

Life Cycle Plan (LCP)

Student Scheduling System Part II

Team 10

Bo Wang: Project Manager / Quality Focal Point / Implementation Team
Bohan Zheng: Prototyper / Builder / Implementation Team
Chenyang Bai: Feasibility Analyst / Operational Concept Engineer /
NDI NCS Evaluator / Implementation Team
Frank Varela: IIV&V / Shaper / Quality Focal Point
Rui Tong: Requirements Engineer / Operational Concept Engineer /
NDI / NCS Acquirer / Implementation Team
Shuai Wang: System/Software Architect / UML Modeler /
Implementation Team
Xiaoran Li: Life Cycle Planner / Tester / Implementation Team

04/03, 2014

Version History

Date	Author	Version	Changes made	Rationale
09/26/13	Xiaoran Li	1.0	<ul style="list-style-type: none"> Title Page Skills 	All the roles and skills of the members of the development team are identified
10/14/13	Xiaoran Li	2.0	<ul style="list-style-type: none"> Section 1-5 Added 	Drafts of documents for Foundation Commitment Package
10/16/13	Xiaoran Li	2.1	<ul style="list-style-type: none"> Bugs Fixed Typo Corrected Resource Estimation Re-analyzed 	Mistakes found after ARB
11/09/13	Xiaoran Li	2.2	<ul style="list-style-type: none"> Revised Section 2 – Milestones and Products 	CS577b schedule was posted
11/18/13	Xiaoran Li	2.3	<ul style="list-style-type: none"> Altered Section 3 – Responsibilities 	Adjustment based on previous performance and 577b tasks
11/22/13	Xiaoran Li	2.4	<ul style="list-style-type: none"> Modified Section 3 – Responsibilities 	Frank won't be taking 577b next semester
11/27/13	Xiaoran Li	2.5	<ul style="list-style-type: none"> Updated Section 3.2 – Skills 	Team members acquired skills during CS577a. Required skills are added for CS577b.
11/30/13	Xiaoran Li	2.6	<ul style="list-style-type: none"> Introduction Section 	The status is about to change
12/02/13	Xiaoran Li	2.7	<ul style="list-style-type: none"> Updated Section 5 – Resources 	Minor EAFs were out of date due to changed situation. Incremental prototype brought solid basis for more accurate code estimation.
12/07/13	Xiaoran Li	3.0	<ul style="list-style-type: none"> Section 6 Added Table of Contents Fixed 	Submission for final DC package after DCR ARB.
02/09/14	Xiaoran Li	4.0	<ul style="list-style-type: none"> Adjusted Section 3 – Responsibilities Refined document 	Personnel turnover and requirement volatility occurred and detected
02/10/14	Xiaoran Li	4.1	<ul style="list-style-type: none"> Adjust strategies for milestones Fixed typo 	Defects found

Date	Author	Version	Changes made	Rationale
03/24/14	Xiaoarn Li	4.2	<ul style="list-style-type: none"> • Section 6.2.1 Added 	For CCD Purpose
03/29/14	Xiaoran Li	4.3	<ul style="list-style-type: none"> • Section 6.2.2 Added 	CCD completed and feedback received from substitute client
04/03/14	Xiaoran Li	4.4	<ul style="list-style-type: none"> • Section 6.3 Added • Improved Layout 	Retrospect For IOC Purpose

Table of Contents

Life Cycle Plan (LCP)	i
Version History	ii
Table of Contents.....	iv
Table of Tables.....	v
Table of Figures	vi
1. Introduction.....	1
2. Milestones and Products.....	2
3. Responsibilities.....	4
3.1 Responsibilities by Phase	4
3.2 Skills	9
4. Approach	11
4.1 Monitoring and Control	11
4.2 Methods, Tools and Facilities	11
5. Resources	13
6. Iteration Plan.....	21
6.1 Plan.....	21
6.2 Iteration Assessment	25
6.3 Adherence to Plan	28

Table of Tables

<i>Table 1: Stakeholder's responsibilities.....</i>	<i>4</i>
<i>Table 2: Skills of Team Members</i>	<i>9</i>
<i>Table 3: Module lists and SLOC of each module</i>	<i>13</i>
<i>Table 4: COCOMOII Scale Driver.....</i>	<i>14</i>
<i>Table 5: COCOMOII Cost Driver – Data Entity Module</i>	<i>14</i>
<i>Table 6: COCOMOII Cost Driver – Data Access Module</i>	<i>15</i>
<i>Table 7: COCOMOII Cost Driver – Data Processing Module</i>	<i>16</i>
<i>Table 8: COCOMOII Cost Driver – Scheduling Solver Module</i>	<i>17</i>
<i>Table 9: COCOMOII Cost Driver – Student Side GUI Module</i>	<i>18</i>
<i>Table 10: COCOMOII Cost Driver – Administrative Side GUI Module.....</i>	<i>19</i>
<i>Table 11: Construction iteration capabilities to be implemented.....</i>	<i>22</i>
<i>Table 12: Construction iteration capabilities to be tested.....</i>	<i>23</i>
<i>Table 13: Capabilities implemented, tested, and results</i>	<i>26</i>
<i>Table 14: Core Capabilities shown and corresponding comments received.....</i>	<i>27</i>

Table of Figures

<i>Figure 1: COCOMO Estimation Result</i>	20
---	----

1. Introduction

The purpose of this Life Cycle Plan is to answer the most common questions about a project or activity: why?, whereas?, what?, when?, who?, where?, how?, and how much?

