

Life Cycle Plan (LCP)

We Are Trojans (WAT) Network

Team01

Team members	Roles
Eirik Skogstad	Project Manager, Life Cycle Planner
Min Li	Feasibility Analyst, Operational Concept Engineer
Pittawat Pamornchaisirikij	NDI/NCS Acquirer & Evaluator, Tester
Punyawee Pakdiying	System Architect, Feasibility Analyst
Saloni Priya	Requirements Engineer, UML Modeler
Ameer Elkordy	IIV&V, Quality Focal Point
Suleyman Erten	Operational Concept Engineer, Requirements Engineer
Kamonphop Srisopha	Prototyper, UML Modeler

11/30/14

Version History

Date	Author	Version	Changes made	Rationale
09/28/14	ES	1.0	Created document from template, updated team roles and section 3.3 skills.	For VCP package submission.
10/11/14	ES	1.5	Updated sections 1-5.	For draft FCP package submission.
10/19/14	ES	2.0	Updated some sections according to feedback from FCR ARB: - Updated artifacts to be delivered before DCR ARB. - Updated team member roles during development phase.	For FCP package submission.
11/30/14	ES	2.5	Updated iteration plan, team member roles, new effort estimation.	For draft DCP submission.

Table of Contents

Life Cycle Plan (LCP)	i
Version History	ii
Table of Contents.....	iii
Table of Tables.....	v
Table of Figures	vi
1. Introduction	1
1.1 Purpose of the LCP	1
1.2 Status of the LCP	1
1.3 Assumptions.....	1
2. Milestones and Products	2
2.1 Overall Strategy	2
2.2 Project Deliverables	3
3. Responsibilities	7
3.1 Project-specific stakeholder's responsibilities	7
3.2 Responsibilities by Phase	7
3.3 Skills	9
4. Approach	12
4.1 Monitoring and Control	12
4.2 Methods, Tools and Facilities	12
5. Resources	14
6. Iteration Plan	20
6.1 Plan.....	20
6.1.1 Capabilities to be implemented	20
6.1.2 Capabilities to be tested.....	20
6.1.3 Capabilities not to be tested	21
6.1.4 CCD Preparation Plans.....	21
6.2 Iteration Assessment	21
6.2.1 Capabilities Implemented, Tested, and Results	21
6.2.2 Core Capabilities Drive-Through Results	22

6.3 Adherence to Plan	22
------------------------------	-----------

Table of Tables

<i>Table 1: Artifact deliverables in Exploration Phase</i>	<i>3</i>
<i>Table 2: Artifact deliverables in Valuation Phase.....</i>	<i>3</i>
<i>Table 3: Artifact deliverables in Foundations Phase</i>	<i>4</i>
<i>Table 4: Artifact deliverable in Development Phase</i>	<i>6</i>
<i>Table 5: Stakeholder's Responsibilities in each phase</i>	<i>7</i>
<i>Table 6: Modules</i>	<i>14</i>
<i>Table 7: COCOMOII Scale Driver.....</i>	<i>14</i>
<i>Table 8: COCOMOII Cost Drivers for WAT point system</i>	<i>15</i>
<i>Table 9: COCOMOII Cost Drivers for Forum Module</i>	<i>16</i>
<i>Table 10: COCOMOII Cost Drivers for Profile Module.....</i>	<i>16</i>
<i>Table 11: COCOMOII Cost Drivers for WAT Store Module</i>	<i>17</i>
<i>Table 12: COCOMOII Cost Drivers for Notification Module.....</i>	<i>18</i>
<i>Table 13: Construction iteration capabilities to be implemented.....</i>	<i>20</i>
<i>Table 14: Construction iteration capabilities to be tested.....</i>	<i>20</i>
<i>Table 15: Capabilities implemented, tested, and results</i>	<i>21</i>

Table of Figures

No table of figures entries found.

1. Introduction

1.1 Purpose of the LCP

The LCP serves as a basis for controlling and monitoring the project's progress. It identifies available people, skills and resources in order to make the best use of these in the project's life cycle. It also gives proof to key stakeholders that the major life cycle issues is known and thought through in advance.

