Life Cycle Plan (LCP)

LEMA Pilot School Integrated Scheduling System

Team No. 12

Name	Primary Role	Secondary Role
David Wiggins	Project Manager	Developer
Aakash Shah	Prototyper	Developer
Kushalpreet Kaur	Developer	Developer
Thammanoon Kawinfruangfukul	Tester	Developer
Eunyoung Hwang	Architect	Developer
Louis Demaria	IIV&V	Developer
Mark Villanueva	QFP	Developer
Sangik Park	Developer	Developer

04/27/2012

Version History

Date	Author	Version	Changes made	Rationale
09/29/11	Harsimran Singh Benipal, Madhulika Mazumdar	1.0	First Draft. Identifying the Roles, Skills and Responsibilities of all the team members.	Initial draft with respect to LEMA Pilot School Integrated Scheduling System Version 1.0
09/30/11	Harsimran Singh Benipal	1.1	Add Introduction part	Correction after grading
10/07/11	Harsimran Singh Benipal	1.2	Fixed bug #4465; Modified skills (section 3.3)	Response to Core VC package; fixing od bugs
10/12/11	Neelima Agarwal	2.0	Identify new skills, describe milestones and products; identifying responsibility for the team members in valuation phase	Added info to comply with the minimum exit criteria for LCP within Core FC
10/18/11	Neelima Agarwal	2.2	Added Stakeholders' responsibilities, monitoring and control, updated methods and tolls, estimated resources using COTIPMO tool; Fixed Bugs	Minimum exit criteria for Draft FC package
10/18/11	Neelima Agarwal	2.3	Changes in the estimated resources	Correction in order for the modules to trace back to the requirements
10/24/2011	Neelima Agarwal	2.4	Detailed project plan for foundations phase, COTIPMO estimate, identification of skills and responsibilities for new members	Minimum exit criteria for FC package; feedback from ARB session
11/07/2011	Neelima Agarwal	2.5	Fixed bugs #5998, #6080 through #6089.	Response to evaluation of FC Package
11/21/2011	Neelima Agarwal	3.0	Updated resource estimates, according to COTIPMO iteration #6	Draft Development Commitment Package (DCP)
11/29/2011	Neelima Agarwal	3.1	Updated resource estimates ; made corrections	Draft Development Commitment Package (DCP)
12/05/2011	Neelima Agarwal	3.2	Updated estimates according to the latest COTIPMO iteration ; fixed bugs	Draft Development Commitment Package (DCP) and Response to evaluation of draft DCP
02/04/2012	Thammanoon	4.0	Updated all sections for rebasedline foundation phase	Updated information following new team structure and members
02/10/2012	Thammanoon	4.1	Updated section 2, 5, and 6	Manage inconsistency of requirements among documents
02/15/2012	Thammanoon	4.2	Update section 5	Updated cost estimate and feedback from ARB session
02/24/2012	Thammanoon	4.3	Update table of contents	Updated following IIV&Ver

Date	Author	Version	Changes made	Rationale
03/25/2012	Thammanoon	5.0	Update section 1.2 and 6	Update document following feedback from TA and add iteration assessment and CCD preparation.
04/03/2012	Thammanoon	5.1	Update section 1.2 and 6.2	Add testing and CCD results
04/11/2012	Thammanoon	5.2	Update section 1.2, 5, and 6.3 Update table 5 and 7	Update document following feedback from TA and plan for second iteration.
04/27/2012	Thammanoon	6.0	Update section 1.2, 5, 6.2 and 6.3	Assess the second iteration

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the LCP

Life Cycle Plan is used to identify the deadlines for the various phases and modules of the project. It also identifies the roles, skills and responsibilities of the various team members.
It provides a detailed description of the milestones, products or deliverables in each phase. It also specifies the life cycle management approach which helps in managing and controlling progress

1.2 Status of the LCP

This is the version 6.0 of the Life Cycle Plan and it is a part of the Initial Operational Capability Package. It identifies all the major milestones regarding the various phases of the project. This version describes the life cycle plan for the development phase, identifies modules for the system and the cost estimation based on the latest iteration of COTIPMO tool. Moreover, this version recorded the test results of capabilities in the second iteration.

1.3 Assumptions

- The duration of the project is 24 weeks, i.e. 12 weeks in Fall 2011 and 12 weeks in Spring 2012
- The team comprises of 8 members including 3 DEN students and 2 unpaid interns.
- The team members, the client and the other success-critical stakeholders are shared goals and have the strong commitments to achieve the goals.
- The team members, the client and the other success-critical stakeholders will be in regular communication with one another, so everyone will be informed of updates their progress, a decision had to be made, or an issue found.
- The requirements will not change drastically during the course of the project

2. Milestones and Products

2.1 Overall Strategy

The LEMA Pilot School Integrated Scheduling System will be following the Architected Agile Methodology because 80% of its functionality has to be developed from scratch and the rest comes from Free Timetabling Software (COTS). The team will be following Incremental Commitment Spiral Model.

Exploration phase

Duration: 09/09/11- 10/03/11

Concept: The team gets to know each other working styles and is introduced to the clients. The team will schedule several meetings with the clients in order to analyze the current system and understand the current business work flow, collect the system scope, make initial life cycle plans, perform benefit chain analysis.

The team also does feasibility analysis, identifies risks and start with initial risk management plans.

Deliverables: Valuation Commitment Package

Milestone: Valuation Commitment Review

Strategy: Scrum Development

Valuation Phase

Duration: 10/03/11- 10/26/11

Concept: The team holds WinWin negotiation sessions with the clients. All win conditions, issues, and options are to be updated on the Winbook. The stakeholders must suggest alternatives and options to address any issues. System requirements are described at a detailed level. Initial prototype is developed here, operational concept is elaborated in detail and system architecture is specified. Feasibility analysis and risk mitigation planning are to be done and the system requirements need to be prioritized.

Deliverables:

- Core Foundation Commitment Package,
- Draft Foundation Commitment Package
- Foundation Commitment Package

Milestone: Architecture Board Review-Foundation Commitment Review

Strategy: Win-win negotiation, initial prototype development, defining architecture

Foundations Phase

Duration: 10/26/11- 12/05/11

Concept: By now, negotiation is done with the client and the system requirements are clearly defined and understood. The team should spend more effort on prototyping, developing the system architecture, specifying architecture styles, patterns and frameworks, managing project quality and assessing and managing project status.

Some effort has to be spent on the transition plans and planning needed for moving to the development phase (choosing development tools, analyzing compatibility of each tools, deciding development environment, etc.)

Deliverables:

- Draft Development Commitment Package
- Development Commitment Package

Milestone: Architecture Board Review-Foundation Commitment Review

Strategy: prototype development, detailed architecture

Rebaselined Foundation Phase

Duration: 01/09/2012 – 02/08/2012

Concept: Updating requirements if it is necessary. Create shared vision and transfer knowledge to new team members. The team revises all requirements, design, and artifacts, and prepares test and transition plans in order to ensure that the project is ready for development phase.

Deliverables: Rebaselined Development Commitment Package

Milestone: Rebaslined Development Commitment Review

Strategy: One short Incremental Commitment Cycle.

Development phase (Construction iteration):

Duration: 02/09/2012 - 04/09/2012

Concept: Develop and implement the system based on previous design and architecture. The development team is able to reuse the prototypes to speed up the process and reduce the risks since the prototypes are developed and tested. Before leaving from this phase, all functions need to finish development and testing, and most of the risks and defects have to eliminate in order to be ready for transition.

Deliverables:

- Initial Operational Capability (IOC) Package
- Core Capability Drive-Thru Report
- Draft Transition Readiness Review Package

Milestones: Core Capability Drive-through

Strategy: Mostly implementation and testing.

Development phase (Transition iteration):

Duration: 04/10/2012 – 04/27/2012

Concept: This involves installing and deploying the system at the operational environment. Training should be provided to the clients and users, so that the users of the system can use it successfully.

Deliverables:

- Initial Operational Capability (IOC) Package
- Support and Transition Set Package
- Close Out Report
- Project Archiving

Milestones: Transition Readiness Review

Strategy: Installation, Transition and training.

2.2 Project Deliverables

This section describes all of the artifacts the team is responsible for delivering and its due date, format, and medium.

2.2.1 Exploration Phase

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	09/29/11	.doc, .pdf	Soft copy
Valuation Commitment Package • Operational Concept Description (OCD) Early Section • Life Cycle Plan (LCP) Early Section • Feasibility Evidence Description (FED) Early Section	09/30/11	.doc, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy
WikiWinWin Report	------	Text	Soft copy
Project Plan	Every Wednesday	.mpp	Soft copy
Initial prototype	------	.doc, .pdf	Soft copy
Evaluation of Valuation Commitment Package	10/03/2011	.xls	Soft Copy; Bugzilla

2.2.2 Valuation Phase

Table 2: Artifact deliverable in Valuation Phase

Artifact	**Due date**	**Format**	**Medium**
Core Foundations Commitment Package • Operational Concept Description (OCD) • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype (PRO) • System and Software Architecture Description (SSAD) • System and Software	10/12/2011	.doc, .pdf	Soft copy

Requirements Definition (SSRD) • Supporting Information Document (SID)			
Progress Report	Every Wednesday	.xls	Soft copy
Project Plan	Every Wednesday	.mpp	Soft copy
Evaluation of Core Commitment Package	10/12/2011	.xls	Soft copy; Bugzilla
Effort Report	Every Monday	-	Effort Reporting system
Draft Foundations Commitment Package • Operational Concept Description (OCD) • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype (PRO) • System and Software Architecture Description (SSAD) • System and Software Requirements Definition (SSRD) • Supporting Information Document (SID)	10/17/2011	.doc, .pdf	Soft copy
Evaluation of Draft FC Package	10/17/2011	.xls	Soft copy; Bugzilla
Foundation Commitment Package: • Operational Concept Description (OCD) • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Supporting Information Document (SID) • System and Software Architecture Description (SSAD) • UML Model	10/24/2011	.doc, .pdf	Soft copy
Evaluation of Foundation Commitment Package	10/31/2011	.xls	Soft copy; Bugzilla

2.2.3 Foundations Phase

Table 3: Artifact deliverable in Foundations Phase

Artifact	Due date	Format	Medium
Draft Development Commitment Package • Operational Concept Description (OCD) • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype (PRO) • System and Software Architecture Description (SSAD) • System and Software Requirements Definition (SSRD) • Supporting Information Document (SID) • Quality Management Plan (QMP) • Test Plan (TP) • Iteration Plan • Acceptance Test Plan (ATP)	11/21/2011	.doc, .pdf	Soft copy
Development Commitment Package • Operational Concept Description (OCD) • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype (PRO) • System and Software Architecture Description (SSAD) • System and Software Requirements Definition (SSRD) • Supporting Information Document (SID)	12/05/2011	.doc, .pdf	Soft copy

• Quality Management Plan (QMP) • Test Plan (TP) • Iteration Plan Acceptance Test Plan (ATP)			
Effort Report	Every Monday	--	Effort Reporting system
Progress Report	Every Wednesday	.xls	Soft Copy
Project Plan	Every Wednesday	.mpp	Soft copy

2.2.4 Rebaselined Development Phase

Table 4: Artifact deliverable in Development Phase

Artifact	Due date	Format	Medium
Draft Rebaselined Development Commitment (RDC) Package • Operational Concept Description (OCD) • System and Software Requirements Definition (SSRD) • System and Software Architecture Description (SSAD) • UML Model • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Supporting Information Document (SID) • Quality Management Plan (QMP) • Test Plan and Cases (TPC) • Transition Plan (TP)	02/08/2012	.doc, .pdf	Soft copy
Prototyping: • User interface prototypes • REST API • OAUTH API • Interface with COTS	02/09/2012	.php, .ep	Soft copy

Rebaselined Development Commitment Package • Operational Concept Description (OCD) • System and Software Requirements Definition (SSRD) • System and Software Architecture Description (SSAD) • UML Model • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Supporting Information Document (SID) • Quality Management Plan (QMP) • Test Plan and Cases (TPC) • Transition Plan (TP)	02/15/2012	.doc, .pdf	Soft copy
Evaluation of Rebaselined Development Commitment Package	02/27/2012	.doc, .pdf, Bugzilla	Soft copy, Bugzilla
Effort Report	Every Monday	--	Effort Reporting system
Progress Report	Every Wednesday	.xls	Soft Copy
Project Plan	Every Wednesday	.mpp	Soft copy