The status of this Life Cycle Plan currently is at Core Capabilities Drive-through Package, version number 4.3. This version will help in CCD purpose. The major changes from Rebaselined Foundation Phase are adding information for CCD preparation and from CCD feedback.

2. Milestones and Products

Exploration Phase

Duration: 09/09/13-09/27/13

Concept: Explore the current system and plan the project. Details include analyzing project feasibility, performing win-win negotiation, detailing system requirements, developing project operational concept, and identifying roles and skills of team members.

Deliverables: Valuation Commitment Package

Milestone: Valuation Commitment Review

Strategy: One Incremental Commitment Cycle, Explore Current System, Evaluate Skills and Responsibilities of team members

Valuation Phase

Duration: 09/28/13-10/21/13

Concept: Identify and prioritize win conditions, analyze project risks, prototype the system (mainly on user interfaces and algorithm), and detail project plan.

Deliverables: Foundation Commitment Package

Milestone: Architecture Review Board, Foundation Commitment Review

Strategy: One Incremental Commitment Cycle; Win-Win Negotiation; Prototype to Mitigate Risks; Design Level-0 Architecture;

Foundation Phase

Duration: 10/22/13-12/09/13

Concept: Assess project status, develop software architecture, perform UI negotiation, design database, develop algorithm, learn required technologies, and transit UI prototype from throwaway to incremental

Deliverables: Development Commitment Package, Database Schema, Algorithm Description

Milestone: Development Commitment Review

Strategy: One Incremental Commitment Cycle; Detail Project Plan; UI Iteration; Assess Architecture; Monitor Risks;

Rebaselined Development Phase

Duration: 01/13/14-02/12/14

Concept: Review works in previous phase, rebaseline prototype, refine requirement, detail CS577b plan, and complete test plan and test cases

Deliverables: Rebaselined Development Commitment Package

Milestone: Rebaselined Development Commitment Review Architecture Review Board

Strategy: One Incremental Commitment Cycle; Adapt Personnel turnover with Requirement Volatility; Refine Project Plan

Development Phase – Construction Iteration

Duration: 02/13/14-04/16/14

Concept: Manage project quality, detail test cases, and perform testing

Deliverables: Transition Commitment Package

Milestone: Core Capability Drive-through; Transition Readiness Review ARB

Strategy: Incremental Commitment Cycles; Collaborate on Construction; Focus on Unit Test, Integration Test, Alpha Test;

Development Phase – Transition Iteration

Duration: 04/17/14-04/30/14

Concept: Transit the system, provides training, get feedback from client/users, and fix bugs.

Deliverables: Final Deliverables including final source code, executable system package, user manual, and maintenance manual

Milestone: Operational Commitment Review

Strategy: Two Incremental Commitment Cycles; Transition along with Training; Completion of Performance Test, Beta Test;

Operation Phase

Duration: 04/30/13-05/05/13

Concept: Help maintain the system.

Deliverables: N/A

Milestone: Acceptance from client

Strategy: Assistance

3. Responsibilities

3.1 Responsibilities by Phase

Table 1: Stakeholder's responsibilities

Name: Bo Wang	
Role: Project Manager / Quality Focal Point / Implementation Team	
Exploration	Organize team meeting; Distribute roles for team members; Set up initial project plan; Correspond with client and DEN team member;
Valuation	Further project plan for valuation and foundation phase; Record project progress; Access Bugzilla repository for team activities; Correspond with client and DEN team member;
Foundations	Detail project plan for development phase; Quality assurance; Assess Bugzilla reports; Correspond with client and DEN team member; Develop incremental prototyping;
Development- Construction Iteration	Conduct system construction on administrative side; Participate in system construction for data entity and data controller; Verify & Validate Construction Status;
Development- Transition Iteration	Direct system transition; Deploy system; Verify & Validate Transition Status;

Name: Bohan Zheng	
Role: Prototyper / Builder / Implementation Team	
Exploration	Raise ideas on UI design;
Valuation	Analyze and Prioritize Capabilities to Prototype; Draft and develop prototype using Balsamiq Mockups; Assess system development environment for construction;
Foundations	Conduct UI iteration; Negotiate with client; Direct incremental prototyping; Assess components; Assist algorithm design;
Development- Construction Iteration	Direct system construction; Conduct system construction on student side; Participate in system construction for GUI; Assess Components; Tailor Components;
Development- Transition Iteration	Conduct system deployment; Fix bugs;