1.2 Status of the LCP

The LCP is currently at the draft DCP version number 2.5. In this version, sections 1 to 5 have been updated and section 6.1 is completed. Effort estimates are updated and the iteration plan is completed.

1.3 Assumptions

- The duration of the project is 24 weeks, which are 12 weeks in fall 2014 and 12 weeks in spring 2015.
- All team members listed on the title page will work on the project during the fall semester and the spring semester.

2. Milestones and Products

2.1 Overall Strategy

The We Are Trojans (WAT) Network is following the architected agile process, as a big part of the project involves modification of existing software to create the online forum core capability.

Exploration phase

Duration: 9/10/14-9/29/14

Concept: Identify operational concept, system and software requirements and architecture, and life-cycle plan.

Deliverables: Client Interaction Report, Valuation Commitment Package

Milestone: Valuation Commitment Review

Strategy: One Incremental Commitment Cycle, Risk assessment analysis, Win-Win Negotiation Sessions.

Valuation phase

Duration: 9/30/14-10/15/14

Concept: Identify Objectives, Constraints and Priorities, Develop operation concept, Explore alternatives, Provide project feasibility evidence, Prototyping, Assess and plans to mitigate risks, Plan and manage project, Perform win-win negotiation, Define quality and configuration policy.

Deliverables: Draft Foundations Commitment Package

Milestone: Foundations Commitment Review

Strategy: One Incremental Commitment Cycle, Risk assessment analysis, Win-Win Negotiation Sessions, Planning Poker.

Foundations phase

Duration: 10/16/14-12/12/14

Concept: Assess Project Status, Plan and Manage Project, Manage Project Quality, Prototyping, Develop Software Architecture.

Deliverables: Foundations Commitment Package, Draft Development Commitment Package, Development Commitment Package.

Milestone: Development Commitment Review

Strategy: One Incremental Commitment Cycle, Risk assessment analysis.

Re-baselined Foundations phase

Duration: 1/12/15-2/13/15

Concept: Re-baseline Project Status, Prepare for Development Phase, Plan for Testing, Plan and Manage Project.

Deliverables: Re-baselined Development Commitment Package

Milestone: Re-baselined Development Commitment Review

Strategy: One Incremental Commitment Cycle, Risk assessment analysis.

Development phase – Construction**Duration:** 1/26/15-3/20/15**Concept:** Construction Iteration, build and test all modules.**Deliverables:** Operation Commitment Package**Milestone:** Operation Commitment Review**Strategy:** Two development cycles.

2.2 Project Deliverables

2.2.1 Exploration Phase

Table 1: Artifact deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	9/19/2014	.doc, .pdf	Soft copy
Program Model & Business Process Model	9/21/2014	.doc, .pdf	Soft copy
Valuation Commitment Package <ul style="list-style-type: none"> • Operational Concept Description (OCD) Early Section • Life Cycle Plan (LCP) Early Section • Feasibility Evidence Description (FED) Early Section 	9/29/2014	.doc, .pdf	Soft copy
Project Effort	Every Monday	Text	Bugzilla
Project Plan	Every other Wednesday	.mpp	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy

2.2.2 Valuation Phase

Table 2: Artifact deliverables in Valuation Phase

Artifact	Due date	Format	Medium
Prototype Presentation	10/3/2014	.pdf	Soft copy
Draft Foundations Commitment Package <ul style="list-style-type: none"> • Operational Concept Description (OCD) All 	10/13/14	.doc, .pdf	Soft copy

sections • Life Cycle Plan (LCP) Sections 1-5 • Feasibility Evidence Description (FED) Sections 1-5 • Prototype Description (PRO) All sections • System and Software Architecture Description (SSAD) Sections 1 & 2			
Project Effort	Every Monday	Text	Bugzilla
Project Plan	Every other Wednesday	.mpp	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy

2.2.3 Foundations Phase

Table 3: Artifact deliverables in Foundations Phase

Artifact	Due date	Format	Medium
Foundations Commitment Package • Operational Concept Description (OCD) All sections • Life Cycle Plan (LCP) Sections 1-5 • Feasibility Evidence Description (FED) Sections 1-5 • Prototype Description (PRO) All sections • System and Software Architecture Description (SSAD) Sections 1 & 2	10/20/14	.doc, .pdf	Soft copy
Client Feedback Form			
Draft Development Commitment Package	12/01/14		
Development Commitment Package	12/08/14		
Project Effort	Every Monday	Text	Bugzilla

Project Plan	Every other Wednesday	.mpp	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy

2.2.4 Rebaselined Foundations Phase

Table 4: Artifact deliverable in Rebaselined Foundations Phase

Artifact	Due date	Format	Medium
Rebaselined Development Commitment Package	9/02/15	.doc, .pdf	Soft copy
Project Effort	Every Monday	Text	Bugzilla
Project Plan	Every other Wednesday	.mpp	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities

3.2 Responsibilities by Phase

Table 5: Stakeholder's Responsibilities in each phase

Team Member / Role	Primary / Secondary Responsibility				
	Exploration	Valuation	Foundations	Development-Construction Iteration	Development - Transition Iteration
Eirik Skogstad Project Manager, Life-cycle planner	Primary Responsibility Create and follow up action items. Record project progress. Secondary Responsibility Detail Project Plan. Identify responsibilities and skills.	Primary Responsibility Create and follow up action items. Record project progress. Secondary Responsibility Identify milestones and products. Estimate project effort and schedule.	Primary Responsibility Create and follow up action items. Record project progress. Secondary Responsibility Provide Process Feasibility Evidence. Develop Iteration Plan.	Primary Responsibility Create and follow up action items. Record project progress. Secondary Responsibility Assess development iteration. Develop Transition Plan.	Primary Responsibility Create and follow up action items. Record project progress. Secondary Responsibility Assess development iteration. Develop Support Plan.
Min Li Feasibility Analyst, Operational Concept Engineer	Primary Responsibility Gather risks. Assess and plan to mitigate risks. Secondary Responsibility Responsibility 3 Responsibility 4	Primary Responsibility Provide evidence of feasibility of architecture. Analyze Business Case.	Primary Responsibility Assess Feasibility Evidence. Provide conclusion and recommendation	Primary Responsibility Build software Secondary Responsibility Testing	Primary Responsibility Build software Secondary Responsibility Testing
Punyawee Pakdiying System Architect, Feasibility Analyst	Primary Responsibility Gather risks. Assess and plan to mitigate risks. Secondary Responsibility	Primary Responsibility Define high-level architecture. Document architecture feasibility.	Primary Responsibility Define platform-dependent architecture. Document architecture feasibility.	Primary Responsibility Assess system architecture. Build software. Secondary Responsibility	Primary Responsibility Assess system architecture. Build software. Secondary Responsibility