2.2.5 Development Phase

Table 5: Artifact deliverable in Development Phase

Artifact	Due date	Format	Medium
Develop system modules: - Online application for students subject registration - Locking certain classes - Student course requests are viewed/approved/rejecte	03/15/2012	.php	Soft copy

d by counselor - Schedule generation - Map teachers to course sections - View alternate schedules - LAUSD constraints - Detailed Database			
Develop system modules: - Authentication to the system - Providing Information to team 04 - Integration with family accountability system	03/22/2012	.php	Soft copy
Test Result and Feedback • Unit testing • Integrated testing	03/24/2012	.doc, .pdf	Soft copy
Initial Operational Capability (IOC) Package Working set #1 • Operational Concept Description (OCD) • System and Software Requirements Definition (SSRD) • System and Software Architecture Description (SSAD) • UML Model • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Supporting Information Document (SID) • Quality Management Plan (QMP) • Test Plan and Cases (TPC) • Transition Plan (TP) • Test Procedure and Results (TPR)	03/26/2012	.doc, .pdf	Soft copy

Core Capability Drive-Thru Report • Core Capability Drive-Thru document (CCD) • Code Count • Code Count Output • COCOMOII Estimation Uncertainty • COCOMOII Output	04/04/2012	.doc, .pdf, .csv	Soft copy
Evaluation of Initial Operational Capability (IOC) Package Working set #1	04/04/2012	.doc, .pdf, Bugzilla	Soft copy, Bugzilla
Draft Transition Readiness Review Package • Transition Plan (TP) • User Manual (UM) • Support Plan (SP) • Training Materials (TM) • Regression Test Package (RTP)	04/09/2012	.doc, .pdf	Soft copy
Develop system modules: - Track the progress of a particular student - Student Progress Enhanced - GPA for various universities - Level of Normalization	04/17/2012	.php	Soft copy
Test Result and Feedback • Unit testing • Integrated testing	04/20/2012	.doc, .pdf	Soft copy
Support and Transition Set Package • Transition Plan (TP) • User Manual (UM) • Support Plan (SP) • Training Materials (TM)	04/16/2012	.doc, .pdf	Soft copy

• Regression Test Package (RTP)			
Evaluation of Support and Transition Set Package	04/23/2012	.doc, .pdf, Bugzilla	Soft copy, Bugzilla
• Initial Operational Capability (IOC) Package Working set #2 • Support and Transition Set Package	04/27/2012	.doc, .pdf	Soft copy
Software system installation • Source Code Files • Executable Components	04/28/2012	.php, .twig, .exe	Soft copy
Close Out Report Project Archiving • Source Code Files • Executable Components • Release Description	05/04/2012	.doc, .pdf, .php, .twig, .exe	Soft copy
Effort Report	Every Monday	--	Effort Reporting system
Progress Report	Every Wednesday	.xls	Soft Copy
Project Plan	Every Wednesday	.mpp	Soft copy

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities

The following table shows the responsibilities of each stakeholder during the project development.

Table 6: Project Specific stakeholder's responsibilities

Role	Responsibilities
All stakeholders	▪ Participate in WinWin negotiation sessions, commitment reviews and weekly meetings ▪ Collaborate and take responsibility for tasks assigned ▪ Commit to the agreed project progress
Client (Mrs. Beth Kennedy, LEMA Pilot School)	▪ Arrange for site visit, and collaborate with the development team Specify win conditions and explain operation concept ▪ Coordinate among user and developer ▪ Review and test the product and provide feedback ▪ Test and deploy the product in operational environment ▪ Support system's transition ▪ Receive training for the new system, provide training for regular users
System Users – Students, Counselor, Teachers, Administrator (Ms. Roberta Mailman)	▪ Explain current business workflow and context ▪ Express win conditions ▪ Review and test prototypes and the product and provide feedback ▪ Test and deploy the product in operational environment
IIV&V	▪ Help in conducting WinWin Sessions ▪ Verify and validate documents and provide feedback to the development team ▪ Ensure the quality of the project

Development team	▪ Get all win conditions from stakeholders ▪ Understand the system and convert the project-related information into requirements, operational concepts, initial architecture, and prototype ▪ Develop project plan, ROI analysis, provide feasibility evidence ▪ Analyze current system environment, identify project risks, analyze project feasibility and mitigate risks ▪ Develop software and project artifacts to meet milestone requirements ▪ Interact with the various stakeholders for better understanding of the system ▪ Participate in Win Win sessions; identify issues/concerns and suggest alternatives/ options ▪ Schedule weekly client meetings and team meetings ▪ Meet the client weekly for showing them various system & documents and taking their reviews about the system ▪ Work together as a team aiming towards a common goal ▪ Plan and conduct testing ▪ Develop the system based on the agreed architecture ▪ Perform system transition, provide training for client and maintainer ▪ Provide product support in operational environment to customer
Maintainer (LEMA school)	▪ Express interests or win conditions ▪ Provide information and show current intranet. ▪ Provide information and feedback, review and test the intranet in early version ▪ Prepare operational environment ▪ Test and deploy the product in operational environment ▪ Receive instruction for the new system, provide training for users ▪ Maintain the system

3.2 Responsibilities by Phase

Table 7: Stakeholder's Responsibilities in each phase

Team Member / Role	**Primary / Secondary Responsibility**					
	Exploration	**Valuation**	**Foundations**	**Rebaselined Foundation**	**Development-** Construction Iteration	**Development-** Transition Iteration
Neelima Agarwal: Project Manager/ Software Architect	**Project Manager :** -Define, plan, assign, and identify tasks & priorities for each team member -Record project proceedings -Record project progress	**Project Manager :** - Planning and control -Record project progress and individual effort - Identify and resolve risks -Life cycle planning **Software Architect:** - Analyze the Proposed System architecture	**Project Manager :** - Planning and assessing project progress - Identify risks -Discuss with clients **Software Architect:** - Specify architecture styles, patterns and frameworks	**N/A**	**N/A**	**N/A**
Aakash Shah: Prototyper/ Feasibility Analyst/ Builder	**Prototyper:** - Search for a COTS - Explore the various available technologies.	**Feasibility Analyst:** - Assess and plan to mitigate risks **Prototyper:** - Develop initial prototype	**Feasibility Analyst:** - Assess feasibility evidence **Prototyper:** - Develop prototype -Analyze and prioritize capabilities to prototype	**Builder:** - Assess capabilities - Study COTS - Implement REST services	**Builder:** - Develop the system components - Perform unit testing	**Builder:** - Deploy the system - Final Project Deliverable
Madhulika Mazumdar: Operational Concept Engineer	**Operational Concept Eng:** - Making the OCD and LCP - Contributing in Progress report	**Operational Concept Eng:** - Identify objectives, constraints and priorities - Develop operational concept	**Operational Concept Eng:** - Identify organizational and operational transformations - Assess operational concept	**N/A**	**N/A**	**N/A**
Eunyoung Hwang: Software Architect/ Requirements Engineer/	**Software Architect:** - Constructing and Maintaining the website	**Software Architect:** - Analyze the Proposed System architecture	**Software Architect:** -Describing the architecture, styles, and	**Builder:** - Assess capabilities - Develop user interface prototypes	**Builder:** - Develop the system components - Perform unit testing	**Builder:** - Deploy the system - Final Project Deliverable

Builder/ Trainer	**Requirements Engineer:** - Help in requirement Analysis	**Requirements Engineer:** - Analyze the current and proposed system	patterns -Defining technology dependent architecture **Requirements Engineer:** **-** Re-evaluate the Win Conditions based on client's feedback	**Software Architect:** - Assess architecture and detail design		**Trainer:** - Prepare and train the system content - Transition the system
Harsimran Benipal: Requirements Engineer/ Life Cycle Planner	**Requirements Engineer:** - Making the CIR - Getting System Requirements and making the SSRD. **Life Cycle Planner:** - Making LCP	**Requirements Engineer:** - Negotiate Win conditions	**Requirements Engineer:** - Re-negotiate Win Conditions based on client's feedback **Life Cycle Planner:** **-** Estimate project effort and schedule	**N/A**	**N/A**	**N/A**
Louis DeMaria: IIV&V / Shaper/ Quality Focal Point/ Builder/ Tester	**IIV&V:** - facilitation in win win sessions - verifying the requirements	**IIV&V:** - Manage Project Quality - Verifying & validation of products or deliverables **Shaper:** - Setup WinWin negotiation context - Negotiate WIOA - Capture new system items and project context	**IIV&V:** **-** Verify and validate documents using Bugzilla **Quality Focal Point:** **-** Construct traceability matrix - Identify configuration management strategy **Shaper:** - Negotiate the Win conditions to validate the feasibility of requirements - Capture new system items and project context	**IIV&V:** **-** Verify and validate documents recording to Bugzilla **Builder:** - Assess capabilities	**IIV&V:** - Verify and validate the work products **Builder:** - Develop the system components - Perform unit testing **Tester:** - Test modules/ system and record test results	**IIV&V:** **-** Manage issues and defects - Verify and validate the work products **Builder:** - Deploy the system - Final Project Deliverable **Tester:** - Test system and record test results
Tianyin Zhou: Prototyper	**N/A**	**N/A**	**Prototyper:** **-** Develop prototype -Prioritize capabilities to prototype	**N/A**	**N/A**	**N/A**

David Wiggins: Project Manager/ Feasibility Analyst/ Builder/ Trainer	**N/A**	**N/A**	**N/A**	**Project Manager:** - Define detail project plan - Track Progress **Feasibility Analyst:** - Assess and evaluate COTS - Track risks throughout life cycle **Builder:** **-** Develop user interface prototypes	**Project Manager:** - Define detail project plan - Track Progress **Builder:** - Develop the system components - Perform unit testing	**Project Manager:** - Define detail project plan - Track Progress **Builder:** - Deploy the system - Final Project Deliverable **Trainer:** - Prepare and train the system content - Transition the system
Thammanoon Kawinfruangfukul: Requirements Engineer/ Life Cycle Planner/ Builder/Tester/ Trainer	**N/A**	**N/A**	**N/A**	**Requirements Engineer:** - Define detail requirements - Handle requirement changes **Life Cycle Planner:** - Assess life cycle plan - Define detail life cycle plan - Define iteration and support plan **Builder:** **-** Develop user interface prototypes - Setup development environment - Study Symfony	**Life Cycle Planner:** - Define detail life cycle plan **Builder:** - Develop the system components - Perform unit testing **Tester:** - Test modules/ system and record test results	**Life Cycle Planner:** - Define detail life cycle plan **Builder:** - Deploy the system - Final Project Deliverable **Tester:** - Test system and record test results **Trainer:** - Prepare and train the system content - Transition the system
Mark Villanueva: Quality Focal Point/ Builder/ Tester	**N/A**	**N/A**	**N/A**	**Quality Focal Point:** - Assess quality management plan and strategies **Builder:** - Study Symfony - Implement REST services	**Quality Focal Point:** - Identify test plan **Builder:** - Develop the system components - Perform unit testing **Tester:** - Test modules/ system , record test results	**Builder:** - Deploy the system - Final Project Deliverable **Tester:** - Test system and record test results

Kushalpreet Kaur: Builder	**N/A**	**N/A**	**N/A**	**Builder:** - Study Symfony **-** Interface with COTS	**Builder:** - Develop the system components - Perform unit testing	**Builder:** - Deploy the system - Final Project Deliverable
Sangik Park: Builder	**N/A**	**N/A**	**N/A**	**Builder:** - Study Symfony **-** Interface with database and Symfony	**Builder:** - Develop the system components - Perform unit testing	**Builder:** - Deploy the system - Final Project Deliverable

3.3 Skills

Table 8: Skills identified for development team (CS577a)

Name	Roles	Skills
Neelima Agarwal	Project Manager/ Software Architect	**Technical Skils:** C, C++, Java, JS, XML, HTML, MySQL -Planning and managerial skills -Inter-personal Skills -Good at assessing and identifying project risks, defects, and concerns -Team Management skills -Mentoring skills -Risk mitigation planning skills -Conflict Resolution Skills -Good at partnering with end users, and clients to ensure progress in the right direction **-** UML modeling skills -Skilled at analyzing the current and proposed system architecture
Aakash Shah	Prototyper / Feasibility Analyst	**Technical Skills:** C#,J#, C, C++, HTML, DB design and SQL scripting to develop the prototype -Mentoring skills - Skilled at analyzing components for implementation and risk -Assessment skills for understanding the system - Skilled in Silverlight UI, C#, C,C++, DB design and SQL scripting to develop the prototype -Good analytical skills for analyzing the cost and benefits of the project **-**NDI analysis skills for checking feasibility - Effective Communication skills