Name: Chenyang Bai	
Role: Feasibility Analyst / Operational Concept Engineer / NDI NCS Evaluator / Implementation Team	
Exploration	Explore Alternatives; Assess project plan; Perform risk assessment analysis; Design program model; Propose solutions to mitigate risks;
Valuation	Assess Feasibility Evidence; Perform risk assessment analysis; Perform cost analysis; Perform benefit analysis; Perform ROI analysis; Propose solutions to mitigate risks; Assess and evaluate NDI and NCS components Candidates;
Foundations	Assess Feasibility Evidence; Perform risk assessment analysis; Perform business analysis; Propose solutions to mitigate risks;
Development- Construction Iteration	Participate in system construction for data entity and data controller; Perform unit testing on administrative side;
Development- Transition Iteration	Participate in system deployment; Train end users;

Name: Frank Varela	
Role: IIV&V / Shaper / Quality Focal Point	
Exploration	Capture progress of win-win negotiation;
Valuation	Verify & validate win conditions; Track quality of project progress; Discover strong points of team members; Mine weak points of team members;
Foundations	Verify & validate project plan and documents; Fix defects in documents; Discover strong points of team members; Mine weak points of team members; Design and access test plan and test cases for win conditions; Construct traceability matrix;

Name: Rui Tong	
Role: Requirements Engineer / Operational Concept Engineer / UML Modeler / Implementation Team	
Exploration	Analyze current system; Identify and clarify system requirement; Explore alternatives;
Valuation	Capture and score win-conditions; Identify objectives constraints and priorities; Establish new operational concept; develop prototype using Balsamiq Mockups;
Foundations	Assess operational concept; Develop algorithm blueprint; Revise architecture artifacts;
Development- Construction Iteration	Participate in system construction; Coordinate construction between front-end and back-end;
Development- Transition Iteration	Participate in manual composition; Train end users;

Name: Shuai Wang	
Role: System/Software Architect / Implementation Team	
Exploration	Assess current system architecture;
Valuation	Build UML Model; Define and specify architecture styles, patterns and frameworks;
Foundations	Develop algorithm blueprint; Assess database schema;
Development- Construction Iteration	Perform algorithm realization; Participate in system construction for GUI;
Development- Transition Iteration	Participate in manual composition; Train end users;

Name: Xiaoran Li	
Role: Life Cycle Planner / Tester / Implementation Team	
Exploration	Identify roles and skills of team member;
Valuation	Assess project plan for CS577a; Identify responsibilities of team member; Identify milestones and products; Identify approaches for project; Estimate software cost; Correspond with client and DEN team member;
Foundations	Assess project plan for CS577b; Estimate software cost; Control system progress; Correspond with client and DEN team member;
Development- Construction Iteration	Develop and further test cases; Perform tests; Supervise system construction progress; Ensure the quality of the project; Review system and provide feedback to the development team;
Development- Transition Iteration	Perform final testing on target system; Participate in manual composition; Ensure the Quality of the Project

Name: David Klappholz	
Role: Client	
Exploration	Analyze Current System; Participate in the WinWin negotiation;
Valuation	Identify objectives, constraints and priorities; Participate in weekly meeting;
Foundations	Provide feedback; Assess prototype and components; Participate in weekly meeting; Participate in commitment review architecture review board;
Development- Construction Iteration	Provide feedback; Participate in commitment review architecture review board;
Development- Transition Iteration	Provide server for transition; Provide support during transition;

Name: Alexey	
Role: Substitute Client / Former Project Participant;	
Exploration	Help analyze current system;
Valuation	Explain current system constraints;
Foundations	Assess the quality of prototype;
Development- Construction Iteration	Help the Team with Play Framework, if possible;
Development- Transition Iteration	Help the Team with Cloud Server, if possible;

Name: Administrator	
Role: Maintainer	
Exploration	Be Represented by Client;
Valuation	Be Represented by Client;
Foundations	Be Represented by Client;
Development- Construction Iteration	
Development- Transition Iteration	Participate in Training; Provide feedback;

Name: Student	
Role: User	
Exploration	Be Represented by Client;
Valuation	Be Represented by Client;
Foundations	Be Represented by Client;
Development-Construction Iteration	Be Represented by Client;
Development-Transition Iteration	Participate in Beta Test; Participate in Training; Provide feedback;

3.2 Skills

Table 2: Skills of Team Members

Team members	Role	Skills
Bo Wang	Project Manager / Quality Focal Point / Implementation Team	Current skills: Project Management, Bugzilla, WinBook, MS Project, C/C++, C#, Java, WinCE, QT, Visual Studio, EA, UX, Play Framework Required skills: Software Defect Tracking, JUnit
Bohan Zheng	Prototyper / Builder / Implementation Team	Current skills: HTML, Javascript, CSS, PHP, MySQL, Java, Eclipse, COCOMO II, UI Design, Play Framework Required skills: Linux/Unix, Defects Fixing, Components Tailoring, JUnit
Chenyang Bai	Feasibility Analyst / Operational Concept Engineer / NDI NCS Evaluator / Implementation Team	Current skills: Java, Eclipse, PHP, Apache, Business Case Analysis, Risks Analysis, Play Framework, Objectives/Constraints/Priorities Identification Required skills: JUnit