	Explore NDI alternatives.	Secondary Responsibility Explore NDI alternatives.	Secondary Responsibility Explore NDI alternatives.	Identify risks and feasibility coming up during development period	Identify risks and feasibility coming up for transition
Pittawat Pamornchaisirikij NDI Acquirer & Evaluator, Tester	Primary Responsibility Gather risks. Assess and plan to mitigate risks. Secondary Responsibility Explore NDI alternatives.	Primary Responsibility Explore NDI alternatives. Assess and evaluate NDI components. Secondary Responsibility Explore NDI alternatives.	Primary Responsibility Acquire NDI components. Assess and evaluate NDI components. Secondary Responsibility Explore NDI alternatives.	Primary Responsibility Develop testing strategies. Test Software. Secondary Responsibility analyze NDI assessment result, provide level of service feasibility	Primary Responsibility Develop testing strategies. Test Software. Secondary Responsibility analyze NDI assessment result, provide level of service feasibility
Saloni Priya Requirements Engineer, UML Modeler	Primary Responsibility Gather win-conditions from stakeholders. Capture win-conditions in win-win session. Secondary Responsibility Perform feasibility evidence for the requirements	Primary Responsibility Document most significant requirements. Prioritize the requirements. Secondary Responsibility Develop the respective UML diagrams	Primary Responsibility Document detailed requirements. Secondary Responsibility Familiarize with the NDI/NCS and COTS	Primary Responsibility Update requirements. Build software. Secondary Responsibility Check whether requirements or being developed as per the negotiations	Primary Responsibility Build software. Secondary Responsibility Perform Feasibility evidence
Kamonphop Srisopha Prototyper, UML Modeler	Primary Responsibility Assess Prototype and Components, Develop Prototype Secondary Responsibility Model a UML component of the system, Analyze the proposed system	Primary Responsibility Develop prototype. Get prototype feedback. Secondary Responsibility Model a UML component of the system, Analyze the proposed system, modified existing UML	Primary Responsibility Tailor components. Secondary Responsibility Model a UML component of the system, Analyze the proposed system, modified existing UML	Primary Responsibility Build software. Secondary Responsibility Model a UML component of the system, Analyze the proposed system, modified existing UML	Primary Responsibility Build software. Secondary Responsibility Model a UML component of the system, Analyze the proposed system, modified existing UML
Suleyman Erten Operational Concept Engineer,	Primary Responsibility Identify shared vision.	Primary Responsibility Develop new operational concept.	Primary Responsibility Assess operational concept.	Primary Responsibility Identify organizational and operational	Primary Responsibility Verified whether the developed project is

Requirements Engineer	Secondary Responsibility Negotiate win-win conditions with the client	Secondary Responsibility Fill in Bugzilla tickets associated to OCD	Secondary Responsibility Perform Feasibility Analysis	transformation. Build software. Secondary Responsibility Analyze the possible changing in requirements	satisfied operational concepts. Build software. Secondary Responsibility Analyze the possible changing in requirements
Ameer Elkordy IIV&V, Quality Focal Point	Primary Responsibility Verify and validate work products. Secondary Responsibility	Primary Responsibility Verify and validate work products. Construct traceability matrix. Secondary Responsibility Nag team on using Bugzilla more consistently.	Primary Responsibility Verify and validate work products. Assess quality management strategy. Identify configuration management strategy. Secondary Responsibility Nag team on using Bugzilla more consistently.	Primary Responsibility Verify and validate work products. Secondary Responsibility Nag team on using Bugzilla more consistently. Test software bugs.	Primary Responsibility Verify and validate work products. Secondary Responsibility
Matthew Wong Client		Review prototype and give feedback. Clarify doubts.	Review prototype and give feedback. Clarify doubts.		
Linda Suen Client		Review prototype and give feedback. Clarify doubts.	Review prototype and give feedback. Clarify doubts.		

3.3 Skills

Team members	Role	Skills
Eirik Skogstad	Project Manager, Life Cycle Planner	Current skills: Software Project Management, C++, Java, HTML, PHP, MySQL, JavaScript, XML, Winbook

		Required skills: MS Project, Life-Cycle tools, Git
Min Li	Feasibility Analyst, Operational Concept Engineer	Current skills: HTML, CSS, PHP, JavaScript, JSP, JQuery, Strut 2.0, Spring 2.0, Ibatis, Java, C++, API application, JSON, XML Required skills: Analytical skills, web development, UML modeling
Pittawat Pamornchaisirikij	NDI/NCS Acquirer & Evaluator, Tester	Current Skills: C, C++, HTML, PHP, JavaScript, CSS, socket programming, device driver programming in Linux, Joomla CMS, Drupal CMS, Thai language, English language Required Skills: Experience in web development domain
Punyawee Pakdiying	System Architect, Feasibility Analyst	Current Skills: Linux/Unix Command line (beginner level), Java, PHP, MySQL, SQL, HTML, JavaScript, CSS, Joomla, Drupal Required Skills: High level Linux/Unix Command line, DBMS, Middleware, Analytical skills, and NDI/NCS used in the project
Saloni Priya	Requirements Engineer, UML Modeler	Current Skills: General purpose modelling language, ArgoUML, Creately for UML, MySQL, PHP, CSS, HTML, rustic JavaScript, C# Required Skills: Winbook, Visual Paradigm,

