Life Cycle Plan (LCP)

Student Scheduling System

TEAM #06

Douglass Kinnes: Project Manager, Quality Focal Point, Implementation Team member

Alexey Tregubov: System Architect, UML Modeler, Implementation Team member

Mihir Daptardar : Operational Concept Engineer, Quality Focal Point, Tester

Ihsan Tolga : Life Cycle Planner, Feasibility Analyst, Implementation Team member

Simone Lojeck : IV&V, Quality Focal Point, Shaper

May, 07, 2013

Version History

Date	Author	Version	Changes made	Rationale
09/30/12	Ihsan Tolga	1.0	Initial draft as project's Life Cycle Plan	Initial draft including Skills section and definitions
10/13/12	Ihsan Tolga	1.1	• Bugs fixed • Section 1 and 3.3 revised. Parts added. • Responsibilities added. • Version history, table descriptions added. • Footer, header defects removed.	Revised skills updated. Complies with ICSM EPG standard.
10/21/2012	Ihsan Tolga	1.12	• Formatting • Approach descriptions added.	Draft approach descriptions (as bullpens) added.
10/22/2012	Ihsan Tolga	2.00	• Milestones and Products section updated. • Current project deliverables added. • Responsibilities updated. • Approaches section added. • Resources section and estimations added.	COCOMO estimation and the results were added. Sections updated for FC Package. Future responsibilities and deliverables are current in this version.
10/24/2012	Ihsan Tolga	2.1	• Captions, footers updated. • COTIPMO screen added.	Footers and table captions are compatible to the document.
10/28/2012	Ihsan Tolga	2.2	• Skills revised, bugs resolved. • Grammatical errors treated.	Needed skills revised, tester role added.
10/30/2012	Ihsan Tolga	2.3	• Captions revised. Formatting.	Current for FCP.
11/03/2012	Ihsan Tolga	2.4	• Defects fixed with respect to ARB comments.	Fixes according to ARB review.
11/12/2012	Ihsan Tolga	2.5	• Defects resolved.	Bugs were fixed and tables were updated.
11/25/2012	Ihsan Tolga	3.0	• Defects resolved with respect to the comments. New skills added. Estimations updated. Life cycle phases updated.	Bugs were resolved. New items added and current ones revised according to the evaluation and ARB comments.
12/04/2012	Ihsan Tolga	3.1	• Bugs fixed. Section 6 added. Estimations revised.	Bugs were resolved. Iteration plans were added. Cost drivers and estimations were updated with respect to the comments.
12/13/2012	Ihsan Tolga	3.2	• Bugs fixed. Tables were revised with respect to the comments of DCR ARB.	Bugs fixed. Tables were updated, improved with respect to the DCR ARB comments.
02/08/2013	Ihsan Tolga	4.0	• Bugs fixed. New phases added. Personnel lists revised. Products updated.	Milestones and new artifacts defined. Personnel turnover updated. Iterations updated.
02/20/2013	Ihsan Tolga	4.1	• Iteration tables revised with respect to the ARB comments.	Tables updated related to upcoming iterations.
02/27/2013	Ihsan Tolga	4.2	• Bugs fixed.	Bugs removed in Section 2.
03/27/2013	Ihsan Tolga	4.3	• CCD related revisions made.	COTIPMO and iteration plans revised.
04/08/2013	Ihsan Tolga	4.5	• CCD feedback added.	Feedback has taken from the client and recorded, added in Section 6.
04/30/2013	Ihsan Tolga	5.0	• Revisions are made before system transition.	Revisions and minor bug fixes before system transition.
05/07/2013	Ihsan Tolga	5.1	• Small bugs fixed.	Minor bug fixes.

Table of Contents

Life Cycle Plan (LCP) 1
Table of Contents 3
Table of Tables 4
Table of Figures 5
1. Introduction 6
1.1 Purpose of the LCP 6
1.2 Status of the LCP 6
1.3 Assumptions 7
2. Milestones and Products 8
2.1 Overall Strategy 8
2.2 Project Deliverables 10
3. Responsibilities 15
3.1 Project-specific stakeholder’s responsibilities 15
3.2 Responsibilities by Phase 15
3.3 Skills 17
4. Approach 18
4.1 Monitoring and Control 18
4.2 Methods, Tools and Facilities 19
5. Resources 20
6. Iteration Plan 29
6.1 Plan 29
6.2 Iteration Assessment 32
6.3 Adherence to Plan 36

Table of Tables

Table 1: Artifacts Deliverables in Exploration Phase 10
Table 2: Artifacts – Deliverables in Valuation Phase 11
Table 3: Artifacts – Deliverables in Foundations Phase 12
Table 4: Artifacts – Deliverables in Rebaselined Foundations Phase 13
Table 5: Artifacts – Deliverables in Development Phase - Construction 13
Table 6: Responsibilities in Each Phase 15
Table 7: Skills of Team Members Related to Their Roles 17
Table 8: Skills Required fir CS577b in case of Personnel Turnover 17
Table 9: Tools Used in the Project 19
Table 10: COCOMOII Project Scale Drivers 21
Table 11: COCOMOII Cost Drivers for Database Module 22
Table 12: COCOMOII Cost Drivers for Constraint Processor Module 23
Table 13: COCOMOII Cost Drivers for Study Plan Construction Module 24
Table 14: COCOMOII Cost Drivers for Degree Requirements Construction Module 25
Table 15: COCOMOII Cost Drivers for View Control Module 26
Table 16: COCOMOII Cost Drivers for Authentication Module 27
Table 17: COCOMOII Cost Drivers for Solver Module 28
Table 18: Construction Iteration Capabilities to be Implemented 29
Table 19: Construction Iteration Capabilities to Be Tested........ 30
Table 20: Construction Iteration Capabilities Not to Be Tested 31
Table 21: Capabilities Implemented, Tested, and Results 32
Table 22: CCD Results and Client Feedbacks 33
Table 23: Explored Short Term Risks for CCD........ 36

Table of Figures

Figure 1: COCOMO Analysis.. 20
Figure 2: COCOMO Analysis Results.. 20
Figure 3: COTIPMO Estimation Screen 21

1. Introduction

1.1 Purpose of the LCP

The purpose of this Life Cycle Plan is to set a basis for project, to clarify borders and activities throughout the whole process and to provide answers about the project: Why? What? When? Who? Where? Whereas? How? How much? It also shows the team members' skills and responsibilities related to project goals.

- Given by the incremental spiral commitment model (and architected agile model selected as the model for Student Scheduling System project); every iteration and phase have to planned and identified before the actions to be taken. LCP serves as guideline for advance through the project.
- For each of the iterations, team members are responsible for their artifacts and documents related to their project roles. These responsibilities are defined in Life Cycle Plan.
- Most recent estimated effort and schedule performed with COCOMO tool are stated in the LCP.

1.2 Status of the LCP

The current status of the LCP is at 3rd Initial Operational Capability Package version number 5.0 and serves as the artifact before transition of the system. This version contains all of revised tables, fixed defect with respect to the evaluations of ARB sessions, revised iteration tables and resolved bugs. Updated skills are also included in this version.

For the IOC3 Package, milestones, iterations and products were inspected again. Development Construction/Transition Phase responsibilities & skills were also examined for integrity. Estimations were re-calculated according to the current state of the project. Most current COTIPMO screenshot was added under its related section. Iteration section was revised and the tables are revised and updated with respect to the comments during the ARB session and package evaluations.

Several minor defects related to some typos and format concerns are removed. In addition, current artifacts are updated with respect to the current state of the project. Cost drivers' integrity is checked with graders for the new version.

Before the transition of Student Scheduling System; required sections which were already in document in advance are revised and examined finally with respect to the comments during the TRR ARB.

1.3 Assumptions

- The duration of the project is 24 weeks, which are 12 weeks in Fall 2012 semester and 12 weeks in Spring 2013 semester.
- There are six members in the developer team, five on-campus and one off-campus students. Yet one member has decided not to take CS577b course.
- Architected Agile Model is being used in the project.
- Instructional Incremental Commitment Spiral Model – Software Electronic Progress Guide is being used as the guideline for the project.
- At least one client meeting is held for each week.
- Client will support transition site and maintenance jobs after project transition.

2. Milestones and Products

2.1 Overall Strategy

Student Scheduling System is following Architected Agile as the ICSM process model, we have chosen this model with respect to that there are not many open-source scheduling software on the market and the required core capabilities are rather specific which do not let us to use COTS.

