI	The developers for this part have done programming related to GUI design before.
LTEX	NOM	Low programming language and software tool experience
PLEX	HI	The GUI may be complex regarding display instructions in a graphical way
TOOL	HI	Developer is familiar with Visual Studio.
SITE	VHI	All team members and client are on campus; whereas the elementary school students are in the same city.
SCED	NOM	There is a nominal schedule constraint.

Table 12: COCOMOII Cost Driver for Translator

Cost Driver	Value	Rationale
RELY	NOM	If the system fails it will deter the elementary students to use the iRobot. However, it will not have a catastrophic effect on the use of iRobot by the students.
DATA	LO	There will not be a lot of data for the Translator to translate.

DOCU	HI	Detailed high-level instructions must be documented so that the translation can be traceable.
CPLX	LO	There are only some simple replacement operations in the logic.
RUSE	LO	This module doesn't need to be reused.
TIME	NOM	The high-level program won't be too long, so time constraint is not severe.
STOR	NOM	The translated program needs to be stored, but it is very small.
PVOL	LO	The platforms are stable and do not require frequent upgrades.
ACAP	NOM	Analysts have basic experience
PCAP	LO	Programmer is unfamiliar with C# which is the main language for this part.
PCON	HI	There will be no personnel change in this semester.
APEX	LO	The developers haven't write something like this before.
LTEX	NOM	Low programming language and software tool experience
PLEX	NOM	Few team members have high platform experience
TOOL	LO	Developers haven't worked with Visual Studio before.
SITE	VHI	All team members and client are on campus; whereas the elementary school students are in the same city.
SCED	NOM	There is a nominal schedule constraint.

Table 13: COCOMOII Cost Driver for Emulator

Cost Driver	Value	Rationale
RELY	HI	There should not be outstanding bugs in the emulator, otherwise it cannot faithfully the functioning of Microcontroller
DATA	LO	There will not be a lot of data to be processed.
DOCU	NOM	Nominal Documentation is required for future maintenance
CPLX	NOM	COM Port programming is easy but the interrupt handling process may be complex.
RUSE	LO	The emulator doesn't need to be reused in the development of GUI.
TIME	HI	The emulator must be efficient enough, otherwise it can't handle interrupt.
STOR	NOM	Only source code and generated program need to be stored.
PVOL	LO	The platforms are stable and do not require frequent upgrades.