		Communication skills for effective negotiation, Knowledge Contributor, Analytical skills, NDI/NCS/COTS experience and knowledge about web server hosting.
Ameer Elkordy	IIV&V, Quality Focal Point	<p>Current skills: Database, Analytics, General Scripting</p> <p>Rusted skills: HTML, CSS, PHP, Java</p> <p>Required skills: Organization, Tracking, Winbook, Loud Voice, and Process & Deliverable Knowledge</p>
Suleyman Erten	Operational Concept Engineer, Requirements Engineer	<p>Current Skills: HTML, JavaScript, Java</p> <p>Required Skills: Goal Decomposition, Use Case Scenarios, UML Modelling, Winbook, Knowledge Contributor</p>
Kamonphop Srisopha	Prototyper, UML Modeler	<p>Current Skills: HTML, Java, PHP, MySQL, Rust, C++</p> <p>Required Skills: Winbook, Advanced UNIX/ Linux Command line, R (statistical programming), NDI/ NCS/COTS integration knowledge</p>

4. Approach

4.1 Monitoring and Control

The project is monitored with a bi-weekly progress report and project plan. The progress report includes a progress description with total time spent on the project. It also includes the top project risks, number of SLOC, COTS software and defects/concerns.

4.1.1 Closed Loop Feedback Control

Draft documents/sketches/diagrams are shared on Google Drive where team members can quickly review and update them. Feedback for a specific artifact is given to the artifact leader either person-to-person or in team meetings.

4.1.2 Reviews

- Team meetings; we have a group meeting every day after CS577a class to debrief and review what we have accomplished, what we should do and what are some difficulties we are facing.
- Win-win negotiation; the negotiation helps us and the client to be on the same page and share the same understanding.
- IIV&V evaluations; Our Den team members make sure that the document looks good and check if time is logged on Bugzilla and provide feedback about the bug/artifact on the ticket for the owner to correct.
- TA feedback; the comments from all TAs are also useful.

4.2 Methods, Tools and Facilities

Tools	Usage	Provider
Github	Provide a version control system to store our team website files.	Github
Google drive	Contain all documents created by our team so that we can collaborate	Google
Visual paradigm	Provide a tool to create various UML diagrams used in the project	Visual paradigm
Balsamiq	The tool was used to develop the user interface prototype	Balsamiq
Microsoft office	The office was used to create docs, presentation and XML files.	Microsoft
Facebook (Group/Chat)	Provide a group feature to collaborate and make an appointment among team members and client	Facebook
Join.me	The tool was used to share screen with DEN student	<u>LogMeIn, Inc.</u>
Skype	Video and/or Voice Chat	Microsoft

Bootstrap	to decorate our team website	Open source
Bugzilla	Provide a tool to create tasks, track task's statuses, and identify effort used in each task	USC
Winbook	Provide a tool to facilitate the team to identify the project's goals, win conditions of all stakeholders and prioritize them	USC

5. Resources

Estimated CSCI577a Effort: 8 team members working 10 hrs./week for 12 weeks

Estimated CSCI577b Effort: 8 team members working 10 hrs./week for 12 weeks

Estimated total effort: 10 hours per week * 6 months = 60 PM

Programming languages: PHP, HTML/CSS, SQL, JavaScript

Budget: assume that this project has no budget for development efforts. There might be a cost for hosting a server and acquire a domain name.

Table 6: Modules

No	Module	Description	SLOC (estimation)	REVL (estimation)
1	Point System	The point system that will provide users incentives to participate in the We Are Trojans network	1,000	15%
2	Forum Module	This module provides the students a platform to interact with each other	3,000	7%
3	Profile Module	The system that allow users to add and modify their information	800	10%
4	Store Module	The store used to redeem WAT points. First iteration will be limited to gift cards.	500	5%
5	Notification Module	The system that notify users with changes, such as liking, disliking, and a special campaign	800	5%

Table 7: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	High	Many examples exist with features similar to this project (i.e. Stackoverflow.com, Facebook)
FLEX	High	Project has a very high flexible set of requirements but the limited schedule pull down FLEX from very high to high.
RESL	Nominal	The project has thoroughness with architectural definition, but the system has a high tendency of risks.