Exploration Phase

Duration: 09/12/2012 – 10/03/2012

Concept: In this phase, the team specifies the initial scope of the project; identifies operational concepts, necessary skills/responsibilities and sets the objectives.

Deliverables: Valuation Commitment Package, Client Interaction Report, Project Effort Reports, Project Plan, Effort Reports

Milestone: Valuation Commitment Review

Strategy: One Incremental Commitment Cycle

Valuation Phase

Duration: 10/04/2012 – 11/05/2012

Concept: In this phase; firstly the success-critical stakeholders have win-win negotiation sessions to evaluate risks, gather requirements, set up mitigation plans, prioritize requirements and define the proposed system by mutual understanding. As a second activity, the development team produces initial prototypes for risk management, showing main points of the system and user interfaces.

Deliverables: Core Foundations Commitment Package, Draft Foundations Commitment Package, Foundation Commitment Package, Project Effort Reports, Project Plan, Progress Reports, User Interface Prototypes (Mockups).

Milestone: Foundations Commitment Review.

Strategy: Incremental Commitment Cycles (Architected Agile)

Foundation Phase

Duration: 11/06/2012 – 12/10/2012

Concept: Continue risk assessment process, regular stakeholder meetings are to be taken every week, regular progress reports and effort reports to be submitted every Monday and Wednesday respectively, project plans are to be prepared and released on project web-page, first actual (non-GUI) prototype to be implemented to benchmark algorithm performance, risk resolution, assessing project status, sharing implementation jobs.

Deliverables: Core Development Commitment Package, Draft Development Commitment Package, Development Commitment Package, Effort Reports, Progress Reports, Project Plan, Prototype Report, Supporting Information Document, Solver Algorithm Prototype

Milestone: Development Commitment Review

Strategy: Incremental Commitment Cycles (Architected-Agile)
Strategy for Rebaselined Foundation Phase: With respect to the comments during the FCR ARB sessions and high risks about constraint solver algorithm, three one-week long iterations of Foundation Phase are rebaselined as forming test/constraint cases for algorithm, reforming user interface mockups and implementing core scheduling algorithm to manage this related high risk. One week long iterations are used and team meetings are held after the iterations.
Milestone for RFP: Rebaselined Development Commitment Review
Deliverables for RFP: Working algorithm/constraint solver, student/administrative interfaces.

Development Phase – Construction Iteration
Duration: 01/14/2013 – 02/28/2013
Concept: Implementation of system algorithm, implementation of architecture, database implementation, integration of problem solver libraries to algorithm, implementation of draft user interface, continue risk assessment process, regular stakeholder meetings are to be taken every week, regular progress reports and effort reports to be submitted every Monday and Wednesday respectively, project plans are to be prepared and released on project web-page, risk resolution,
Deliverables: Scheduling Algorithm, Core Test Case, Effort Reports, Progress Reports, Project Plan, Core Schedule Algorithm, Core User Interface, Rebaselined Development Commitment Package, Operational Concept Definition, System and Software Architecture Description, Life Cycle Plan, Feasibility Evidence Description, Supporting Information Document, Quality Management Plan, Test and Plan Cases, Transition Plan, Controller Modules. Constraint Solver Module, Database Module Draft Version
Milestone: Core Capability Drive-through – Transition Readiness Review
Strategy: Incremental Commitment Cycles, Iterations (Architected-Agile)

Development Phase – Transition Iteration
Duration: 03/01/2013 – 04/29/2013
Concept: Future development of system algorithm, assessing the system capability, implementation of user interface, implementation of login system, database integration, continue risk assessment process, system transition, regular stakeholder meetings are to be taken every week, regular progress reports and effort reports to be submitted every Monday and Wednesday respectively, project plans are to be prepared and released on project web-page, risk resolution,
Deliverables: Operation Commitment Package, Final Scheduling System software product, Initial Operational Capability Working Set, Transition Package, IOC 3 Package, Source Code Files, Final Development Documents (For Maintenance Uses), Executable Components.
Milestone: Product Delivery, Client Evaluation
Strategy: Incremental Commitment Cycles, Implementation Iterations (Architected-Agile), Team feedbacks and bug resolving, Working by weekly sprints, sprint backlogs delivered via email among team members.
Milestone: Product Delivery, Client Evaluation

2.2 Project Deliverables

2.2.1 Exploration Phase

Artifacts which are to be given during Exploration Phase are stated in Table 1.

Dates are given according to the ICSM EPG guidelines and CS577b course schedule, small changes may occur due to schedule changes and distance between client and development team.

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	9/19/2012	.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	10/03/2012	.doc, .pdf	Soft copy
Project Website	09/19/2012	Web-based	
Effort Report	Every Monday	E-form	ER system
Project Plan	Every Wednesday	.mpp	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy

2.2.2 Valuation Phase

Artifacts which are to be given during Valuation Phase are stated in Table 2.

Table 2: Artifacts – Deliverables in Valuation Phase

Artifact	Due date	Format	Medium
Evaluation of Valuation Commitment Package	10/08/2012	E-form, .pdf	Soft copy
Core Foundation Commitment Package • OCD • SSAD • LCP • FED • PRO • SID	10/15/2012	.doc, .pdf	Soft copy
Quality Management Plan	10/26/2012	.doc, .pdf	Soft copy
Client Meeting Notes	Every week	.doc, .pdf	Soft copy
Draft Foundation Commitment Package • OCD • SSAD • LCP • FED • PRO • SID	10/22/2012	.doc, .pdf	Soft copy
Foundations Commitment Package • OCD • SSAD • LCP • FED • PRO • SID	11/05/2012	.doc, .pdf	Soft copy
Effort Report	Every Monday	E-form	ER System
Progress Report	Every Wednesday	.xls	Soft Copy
Project Plan	Every week	.mpp	Soft copy
COTIPMO Report	Every Wednesday	Web-based	Web-based
Algorithm Test-Case	11/05/2012	.xls	Soft copy
User Interface Prototype Mockups	10/28/2012	Xml, .jpeg	Soft copy

2.2.3 Foundations Phase

Artifacts which are to be given during Foundations Phase are stated in Table 3.

Table 3: Artifacts – Deliverables in Foundations Phase

Artifact	Due date	Format	Medium
Evaluation of FC Package	11/12/2012	E-form, .pdf	Soft copy
Quality Management Plan	11/19/2012	.doc, .pdf	Soft copy
Draft DC Package • OCD • SSAD • LCP • FED • PRO • SID	11/26/2012	.doc, .pdf	Soft copy
Evaluation of Draft DC Package	12/03/2012	E-form, .pdf	Soft copy
DC Package • OCD • SSAD • LCP, FED • TP, TPC • PRO, QMP • SID	12/10/2012	.doc, .pdf	Soft copy
Effort Report	Every Monday	E-form	ER System
Progress Report	Every Wednesday	.xls	Soft Copy
Client Meeting Notes	Every week	.doc, .pdf	Soft copy
Project Plan	Every week	.mpp	Soft copy
COTIPMO Report	Every Wednesday	Web-based	Web-based
Algorithm Prototype	12/10/2012	.xls	Soft copy
Core Constraint Solver Module	12/10/2012	.java	Soft copy
Core Database Module	12/10/2012	.doc	Soft copy
Core Schedule Plan Constructor Module	12/10/2012	.java	Soft copy
Core Degree Requirements Construction Module	12/10/2012	.java	Soft copy
Formalism for Specifying Requirements	11/24/2012	.doc	Soft copy
Schedule Constraints Core Test Case	11/22/2012	.xls	Soft copy

2.2.4 Rebaselined Foundations Phase

The artifacts which are to be given during Rebaselined Foundations Phase are stated in Table 4.