Madhulika Mazumdar	Operation Concept Engineer / Project Manager	**Technical Skills:** C,C++, Java, HTML, XML, JS -Skilled at identifying the goals, restrictions, and priorities of the system -Good at evaluation of the new concepts for the proposed system as well as the current system -Effective communication skills -Team collaboration skills -Inter-personal skills.
Harsimran Singh Benipal	Requirements Engineer / Life Cycle Planner	**Technical Skills** C, C++, C#, ASP.NET, SQL Server, MySQL -Negotiation skills -Analytical skills to understand and comprehend the proposed system -Resource management skills -Listening skills -Communication skills -Team collaboration skills
Eunyoung Hwang	Software Architect / Requirements Engineer	**Technical Skills** Java, SQL Server , Javascript , C, C++, J#, HTML **-**Skilled in system analysis - Analytical skills to understand how the users will interact with the system -Skilled at mapping requirements to architecture and detailed designing. -Problem solving skills - Listening skills -Communication skills -UML modeling skills
Tianyin Zhou	Prototyper	**Technical Skills:** Python, PHP - Team collaboration skills - Listening skills -Communication skills -Analytical skills to understand and comprehend the proposed system

Table 9: Skills identified for development team (CS577b)

Name	Roles	Skills
David Wiggins	• Project Manager • Feasibility Analyst • Builder • Trainer	**Current skills:** • PHP, CSS, HTML, MySQL • Project management skills • People skills • Ability to handle conflicts • Ability to analyze business and investment • Ability to assess and mitigate risks • Communication skills **Required skills:** • Symfony
Aakash Shah	• Builder	**Current skills:** • C#,J#, C, C++, HTML, DB design and SQL scripting to develop the prototype • Mentoring skills • Skilled at analyzing components for implementation and risk • Assessment skills for understanding the system • Skilled in Silverlight UI, C#, C,C++, DB design and SQL scripting to develop the prototype • Good analytical skills for analyzing the cost and benefits of the project • NDI analysis skills for checking feasibility • Effective Communication skills **Required skills:** • Symfony • PHP
Eunyoung Hwang	• Software Architect • Requirements Engineer • Builder	**Current skills:** • Java, SQL Server , Javascript , C, C++, J#, HTML • Skilled in system analysis • Analytical skills to understand how the users will interact with the system • Skilled at mapping requirements to architecture and detailed designing. • Problem solving skills • Listening skills • Communication skills • UML modeling skills **Required skills:** • Symfony • PHP

Thammanoon Kawinfruangfukul	• Requirements Engineer • Life Cycle Planner • Builder • Tester • Trainer	**Current skills** • Planning skills • Project coordination • COTIPMO • MS project • UML, RSM • PHP, MySQL • Communication skills • Analytical skills • Negotiation skills **Required skills** • Symfony • Testing methodology • Bug tracking and removal techniques
Mark Villanueva	• Quality Focal Point • Builder • Tester	**Current skills** • Analytical skills • Communication skills • Testing methodology • Bug tracking and removal techniques • UML, RSM • WinBook • PHP, HTML **Required skills:** • Symfony • PHP
Louis Demaria	• IIV&V • Shaper • Quality Focal Point • Builder • Tester	**Current skills** • C, C++ , HTML, J script, JAVA, Visual Basic, C++, C#, J#, MySQL • Communication and negotiation skills • Analytical skills to understand the system and finding any potential errors. • Listening skills • Communication skills • Team collaboration skills • Inter-personal skills • Defect tracking and reporting skills **Required skills:** • Symfony • PHP
Kushalpreet Kaur	• Builder	**Current skills** • PHP, CSS, HTML, MySQL • Analytical skills • Communication skills **Required skills:** • Symfony

Sangik Park	• Builder	**Current skills** • PHP, CSS, HTML, MySQL • Analytical skills • Communication skills **Required skills:** • Symfony

4. Approach

4.1 Monitoring and Control

The team is adopting various approaches for monitoring and control. Weekly meetings with the client provide the necessary feedback, monitoring and control. Further, regular sync ups with the team members through meetings, and emails allow for effective communication. The summary of report will be recorded on the meeting minutes sent to group email, progress report, effort report, and project plan.

4.1.1 Closed Loop Feedback Control

- Weekly Progress reports identifying risks and risk mitigation plans; defects, problems and concerns and progress in terms of tasks completed last week and tasks to be completed in the following week. Any deviation from the baseline is identified, its severity is analyzed and action, as appropriate, is taken.
- Weekly detailed project plan stating the plan for the coming weeks and tasks for each of the team members. Moreover, MS Project plan gives the tasks to be completed by calendar year, this also helps the manager to record any lags and if any, compensate them as soon as possible, so as to avoid future major delays.
- Weekly effort reports for assessing the number of hours spent per week by the development team members on the project development.
- Regular team meetings to discuss and plan the tasks for the coming week and for follow-ups on the tasks completed in between two consecutive meetings
- Use of emails, phone calls, wikispace, and CSCI 577 Team 12 Google group page(http://groups.google.com/group/csci577-team12) for effective communication.

4.1.2 Reviews

Reviews and recommendations for every artifact are done at various levels.
First, IIV&V activities are carried out and IIV&V files a bug report in case there is a bug in the artifact being reviewed. Then, the artifacts are reviewed and graded by the teaching staff members. The team also uses Architecture Review Boards which are performed after every major milestone of project development.

Sometimes, the team uses peers review where team members review each other’s work and provide suggestions.

4.2 Methods, Tools and Facilities

Table 10: Methods, Tools and Facilities

Tools	Usage	Provider
Bugzilla	Utilized to detect bugs, and errors	USC
CSE Effort Reporting System and iCard system	Record and keep track of the cumulative weekly effort reports for each team member	USC
Google Docs	Utilized for cohesive collaboration on documents	Google
Google Groups	Tool for communication and collaboration among the team members	Google
Project Website	Uploading documentation	USC
Rational Software Modeler	Create UML models (use-case diagrams, sequence diagrams, etc) for SSAD	USC
Microsoft Word	Create documents for the documentation purposes	Developers
Microsoft Project	Used for documenting detailed weekly project plan, defining tasks for each team member and setting milestones or deadlines	USC
Microsoft PowerPoint	Create presentation for ARB sessions	Developers
Winbook	Facilitates and support WinWin Negotiation	USC
COTIPMO	Cost Estimation tool	USC
Subversion	Configuration manager	CollabNet

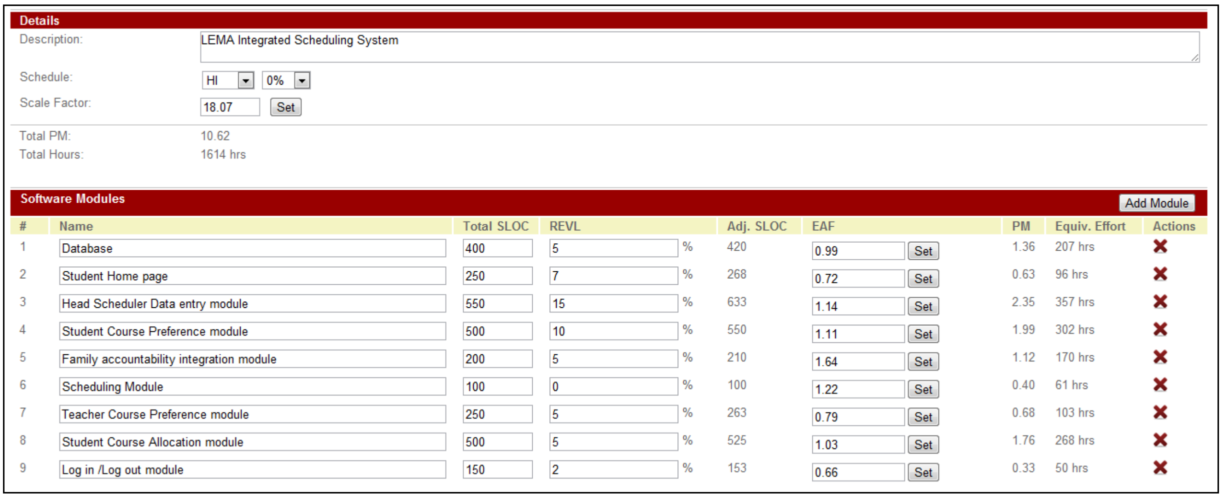
5. Resources

Estimate on the software cost (estimated for Iteration #14 using COTIPMO)

- Estimated CSCI577a Effort : 7 team members at 20 hrs/week for 12 weeks
- Estimated CSCI577b Effort : 8 team members at 8 hrs/week for 12 weeks
- Estimated effort from COTIPMO Iteration #10 = 4.31 PM (656 hours)
- 8 person development team is required to complete all the requirements of the project (Estimation required team members are 3.02 (6.04 /1.67) and there are 8 team members, so the project is feasible enough to be delivered)
- Project duration – 24 weeks
- Component modules in your development project :
    - Student Progress Tracking module
    - Student Course Preference module
    - Student Course Allocation module
    - Teacher Course Allocation module
    - csv import/export for integration
    - Head Scheduler data entry module
    - Family accountability integration module
    - Database
- Programming language used: PHP
- Development framework used: Symfony 2.0

Initial Estimates:

Figure 1: Software Modules (Initial Estimates)



Details

Description: LEMA Integrated Scheduling System

Schedule: HI 0%

Scale Factor: 18.07 Set

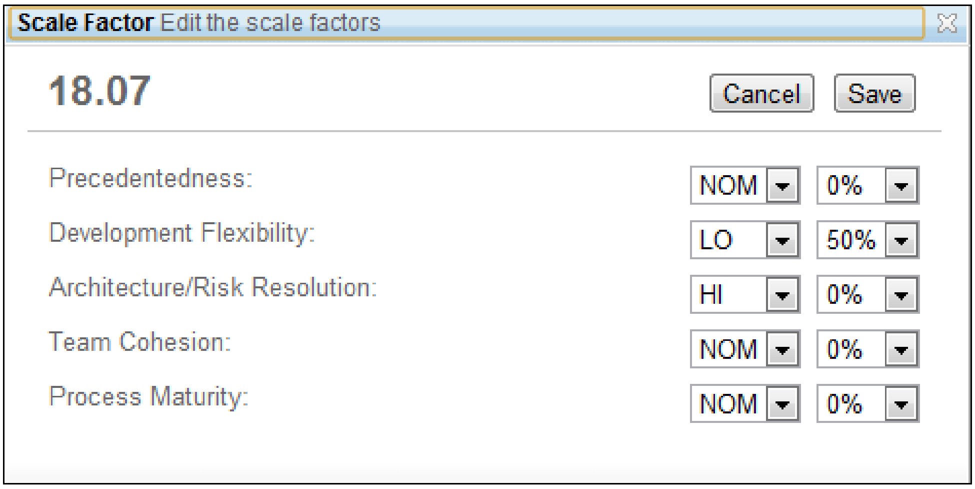
Total PM: 10.62

Total Hours: 1614 hrs

Software Modules Add Module

#	Name	Total SLOC	REVL		Adj. SLOC	EAF		PM	Equiv. Effort	Actions
1	Database	400	5	%	420	0.99	Set	1.36	207 hrs	✖
2	Student Home page	250	7	%	268	0.72	Set	0.63	96 hrs	✖
3	Head Scheduler Data entry module	550	15	%	633	1.14	Set	2.35	357 hrs	✖
4	Student Course Preference module	500	10	%	550	1.11	Set	1.99	302 hrs	✖
5	Family accountability integration module	200	5	%	210	1.64	Set	1.12	170 hrs	✖
6	Scheduling Module	100	0	%	100	1.22	Set	0.40	61 hrs	✖
7	Teacher Course Preference module	250	5	%	263	0.79	Set	0.68	103 hrs	✖
8	Student Course Allocation module	500	5	%	525	1.03	Set	1.76	268 hrs	✖
9	Log in /Log out module	150	2	%	153	0.66	Set	0.33	50 hrs	✖