TEAM	High	Client's limited schedule availability could pose complexities
PMAT	Nominal	Relatively new team with limited experience working together under stressful deadlines

Table 8: COCOMOII Cost Drivers for WAT point system

Cost Driver	Value	Rationale
RELY	Nominal	All other modules rely on this module. However, it is not risk to human life.
DATA	Low	We do not have much test data.
CPLX	Nominal	The analysis relating to the "WAT" Point system requires some numerical analysis that is prone to precision problems needed to be handled.
RUSE	Nominal	This WAT point module is developed specifically for this project. So it is only reusable across project (nominal)
DOCU	Nominal	The documentation was developed for each phase of development.
TIME	Nominal	The system is expected to use less than 50% of the available execution time.
STOR	Nominal	It will take up at most 20% use of available storage.
PVOL	Nominal	In this project, we do not update the change frequently.
ACAP	High	The team cohesively works on gathering requirements and high level design of the system.
PCAP	High	Most of our team members are capable. We have mix skills from 90th percentile to 55 percentile so we are going to estimate in between = 75 percentile.
PCON	Very High	There is no personnel turnover.
APEX	Nominal	The team is not fully familiar with the COTS/NDI that will be used in developing the system.
PLEX	Nominal	In the team, there are mixes of people familiar and unfamiliar with the platforms
LTEX	High	Every member of the team has used the language that will be used in development of this system.
TOOL	Nominal	Basic life-cycle tools, moderately integrated.
SITE	High	The clients and the DEN students are not present at the same site, but connection is achieved using Skype.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 9: COCOMOII Cost Drivers for Forum Module

Cost Driver	Value	Rationale
RELY	Nominal	This module is crucial for the system, but a failure would only result in moderate losses.
DATA	Nominal	In forum functionality, we may have some amount of data like images and texts for testing.
CPLX	Nominal	Development of this module will mostly involve modification of an existing COTS forum.
RUSE	Low	Forum will be tailored for this specific project, no reuse.
DOCU	Nominal	The documentation was developed for each phase of development.
TIME	Nominal	The system is expected to use less than 50% of the available execution time.
STOR	Very High	It will take up at < 80% use of available storage
PVOL	Nominal	In this project, we do not update the change frequently.
ACAP	High	The team cohesively works on gathering requirements and high level design of the system.
PCAP	High	Most of our team members are capable. We have mix skills from 90th percentile to 55 percentile so we are going to estimate in between = 75 percentile.
PCON	Very High	There is no personnel turnover.
APEX	Nominal	The team is not fully familiar with the COTS/NDI that will be used in developing the system.
PLEX	Nominal	In the team, there are mixes of people familiar and unfamiliar with the platforms
LTEX	High	Every member of the team has used the language that will be used in development of this system.
TOOL	Nominal	Basic life-cycle tools, moderately integrated.
SITE	High	The clients and the DEN students are not present at the same site, but connection is achieved using Skype.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 10: COCOMOII Cost Drivers for Profile Module

Cost Driver	Value	Rationale
RELY	Nominal	moderate, easily recoverable losses
DATA	Low	The amount of test data will be relatively low compared to SLOC.