Team members	Role	Skills
Frank Varela	IIV&V / Shaper / Quality Focal Point	Current skills: Java, HTML, XML, CSS, Bugzilla, WinBook, Report Generation, Software Defect Tracking, Algorithm Benchmarking Required skills: Play Framework
Rui Tong	Requirements Engineer / Operational Concept Engineer / UML Modeler / Implementation Team	Current skills: MySQL, Apache, PHP, C/C++, C#, Java, Perl, Eclipse, Dreamweaver, WinBook, Win- conditions Capturing/Scoring, Win-Win Negotiation, Algorithm Analysis/Design, Objectives/Constraints/Priorities Identification, UML Modeling, Play Framework Required skills: JUnit
Shuai Wang	System/Software Architect / UML Modeler / Implementation Team	Current skills: C++, MFC, Java, HTML, CSS, Javascript, JSP, MySQL, Eclipse, Dreamweaver, Visual Studio, UML Modeling, System/Software Analysis, Play Framework Required skills: JUnit
Xiaoran Li	Life Cycle Planner / Tester / Implementation Team	Current skills: C/C++, Java, Linux/Unix, SQL, Cost/Benefit Analysis, Effort/Schedule Estimate, Team Organizing, Project Planning, JUnit, Test Case Implementation, Play Framework Required skills: Documentation Composition

Personnel Turnover:

Only Frank Varela left the team. Other stakeholders remain who they are.

4. Approach

4.1 Monitoring and Control

- Progress Report
- Project Plan
- Effort Report
- Bugzilla
- Winbook.

4.1.1 Closed Loop Feedback Control

- Using Bugzilla to Assign Tasks;
- Having Frequent Meetings to Collaborate and Share Ideas;
- Communicating via Facebook Private Group and Email;

4.1.2 Reviews

- Commitment Reviews
- Schedule Reviews
- Task Reviews;

4.2 Methods, Tools and Facilities

Tools	Usage	Provider
Microsoft Office Project	Generates Project plans	Microsoft
Bugzilla	Distributes assignments among team members	USC CS577
Trello	Creates Program Model	Fog Creek
WinBook	Helps reach win conditions during negotiation.	
Planning Poker	Provides an online environment for team for playing planning poker to prioritize the ease of realization.	Mountain Goat Software, LLC
Balsamiq Mockups	Provides templates and components for prototyping user interface	Balsamiq
Visual Paradigm	Helps create UML by providing templates	Visual Paradigm International
COINCOMOII	Estimate Project Cost and Schedule	USC CS577
JAVA SDK	Development Kit for Java	Oracle

Tools	Usage	Provider
Eclipse	Integrated Development Environment	The Eclipse Foundation
Play framework	A Website Development Framework Supports Java	Play
Apache Web Server	Web Server	The Apache Software Foundation
MySQL	Lightweight Database	Oracle

5. Resources

Information for estimating the software cost is listed as follow:

- Numbers of modules: 6 (listed in Table 2)
- Programming language used: Java
- Budget information: No budget
- Construction duration: 12 weeks (10 from 577b, 2 from Winter Recess)
- Available staff: 6 team members
- Probable working hours/week/person: 15 hours (2.5 hours/day at 6 days)

Table 3: Module lists and SLOC of each module

No.	Module Name	Brief Description	SLOC	REVL
1	Data Entity (Database)	Preserves essential information to support the whole system	66	5%
2	Data Access (Model)	Provides interface to interact with database	820	5%
3	Data Processing (Controller)	Process data from Model for multiple needs requested by View and Other Controller	600	5%
4	Scheduling Solver (Controller)	Generates study plans under specified constraints	1500	15%
5	Student Side GUI (View)	Friendly user interface of student side	1780	20%
6	Administrative Side GUI (View)	Friendly user interface of administrative side	2240	5%

Table 4: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	High	This system is generally familiar to the kind of management system that we have done before, except the algorithm part.
FLEX	Low	The schedule is rigorous while some of the requirements are occasionally flexible.
RESL	Very High	Critical risks were identified and a prototype was made.
TEAM	High	The implement team is cooperative and willing to pay extra effort due to difficulties. However, the client is remote to touch with 3 hours of time difference.
PMAT	Very High	This is the Part II of building the system.

Table 5: COCOMOII Cost Driver – Data Entity Module

Cost Driver	Value	Rationale
Product		
RELY	Low	Easily recoverable with backup data.
DATA	High	Needs to be tested with amount of course records.
CPLX	Low	Just a few tables in database
RUSE	Low	No needs to reuse this module
DOCU	Very Low	Major description has been composed for this module
Platform		
TIME	Nominal	No time constraints for this module
STOR	Nominal	No storage constraints for this module
PVOL	Low	Low frequency to have platform changed.
Personnel		
ACAP	Very High	All team members have good analysis on database
PCAP	Very High	All team members are good programmers
PCON	Very High	All members will participate in CS577b
APEX	High	3 years experience on average of database management
PLEX	Nominal	1 years experience of server platform
LTEX	High	3 years experience on average of SQL and MySQL
Project		
TOOL	Nominal	Uses MySQL to manage the database
SITE	Extra High	Implementation team collaborates at same location
Schedule Compression		
SCED	Nominal	Basically, the schedule is fixed for 2 semesters