Table 4: Artifacts – Deliverables in Rebaselined Foundations Phase

Artifact	Due date	Format	Medium
Updated Team Website	01/28/2013	Html	Soft copy
Effort Report	Every Monday	E-form	ER System
Progress Report	Every Wednesday	.xls	Soft Copy
Client Meeting Notes	Every week	.doc, .pdf	Soft copy
COTIPMO Report	Each Two Weeks Wednesday	Web-based	Web-based
Project Plan	Every Wednesday	.mpp	Soft copy
Draft RDCR Package	02/11/2013	.doc, .pdf, .xls	Soft copy
Core Constraint Solver Module	02/15/2013	.java	Soft copy
Core Database Module	02/15/2013	.java, MySQL	Soft copy
Core Schedule Plan Constructor Module	02/15/2013	.java	Soft copy
Core Degree Requirements Constructor Module	02/15/2013	.java	Soft copy
Core Authenticator Module	02/15/2013	.java	Soft copy
Core View Control Module	02/15/2013	.java	Soft copy

2.2.5 Development Phase – Construction

The artifacts which are to be given during Development Phase-Construction are stated in Table 4.

Table 5: Artifacts – Deliverables in Development Phase - Construction

Artifact	Due date	Format	Medium
IOC Package	02/28/2013	.doc, .pdf	Soft copy
Effort Report	Every Monday	E-form	ER System
Progress Report	Every Wednesday	.xls	Soft Copy
Client Meeting Notes	Every week	.doc, .pdf	Soft copy
COTIPMO Report	Every Wednesday	Web-based	Web-based
Project Plan	Every week	.mpp	Soft copy
Constraint Solver Module	02/28/2013	.java	Soft copy
Database Module	02/28/2013	.java, MySQL	Soft copy
Schedule Plan Constructor Module	02/28/2013	.java	Soft copy

Degree Requirements Constructor Module	02/28/2013	.java	Soft copy
Core Authenticator Module	02/28/2013	.java	Soft copy
Core View Control Module	02/28/2013	.java	Soft copy

2.2.6 Development Phase – Transition

The artifacts which are to be given during Development Phase – Transition are given in Table 5.

Table 6: Artifacts – Deliverables in Development Phase - Transition

Artifact	Due date	Format	Medium
Final Deliverables	04/29/2013	.java, .doc, MySQL	Soft copy
Core Capability Drive-through Documents-System	03/27/2013	Java, MySQL	Soft copy
Effort Report	Every Monday	E-form	ER System
Progress Report	Every Wednesday	.xls	Soft Copy
COTIPMO Report	Every Wednesday	Web-based	Web-based
Client Meeting Notes	Every week	.doc, .pdf	Soft copy
Project Plan	Every week	.mpp	Soft copy
Initial Database Information	04/25/2013	.doc	Soft copy
User's Manual	04/25/2013	.doc, .pdf	Soft copy
Close Out Report	04/30/2013	.doc	Soft copy

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities

The client is Professor David Klappholz, from Stevens Institute of Technology. The users of the project are students, administrative and professors in Stevens Institute of Technology. The maintainers will be attended by the institute. Developer is Team#06 and Simone Lojeck is in the role of IIV&V.

Professor David Klappholz is project-specific stakeholder and his role is to supply schedule/course charts and platform information to developers.

Operation phase responsibilities including maintenance jobs belong to client side.

Transition phase responsibilities and detailed requirements are stated in Transition Plan document in more clear view.

3.2 Responsibilities by Phase

Team members' responsibilities related to their roles are given in the Table 6 below.

Table 6: Responsibilities in Each Phase

Team Member / Role	Primary / Secondary Responsibility					
	Exploration	**Valuation**	**Foundations**	**Rebaselined Foundations**	**Development-** Construction Iteration	**Development-** Transition Iteration
Douglass Kinnes: Project Manager, Quality Focal Point, Implementer / Builder	**Primary Responsibility** -Detail Project Plan -Track Project Progress **Secondary Responsibility** -Set up client meetings -Documenting OCD	**Primary Responsibility** -Detail project plan -Track project progress -Lead win-win negotiations **Secondary Responsibility** SID Set up client meetings	**Primary Responsibility** -Detail project plan -Track project progress -Assess project feasibility and schedule **Secondary Responsibility** -Task assignment	**Primary Responsibility** -Forming constraints test case -Track project progress **Secondary Responsibility** -Hold team meetings	**Primary Responsibility** -Implementing algorithm. -Track project progress -Assess project feasibility and schedule **Secondary Responsibility** -Task assignment	**Primary Responsibility** -Track project progress -Implement software modules -Prepare close out report. **Secondary Responsibility** -Hold stakeholder meetings

Alexey Tregubov: System Architect, UML Modeler, Implementer / Builder	**Primary Responsibility** -Analyze current system **Secondary Responsibility** -Track individual effort -Documenting OCD	**Primary Responsibility** -SSAD documenting. -UML modeling **Secondary Responsibility** -Analyze the proposed system -Track individual effort -UML modeling	**Primary Responsibility** -Draft architecture implementation -Solver library integration **Secondary Responsibility** -Analyze the proposed system	**Primary Responsibility** -Prepare formalism for specifying requirements -Implementing algorithm prototype **Secondary Responsibility** -Updating architecture	**Primary Responsibility** -Implementing software modules -Solver library integration **Secondary Responsibility** -Architecture implementing	**Primary Responsibility** -Implement software modules -Module integration **Secondary Responsibility** -Task assignment Prepare user's manual
Ihsan Tolga: Life Cycle Planner, Feasibility Analyst, Implementer / Builder	**Primary Responsibility** -Assess and plan to mitigate risks -Documenting FED, LCP **Secondary Responsibility** -Assessing Iteration Plan	**Primary Responsibility** -Provide project feasibility evidence -Explore alternative **Secondary Responsibility** -Life Cycle Planning	**Primary Responsibility** -Assessing life cycle -Detail effort and schedule estimation -Assessing feasibility evidences **Secondary Responsibility** -Feasibility Evidence Description -Assess risks	**Primary Responsibility** -Track project progress -Evaluate risks, manage risks. -Forming constraints test case. -Revising project estimations. **Secondary Responsibility** -Implementing algorithm prototype	**Primary Responsibility** -Assessing life cycle artifacts. -Implementing software modules. **Secondary Responsibility** -Iteration tasks defining -Track project progress	**Primary Responsibility** -Track project progress -Assess project feasibility and schedule -Implementing software modules **Secondary Responsibility** -Preparing user's manual
Mihir Daptardar: Operational Concept Engineer, Quality Focal Point, Tester, Implementer	**Primary Responsibility** -Analyze current system **Secondary Responsibility** -Track project individual effort	**Primary Responsibility** -Define operational concept. -Identify objectives, constraints and priorities **Secondary Responsibility** -Define requirements	**Primary Responsibility** -Integrate solver library to algorithm -Identify objectives, constraints and priorities **Secondary Responsibility** -Detail requirements	**Primary Responsibility** -Preparing constraints test case -Prototype algorithm implementation **Secondary Responsibility** -Requirements satisfaction checking.	**Primary Responsibility** -Integrate solver library to algorithm. -Implementing software modules. **Secondary Responsibility** -Algorithm benchmarking.	**Primary Responsibility** -Implementing software modules -Software benchmarking -Preparing initial database info -Database module interconnection **Secondary Responsibility** -Requirements satisfaction checking
Simone Lojeck: IV&V, Quality Focal Point, Shaper	**Primary Responsibility** -Verify and validate work products -Project web page **Secondary Responsibility** -Track individual effort.	**Primary Responsibility** -Verify and validate work products. -Track defects **Secondary Responsibility** -Shape win-win negotiations.	**Primary Responsibility** -Verify and validate work products. -Track defects **Secondary Responsibility** -Assess risks.	**Primary Responsibility** -Preparing quality management plan -Track prototype defects **Secondary Responsibility** -Defect reporting	**Primary Responsibility** -Verify and validate work products -Track defects **Secondary Responsibility** -Team reviews shaping -Defect reporting	**Primary Responsibility** -Verify and validate work products -Track software/document defects -Website update **Secondary Responsibility** -Team reviews shaping -Defect reporting
***** The responsibilities and jobs of non-existing team member were distributed among current team members. There is no new team member for Spring 2013 semester.						

3.3 Skills

Project and role related skills of team members are given in Table 7 and needed skills in case of personnel turnovers are given in Table 8 below.