Figure 2: Scale factors

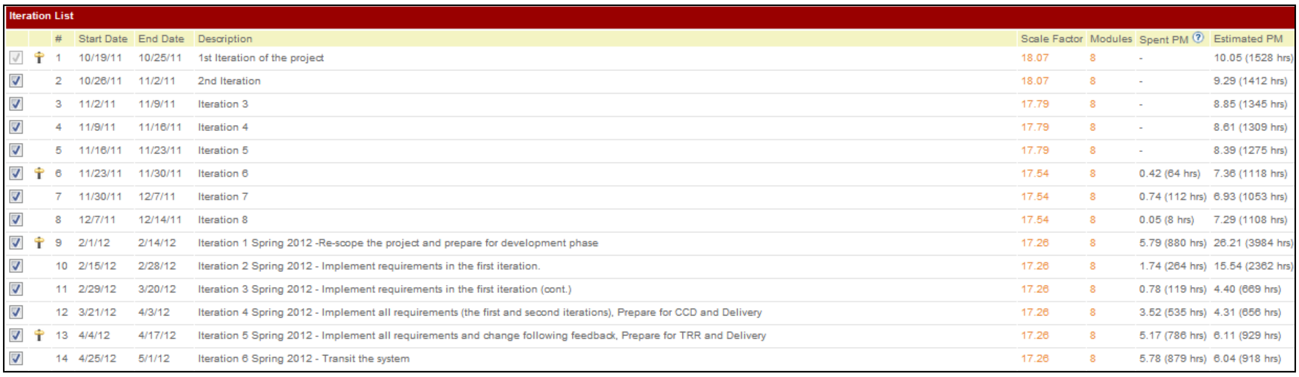


Scale Factor Edit the scale factors

18.07

Precedentedness:	NOM	0%
Development Flexibility:	LO	50%
Architecture/Risk Resolution:	HI	0%
Team Cohesion:	NOM	0%
Process Maturity:	NOM	0%

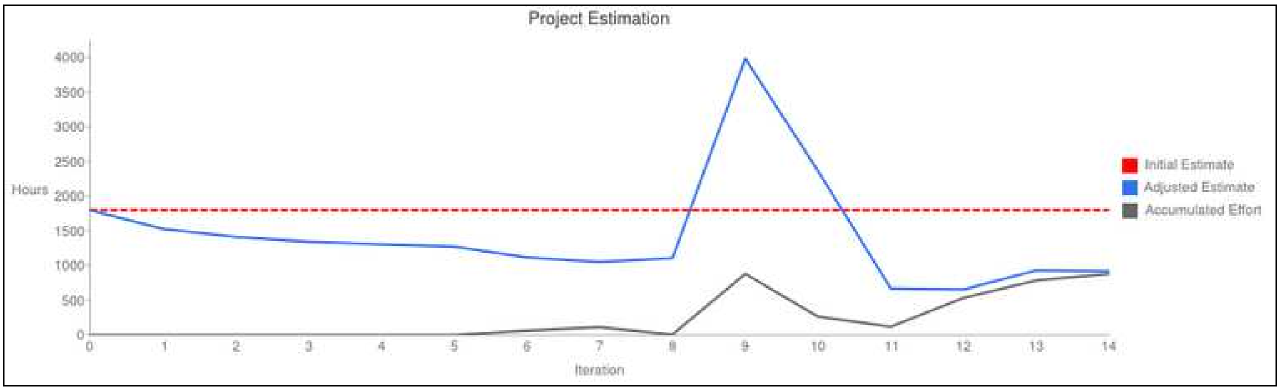
COTIPMO Iteration List



Iteration List

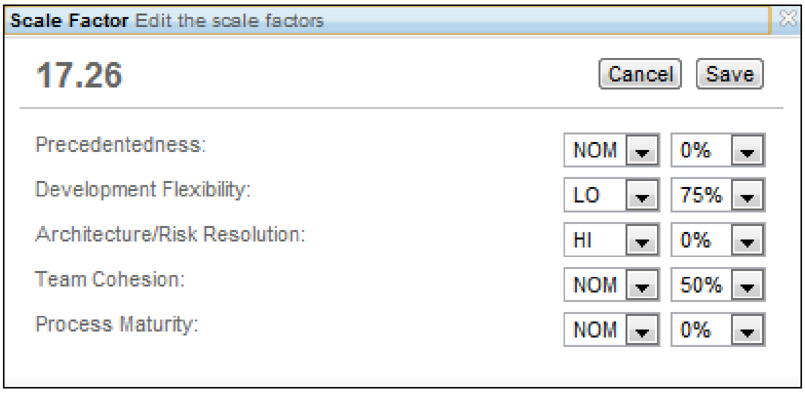
		#	Start Date	End Date	Description	Scale Factor	Modules	Spent PM	Estimated PM
☑	📌	1	10/19/11	10/25/11	1st Iteration of the project	18.07	8	-	10.05 (1528 hrs)
☑		2	10/26/11	11/2/11	2nd Iteration	18.07	8	-	9.29 (1412 hrs)
☑		3	11/2/11	11/9/11	Iteration 3	17.79	8	-	8.85 (1345 hrs)
☑		4	11/9/11	11/16/11	Iteration 4	17.79	8	-	8.61 (1309 hrs)
☑		5	11/16/11	11/23/11	Iteration 5	17.79	8	-	8.39 (1275 hrs)
☑	📌	6	11/23/11	11/30/11	Iteration 6	17.54	8	0.42 (64 hrs)	7.36 (1118 hrs)
☑		7	11/30/11	12/7/11	Iteration 7	17.54	8	0.74 (112 hrs)	6.93 (1053 hrs)
☑		8	12/7/11	12/14/11	Iteration 8	17.54	8	0.05 (8 hrs)	7.29 (1108 hrs)
☑	📌	9	2/1/12	2/14/12	Iteration 1 Spring 2012 -Re-scope the project and prepare for development phase	17.26	8	5.79 (880 hrs)	26.21 (3984 hrs)
☑		10	2/15/12	2/28/12	Iteration 2 Spring 2012 - Implement requirements in the first iteration.	17.26	8	1.74 (264 hrs)	15.54 (2362 hrs)
☑		11	2/29/12	3/20/12	Iteration 3 Spring 2012 - Implement requirements in the first iteration (cont.)	17.26	8	0.78 (119 hrs)	4.40 (669 hrs)
☑		12	3/21/12	4/3/12	Iteration 4 Spring 2012 - Implement all requirements (the first and second iterations), Prepare for CCD and Delivery	17.26	8	3.52 (535 hrs)	4.31 (656 hrs)
☑	📌	13	4/4/12	4/17/12	Iteration 5 Spring 2012 - Implement all requirements and change following feedback, Prepare for TRR and Delivery	17.26	8	5.17 (786 hrs)	6.11 (929 hrs)
☑		14	4/25/12	5/1/12	Iteration 6 Spring 2012 - Transit the system	17.26	8	5.78 (879 hrs)	6.04 (918 hrs)

Project Progress Graph



New estimates based on COTIPMO Iteration#14 and survey results:

Scale Factors



Scale Factor Edit the scale factors

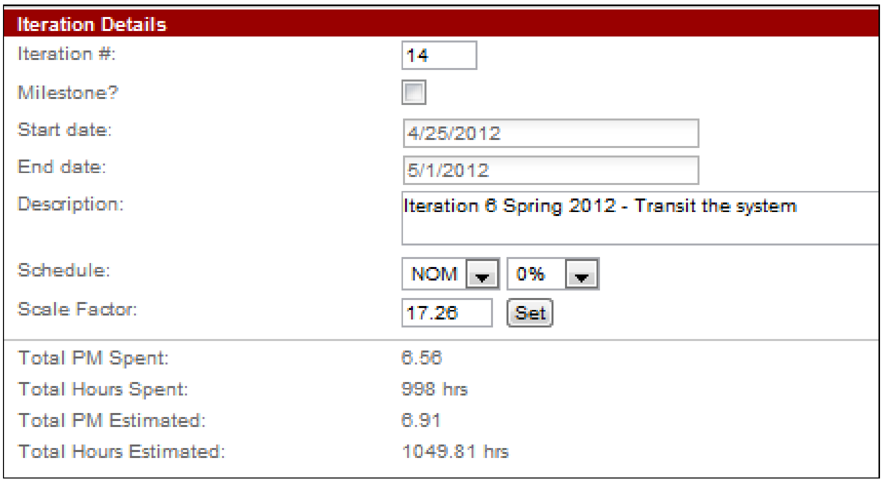
17.26

Precedentedness:	NOM	0%
Development Flexibility:	LO	75%
Architecture/Risk Resolution:	HI	0%
Team Cohesion:	NOM	50%
Process Maturity:	NOM	0%

Table 11: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	NOM	The product is somewhat similar to several that have been developed before
FLEX	LOW+75%	There is considerable flexibility on the requirements, schedules, and interfaces. However, most of the requirements are a must have.
RESL	HIGH	Risk resolution is done extensively throughout the project lifecycle. A good amount of effort is spent in reviewing architecture and mitigating risk.
TEAM	NOM+50%	The team members have a communication and meetings and the team uses many collaborative tools to synchronize among team members.
PMAT	NOM	The level of detail done for 577 roughly corresponds to CMM level 2. Nominal amount of project management and life cycle tasks are performed.

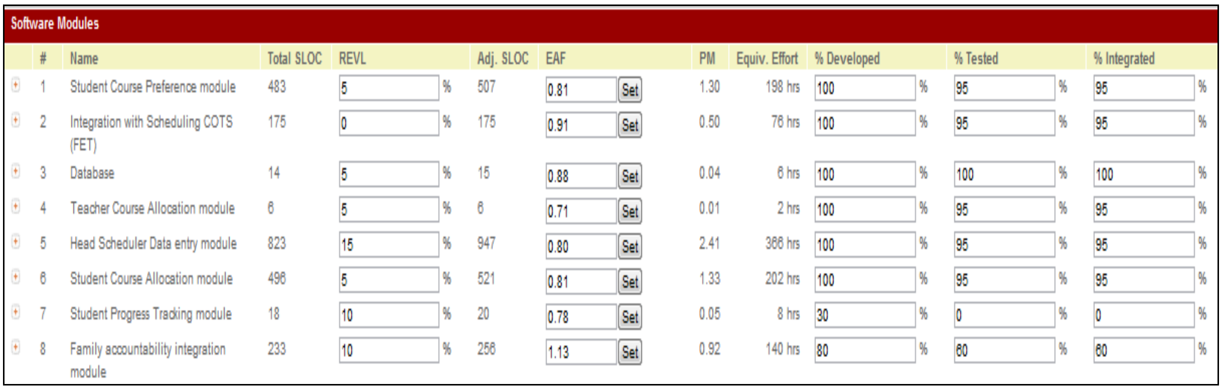
Iteration #14 Resource Details



Iteration Details

Iteration #:	14
Milestone?	□
Start date:	4/25/2012
End date:	5/1/2012
Description:	Iteration 6 Spring 2012 - Transit the system
Schedule:	NOM 0%
Scale Factor:	17.26
Total PM Spent:	6.56
Total Hours Spent:	998 hrs
Total PM Estimated:	6.91
Total Hours Estimated:	1049.81 hrs

Software Modules (Iteration #14):



Software Modules

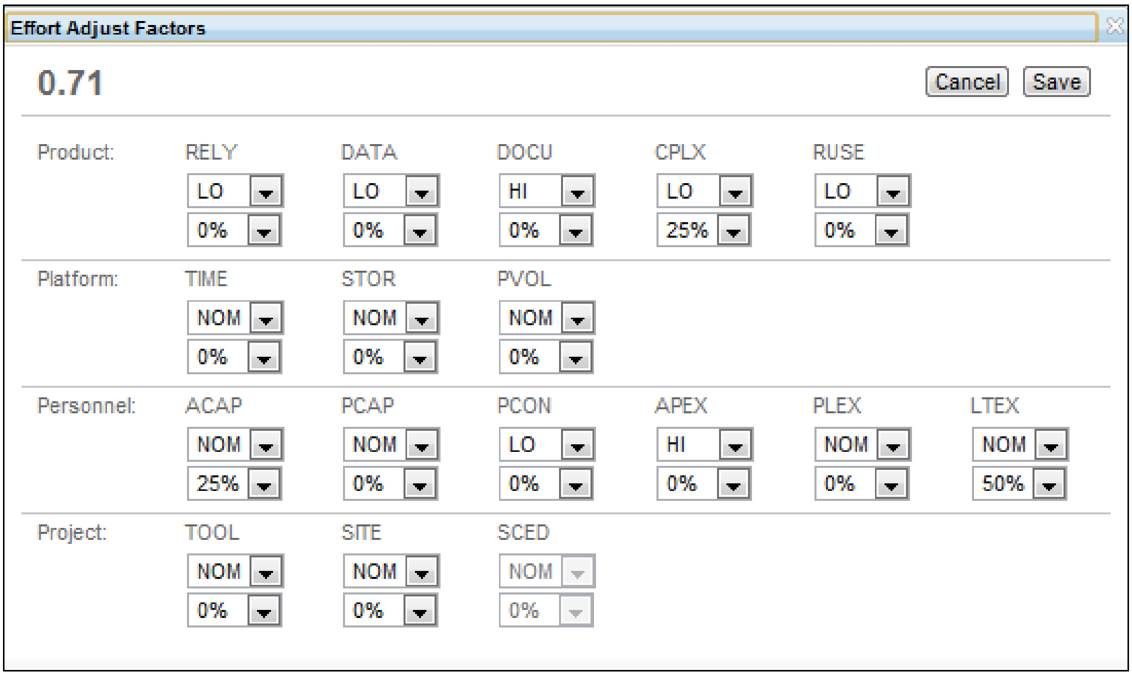
#	Name	Total SLOC	REVL	Adj. SLOC	EAF	PM	Equiv. Effort	% Developed	% Tested	% Integrated
1	Student Course Preference module	483	5 %	507	0.81	1.30	198 hrs	100 %	95 %	95 %
2	Integration with Scheduling COTS (FET)	175	0 %	175	0.91	0.50	76 hrs	100 %	95 %	95 %
3	Database	14	5 %	15	0.88	0.04	6 hrs	100 %	100 %	100 %
4	Teacher Course Allocation module	6	5 %	6	0.71	0.01	2 hrs	100 %	95 %	95 %
5	Head Scheduler Data entry module	823	15 %	947	0.80	2.41	366 hrs	100 %	95 %	95 %
6	Student Course Allocation module	496	5 %	521	0.81	1.33	202 hrs	100 %	95 %	95 %
7	Student Progress Tracking module	18	10 %	20	0.78	0.05	8 hrs	30 %	0 %	0 %
8	Family accountability integration module	233	10 %	256	1.13	0.92	140 hrs	80 %	60 %	60 %