Table 6: COCOMOII Cost Driver – Data Access Module

Cost Driver	Value	Rationale
Product		
RELY	Very Low	Slight inconvenience and easily recoverable
DATA	Low	Needs a few testing data.
CPLX	Low	Data Access Objects
RUSE	Low	No needs to reuse this module
DOCU	Very Low	Only needs to be covered a little bit for this module
Platform		
TIME	Nominal	No time constraints for this module
STOR	Nominal	No storage constraints for this module
PVOL	Low	Low frequency to have platform changed.
Personnel		
ACAP	Very High	90% of team members have good analysis on data access
PCAP	Very High	All team members are good programmers
PCON	Very High	All members will participate in CS577b
APEX	Very Low	No experience of Play Framework
PLEX	Nominal	1 years experience of server platform
LTEX	High	3 years experience on average of Eclipse
Project		
TOOL	High	Uses Eclipse and MySQL for interaction
SITE	Extra High	Implementation team collaborates at same location
Schedule Compression		
SCED	Nominal	Basically, the schedule is fixed for 2 semesters

Table 7: COCOMOII Cost Driver – Data Processing Module

Cost Driver	Value	Rationale
Product		
RELY	Nominal	Easily recoverable
DATA	Low	Needs a few testing data.
CPLX	Nominal	This module adapts data structures between modules
RUSE	Low	No needs to reuse this module
DOCU	Very Low	Only needs to be covered a little bit for this module
Platform		
TIME	Nominal	No time constraints for this module
STOR	Nominal	No storage constraints for this module
PVOL	Low	Low frequency to have platform changed.
Personnel		
ACAP	Nominal	60% of team members are good analyst in this field
PCAP	Very High	All team members are good programmers
PCON	Very High	All members will participate in CS577b
APEX	High	3 years experience on average of Java application
PLEX	Nominal	1 years experience of server platform
LTEX	High	3 years experience on average of Eclipse
Project		
TOOL	Very High	Uses Eclipse for development
SITE	Extra High	Implementation team collaborates at same location
Schedule Compression		
SCED	Nominal	Basically, the schedule is fixed for 2 semesters

Table 8: COCOMOII Cost Driver – Scheduling Solver Module

Cost Driver	Value	Rationale
Product		
RELY	High	Chance existent to cause financial loss
DATA	High	Needs to be tested with amount of course records.
CPLX	Very High	Complex algorithm with numerous required constraints.
RUSE	Low	No needs to reuse this module
DOCU	High	Excessive for life-cycle needs.
Platform		
TIME	Extra High	Consumes most available execution time.
STOR	Nominal	No storage constraints for this module
PVOL	Low	Low frequency to have platform changed.
Personnel		
ACAP	Nominal	60% of team members are algorithm analyst
PCAP	Very High	All team members are good programmers
PCON	Very High	All members will participate in CS577b
APEX	High	3 years experience on average of Java application
PLEX	Nominal	1 years experience of server platform
LTEX	High	3 years experience on average of Eclipse
Project		
TOOL	Very High	Uses Eclipse for development.
SITE	Extra High	Implementation team collaborates at same location.
Schedule Compression		
SCED	Nominal	Basically, the schedule is fixed for 2 semesters