Table 7: Skills of Team Members Related to Their Roles

Team members	Role	Skills
Douglass Kinnes	Project Manager, Quality Focal Point, Implementer / Builder	Project management, Architecture Design Java (good) MySQL WinBook, MS Project
Alexey Tregubov	System Architect, UML Modeler, Implementer / Builder	Architecture Design Java, XML, PHP UML Modeling UP for UML SW Module Integration
Mihir Daptardar	Operational Concept Engineer, Quality Focal Point, Tester, Implementer	System Analysis Java, PHP, MySQL Algorithm Benchmarking Test Case Implementation Bugzilla
Ihsan Tolga	Life Cycle Planner, Feasibility Analyst, Implementer / Builder	Java, MySQL Life Cycle Planning Cost/Benefit/ROI Analysis Risk Management COCOMO, COTIPMO
Simone Lojeck	IV&V, Quality Focal Point	Bugzilla WinBook Software Defect Tracking Algorithm Benchmarking HTML

Table 8: Skills Required fir CS577b in case of Personnel Turnover

Role	Skills
Builder	Java, PHP, MySQL, Play Framework
Operational Concept Engineer	System Analysis, UML
System Architect	Architecture Design, UML
IIV&V	Testing, Bugzilla, WinBook
Life Cycle Planner	COCOMO, Cost/Benefit/ROI Analysis, COTIPMO
Tester	Bugzilla, Benchmarking, Debugging

4. Approach

4.1 Monitoring and Control

The Team 06 will be utilizing various tools and documentation to assist in the monitoring and control of the project. The key items are listed below and described in the following sections.

- Progress Reports – Weekly submissions about sum of the team members' efforts on the project.
- Effort Reports – Weekly submission about each team member's total effort on the project.
- Winbook Negotiations – Prioritizing requirements and win conditions to reflect the changes in requirements and win conditions.
- Bugzilla Defect Tracking and Resolving – Providing feedback to team members about their artifacts and documentations.
- Project Plans – Tracking the project progress and future increments/events.
- Commitment Reviews – Provide feedback and grading for the current stage of the project and artifacts.
- Client Meetings – Weekly negotiations and discussions about project progress and stakeholder collaboration.
- SubVersion – The tool is used for remote developers during the development iterations. Development team members can check the others work and track the differences. It brings many advantages for project's management and leading.

4.1.1 Closed Loop Feedback Control

Bug tracking by IIV&V is used for internal feedback within the project team. Each member is responsible to resolve the defects related to his/her roles in the project.

Bugzilla is used for bug tracking; they are tracked by IIV&V member and each member who is responsible for the particular defect gets a notification about this bug to resolve or take the necessary action.

Client requirements and prioritized requirements exist in Winbook page of the project team to serve as a basis for commitments.

Every week, at least one client meeting is performed with client and project team members.

Skype, WebEx, Trello and TeamViewer screen sharing web site and tools are used for remote client meetings and desktop sharing.

Independently from other methods; each member is encouraged to report defects and shortfalls he/she encounters.

4.1.2 Reviews

- IIV&V reviews.
- Commitment Package Evaluations
- Stakeholder commitment review
- Weekly stakeholder meetings
- Bugzilla
- Arbitrary / weekly individual reviews
- Core Capability Drive-through

4.2 Methods, Tools and Facilities

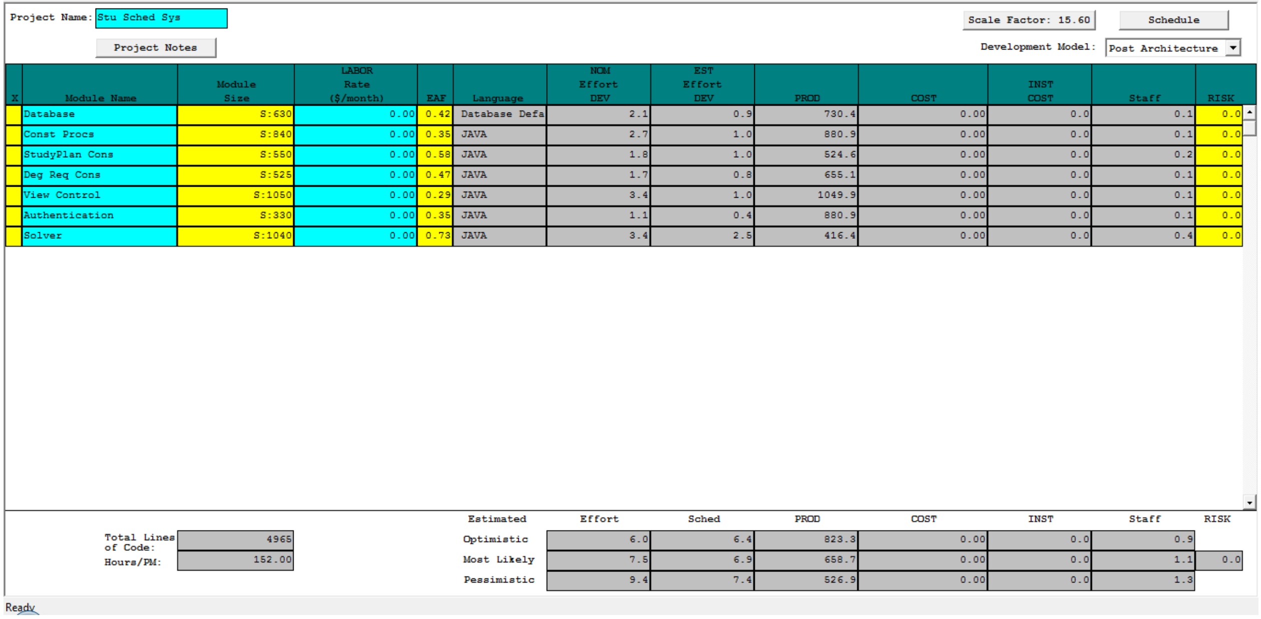
Major tools used by team members during the project are listed in Table 9.

Table 9: Tools Used in the Project

Tools	Usage	Provider
ECLIPSE	Java compiler and editor diversely used for building framework, source code, libraries and user interface.	Open source
MS Project 2010	Project Plan, Life Cycle Plan	USC License
Balsamiq	Used to form user interface prototypes.	Demo version
Subversion	Used for project administration, control/monitoring and implementing from remote locations on the same project.	USC License
MySQL Workbench	Used for MySQL database implementing and identifying student information for software algorithm.	Open source
MS Word	Used for documenting deliverables, LCP, OCD, FED.	Team members
TeamViewer	Used for screen sharing during client meetings, ARB sessions, internal meetings.	Team members
Bugzilla	Used for defect/bug reporting to team members and it serves as a guideline for bug resolving.	USC License
Winbook	Used for stating stakeholder requirements, win-win negotiations and prioritizing the requirements.	USC License
MS Excel	Used for testing schedule constraints, preparing test case constraints, constraint mismatch detecting.	Team members

5. Resources

- Team members :5 (3 on-campus, 2 off-campus)
- Project duration :12 weeks in Fall 2012 and 12 weeks in Spring 2013
- Programming Language :Java, MySQL