Note: The *csv import/export for integration* is the module for integration with Columbia (legacy system used by LAUSD) and the scheduling COTS (FET)

Table 12: Cost Drivers - Teacher Course Allocation Module

Cost Driver	Value	Rationale
RELY	LOW	Effect of software failure causes little inconvenience and losses can be recovered easily.
DATA	LOW	The data size is small. The database contains around maximum of 50 teachers.
DOCU	HIGH	The team produces and maintains documents. Comparing with project’s complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	LOW+25%	Simple UI designs; no direct hardware operations; simple queries for retrieval and update to database; no complex data structure designs
RUSE	LOW	Not developing for product line; independent standalone application; very little component reuse.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. A lot of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite good. There is a considerable amount of cooperation and communication among team members
PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project’s annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	HIGH	All computer science students in the development team, so good amount of application experience of the development team. All team members have experiences about web application and related domain.
LTEX	NOM+50%	Most of the team members have experience with programming languages such as PHP, Java and databases like MySQL, GUI development etc. Six members out of eight are experienced in PHP.
PLEX	NOM	Team has nominal experience with development of database (MySQL) and GUIs.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: Teacher Course Allocation Module



Effort Adjust Factors

0.71

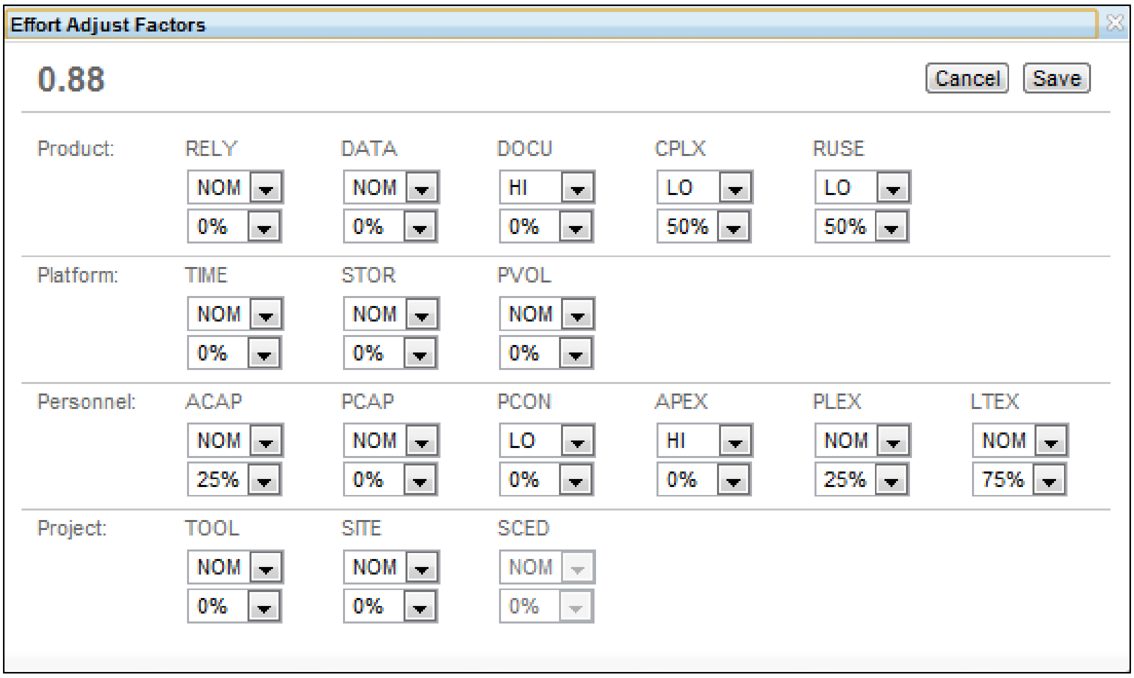
Product:	RELY	DATA	DOCU	CPLX	RUSE	
	LO	LO	HI	LO	LO	
	0%	0%	0%	25%	0%	
Platform:	TIME	STOR	PVOL			
	NOM	NOM	NOM			
	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	PLEX	LTEX
	NOM	NOM	LO	HI	NOM	NOM
	25%	0%	0%	0%	0%	50%
Project:	TOOL	SITE	SCED			
	NOM	NOM	NOM			
	0%	0%	0%			

Table 13: Cost Drivers - Database Module

Cost Driver	Value	Rationale
RELY	NOM	Effect of software failure causes moderate inconvenience and losses can be recovered easily. Database should be reliable for the various operations to be carried out successfully.
DATA	NOM	The system needs to maintain database for 450 students. The database also has to maintain the course related data for students, teacher preferences, and constraints.
DOCU	HIGH	The team produces and maintains documents. Comparing with project's complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	LOW+50%	Simple UI designs; no direct hardware operations; simple queries for retrieval and update to database; no complex data structure designs
RUSE	LOW + 50%	Not developing for product line; independent standalone application; very little component reuse.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. Lots of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite good. There is a considerable amount of cooperation and communication among team members

PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project’s annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	HIGH	All computer science students in the development team, so good amount of application experience of the development team. All team members have experiences about web application and related domain.
LTEX	NOM + 75%	Most of the team members have experience with programming languages such as PHP, Java and databases like MySQL, GUI development etc.
PLEX	NOM+25%	Team has nominal experience with development of database (MySQL) and GUIs.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: Database Module



0.88

Product: RELY NOM 0% DATA NOM 0% DOCU HI 0% CPLX LO 50% RUSE LO 50%

Platform: TIME NOM 0% STOR NOM 0% PVOL NOM 0%

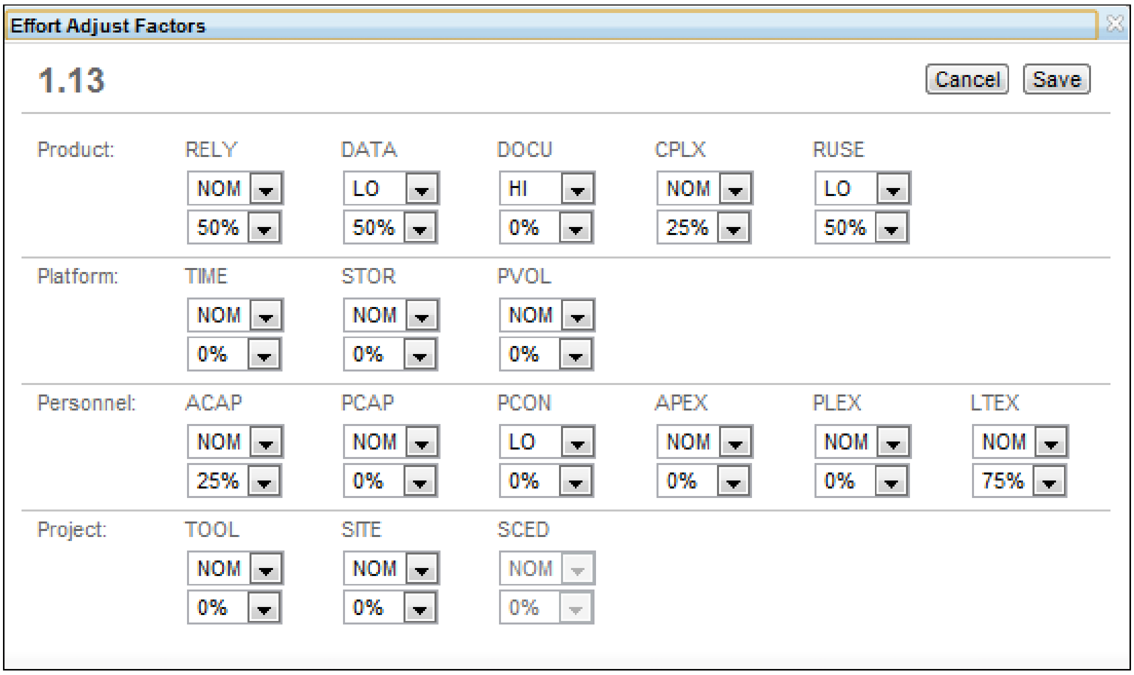
Personnel: ACAP NOM 25% PCAP NOM 0% PCON LO 0% APEX HI 0% PLEX NOM 25% LTEX NOM 75%

Project: TOOL NOM 0% SITE NOM 0% SCED NOM 0%

Table 14: Cost Drivers - Family Accountability Integration Module

Cost Driver	Value	Rationale
RELY	NOM+50%	Effect of software failure causes moderate inconvenience and losses can be recovered easily. Reliability should be on the higher side because this module has to interface with the external system to get information from the external database. Failure can cause the system to be unusable.
DATA	LOW+50%	The system needs to maintain database for 450 students. However, we need very specific information from the external system such as student ID, course ID etc.
DOCU	HIGH	The team produces and maintains documents. Comparing with project's complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	NOM+25%	Complex DB queries and updates, simple callbacks, distributed processing
RUSE	LOW+50%	Not developing for product line; independent standalone application; very little component reuse. But, might be re-used across project.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. Lots of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite good. There is a considerable amount of cooperation and communication among team members
PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project's annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	NOM	All computer science students in the development team, so some amount of application experience of the development team. But, most of the members don't have experience with REST services.
LTEX	NOM+75%	Most of the team members have experience with programming languages such as PHP, Java and databases like MySQL, GUI development etc. Six members out of eight are experienced in PHP.
PLEX	NOM	Team has nominal experience with development of database (MySQL) and GUIs.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: Family Accountability Integration Module



Effort Adjust Factors

1.13

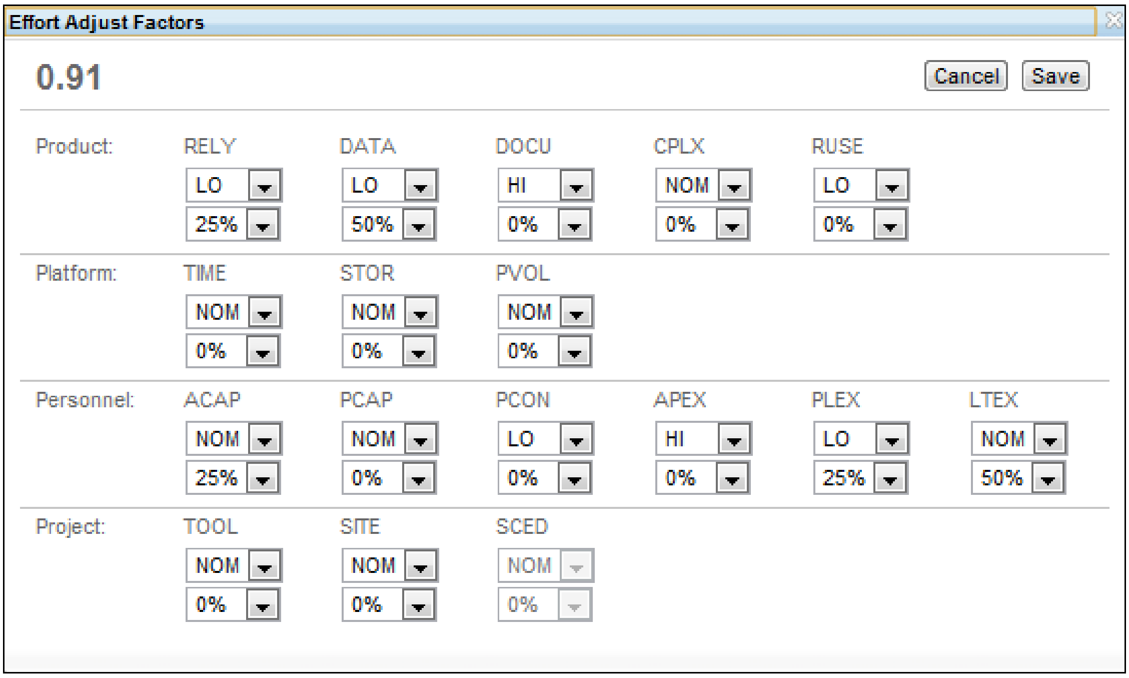
Product:	RELY	DATA	DOCU	CPLX	RUSE	
	NOM	LO	HI	NOM	LO	
	50%	50%	0%	25%	50%	
Platform:	TIME	STOR	PVOL			
	NOM	NOM	NOM			
	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	PLEX	LTEX
	NOM	NOM	LO	NOM	NOM	NOM
	25%	0%	0%	0%	0%	75%
Project:	TOOL	SITE	SCED			
	NOM	NOM	NOM			
	0%	0%	0%			