Table 9: COCOMOII Cost Driver – Student Side GUI Module

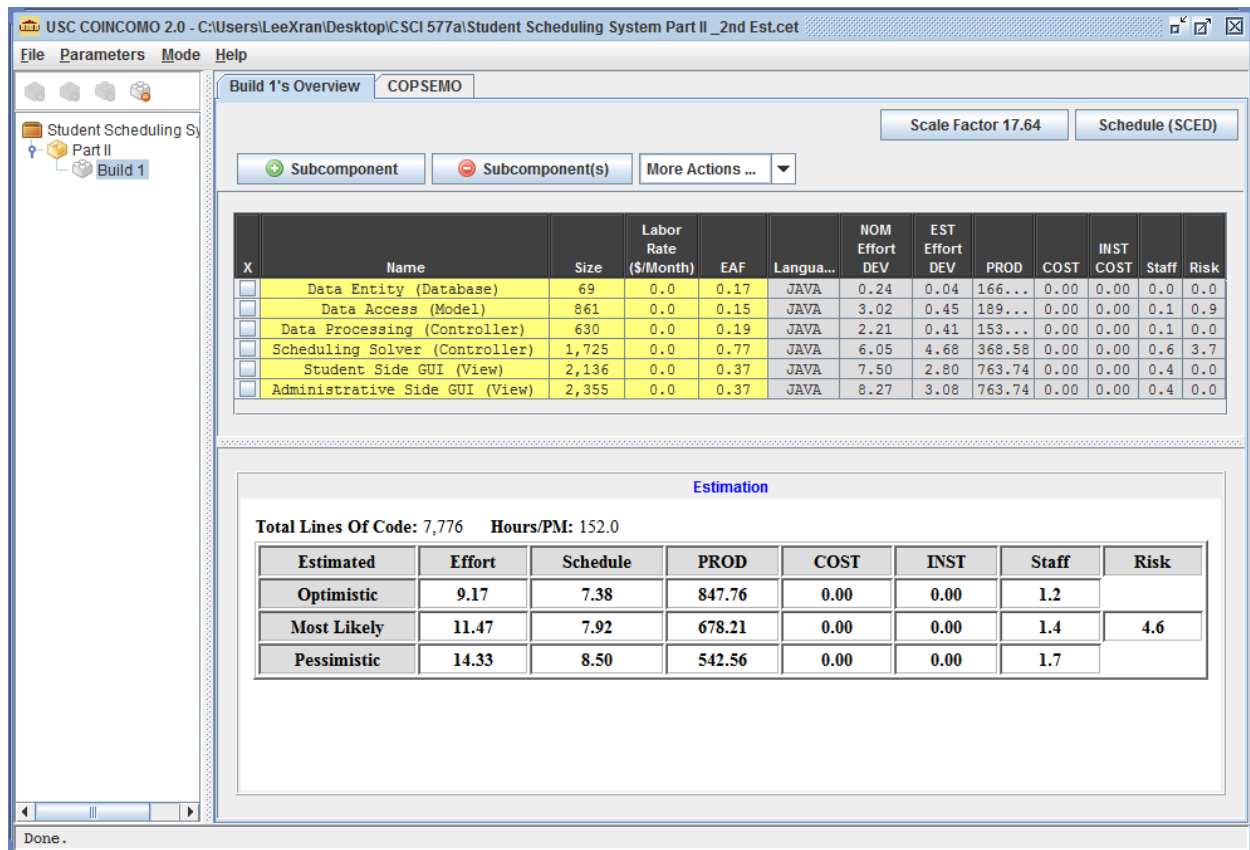
Cost Driver	Value	Rationale
Product		
RELY	Nominal	Easily recoverable
DATA	Low	Needs a few testing data.
CPLX	High	Delicate UI with elegant layout based on dynamic content
RUSE	High	Major part could be reused within GUI
DOCU	High	Excessive for life-cycle needs.
Platform		
TIME	Nominal	No time constraints for this module
STOR	Nominal	No storage constraints for this module
PVOL	Low	Low frequency to have platform changed.
Personnel		
ACAP	High	80% of team members are webpage analyst
PCAP	Very High	All team members are good programmers
PCON	Very High	All members will participate in CS577b
APEX	Nominal	1 years experience on webpage UI design
PLEX	Nominal	1 years experience of server platform
LTEX	High	3 years experience on average of Eclipse
Project		
TOOL	High	Uses Eclipse for webpage UI realization
SITE	Extra High	Implementation team collaborates at same location
Schedule Compression		
SCED	Nominal	Basically, the schedule is fixed for 2 semesters

Table 10: COCOMOII Cost Driver – Administrative Side GUI Module

Cost Driver	Value	Rationale
Product		
RELY	Nominal	Easily recoverable
DATA	Low	Needs a few testing data.
CPLX	High	Delicate UI with elegant layout based on dynamic content
RUSE	High	Major part could be reused within GUI
DOCU	High	Excessive for life-cycle needs.
Platform		
TIME	Nominal	No time constraints for this module
STOR	Nominal	No storage constraints for this module
PVOL	Low	Low frequency to have platform changed.
Personnel		
ACAP	High	80% of team members are webpage analyst
PCAP	Very High	All team members are good programmers
PCON	Very High	All members will participate in CS577b
APEX	Nominal	1 years experience on webpage UI design
PLEX	Nominal	1 years experience of server platform
LTEX	High	3 years experience on average of Eclipse
Project		
TOOL	High	Uses Eclipse for webpage UI realization
SITE	Extra High	Implementation team collaborates at same location
Schedule Compression		
SCED	Nominal	Basically, the schedule is fixed for 2 semesters

The screenshot of result estimated by COINCOMO based on above scale factors and cost drivers is pasted below.

Figure 1: COCOMO Estimation Result



- According to COINCOMO, the estimated total effort this project requires is:
11.47 person months, most likely;
- According to the given available resources this team has, single member effort is:
(10 weeks) (15 hours/week) = 150 hours
- According to the assumption, estimated COCOMO II person month is
(100 hours) (72%) = 72 hours;
- Since:
(2.08) (72 hours) = 150 hours;
(2.08) (6 members) = 12.48 person months > 11.47 person months;
Construction had been started in Nov 2013 and will be continued in winter recess;
- Thus:
This project is able to be finished in time.

6. Iteration Plan

6.1 Plan

This iteration plan mainly focuses on construction iterations in development phase. The student scheduling system website project consists of two major parts: administrative side, and, student side. In details, according to the design, administrative side can be further divided into five modules – Course Management module, Course Group management module, Requirement Management module, Degree Management module, and Help module, while student side encompasses Scheduling Solver module as its core capability. Since administrative side serves as the foundation of the whole system, the construction of this system will start from the development of administrative side.