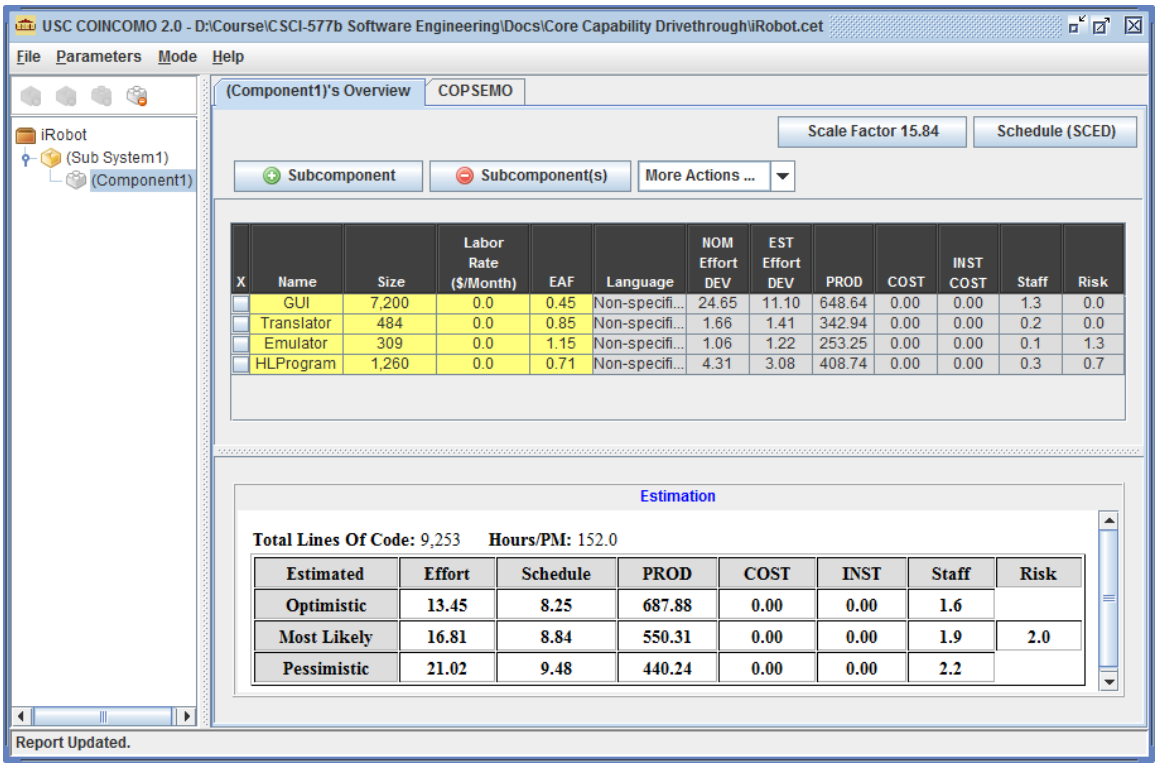
ACAP	LO	All of the team members have low experience related to this kind of programming and analyzing requirement of such field.
PCAP	LO	Developer of this module has little experience in such field.
PCON	HI	There will be no personnel change in this semester.
APEX	LO	Developers have low experience related to hardware programming.
LTEX	NOM	Low programming language and software tool experience
PLEX	NOM	Few team members have high platform experience
TOOL	LO	Developers haven't worked with Visual Studio before.
SITE	VHI	All team members and client are on campus; whereas the elementary school students are in the same city.
SCED	NOM	There is a nominal schedule constraint.

Table 14: COCOMOII Cost Driver for Controller

Cost Driver	Value	Rationale
RELY	HI	The controller must navigate all modules through the workflow correctly.
DATA	LO	There will not be a lot of data to be processed.
DOCU	NOM	Nominal Documentation is required for future maintenance
CPLX	LO	The logic is straightforward.
RUSE	LO	This module doesn't need to be reused.
TIME	NOM	Real-time response is not required.
STOR	NOM	Functional module, no storage is need for it.
PVOL	LO	The platforms are stable and do not require frequent upgrades.
ACAP	LO	All of the team members have low experience related to this kind of programming and analyzing requirement of such field.
PCAP	NOM	This module only need some basic programming logic which does not require high programming ability.
PCON	HI	There will be no personnel change in this semester.
APEX	NOM	Developers of this module have some basic knowledge about this module.
LTEX	NOM	Low programming language and software tool experience
PLEX	NOM	Few team members have high platform experience
TOOL	LO	Developers haven't worked with Visual Studio before.

SITE	VHI	All team members and client are on campus; whereas the elementary school students are in the same city.
SCED	NOM	There is a nominal schedule constraint.

Fig. 1: COCOMO II Analysis Result



According to COCOMO II Estimates for CSCI577,
N COCOMO II PM's / 1.67 = N CS577b team members needed
The most likely result is 16.81 / 1.67 = 10.1
Therefore, the number of people required is much greater than we currently have. This may explain why the product quality is not very satisfactory.

6. Iteration Plan

6.1 Plan

There are two iterations in the construction phase. The first iteration is for Core Capability which includes all four modules, testing, and quality assurance. The second one is Full Capability Iteration including improving products, process, and testing all features.