Table 15: Cost Drivers – csv import/export for integration

Cost Driver	Value	Rationale
RELY	LOW+25%	Effect of software failure causes little inconvenience and losses can be recovered easily.
DATA	LOW+50%	It uses data related to course, teacher preferences, constraints, student courses, rooms etc.
DOCU	HIGH	The team produces and maintains documents. Comparing with project's complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	NOM	Simple UI designs; no direct hardware operations; simple queries for retrieval and update to database; no complex data structure designs
RUSE	LOW	Not developing for product line; independent standalone application; very little component reuse.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. Lots of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite good. There is a considerable amount of cooperation and communication among team members

PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project’s annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	HIGH	All computer science students in the development team, so good amount of application experience of the development team. All team members have experiences about web application and related domain.
LTEX	NOM+50%	Most of the team members have experience with programming languages such as PHP, Java, Perl and databases like MySQL, GUI development etc. Six members out of eight are experienced in PHP.
PLEX	LOW+25%	Team has nominal experience with development of database (MySQL) and GUIs. Most developers don’t have experience with COTS products.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: csv import/export for integration



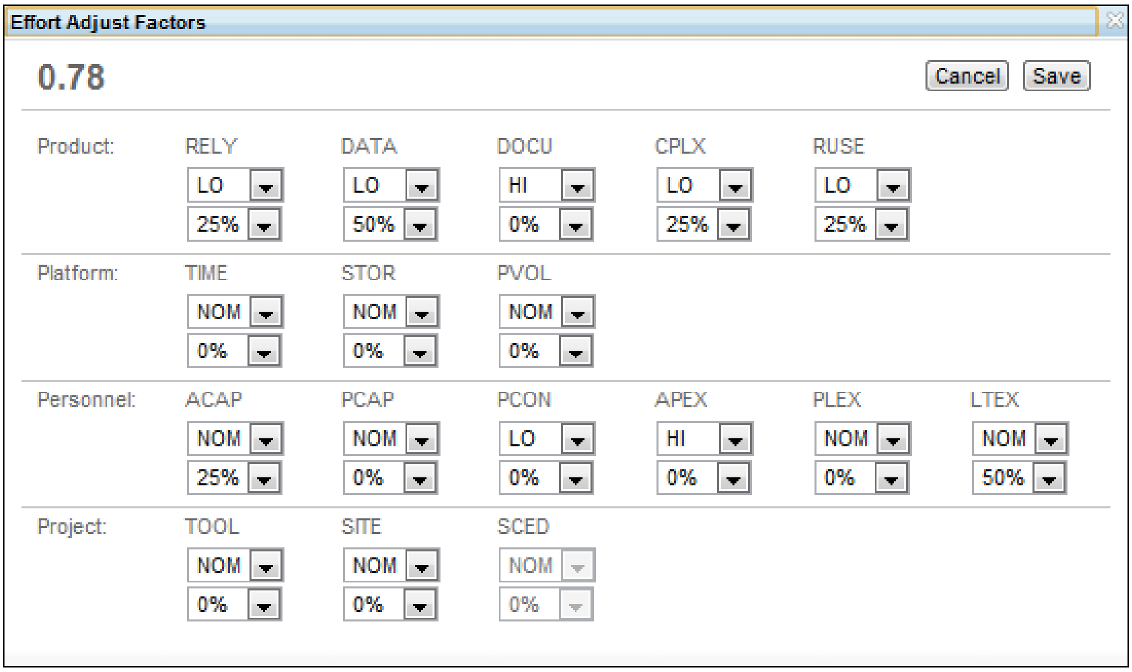
0.91

Product:	RELY	DATA	DOCU	CPLX	RUSE	
	LO	LO	HI	NOM	LO	
	25%	50%	0%	0%	0%	
Platform:	TIME	STOR	PVOL			
	NOM	NOM	NOM			
	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	PLEX	LTEX
	NOM	NOM	LO	HI	LO	NOM
	25%	0%	0%	0%	25%	50%
Project:	TOOL	SITE	SCED			
	NOM	NOM	NOM			
	0%	0%	0%			

Table 16: Cost Drivers - Student Progress Tracking

Cost Driver	Value	Rationale
RELY	LOW+25%	Effect of software failure causes little inconvenience and losses can be recovered easily.
DATA	LOW+50%	The system needs to maintain database for 450 students. The database also has to maintain the course related data for students and their grades.
DOCU	HIGH	The team produces and maintains documents. Comparing with project's complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	LOW+25%	Simple UI designs; no direct hardware operations; simple queries for retrieval and update to database; no complex data structure designs
RUSE	LOW+25%	Not developing for product line; independent standalone application; very little component reuse.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. Lots of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite high. The efficiency and throughput of the team members, level of cooperation and communication is nominal.
PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project's annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	HIGH	All computer science students in the development team, so good amount of application experience of the development team. All team members have experiences about web application and related domain.
LTEX	NOM+50%	Most of the team members have experience with programming languages such as PHP Java and databases like MySQL, GUI development etc. Six members out of eight are experienced in PHP.
PLEX	NOM	Team has nominal experience with development of database (MySQL) and GUIs.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: Student Progress Tracking



0.78

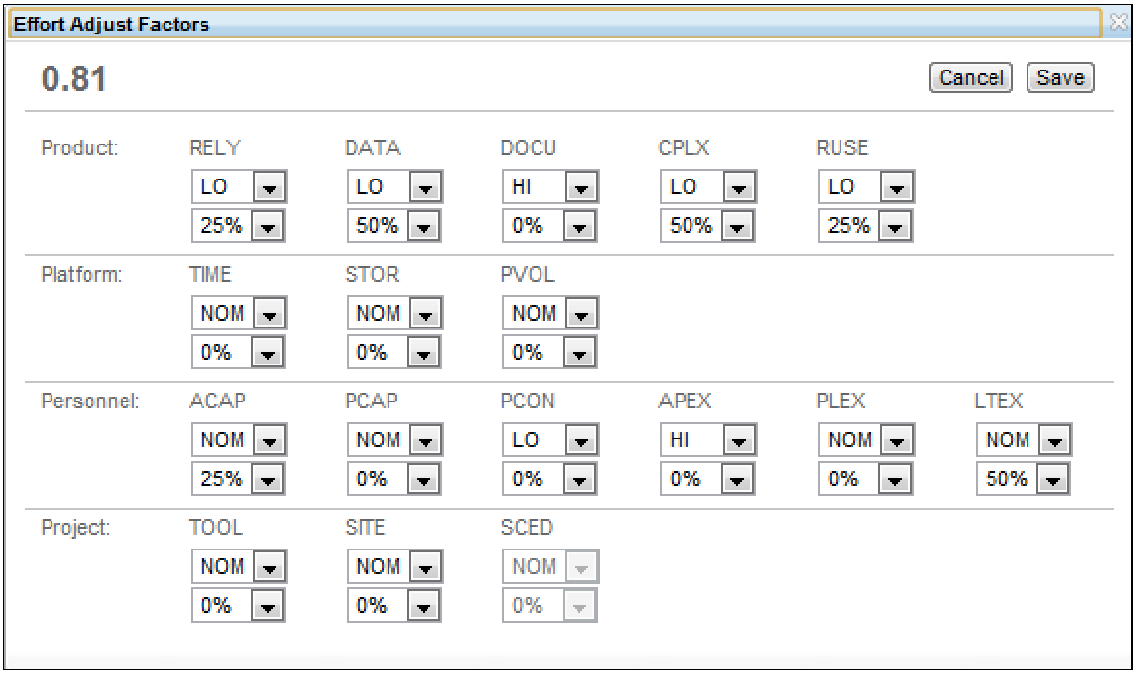
Product:	RELY	DATA	DOCU	CPLX	RUSE	
	LO	LO	HI	LO	LO	
	25%	50%	0%	25%	25%	
Platform:	TIME	STOR	PVOL			
	NOM	NOM	NOM			
	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	PLEX	LTEX
	NOM	NOM	LO	HI	NOM	NOM
	25%	0%	0%	0%	0%	50%
Project:	TOOL	SITE	SCED			
	NOM	NOM	NOM			
	0%	0%	0%			

Table 17: Cost Drivers - Student Course Allocation module

Cost Driver	Value	Rationale
RELY	LOW+25%	Effect of software failure causes little inconvenience and losses can be recovered easily.
DATA	LOW+50%	The data size is small. The database contains around maximum of 50 teachers and course details.
DOCU	HIGH	The team produces and maintains documents. Comparing with project's complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	LOW+50%	Simple UI designs; no direct hardware operations; simple queries for retrieval and update to database; no complex data structure designs
RUSE	LOW+25%	Not developing for product line; independent standalone application; very little component reuse.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. Lots of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite high. The efficiency and throughput of the team members, level of cooperation and communication is nominal.

PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project’s annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	HIGH	All computer science students in the development team, so good amount of application experience of the development team. All team members have experiences about web application and related domain.
LTEX	NOM+50%	Most of the team members have experience with programming languages such as PHP, Java and databases like MySQL, GUI development etc. Six members out of eight are experienced in PHP.
PLEX	NOM	Team has nominal experience with development of database (MySQL) and GUIs.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: Student Course Allocation module



0.81

Product: RELY LO 25%; DATA LO 50%; DOCU HI 0%; CPLX LO 50%; RUSE LO 25%

Platform: TIME NOM 0%; STOR NOM 0%; PVOL NOM 0%

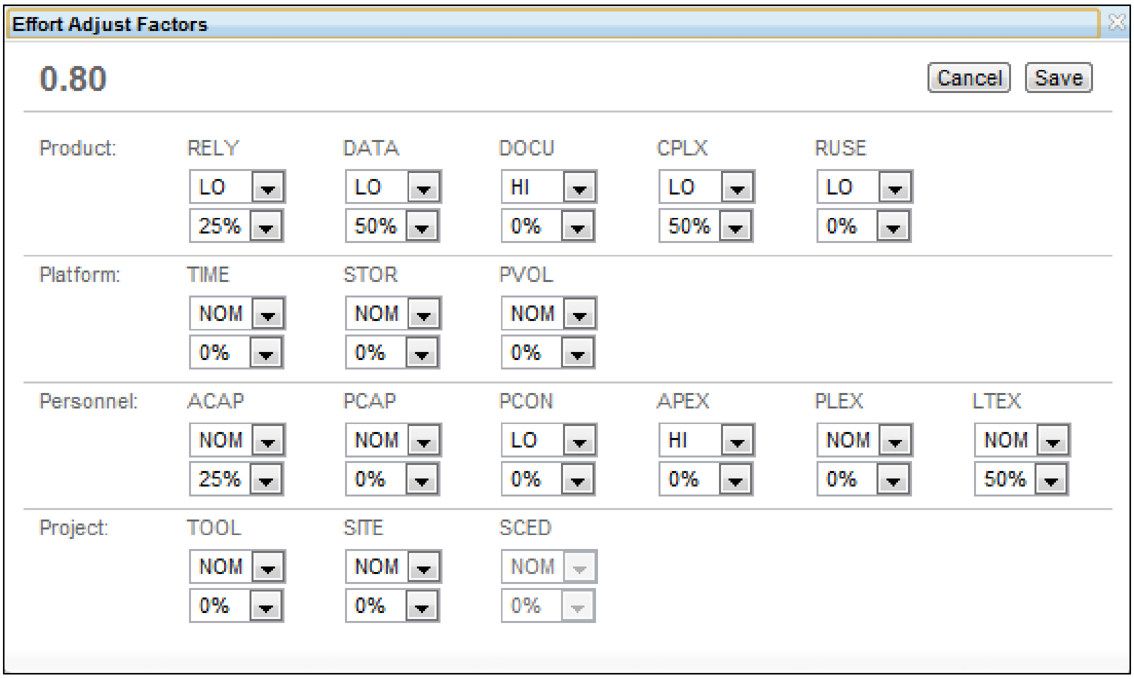
Personnel: ACAP NOM 25%; PCAP NOM 0%; PCON LO 0%; APEX HI 0%; PLEX NOM 0%; LTEX NOM 50%