- Construction – 1st Iteration
Duration: 12/27/14 – 01/12/14
Milestone: The End of Winter Recess (Right before Rebaseline Foundation Phase)
Activity: Set up system database. Develop Course Management module and Course Group Management module. Perform unit testing.
- Construction – 2nd Iteration
Duration: 01/13/14 – 01/27/14
Milestone: Rebaseline Development Commitment Review
Activity: Develop Requirement Management module, Degree Management module, and Authentication module. Perform unit testing and integration testing.
- Construction – 3rd Iteration
Duration: 02/11/14 – 03/25/14
Milestone: Core Capability Drivethrough
Activity: Refine Administrative Site. Develop Scheduling Solver module and other non-critical parts of student side. Perform unit testing, integration testing, and system alpha testing.

6.1.1 Capabilities to be implemented

Table 11: Construction iteration capabilities to be implemented

ID	Capability	Description	Priority	Iteration
1	Database	Contains information that supports the whole scheduling system	1	1
2	Test Cases	Database correctness checking	2	1
3	Course Management	Create, modify, delete, and view courses that would be provided to students.	3	1
4	Course Group Management	Create, modify, delete, and view course group that contains courses.	4	1
5	Requirement Management	Create, modify, delete, and view requirements that are needed to be satisfied for pursuing particular degree.	1	2
5	Degree Management	Create, modify, delete, and view degree program that are set by administration.	2	2
6	Test Cases	For UI of administrative side	3	2
7	Test Cases	For back-end of Administrative side	1	3
8	Student Side Skeleton	UI of student side for students to interact with the system and reduce complexity	1	3
9	Scheduling Solver	Core capability of system. Automated scheduling of study plan based on constraints provided by school and desire given by students.	1	3
10	Authentication	Allow administrators and directors to do login on administrative side	3	3
11	Administrative Side	Refine administrative side.	2	3
12	Test Cases	For student side	3	3

6.1.2 Capabilities to be tested

Table 12: Construction iteration capabilities to be tested

ID	Capability	Description	Priority	Iteration
1	Database	Database should be built according to database schema. All changes made by program should be reflected correctly onto table and filed in the database.	1	1
2	Add/Modify Course	An administrator should be able to add/modify a course.	2	1
3	Create/Modify Course Groups	An administrator should be able to create/modify course groups by selecting the desired courses.	2	1
4	Create/Modify Requirement	An administrator should able to create/modify a requirement for course groups.	1	2
5	Create/Modify Degree Programs	An administrator should be able to create/modify a degree program.	1	2
6	Co-/Pre-requisite Hints	An administrator should be provided with hints as to how corequisite and prerequisite information is populated	1	3
7	View Study Plan (Level of Service)	Students should be able to view the generated study plan within an acceptable amount of time.	1	3
8	Manual Course Selection	Students should be able to select individual courses for their desired degree program.	1	3
9	Semester Criteria	Students should be able to specify their desired semesters and units per semester for their degree program.	2	3
10	Insufficient Course Selections	Students should be able to be notified when the selected courses are insufficient for a degree program.	2	3
11	Auto Course Selection	Students should be able to request the system to automatically selected courses for their desired degree program.	3	3
12	Required Relaxation	Students should be able to be notified when their selected courses and semester criteria must be relaxed in order for a valid study plan to be generated by the system in an acceptable amount of time.	3	3
13	User Login	Administrator and director users should be able to login and see the correct information based on the user type.	4	3

6.1.3 Capabilities not to be tested

- Capability: Layout for Administrative Side
Initially Implemented Iteration: 1
Reason: Layout would be finally refined in 3rd iteration.
- Capability: User Login for Administrative Side
Initially Implemented Iteration: 3
Reason: Authentication would be a dummy feature until 3rd iteration completion.
- Capability: Hints for Users
Initially Implemented Iteration: 1, 2, 3
Reason: Hints contents would be added right before CCD and revised later after CCD.
- Capability: Scheduling Response Time Consumption
Initially Implemented Iteration: 3
Reason: Results won't be accurate until deployed onto SIT server with official data. (Beta Testing after Transition)
- Capability: Resistance to Pressure
Initially Implemented Iteration: 3
Reason: Results won't be accurate until deployed onto SIT server. (Beta Testing after Transition)

6.1.4 CCD Preparation Plans

- Attendant Stakeholders
Client: Professor David Klappholz (Play Roles as Admin and Student)
Development Team: Team 10
- Methods for Drive-through
Video Chatting using Google Hangout
Remote Control by Client using Join.me
- Supporting Materials and Data
Training Materials: User Manual and Tutorial
Test Data: Part of Real Data from SIT
- Capabilities and Procedures to be Drive-through
As Admin: Adding Course -> Adding Course Group -> Adding Simple Requirement ->
Adding Requirement -> Adding Degree Program
As Student: Choosing Degree Program -> Manually Selecting Courses -> Specifying
Semester Criteria
- Plan B
Using SQL Scripts to Restore Database with Real Data

- Target Information for Risk Mitigation Plan
Fluency of Operations, User Feedback and Experience, Potential Defects

6.2 Iteration Assessment

6.2.1 Capabilities Implemented, Tested, and Results

Capabilities that are implemented are listed below. Other capabilities such as Auto Course Selection, Semester Criteria, etc