CPLX	Low	Straightforward nesting of structured. Simple GUI builder, straightforward database structure.
RUSE	Nominal	Reusable across project.
DOCU	Nominal	The documentation was developed for each phase of development.
TIME	Nominal	The system is expected to use less than 50% of the available execution time.
STOR	Nominal	It will take up at most 20% use of available storage
PVOL	Nominal	In this project, we do not update the change frequently.
ACAP	High	The team cohesively works on gathering requirements and high level design of the system.
PCAP	High	Most of our team members are capable. We have mix skills from 90th percentile to 55 percentile so we are going to estimate in between = 75 percentile.
PCON	Very High	There is no personnel turnover.
APEX	Nominal	The team is not fully familiar with the COTS/NDI that will be used in developing the system.
PLEX	Nominal	In the team, there are mixes of people familiar and unfamiliar with the platforms
LTEX	High	Every member of the team has used the language that will be used in development of this system.
TOOL	Nominal	Basic life-cycle tools, moderately integrated.
SITE	High	The clients and the DEN students are not present at the same site, but connection is achieved using Skype.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 11: COCOMOII Cost Drivers for WAT Store Module

Cost Driver	Value	Rationale
RELY	Low	It does not affect the system too much
DATA	Low	We do not have much test data.
CPLX	Low	Straightforward nesting of structured. Simple GUI builder, straightforward database structure.
RUSE	Nominal	Reusable across project.
DOCU	Nominal	The documentation was developed for each phase of development.
TIME	Nominal	The system is expected to use less than 50% of the available execution time.
STOR	Nominal	It will take up at most 40% use of available storage

PVOL	Nominal	In this project, we do not update the change frequently.
ACAP	High	The team cohesively works on gathering requirements and high level design of the system.
PCAP	High	Most of our team members are capable. We have mix skills from 90th percentile to 55 percentile so we are going to estimate in between = 75 percentile.
PCON	Very High	There is no personnel turnover.
APEX	Nominal	The team is not fully familiar with the COTS/NDI that will be used in developing the system.
PLEX	Nominal	In the team, there are mixes of people familiar and unfamiliar with the platforms
LTEX	High	Every member of the team has used the language that will be used in development of this system.
TOOL	Nominal	Basic life-cycle tools, moderately integrated.
SITE	High	The clients and the DEN students are not present at the same site, but connection is achieved using Skype.
SCED	Nominal	The schedule is fixed for 12 weeks.

Table 12: COCOMOII Cost Drivers for Notification Module

Cost Driver	Value	Rationale
RELY	Low	No other module relies on this. Failure will only cause slight inconvenience.
DATA	Low	We do not have much test data.
CPLX	Nominal	Mostly simple nesting. Some intermodule control.
RUSE	Nominal	Reuse across project.
DOCU	Nominal	The documentation was developed for each phase of development.
TIME	Nominal	The system is expected to use less than 50% of the available execution time.
STOR	Nominal	It will take up at most 20% use of available storage
PVOL	Nominal	In this project, we do not update the change frequently.
ACAP	High	The team cohesively works on gathering requirements and high level design of the system.
PCAP	High	Most of our team members are capable. We have mix skills from 90th percentile to 55 percentile so we are going to estimate in between = 75 percentile.
PCON	Very High	There is no personnel turnover.

APEX	Nominal	The team is not fully familiar with the COTS/NDI that will be used in developing the system.
PLEX	Nominal	In the team, there are mixes of people familiar and unfamiliar with the platforms
LTEX	High	Every member of the team has used the language that will be used in development of this system.
TOOL	Nominal	Basic life-cycle tools, moderately integrated.
SITE	High	The clients and the DEN students are not present at the same site, but connection is achieved using Skype.
SCED	Nominal	The schedule is fixed for 12 weeks.

System's Overview COPSEMO

Scale Factors 15.62 Schedule (SCED)

Subcomponent Subcomponent(s) More Actions ...