Project Name: Stu Sched Sys

Scale Factor: 15.60

Development Model: Post Architecture

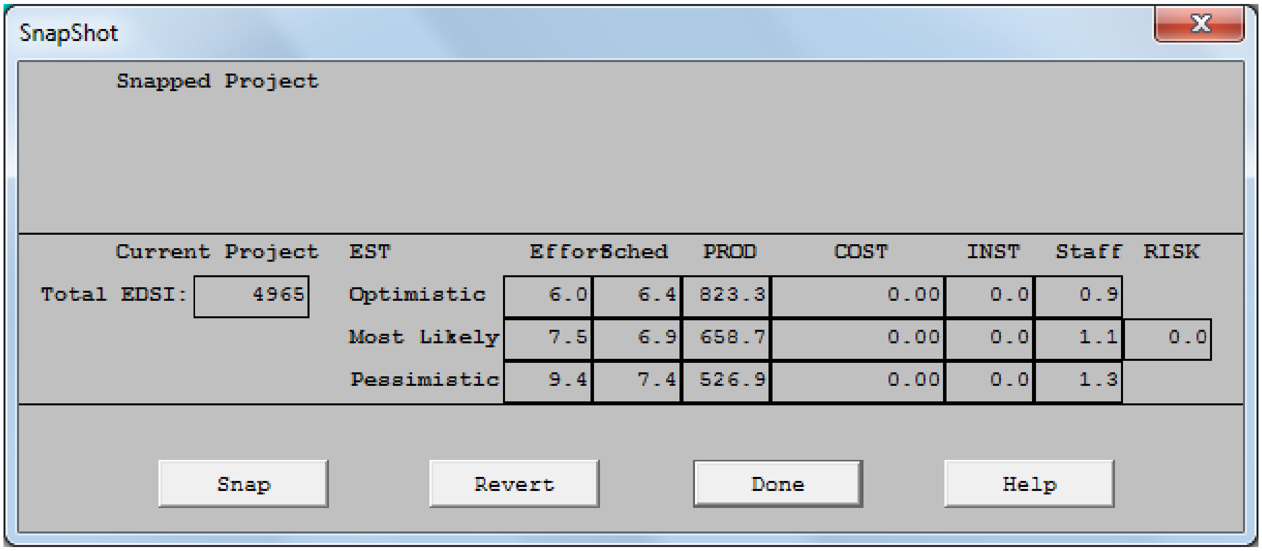
X	Module Name	Module Size	LABOR Rate ($/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Database	S:630	0.00	0.42	Database Defa	2.1	0.9	730.4	0.00	0.0	0.1	0.0
	Const Procs	S:840	0.00	0.35	JAVA	2.7	1.0	880.9	0.00	0.0	0.1	0.0
	StudyPlan Cons	S:550	0.00	0.58	JAVA	1.8	1.0	524.6	0.00	0.0	0.2	0.0
	Deg Req Cons	S:525	0.00	0.47	JAVA	1.7	0.8	655.1	0.00	0.0	0.1	0.0
	View Control	S:1050	0.00	0.29	JAVA	3.4	1.0	1049.9	0.00	0.0	0.1	0.0
	Authentication	S:330	0.00	0.35	JAVA	1.1	0.4	880.9	0.00	0.0	0.1	0.0
	Solver	S:1040	0.00	0.73	JAVA	3.4	2.5	416.4	0.00	0.0	0.4	0.0

Total Lines of Code: 4965

Hours/PM: 152.00

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	6.0	6.4	823.3	0.00	0.0	0.9	
Most Likely	7.5	6.9	658.7	0.00	0.0	1.1	0.0
Pessimistic	9.4	7.4	526.9	0.00	0.0	1.3	

Figure 1: COCOMO Analysis



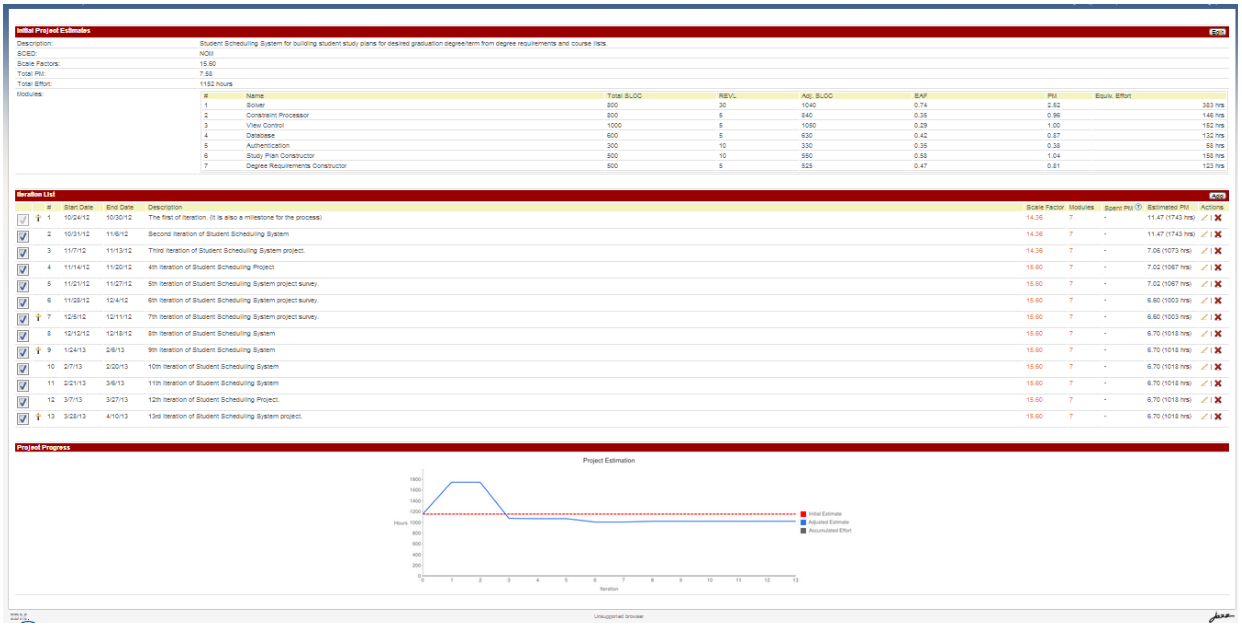
Snapped Project

Current Project

Total EDSI: 4965

EST	EfforSched		PROD	COST	INST	Staff	RISK
Optimistic	6.0	6.4	823.3	0.00	0.0	0.9	
Most Likely	7.5	6.9	658.7	0.00	0.0	1.1	0.0
Pessimistic	9.4	7.4	526.9	0.00	0.0	1.3	

Figure 2: COCOMO Analysis Results



Initial Project Estimates Edit

Description:	Student Scheduling System for building student study plans for desired graduation degree/term from degree requirements and course lists.
SCED:	NOM
Scale Factors:	15.60
Total PM:	7.58
Total Effort:	1152 hours

Modules:

#	Name	Total SLOC	REVL	Adj. SLOC	EAF	PM	Equiv. Effort
1	Solver	800	30	1040	0.74	2.52	383 hrs
2	Constraint Processor	800	5	840	0.35	0.96	146 hrs
3	View Control	1000	5	1050	0.29	1.00	152 hrs
4	Database	600	5	630	0.42	0.87	132 hrs
5	Authentication	300	10	330	0.35	0.38	58 hrs
6	Study Plan Constructor	500	10	550	0.58	1.04	158 hrs
7	Degree Requirements Constructor	500	5	525	0.47	0.81	123 hrs

Iteration List Add

	#	Start Date	End Date	Description	Scale Factor	Modules	Spent PM	Estimated PM	Actions
☑	1	10/24/12	10/30/12	The first of iteration. (It is also a milestone for the process)	14.36	7	-	11.47 (1743 hrs)	✎ | ✖
☑	2	10/31/12	11/6/12	Second iteration of Student Scheduling System	14.36	7	-	11.47 (1743 hrs)	✎ | ✖
☑	3	11/7/12	11/13/12	Third iteration of Student Scheduling System project.	14.36	7	-	7.06 (1073 hrs)	✎ | ✖
☑	4	11/14/12	11/20/12	4th iteration of Student Scheduling Project	15.60	7	-	7.02 (1067 hrs)	✎ | ✖
☑	5	11/21/12	11/27/12	5th iteration of Student Scheduling System project survey.	15.60	7	-	7.02 (1067 hrs)	✎ | ✖
☑	6	11/28/12	12/4/12	6th iteration of Student Scheduling System project survey.	15.60	7	-	6.60 (1003 hrs)	✎ | ✖
☑	7	12/5/12	12/11/12	7th iteration of Student Scheduling System project survey.	15.60	7	-	6.60 (1003 hrs)	✎ | ✖
☑	8	12/12/12	12/18/12	8th iteration of Student Scheduling System	15.60	7	-	6.70 (1018 hrs)	✎ | ✖
☑	9	1/24/13	2/6/13	9th iteration of Student Scheduling System	15.60	7	-	6.70 (1018 hrs)	✎ | ✖
☑	10	2/7/13	2/20/13	10th iteration of Student Scheduling System	15.60	7	-	6.70 (1018 hrs)	✎ | ✖
☑	11	2/21/13	3/6/13	11th iteration of Student Scheduling System	15.60	7	-	6.70 (1018 hrs)	✎ | ✖
☑	12	3/7/13	3/27/13	12th iteration of Student Scheduling Project.	15.60	7	-	6.70 (1018 hrs)	✎ | ✖
☑	13	3/28/13	4/10/13	13rd iteration of Student Scheduling System project.	15.60	7	-	6.70 (1018 hrs)	✎ | ✖

Project Progress

Figure 3: COTIPMO Estimation Screen

As can be seen from the Figure 1 and Figure 2 and Figure 3;

- the project consists of 7 modules
- the scale factor is 15.6
- total line of source code is 4965
- total effort estimated is 6.6 person/month most likely
- PM/1.67 = 4.20 persons are needed to complete the project within the time limits of CS577a and CS577b.