After the Core Capability Iteration, implement team and clients would check and record the accomplishments and take use of it as the input for the second Development Iteration. At the same time, they would also test the core capability and make use of it as the input for the next iteration. After the Core Capability Iteration, there is a milestone, CCD and after the Full Capability Iteration, there would be a milestone, TRR.

6.1.1 Capabilities to be implemented

Table 15: Construction iteration capabilities to be implemented

ID	Capability	Description	Priority Level	Iteration
OC-1	Navigation Programmability	The iRobot must be able to navigate in all the four directions	Must Have	2
OC-2	Sensor Programmability	The iRobot must sense various objects in its operating environment.	Must Have	1
OC-3	Loop and Wait Constructs	Instructions given to the iRobot will run in if-then, for and while loop constructs	Should Have	1
OC-4	Sounds and Lights Programmability	Robots must display light from the in-built LEDS and generate music	Should Have	1
OC-5	Demo Mode Programmability	The iRobot should run certain pre-programmed demo modes	Should Have	1
LOS-2	Detect and report ambiguous instructions	Detect and report ambiguous and conflicting instructions in an understandable way.	Must Have	1
LOS-3	Reasonable frequency of reading sensor data.	The iRobot must sense the data from sensors and react without any delays.	Must Have	1
LOS-4	Portability above Windows 7	The GUI should be easily ported on any Microsoft Windows Operating System above Windows 7.	Should Have	1

6.1.2 Capabilities to be tested

Table 16: Construction iteration capabilities to be tested

ID	Capability	Description	Priority Level	Iteration
OC-1	Navigation Programmability	The iRobot must be able to navigate in all the four directions	Must Have	1
OC-2	Sensor Programmability	The iRobot must sense various objects in its operating environment.	Must Have	1
OC-3	Loop and Wait Constructs	Instructions given to the iRobot will run in if-then, for and while loop constructs	Should Have	1

OC-4	Sounds and Lights Programmability	Robots must display light from the in-built LEDS and generate music	Should Have	1
OC-5	Demo Mode Programmability	The iRobot should run certain pre-programmed demo modes	Should Have	1
LOS-1	Seamless interoperability between GUI and compiler.	The system shall generate instructions for iRobot in C which is then later compiled for deployment on the microcontroller using the APIs of iRobot.	Must Have	1
LOS-2	Detect and report ambiguous instructions	Detect and report ambiguous and conflicting instructions in an understandable way.	Must Have	1
LOS-3	Reasonable frequency of reading sensor data.	The iRobot must sense the data from sensors and react without any delays.	Must Have	1
LOS-4	Portability above Windows 7	The GUI should be easily ported on any Microsoft Windows Operating System above Windows 7.	Should Have	1

6.1.3 Capabilities not to be tested

All capabilities will be tested after two iterations.

6.1.4 CCD Preparation Plans

Core Capability Drive through of 577b is scheduled on March 25, so we plan to rehearse the process two days before the formal one. From developers' end we will be delivering user guides, troubleshooting guides and also training the undergraduates. Before this, we would take undergraduates' feedback on the interface for sure. Undergraduates and Mission Science will carry out demonstrations in elementary schools for publicity and collect feedback in a way they see fit.

6.2.2 Core Capabilities Drive-Through Results

Positive feedback:

-Program meets expectation of what is required
-Program will make it easier for elementary students to become interested in robotics
-Program can be used as a framework for developing similar application for Arduino redbot

Improvements needed:
-Hover identification of functions
-Function pop up for parameter when dragged
-Default program and video

-Narrative accompanies documentation for future development

Risks:
-Not completing all features
-Interface not intuitive enough for elementary school students

For more details, see the concern logs.

6.3 Adherence to Plan

The iteration did not really adhere to the plan. Some features are still missing from CCD, like COM Port auto-detection and output information re-direction. Besides, navigation instructions have not been properly tested.
Development were delayed due to Operation System course which two team members takes. The solution is to accelerate the development process. At the same time, the team should increase the frequency of releases so that the client and users can test them in time.