Project: TOOL NOM 0%; SITE NOM 0%; SCED NOM 0%

Table 18: Cost Drivers - Head Scheduler Data entry module

Cost Driver	Value	Rationale
RELY	LOW+25%	Effect of software failure causes little inconvenience and losses can be recovered easily.
DATA	LOW+50%	The database contains teacher, student, and course information. This module is only used for data entry and not for retrieval of data.
DOCU	HIGH	The team produces and maintains documents. Comparing with project's complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	LOW+50%	Simple UI designs; no direct hardware operations; simple queries for retrieval and update to database; no complex data structure designs
RUSE	LOW	Not developing for product line; independent standalone application; very little component reuse.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. Lots of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite high. The efficiency and throughput of the team members, level of cooperation and communication is nominal.
PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project's annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	HIGH	All computer science students in the development team, so good amount of application experience of the development team. All team members have experiences about web application and related domain.
LTEX	NOM+50%	Most of the team members have experience with programming languages such as PHP, Java and databases like MySQL, GUI development etc. Six members out of eight are experienced in PHP.
PLEX	NOM	Team has nominal experience with development of database (MySQL) and GUIs.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: Head Scheduler Data entry module



Effort Adjust Factors

0.80

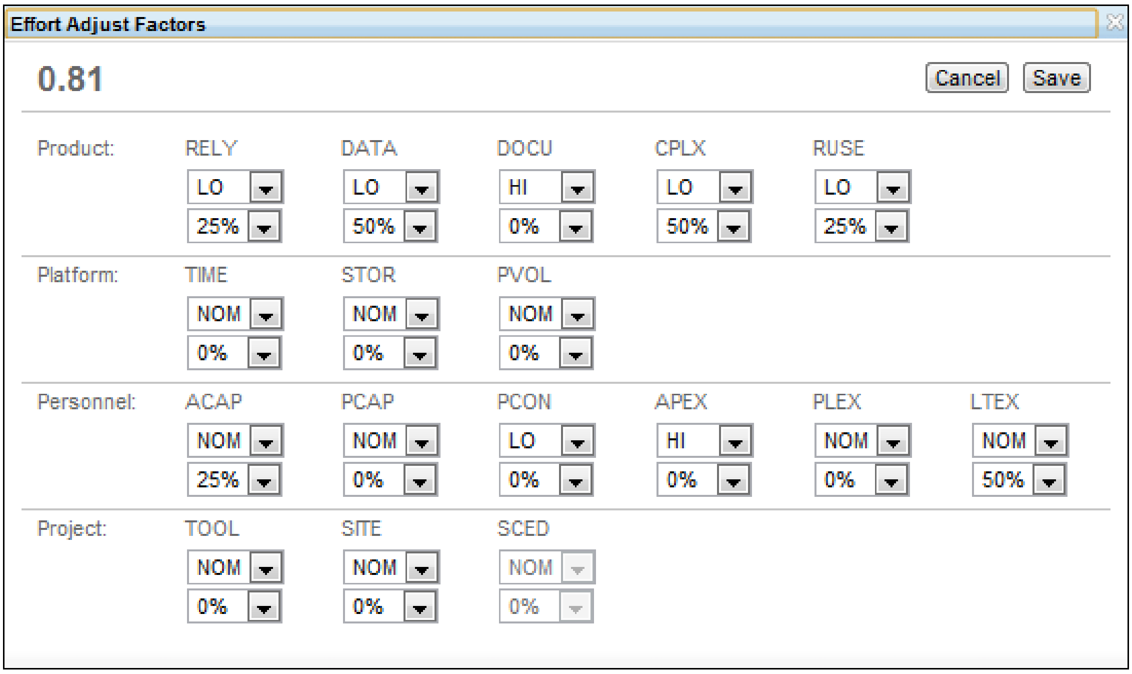
Product:	RELY	DATA	DOCU	CPLX	RUSE	
	LO	LO	HI	LO	LO	
	25%	50%	0%	50%	0%	
Platform:	TIME	STOR	PVOL			
	NOM	NOM	NOM			
	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	PLEX	LTEX
	NOM	NOM	LO	HI	NOM	NOM
	25%	0%	0%	0%	0%	50%
Project:	TOOL	SITE	SCED			
	NOM	NOM	NOM			
	0%	0%	0%			

Table 19: Cost Drivers - Student Course Preference module

Cost Driver	Value	Rationale
RELY	LOW+25%	Effect of software failure causes little inconvenience and losses can be recovered easily.
DATA	LOW+50%	The database contains 450 students and course details.
DOCU	HIGH	The team produces and maintains documents. Comparing with project's complexity, there are excessive amount of documentation for life-cycle needs.
CPLX	LOW+50%	Simple UI designs; no direct hardware operations; simple queries for retrieval and update to database; no complex data structure designs
RUSE	LOW+25%	Not developing for product line; independent standalone application; very little component reuse.
TIME	NOM	Execution time constraint is not very strict and is expected to be less than or equal to 50%
STOR	NOM	It is expected to be less than 50% use of available storage. Little computation resource is required by the system. Lots of available execution time and main storage makes this of less relevance.
PVOL	NOM	The system will use stable platforms (i.e. stable OS and DBMS)
ACAP	NOM+25%	The ability in analysis, design and requirements gathering is quite high. The efficiency and throughput of the team members, level of cooperation and communication is nominal.

PCAP	NOM	Some team members have industrial experience in dealing and developing various COTS package, SDKs, and frameworks.
PCON	LOW	The project’s annual personnel turnover is around 25%/year. Only two people will continue to the 577b
APEX	HIGH	All computer science students in the development team, so good amount of application experience of the development team. All team members have experiences about web application and related domain.
LTEX	NOM+50%	Most of the team members have experience with programming languages such as PHP Java and databases like MySQL, GUI development etc. Six members out of eight are experienced in PHP.
PLEX	NOM	Team has nominal experience with development of database (MySQL) and GUIs.
TOOL	NOM	Basic lifecycle tools for development and report are used. Some software engineering aid tools like Bugzilla, COCOMO tool, etc. are also used
SITE	NOM	The team uses many collaborative tools, such as skype, google group, and wikispaces to cooperate among team members. Some of team members live different cities and time zones.
SCED	NOM	Schedule of the project is fixed i.e. 24 weeks. So, schedule stretch-out is 100% .

EAF: Student Course Preference module



0.81

Product:	RELY	DATA	DOCU	CPLX	RUSE	
	LO	LO	HI	LO	LO	
	25%	50%	0%	50%	25%	
Platform:	TIME	STOR	PVOL			
	NOM	NOM	NOM			
	0%	0%	0%			
Personnel:	ACAP	PCAP	PCON	APEX	PLEX	LTEX
	NOM	NOM	LO	HI	NOM	NOM
	25%	0%	0%	0%	0%	50%
Project:	TOOL	SITE	SCED			
	NOM	NOM	NOM			
	0%	0%	0%			

6. Iteration Plan

6.1 Plan

The development of the project is divided into two iterations, and the main milestone is the Core Capability Drive-through (CCD). The capabilities which have higher priorities and risks will be implemented in the first iteration, and the less priority capabilities will be implemented in the second iteration, which occurs after the CCD.

The development phase has a transition iteration which comes after the construction iteration. It begins at the end of the construction iteration two and continues until the end of the semester. The transition iteration accounts for the transition of the system and training if the users and clients.

A detailed plan for the construction and transition iterations of the development phase has been documented in the project plan. Major milestones and products for each iteration can be found in the life cycle plan.

Estimated dates for each iteration are shown below:
- Construction iteration 1: 02/10/2012 - 03/23/2012
- Construction iteration 2: 03/24/2012 - 04/09/2012
- Transition iteration: 04/10/2012- 04/27/2012

6.1.1 Capabilities to be implemented

Iteration 1:
The capabilities to be implemented in the first iteration will be the capability requirements that have a high priority, with low ease of implementation or are required for development of other capability requirements. Also, since the system needs to be integrated with the system being developed by team 4, the capabilities which are required for integration of the two systems are also implemented during the first iteration.

Table 20: Capabilities to be implemented in the iteration 1

Requirement	Description	Priority
CR- 1: Online application for students subject registration	Students should be able to fill in their choices for various subjects.	M (Must have)
CR-2: locking certain classes.	The administration should be able to lock certain classes to certain time frames	M (Must have)
CR-3: Student course requests are viewed/approved/rejected by counselor	The counselor can review all the courses requested by the student and can edit or approve or disapprove the student request	M (Must have)

CR-4: Schedule generation	The system helps the administration in scheduling a time table for the school	M (Must have)
CR-7: Map teachers to course sections	Teachers are assigned a particular course section by the scheduler	M (Must have)
CR-8: View alternate schedules	The administrator can view alternate schedules	M (Must have)
CR-10: LAUSD constraints	The various LAUSD constraints should be entered into the database through a simple UI	M (Must have)
CR-11: Detailed Database	Database to contain student id, course details and teachers id	M (Must have)
LOS-1: System usability	The system will provide an intuitive User interface in order to provide ease of usability	M (Must have)
SR-1: Use given templates by LEMA Pilot School	It is advised to make the user interface according to the given templates and it should be user friendly	M (Must have)
SR-2: Integration with family accountability system	The system should interact with the system being developed by team 04 and get teacher information, student information, and authenticated token	M (Must have)

Iteration 2:
The capabilities to be implemented in the second iterations are all the remaining capability requirements.

Table 21: Capabilities to be implemented in the iteration 2

Requirement	Description	Priority
CR-4: Schedule generation	The system helps the administration in scheduling a time table for the school Note: this requirement has been developed in the first iteration but it cannot satisfy the requirement, so this requirement needs to be redeveloped and tested.	M (Must have)
CR-10: LAUSD constraints	The various LAUSD constraints should be entered into the database through a simple UI Note: this requirement has been developed in the first iteration but it cannot satisfy the requirement, so this requirement needs to be redeveloped and tested.	M (Must have)
SR-3: Authentication to the system	The user logs in from the interface from team 04 and in an authenticated session from where it is redirected to our system. Thus we check the cookie to confirm that they were in an authenticated session. The cookie would have user type, user id and authentication key. Note: this must have requirement cannot be delivered at the first iteration because it need to be integrated with	M (Must have)

	team 4, but team 4 cannot finish related functionality by CCD	
LOS-1: System usability	The system will provide an intuitive User interface in order to provide ease of usability	M (Must have)
CR- 5 : Track the progress of a particular student	Track student progress from A-G requirements for mapping	S (Should have)
CR- 6: Student Progress Enhanced	Ability of students to see how many more credits they need to complete a particular grade	S (Should have)
CR-9: GPA for various universities	Displays how much GPA is required for CSU, U of California and other universities	S (Should have)
CR-12: Level of Normalization	The introduction of a school code which will enable the scheduling system to be ubiquitous	W (Want to have)

6.1.2 Capabilities to be tested

All the capability requirements that are implemented need to be tested. The list is the same as given in tables above respective to the iterations for which they have been mentioned.

6.1.3 Capabilities not to be tested

There is no capability that does not need to be tested. All capabilities require testing and modules need to be integrated and tested as well.

6.1.4 CCD Preparation Plans

6.1.4.1 CCD Stakeholders

LEMA Pilot School Integrated Scheduling System provides four types of users which are scheduler, counselor, teacher, and student. The client and maintainer are included as CCD stakeholders.

All stakeholders who will be involved in Core Capability Drive-through are summarized as followed.

Table 22: CCD Stakeholders

Stakeholder name	Role
Dr. Supannika Koolmanojwong	• Instructor
Pongtip Aroonvatanaporn	• TA
Roberta Mailman	• Client • Scheduler • Teacher
Beth Kennedy	• Client

	- Maintainer - Counselor - Student
Thammanoon Kawinfruangfukul	- Developer
Eunyoung Hwang	- Developer
David Wiggins	- Developer
Mark Villanueva	- Developer - Tester
Louis Demaria	- IIV&V - Tester
Kushalpreet Kaur	- Developer
Sangik Park	- Developer

6.1.4.2 CCD usage scenarios

The usage scenarios of each capability in the first iteration are shown below.