Thus, (4.2 < 5) the project estimates shows that it can be done within the CS577a and CS577b course periods and with 5 team members.

Scale drivers and cost drivers for each module of the project related are given in the Table 10 – Table 17.

Table 10: COCOMOII Project Scale Drivers

Scale Driver	Value	Rationale
PREC	LO	It is hard to find any former similar algorithm as an example; team members have some experience with constraint solver algorithms. Thus the PREC is low.
FLEX	HI	The client does not mandate any platform or development environment so the flexibility is high.
RESL	HI	Possible risks related to the projects are defined and there mitigation plans for them, so the resolution is high.
TEAM	VHI	All project team members are eager to take the CS577b course, thus it is high.
PMAT	NOM	ICSM principles are strictly followed by team members and there are plenty of guidelines for the process, so PMAT driver is nominal.

Table 11: COCOMOII Cost Drivers for Database Module

Cost Driver	Value	Rationale
RELY	NOM	This module's performance is moderately critical since it is the domain of the system.
DATA	HI	The ratio of "Testing DB bytes/Program SLOC" is between 100 and 1000; thus it is high.
DOCU	VLO	Most of the life-cycle needs of this module uncovered.
CPLX	LO	The system needs below moderate details to be processed.
RUSE	LO	There is no actual need to reuse this module.
TIME	NOM	There is no strict requirement for computation time yet it has to be relatively fast.
STOR	NOM	The module needs <50% use of available storage.
PVOL	NOM	MySQL provides the team flexibility for different platforms.
ACAP	NOM	More than half of the team members are capable to capture requirements for detailed assessment and design. There is good communication/cooperation among them.
PCAP	HI	Majority of the team members have high programming skills for implementation / integration.
PCON	HI	Team members are eager to take the CS577b course so no turnover is expected.
APEX	NOM	Some of the team members have moderate application experience related to this module.
LTEX	HI	Implementation members have good programming skills for this module.
PLEX	HI	Team members related to this database module have very good experience with the project platform.
TOOL	HI	All project members are familiar with the tools used in the project and trained for the new tools during the CS577a course.
SITE	HI	The customer provides the team the required data for the project via emails for this particular module; and except the remote client, there is good infrastructure for team members' collaboration and data sharing.

Table 12: COCOMOII Cost Drivers for Constraint Processor Module

Cost Driver	Value	Rationale
RELY	NOM	This module is essential for connection between database and construction modules but not hard to implement.
DATA	NOM	The ratio of "Testing DB bytes/Program SLOC" is between 10 and 100; thus it is nominal.
DOCU	LO	Some of the life-cycle needs of this module uncovered.
CPLX	LO	The complexity of implementing of this module is low due to ease of using Java for objects.
RUSE	LO	There is no current need to reuse this module in the future.
TIME	NOM	This module should run fast since it is the bridge between database and other modules.
STOR	NOM	The module needs <50% use of available storage.
PVOL	LO	Using Java provides good platform volatility and there is no strong need for frequent updates.
ACAP	NOM	More than half of the team members are capable to capture requirements for detailed assessment and design. There is good communication/cooperation among them.
PCAP	HI	Majority of the team members have high programming skills (Java) for implementation / integration.
PCON	HI	Team members are eager to take the CS577b course so no turnover is expected.
APEX	HI	Some team members have application experience from similar projects but all team members have good experience with this kind of modules.
LTEX	HI	All implementation members have good programming skills related to this module.
PLEX	NOM	All team members have good experience with the project platform.
TOOL	HI	All project members are familiar with the tools used in the project and trained for the new tools during the CS577a course.
SITE	HI	The customer provides the team the required data for the project via emails for this particular module; and except the remote client, there is good infrastructure for team members' collaboration and data sharing.

Table 13: COCOMOII Cost Drivers for Study Plan Construction Module

Cost Driver	Value	Rationale
RELY	NOM	This is the most essential module of the project thus its reliability is important.
DATA	NOM	The ratio of "Testing DB bytes/Program SLOC" is between 10 and 100; thus it is nominal.
DOCU	NOM	It is right-sized to life-cycle needs for this module.
CPLX	HI	This module maintains a high complexity resulted from scheduling problems and constraints.
RUSE	LO	There is no current need to reuse this module in the future.
TIME	NOM	There is no strong computation time constraint for this module.
STOR	NOM	The module needs <50% use of available storage.
PVOL	LO	Using Java provides the system good platform volatility and there is no strong need for frequent updates.
ACAP	NOM	More than half of the team members are capable to capture requirements for detailed assessment and design. There is good communication/cooperation among them.
PCAP	HI	Majority of the team members have high programming skills (Java) for implementation / integration.
PCON	HI	Team members are eager to take the CS577b course so no turnover is expected.
APEX	NOM	Some team members have limited application experience for this module.
LTEX	HI	All implementation members have good programming skills related to this module.
PLEX	NOM	All team members have good experience with the project platform.
TOOL	HI	All project members are familiar with the tools used in the project and trained for the new tools during the CS577a course.
SITE	HI	The customer provides the team the required data for the project via emails for this particular module; and except the remote client, there is good infrastructure for team members' collaboration and data sharing.

Table 14: COCOMOII Cost Drivers for Degree Requirements Construction Module

Cost Driver	Value	Rationale
RELY	NOM	This is the second most essential module of the project thus its reliability is important.
DATA	NOM	The ratio of "Testing DB bytes/Program SLOC" is between 10 and 100; thus it is nominal.
DOCU	LO	Some of the life-cycle needs of this module uncovered.
CPLX	HI	This module maintains a high complexity resulted from scheduling problems and constraints.
RUSE	LO	There is no current need to reuse this module in the future.
TIME	NOM	There is no strong computation time constraint for this module.
STOR	NOM	The module needs <50% use of available storage.
PVOL	LO	Using Java provides the system good platform volatility and there is no strong need for frequent updates.
ACAP	NOM	More than half of the team members are capable to capture requirements for detailed assessment and design. There is good communication/cooperation among them.
PCAP	HI	Majority of the team members have high programming skills (Java) for implementation / integration.
PCON	HI	Team members are eager to take the CS577b course so no turnover is expected.
APEX	HI	Some team members have limited application experience for this module.
LTEX	HI	All implementation members have good programming skills related to this module.
PLEX	NOM	All team members have good experience with the project platform.
TOOL	HI	All project members are familiar with the tools used in the project and trained for the new tools during the CS577a course.
SITE	HI	The client is remote, yet there is a good infrastructure for team members' collaboration and data sharing. There is no major problem with communication to the client.

Table 15: COCOMOII Cost Drivers for View Control Module

Cost Driver	Value	Rationale
RELY	NOM	This is an important module for user interface but can be easily treated in a possible crash.
DATA	NOM	The ratio of "Testing DB bytes/Program SLOC" is between 10 and 100; thus it is nominal.
DOCU	LO	Some of the life-cycle needs of this module uncovered.
CPLX	VLO	The complexity of implementing of this module is low due to ease of using Java for objects.
RUSE	LO	There is no current need to reuse this module in the future.
TIME	NOM	There is not strict execution time constraint for this module.
STOR	NOM	The module needs <50% use of available storage.
PVOL	LO	Using Java provides good platform volatility and there is no strong need for frequent updates.
ACAP	NOM	More than half of the team members are capable to capture requirements for detailed assessment and design. There is good communication/cooperation among them.
PCAP	HI	Majority of the team members have high programming skills (Java) for implementation / integration.
PCON	HI	Team members are eager to take the CS577b course so no turnover is expected.
APEX	HI	Some team members have good application experience for this module.
LTEX	HI	All implementation members have good programming skills related to this module.
PLEX	NOM	All team members have good experience with the project platform.
TOOL	HI	All project members are familiar with the tools used in the project and trained for the new tools during the CS577a course.
SITE	HI	The client is remote, yet there is a good infrastructure for team members' collaboration and data sharing. There is no major problem with communication to the client.