Table 13: Capabilities implemented, tested, and results

ID	Capability	Test Case	Test Results	Comments
1	Database: Contains information that supports the whole scheduling system and should be built according to database schema. All changes made by program should be reflected correctly onto table and filed in the database.	TC-02-01, TC-02-03, TC-02-04, TC-02-05, TC-02-06	Pass	Perfectly Matches
2	Course Management: As an admin, I can create, edit, delete, and view courses that would be provided to students	TC-02-01	Pass	Except for Robustness in Validating Pre-/Co-requisites
3	Course Group Management: As an admin, I can add, edit, delete, and view course group that contains courses	TC-02-03	Pass	Except for Robustness in Title Validation
4	Requirement Management: As an admin, I can add, edit, delete, and view requirements that are needed to be satisfied for pursuing particular degree	TC-02-04, TC-02-06	Pass	Except for Robustness in Inputs Validation
5	Degree Program Management: As an admin, I can add, edit, delete, and view degree program that are set by administration	TC-02-05	Pass	Except for Robustness in Validating Degree Title
6	Pre-/Co-requisite Hints: As an admin, I should be provided with hints as to how co-requisite and prerequisite information is populated	TC-02-02	Pass	Waiting for Client's Feedback
7	Manual Course Selection: As a student, I should be able to select individual courses for my desired degree program	TC-01-01	Pass	Except for Robustness in Choosing courses that are already taken by the students

6.2.2 Core Capabilities Drive-Through Results

Since our client got sick right before our CCD session, the results that we recorded includes the feedback from our TA – Daniel – acting as substitute client, as his valuable opinions can be viewed as the feedback from the perspectives of students and administrators from Stevens Institute of Technology, who will use this system in the future.

All implemented capabilities have been shown and discussed during the session; the main focus is on those critical ones listed in Section 6.2.1. The scenarios we created are: 1) letting the client act as administrator who manipulates data on administrative side of the system to set courses, course group, simple requirement, requirement (complex), and degree program for students; 2) letting the client act as student who chooses several desired courses under the corresponding degree program on student side of the system in order to have a constraints-satisfied study plan generated by the system. Both scenarios were supported by preprinted user manual and tutorial.

The positive feedback includes the following:

- Well functional system under both scenarios
- Good robustness performance of core capabilities
- Concise user interface for human-computer interaction
- Impressive time consumption for study plan generation

Improvements and suggestions contain the following:

- For administrative side
 - To list all courses in sorted order on *Course Page*
 - To remove assigned courses from course list on the left side on *Course Group Page*
 - To make all buttons look like buttons rather than labels or plain text (i.e., a button should have clickable style)
- For student side, on step 1 – course selection
 - To mark odd rows and even rows in course list with two colors for a more clear view
 - To add mouse-over-event-hints for “+” and “x” button beside courses
 - To mark chosen courses in course list for a more clear view
 - To move “Next” button out of course bin since it serves the whole page
 - To allow “Auto” button clickable more than once
 - To avoid confusion between the “x” button in course bin and the one in course group
 - To use “-” button for collapsing course group instead of using “x” button

For changes to-be considered and updated risks, we have to wait for our client’s recovery and for his earliest available to schedule a meeting.

Details of core capabilities shown and corresponding comments received during CCD are described below

Table 14: Core Capabilities shown and corresponding comments received

Capability	Comments
Database	No errors found
Course Management	No defects found, except for tiny UI improvements
Course Group Management	No defects found, except for tiny UI improvements
Requirement Management	No defects found
Degree Program Management	No defects found
Pre-/Co-requisite Hints	Functional and understandable
Manual Course Selection	No defects found, except for tiny UI improvements

6.3 Adherence to Plan

In general, our team sticks with our initial project plan for this Spring 2014 semester. The reason is quite objective: all required features of the system can be viewed as core capabilities. The target goal of this system is study plan generation, which closely relies on degree program management as the infrastructure. Besides, the algorithm that used to solve this study plan generation problem lives throughout the whole system, the generation feature and management feature, thus, can only be broken down into step-by-step capabilities, meaning that all these smaller parts should be viewed as a whole. As a result, we have to finish construction in advance; at least 95% of it should be done before the Core Capabilities Drive-through session.

Our project plan includes 3 construction iterations. The first iteration started from the New Year to the beginning of this semester, in which we finished two components – Course Management and Course Group Management – of the administrative side as planned. The second iteration began after and stopped right before the Rebaselined Development Commitment Review session, where the administrative side was almost done – leading our progress to 45% of done as intended. The third iteration – served as the last one – then commenced and the study plan was able to be generated at our CCD session, as expected. In terms of total, the project is on schedule.

The only uncertainty in the development phase is about the possibility of one feature that was added in the middle of the project. The reason that causes this issue is that, due to the illness of our client, in the week of the CCD session, followed by a two-week busy travel, we have to wait until Apr. 9 to have a meeting for discussing the details, and the feedback of the system as well. This is also the reason that we consider extending the construction phase and postpone the transition phase until Transition Readiness Review. It won't hurt too much, however, since this feature is considered as a win-condition.

Perhaps in the future, we should consider scalability and compatibility in advance, although it would require sophisticated experience in order to think non-functional properties in a thorough way. And it would help a lot if we asked for an authorized substitute client at the beginning of this project in case the chief client is not available against any accident or emergency.