Table 23: CCD usage scenarios

Requirement	Description
CR- 1: Online application for students subject registration	1. The student goes to student course registration page 2. The student selects courses which are shown following student grade level. 3. The student clicks at confirm button
CR-2: locking certain classes.	After the scheduler selects courses that will be open in the current semester, the scheduler can click at lock button to set available courses and allow student to register.
CR-3: Student course requests are viewed/approved/rejected by counselor	After the student selects and registers courses, the counselor can review all the student registered courses. 1. The counselor goes to summary page 2. The counselor clicks a link on student row that the counselor would like to review. 3. The counselor can change selected courses and approval status. 4. The counselor clicks at confirm button
CR-4: Schedule generation	After the scheduler import activity list csv file to FET, the scheduler can add constrains and generate the schedule.
CR-7: Map teachers to course sections	1. The scheduler goes to teacher assignment page in schedule management 2. The scheduler selects the current semester 3. The scheduler assigns the teacher to the course. 4. The scheduler can click save button to continue finishing teacher assignment later or can click lock button to enable export activity to csv file in Import/Export page 5.

CR-8: View alternate schedules	After scheduler generates the schedule with FET, FET provides website to see possible schedules.
CR-10: LAUSD constraints	Overall student grade level courses: 1. The scheduler goes to constrains management page to set overall minimum and maximum number of students for student grade level courses 2. The scheduler clicks save button Specific courses: 1. The scheduler goes to course profile page 2. The scheduler clicks a link under course name to go to edit course profile page 3. The scheduler sets minimum and maximum number of students 4. The scheduler clicks save button
CR-11: Detailed Database	The scheduler inputs all data using the phpmyadmin UI into the database.
LOS-1: System usability	The user uses the system to justify the system usability
SR-1: Use given templates by LEMA Pilot School	The user uses the system to review that user interface follow LEMA Pilot School Templates.
SR-2: Integration with family accountability system	The user is able to see information exist in family accountability system

6.1.4.3 CCD preparation plans

1. Implement and test all capabilities in the first iteration, using both informal unit testing and formal testing with test plan by testers.
2. Prepare sample data on the development server for using in the CCD session. The data is provided by the clients and include course information and user id.
3. Prepare user manual and feedback form for all participants
4. In the CCD session, the team introduces and explains the capabilities, test data, and provide all participants with user manual which is on how to use the system, and feedback form which will focus mainly on the design and usability of the system.
5. While participants are using the system, the team observes and records the result and behavior of the participant.
6. Gather the feedback form

6.1.4.4 CCD preparation site and dry run

Before performing the CCD session, the team will setup the system with sample data to the setting environment by connecting the server and client devices via Wireless connection.

In CCD session, the team will present and demonstrate all capabilities in the first iteration of the project and then give the user manual to a participant. Participants test the proposed system following the user manual.

After CCD session, the participants give feedback by filling in the feedback form.

6.1.4.5 Feedback form

Feedback form should include at least entities shown below

Table 24: CCD feedback form layout

Content	Description
No	Document number
Time	Document time
Capability	Which capability will be tested
Description	Detail and guideline for testing
Correctness	Correctness of module, for example, the functionality perform and response correctly based on input data in scale of 0-10
Performance	Performance of module in scale of 0-10
Satisfaction	Satisfaction of module in scale of 0-10
Overall	The overall result of testing in scale of 0-10
Comment	Suggestion or comment added
Signature	Signature of the participant
Date	Date

Note 0 for the lowest score and 10 for the highest score

6.1.4.6 CCD preparation risk management plans

Table 25: CCD preparation risk management plans

Risk	Risk description	Risk management plan
Schedule conflict	Some team members have many assignments due before CCD session, so team members may not have enough time to develop and test some capabilities lead to be not able to finish all capabilities in the first iteration	1. Prioritize the capabilities according to business value, detail of implementation, and risks. 2. Implement and test the higher prioritized requirements. If there is not enough time to finish all capabilities, the rest of capabilities will be implemented in next iteration.

Requirement mismatch	The team and client may misunderstand the concept of the requirements, so some user interfaces may not easy to use or some capabilities cannot perform properly	The team develops user interface mockup to show and get feedback from the client. Moreover, the team encourages the clients to experience some part of development before CCD session.

6.2 Iteration Assessment

This part of the document records the assessment of implemented capability at the end of each iteration. This version records the end of the first iteration and assessment activity at Core Capability Drive-through (CCD) which allows the client and end-user to have a hand-on experience on the proposed system.

6.2.1 Capabilities Implemented, Tested, and Results

Most of capabilities mentioned in the first iteration plan have been implemented and tested according to test cases shown in Acceptance Test Plan and Cases document; however, CR-4 and CR-10 have been pushed to the second iteration due to requirement changes and less support from the clients.

Table 26: Capabilities implemented, tested, and results in the first iteration

ID	Capability	Test Case	Test Results	If fail, why?
CR- 1	Online application for students subject registration	TC-01-01, TC-01-02, TC-01-03, TC-01-04, TC-02-01, TC-02-02	Failed	This requirement cannot pass some test cases because the authentication does not implement (second iteration) and client needs to change the layout and some functionalities in this requirement
CR-2	Locking certain classes.	TC-03-01, TC-03-02, TC-03-04, TC-03-05, TC-03-08, TC-03-09	Failed	The implementation of this requirements need to be redeveloped and retested due to the completeness and correctness of the functionality and response
CR-3	Student course requests are viewed/approved/rejected by counselor	TC-04-01, TC-04-02, TC-04-02a	Passed	

CR-7	Map teachers to course sections	TC-03-01, TC-03-03, TC-03-03a, TC-05-01	Failed	The functionality does not response correctly when the user would like to undo assign the teachers
CR-8	View alternate schedules	TC-08-01	Failed	The FET can provide alternated schedule, but the schedule generation need to be redeveloped and retested.
CR-11	Detailed Database	TC-06-01	Passed	
LOS-1	System usability	TC-07-01	Passed	
SR-1	Use given templates by LEMA Pilot School	N/A	Passed	
SR-2	Integration with family accountability system	TC-02-01, TC-02-02, TC-02-03, TC-03-01, TC-03-03, TC-03-03a, TC-03-04, TC-03-08, TC-04-01, TC-04-02, TC-04-02a, TC-04-03, TC-05-01	Failed	There are some information needed from team 4, such as student performance, so the team cannot meet this requirement.

Some of capabilities mentioned in the second iteration plan have been implemented and tested according to test cases shown in Acceptance Test Plan and Cases document; however, CR-5, 6, 9 and CR-12 have become evolutionary requirements because team 4 cannot provide relative features as an infrastructure. Moreover, SR-3 cannot finish due to dependencies from team 4. The team collaborates with team 4 to deliver the system.

Table 27: Capabilities implemented, tested, and results in the second iteration

ID	Capability	Test Case	Test Results	If fail, why?
CR- 4	Schedule generation	TC-02-04, TC-03-06, TC-03-07a, TC-03-07b, TC-05-01, TC-03-08a	Passed	
CR-10	LAUSD constraints	TC-03-10, TC-03-10a	Passed	
SR-3	Authentication to the system	TC-01-01, TC-01-02,	Failed	This requirement cannot pass because team 4

		TC-01-03, TC-01-04		cannot provide the support functionalities on time.
LOS-1	System usability	TC-07-01	Passed	

6.2.2 Core Capabilities Drive-Through Results

The core capabilities drive-through results are captured from both the client and developers during the CCD session. There are some capabilities that the client could not capture during the CCD session because of running out time during the CCD, so the team gathered all comments recorded from the team members in the CCD session and then combined with the client's comments as the CCD results.

Most of capabilities provided in the CCD are met the clients' expectation and business domain. Some capabilities, such as "Online application for students subject registration" are needed to update to reflect the clients' needs and be more intuitive when they are displayed in the website. Each page particularly for students should have instructions telling the user what to do. The detail feedback is shown in the table 27.

Table 28: **Core capabilities drive-through results**

Requirement	User experience	Result
CR- 1: Online application for students subject registration	Driven-through	**Positive feedbacks:** - The user interface is easy to use **Improvements needed:** - The students should be able to see the default grade level and advanced courses at the same page - The layout should be updated to reflect the domain business - The user interface should display the number of courses that students can take and course name with A – G requirement - The user interface should display default courses with selected check box. **Changes-to-be-considered:** - In the user interface, is should have a mechanism to detect conflicts and errors, such as conflict courses.
CR-2: locking certain classes.	Demonstrated	**Positive feedbacks:** - The functionality works properly **Improvements needed:** N/A

		Changes-to-be-considered: - Exporting the activities and locking certain classes should work correspondingly
CR-3: Student course requests are viewed/approved/rejected by counselor	Driven-through	**Positive feedbacks:** - The user interface is easy to navigate - The client was satisfied with the performance and correctness of this capability **Improvements needed:** - An approval status should be approved or remain unapproved **Changes-to-be-considered:** - The user interface should be similar to the student subject registration when the counselor would like to change the selected courses
CR-7: Map teachers to course sections (this feedback was captured by the team)	Driven-through	**Positive feedbacks:** - The user interface is intuitive and easy to use **Improvements needed:** N/A **Changes-to-be-considered:** N/A
CR-8: View alternate schedules (this feedback was captured by the team)	Shown	**Positive feedbacks:** N/A **Improvements needed:** - It should integrate with the schedule generation requirement **Changes-to-be-considered:** N/A
CR-11: Detailed Database (this feedback was captured by the team)	Driven-through	**Positive feedbacks:** - The sample data helps the clients understand the system better **Improvements needed:** - The database schema need to be modified by adding isDefaultCourse field in course_detail table and removing denied approval status in approval_status table **Changes-to-be-considered:** - Reflect modified database schema in related pages
LOS-1: System usability	Driven-through	**Positive feedbacks:** - The system is easy to navigate and use

		Improvements needed: - Each page should have instructions telling the user what to do. **Changes-to-be-considered:** - The users in the system have different background and skills. Student's user interface should be the easiest and simplest to use.
SR-1: Use given templates by LEMA Pilot School (this feedback was captured by the team)	Driven-through	**Positive feedbacks:** - The clients are familiar with the user interfaces since the website includes LEMA Pilot School templates **Improvements needed:** N/A **Changes-to-be-considered:** N/A
SR-2: Integration with family accountability system	Driven-through	**Positive feedbacks:** - The information from family accountability system is displayed correctly **Improvements needed:** - The student performance should be displayed in the student progress tracking properly **Changes-to-be-considered:** N/A

According to the concern log provided by Ms. Beth Kennedy, all capabilities in the second iteration are important. As a result, the priority of each capability is still the same and needed to be done.

The team realized that there was some misunderstanding in the domain knowledge and it might lead to cause the problems in the future, so the team will identify all possible risks from the CCD session and create mitigation plan.

Table 29: CCD risk management plans

Risk	Risk description	Risk management plan
User interface mismatch	The team may misunderstand the concept of the domain knowledge led to create the user interface which is not reflect clients' needs	The team develops user interface mockup to show and get feedback from the client.
Schedule shortfall	After finishing the CCD session, there are many changes required in the system and the project schedule is compressed, so it is possible that the team cannot	The team asked the clients to reprioritize capabilities left and focus on the higher priority capabilities.

	provide all the capabilities	
Low productivity	The team members have many assignments and get many changes from the CCD, so it is possible that the team productivity will be reduced.	The project manager plans ahead and gives buffer time to finish team member’s responsibilities

6.3 Adherence to Plan

Most of capabilities mentioned in the first iteration plan are on the schedule. However, some capabilities need to be redeveloped and retested to meet the project requirements because some of capabilities were required to be updated following the client’s feedback. Three capabilities in the first iteration were deferred to implement and test in the second iteration due to schedule conflicts and developmental dependency on team 4. The team put capabilities which were required redeveloping and retesting and capabilities which were deferred in the second iteration, so these capabilities will be developed and tested.

During developing the project in the first iteration, there were uncertainties occurred such as requirement changes and misunderstanding the concepts between the client and development team. To avoid mistakes, the team will do the prototypes helped all stakeholders understand and give feedback.

In second iteration, team spent time redeveloping and retesting some requirements of the first iteration, so it affected the plan and cause the team not to be able to follow the plan. Moreover, the team 4’s collaboration and commitment played an important role in delivering some requirements. CR-5, 6, 9 and CR-12 became the evolutionary requirements because team 4 could not provide support functionalities. Moreover, SR-3 also depended on team 4. The team tried to mitigate risks by having a backup plan to implement by the team and increasing communication between teams.