Table 16: COCOMOII Cost Drivers for Authentication Module

Cost Driver	Value	Rationale
RELY	NOM	This module completes the system integrity but is not essential except some periods of the system's run-time period.
DATA	NOM	The ratio of "Testing DB bytes/Program SLOC" is between 10 and 100; thus it is nominal.
DOCU	LO	Some of the life-cycle needs of this module uncovered.
CPLX	LO	The complexity of implementing of this module is low due to ease of using Java for objects.
RUSE	LO	There is no current need to reuse this module in the future.
TIME	NOM	There is no strict execution time constraint for this module.
STOR	NOM	The module needs <50% use of available storage.
PVOL	LO	Using Java provides good platform volatility and there is no strong need for frequent updates.
ACAP	NOM	More than half of the team members are capable to capture requirements for detailed assessment and design. There is good communication/cooperation among them.
PCAP	HI	Majority of the team members have high programming skills (Java) for implementation / integration.
PCON	HI	Team members are eager to take the CS577b course so no turnover is expected.
APEX	HI	Some team members have good application experience for this web related module.
LTEX	HI	All implementation members have good programming skills related to this module.
PLEX	NOM	All team members have good experience with the project platform.
TOOL	HI	All project members are familiar with the tools used in the project and trained for the new tools during the CS577a course.
SITE	HI	The client is remote, yet there is a good infrastructure for team members' collaboration and data sharing. There is no major problem with communication to the client.

Table 17: COCOMOII Cost Drivers for Solver Module

Cost Driver	Value	Rationale
RELY	NOM	This module is essential for the algorithm and mathematical constraint solving but does not have any vital effects in general.
DATA	NOM	The ratio of "Testing DB bytes/Program SLOC" is between 10 and 100; thus it is nominal.
DOCU	VLO	Some of the life-cycle needs of this module uncovered.
CPLX	HI	The complexity of implementing of this module is high due to the complexity of forming mathematical constraints.
RUSE	LO	There is no current need to reuse this module in the future.
TIME	NOM	This module should run fast. The ratio of consumed and available execution time is below 50%.
STOR	NOM	The module needs <50% use of available storage.
PVOL	LO	Using Java provides good platform volatility and there is no strong need for frequent updates.
ACAP	LO	More than half of the team members are capable to capture requirements for detailed assessment and design yet algorithm issues are hard to capture/analyze.
PCAP	LO	Majority of the team members have high programming skills (Java) for implementation / integration. But the Java library used (CHOCO) is new to the developers.
PCON	HI	Team members are eager to take the CS577b course so no turnover is expected.
APEX	NOM	Some team members have application experience from similar projects but a constraint solver module is not so familiar for the members to work on.
LTEX	HI	All implementation members have good programming skills related to this module.
PLEX	NOM	All team members have good experience with the project platform.
TOOL	HI	All project members are familiar with the tools used in the project and trained for the new tools during the CS577a course.
SITE	HI	The customer provides the team the required data for the project via emails for this particular module; and except the remote client, there is good infrastructure for team members' collaboration and data sharing.

6. Iteration Plan

6.1 Plan

Upcoming iterations are planned for the Development Commitment milestone. During the Foundation Phase, project team is implementing the mathematical model of the solver algorithm, evaluation of CHOCO Java library for the compatibility concerns, test-case for early testing of constraint solver and user interface prototypes. Besides, the team is working on a formalism document to provide a guideline for future implementations.

- **First Iteration :** This iteration has started with the DCR ARB and will end with the CCD activity on 04/08/2013. It mostly covers the initial capability of system with user interface modules for data input/log in capabilities, controller sub-modules between database-solver module-user interface and constraint solving (80% of overall). Testing activities will be going simultaneously with the implementation. The implementation process is being held as weekly sprints. On Fridays, new information for the next sprint and backlog is distributed among team members.

- **Second Iteration :** This iteration will start right after CCD on 04/08/2013 and will cover the period until starting transition of the system. The main focus will be on problems tracked during the CCD and completion of the full capability of the system modules before the end product delivery. Testing activities will be conducted again simultaneously with implementation yet more focus will be given to it.

6.1.1 Capabilities to be implemented

For the upcoming milestone, the capabilities that will be implemented in the next iterations are listed in Table 18.

Table 18: Construction Iteration Capabilities to be Implemented

ID	Capability	Description	Priority	Iteration
1	Solving schedule constraints.	The prototype algorithm will solve the course constraints.	1	1
2	User Interface Prototype	We are forming mock-up user interface prototypes before starting implementing them in Java.	2	1
3	Test-case for constraint solver	We are implementing a draft test-case (courses, constraints) to test the scheduling algorithm.	1	1
4	Formalism for Specifying Requirements	We are documenting a formalism document as an implementation guideline for future development phase.	2	1
5	Mathematical Model	We are forming a mathematical model in which we will define the course information	3	2

		and constraints.		
6	Solver Library	We are defining the software architecture for best interconnection between the modules and database connection.	1	2
7	Entering Course Information Input	Administrative side will be able to enter course information. Thus we are implementing prototypes related to input format and specifications.	2	2
8	Course Data Database	Database implementation for storing course data and administrative authentication log files.	2	2

6.1.2 Capabilities to be tested

Below is the list of Win Conditions which the project stakeholders has agreed upon and will be tested during the next iteration.

- WC_1512: System must be able to construct a study plan based upon the inputs from the student within the degree requirements maintained by the advisers.
- WC_1357: Student should get a response from the system approximately in 1-2 minutes dependent on the solver order of the algorithm.
- WC_1352: Student specifies year of entry and desired degree.
- WC_1348: Student can specify a semester for courses to be taken.
- WC_1353: Student specifies desired degree completion semester.

Construction iteration capabilities (related to the win conditions above) which will be tested in iterations before the next milestone are listed in Table 19.

Table 19: Construction Iteration Capabilities to Be Tested

ID	Capability	Description	Priority	Iteration
1	System constructs a study plan within the constraints.	System must be able to construct a sensible study plan based upon the inputs from the student within the degree requirements maintained by the administrative side.	1	1
2	System produces the results within 2 minutes at maximum.	Student should be able to get a response (recommended study plan) from the system in 2 minutes at maximum. (Depends on constraint solver performance of the algorithm.)	2	2
3	Specifying year of entry and desired degree.	The system should be able to take year of entry and desired degree as the inputs before running the scheduling algorithm.	1	1
4	Specifying desired	Student should be able to specify a desired semester for courses to be taken.	3	3

	semesters for courses.			
5	Specifying desired degree completion semester.	Student should be able to specify a desired semester to graduate.	1	1

6.1.3 Capabilities not to be tested

Table 20: Construction Iteration Capabilities Not to Be Tested

ID	Capability	Rationale	Priority
1	System must provide user privileges according to his/her role.	It is planned to be tested in second iteration via demonstration.	4
2	No monetary budget, lots of hours on part of team and client.	It is not possible to test this win condition. They are listed in FED document.	4
3	System will return issue resolution information, if a solution could not be determined.	It is planned to be tested after complete algorithm implementation.	2
4	Student must be able to request that the system construct a schedule when he/she feels that he/she entered as much information as he/she need to into the system.	It is planned to be tested after algorithm implementation by inspecting this ability and demonstrating.	3
5	Administrator can input degree requirements for each year of entry and degree combinations.	It is planned to be tested during first iteration while it is implemented.	2
6	Administrator can input information about individual courses	It is planned to be tested after the first iteration. It will be inspected by testers.	1
7	Student can specify on campus/online for courses to be taken.	It needs a tested solver algorithm before implementation and testing.	3
8	Student can specify completed courses to be applied to the graduation requirements.	It is planned to be implemented and tested after algorithm implementation.	2
9	Student can specify particular course(s) he/she wants to take to meet degree specific requirements.	It is planned to be implemented and tested after algorithm implementation.	3
10	As an administrator, he/she must be able to enter degree requirements as complex as those that have been in effect.	It is planned to implemented/tested after a final agreement is set upon the interface by tester/client inspection.	1

6.1.4 CCD Preparation Plans

For the Core Capability Drive-through, following stakeholders will be involved:

- Client: David Klappholz
- Development Team Members (Team 06)

The CCD preparation plan:

- There will be a briefing to inform the client about the recent iterations and general view which was already discussed during the client meetings. (A dry run)
- The client, as an administrator, will be asked to input related course information to test the administrative side of the system.
- The client, as a user, will be asked to input proper information for system to build a study plan. The information should include the followings:
    - Year of entry
    - Desired graduation semester
    - Desired courses to be taken
    - Selection of on campus/online courses.
    - Courses taken before (for transferred students)
- Client will be asked to share his experience with the system and his opinions. He will be also asked to give a feedback.
- The risk mitigation plan will be revised by team with respect to the client feedback.

6.2 Iteration Assessment

6.2.1 Capabilities Implemented, Tested, and Results

The capabilities began to being implemented and tested are listed in Table 21.

Table 21: Capabilities Implemented, Tested, and Results

ID	Capability	Test Case	Test Results	If fail, why?
1	System constructs a study plan within the constraints.	TC-01	Pass (Works well within the real initial courses and constraints.)	Passed.
2	System produces the results within 2 minutes at maximum	TC-02	Pass	In most cases, prototype algorithm gives the results in seconds but fails with complex constraints. (Halts after given time, client agreed.)
3	Users / Administrator can enter the courses / course information.	TC-03	Pass	An agreement was set upon the user interface of the system between

				developers and the clients.
4	Solver library works well with the given constraints.	TC-04	Pass	In-house developed algorithm works well with real course data.

6.2.2 Core Capabilities Drive-Through Results

The Core Capabilities Drive-Through results were captured during the CCD session held on April 8th 2013. The client David Klappholz attended the CCD session via video conference and screen sharing. The control of the terminal was given to the customer to do the tasks/actions written in demonstration guideline document given to him before the CCD.

The Student Scheduling System was run on USC's Longbeach server to demonstrate also the real usage experience of the system. Only the initial course/requirement/degree program data (which represents the actual course data) was loaded into the system to allow the client to experience administrative side actions himself.

The all CCD session was recorded by the team and notes related to customer feedback and comments were taken simultaneously which are already written in Concern Logs document. Since the duration of CCD is limited; it is not possible to cover/test all win conditions/requirements before it ends. So the team designated the most critical capabilities to be demonstrated during the session and decided to use win conditions and some requirements as basis for Concern Log document.

The critical capabilities mentioned above are mostly related to study plan construction algorithm and key administrative side action such as course adding, course group adding, adding/editing requirements and degree programs. The justification for this particular decision and filtering is the client's preference which is a properly working scheduling algorithm and administrative side controllers are way more important than a user-friendly / nice looking user interface.

Captured list of these win conditions / requirements tested/demonstrated during the CCD are given in the table below.

Table 22: CCD Results and Client Feedbacks

Win Condition / Requirement	User Experience	Result
WC_1512: System must be able to construct a study plan based upon the inputs from the student within the degree requirements maintained by the advisers.	Driven-through	**Positive Feedbacks:** **-** The client is pleased with the output of study plan construction - Client is agreed with the plain looking of constructed study plan. He stated that he is more interested in a properly working algorithm than sharp-looking interface.

		Improvements Needed: The client stated that a more realistic (in terms of semesters/time) study plan would be better **Changes Needed:** N/A
WC_1357: Student should get a response from the system approximately in a reasonable amount of time for the simple case	Driven-through	**Positive Feedbacks:** **-** The system constructed study plans in several seconds as it is desired. **-** The client stated that he is happy with the overall performance of study plan construction **Improvements Needed:** N/A **Changes Needed:** **-** The TA and client agreed that an animation running while waiting for the construction of study plan can make sense to assure the users that system is running or there is no error
WC_1356: No monetary budget, lots of hours on part of team and client	Shown/Demonstrated	**Positive Feedbacks:** **-** The client is content and happy about the satisfaction of this condition/requirement and the development team's commitment to the project **Improvements Needed:** N/A **Changes Needed:** N/A
WC_1355: System will return issue resolution information (list of issues with the student's inputs and/or alternate plan and suggestions), if a solution could not be determined	Driven through	**Positive Feedbacks:** N/A **Improvements Needed:** **-**The client/TA stated that a system function which shows the possible reasons in case of a execution error/failure of constructing study plan would be better to lead the users in the right way **Changes Needed:** **-** The client/TA also stated that courses selected by the user and added by the system should be in different colors if possible; so that users can have a better idea about the proposed study plan
WC_1350: Administrator can input degree requirements for each year of entry and degree combinations	Driven through	**Positive Feedbacks:** **-**The client is satisfied with the way of entering degree requirements in administrative side of the system **Improvements Needed:** N/A **Changes Needed:** N/A
WC_1349: Administrator can input information about individual courses.	Demonstrated	**Positive Feedbacks:** **-**The client is happy with the course adding/editing/deleting functions in administrative side **Improvements Needed:** **-**There should be control mechanism to

		prevent users to try deleting a course twice while waiting for the system to process **Changes Needed:** N/A
WC_1347: Student can specify on campus/online for courses to be taken	Driven through	**Positive Feedbacks:** -The client got very pleased with the overall performance of student side of the system - The client liked the user interface for selecting desired courses online/on campus along with other related information **Improvements Needed:** N/A **Changes Needed:** N/A
WC_1345: Student can specify particular course(s) s/he wants to take to meet degree specific requirement(s) (Degree such as CS, IS, and CyS is also specified by student)	Driven through	**Positive Feedbacks:** **-**The client stated that he was pleased with the overall performance of student side of the system **Improvements Needed:** N/A **Changes Needed:** N/A
WC_1329: As an administrator I must be able to enter degree requirements as complex as those that have been in effect, for the CS, IS, and CyS undergrad degrees and are listed, on the Stevens CS dept. website as well as additional clarifying details provided by the client to the team	Demonstrated	**Positive Feedbacks:** **-**The client was satisfied by the adding complex degree requirements besides simple degree requirements. -The client found the working mechanism of complex degree requirements useful and sensible. **Improvements Needed:** -The related course/requirement information at the bottom section of the pages is not clearly visible. So it should be at the right side of the page or somewhere else which it can be seen more easily **Changes Needed:** N/A
OG-1: Eliminate frustration of the advisors when creating student schedules	Shown/Demonstrated	**Positive Feedbacks:** **-** The client found the way of adding course/course groups/requirements and degree programs fast and useful. **Improvements Needed:** **-**More information on User Interface can be better to help users of the system to keep on right track considering not every user of the system will be familiar with it. **Changes Needed:** **-**The “abbreviations” section a little confusing. A control mechanism named “Automatic Abbreviation Generation” which produces an

		abbreviation from the title of the course/course group/requirement can be useful to eliminate confusion
OG-3: Reduces mistakes in choosing courses	Shown	**Positive Feedbacks:** N/A **Improvements Needed:** **-**It is necessary to add some control structures to treat possible execution error resulted from mistyped / wrong inputs **Changes Needed:** N/A

With respect to the feedbacks received from the client; emerging low-profile risks and mapped risk mitigation plans are defined. These risks and related mitigation plans are given in the table below.

Table 23: Explored Short Term Risks for CCD

Risk	Risk Description	Mitigation Plan
Issue resolution information	The study plan constructor should give reasons in case a study plan cannot be constructed to lead the user in right way.	Giving separate colors to the courses added by user and system. Putting try-catch blocks to show messages with possible reasons cause errors.
Confusing user interface pages	Some user interface pages (i.e. related course groups below) and blank waiting screen can be confusing for users who are not familiar with the system. So they may tend to enter wrong inputs.	Moving tables/items lead confusion to another place in the pages, putting more explanations in user interface, putting "in progress" animation via Ajax.

6.3 Adherence to Plan

- The iterations are going on track according to the plan.
- The risk with the highest magnitude and possibility (Constraint solver algorithm) was focused on during the last iterations and current focus is still on this item. Initial tests resulted well for this particular risk.
- All of the team members are going to take 577b so there will be no personnel turnover as expected.
- A strong agreement was set upon the user interfaces between the project team and the client. We plan to implement user interface modules during the next iteration.
- Initial test-cases and `formalism for specifying requirements` were implemented ahead of the schedule and they are expected to be helpful during development phase.

- Definition of Done table is prepared and will be used through the Core Capability Drive-through session.
- There is one missing team member through the CS577b course, yet the current estimation is still within the course period time limits.
- Sub-teams were formed for efficiency, they are as below:
    - Alexey Tregubov, Mihir Daptardar:
        - Solver Algorithm Implementation, Solver Library Integration, Database Implementation
    - Douglass Kinnes, Ihsan Tolga:
        - User Interface, Templates Implementation, Controller Modules, Authentication, UI-Database Integration, User Manual
    - Simone Lojeck:
        - Test Plans and Cases, CCD Preparation