X	Name	Size	Labor Rate (\$/Month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	Risk
<input type="checkbox"/>	Forum Module	3,210	0.0	0.57	Non-s...	10.69	6.09	52...	0.00	0.00	0.8	0.0
<input type="checkbox"/>	Point System	1,150	0.0	0.46	Non-s...	3.83	1.77	65...	0.00	0.00	0.2	0.0
<input type="checkbox"/>	Notificatio...	840	0.0	0.42	Non-s...	2.80	1.19	70...	0.00	0.00	0.2	0.0
<input type="checkbox"/>	Profile Module	880	0.0	0.40	Non-s...	2.93	1.18	74...	0.00	0.00	0.2	0.0
<input type="checkbox"/>	Store Module	525	0.0	0.37	Non-s...	1.75	0.65	81...	0.00	0.00	0.1	0.0

Estimation

Total Lines Of Code: 6,605 Hours/PM: 152.0

Estimated	Effort	Schedule	PROD	COST	INST	Staff	Risk
Optimistic	8.70	7.20	759.29	0.00	0.00	1.2	
Most Likely	10.87	7.71	607.43	0.00	0.00	1.4	0.0
Pessimistic	13.59	8.27	485.94	0.00	0.00	1.6	

Total number of week = 24 weeks (2 semesters).

One person can work 10 hours per week.

According to COINCOMO II;

The COCOMO II estimation effort calculated from the 5 modules gives an effort of **11.07 PM**

$13.59 \text{ PM} \times 152 \text{ hrs/PM} = 2065.68 \text{ hr to do the work}$

$10 \text{ hrs/week/person} \times 24 \text{ weeks} = 240 \text{ hr/person}$

$2065.68 \text{ hrs} / 240 \text{ hrs / person} = 8.6 \text{ person}$

We have 8 team members, so according to the pessimistic estimate we will be able to finish the project within the time constraint if we work a little extra.

6. Iteration Plan

6.1 Plan

<< Provide a high-level overview of the content of the given iteration. Indicate which Life cycle milestones will be addressed. >>

6.1.1 Capabilities to be implemented

<< For the milestone identified above, identify the capabilities that will be implemented in the upcoming iteration. Identify the features, requirements or use-cases that are being developed (implemented, tested, etc.) for this iteration. Each component should be accounted for in at least one iteration. All requirements should be implemented and tested (or re-negotiated) by the completion of all the iterations. Be mindful of implementation dependencies. Document complex dependencies and communicate them to the appropriate development staff. >>

Table 13: Construction iteration capabilities to be implemented

ID	Capability	Description	Priority	Iteration
1	OC-1	Q&A Forum	High	1
2	OC-2	WAT Point System	High	1
3	OC-3	Notification System	High	1
4	OC-4	Profile	Medium	1
5	OC-5	Leaderboard	Medium	2
6	OC-6	Redemption	Medium	2
7	OC-7	Event System	Low	2

6.1.2 Capabilities to be tested

<< For the milestone identified above, identify the capabilities that will be tested in the upcoming iteration.

Identify the software features and combinations of software features to be tested this iteration. This may also include non-functional requirements or extra-functional requirements, such as performance, portability, and so forth.

Additionally you may need to test every requirement listed in the WinWin Agreements DC package, non-requirement component features such as COTS capabilities and quality, API functionality, etc. >>

Table 14: Construction iteration capabilities to be tested

ID	Capability	Description	Priority	Iteration
1	OC-1	Q&A Forum	High	1

2	OC-2	WAT Point System	High	1
3	OC-3	Notification System	High	1
4	OC-4	Profile	Medium	1
5	OC-5	Leaderboard	Medium	2
6	OC-6	Redemption	Medium	2
7	OC-7	Event System	Low	2

6.1.3 Capabilities not to be tested

<< Identify notable features, and significant combinations of features, which will not be tested this iteration and why (e.g. a given feature uses a feature which will be implemented in following iteration). >>

6.1.4 CCD Preparation Plans

<< Identify the clients and other users who will be involved in the Core Capability Drive-through, the usage scenarios that it will support, and the specific CCD preparation plans and milestones. These may include

- user context-setting
- site preparation dry runs,
- feedback forms, and
- CCD risk management plans. >>

6.2 Iteration Assessment

6.2.1 Capabilities Implemented, Tested, and Results

<< Describes, in brief, the capabilities that were implemented and the test results. The capabilities implemented and tested do not necessarily need to match the ones listed in section 6.1 because some capabilities may have been pushed to the next iteration. >>

Table 15: Capabilities implemented, tested, and results

ID	Capability	Test Case	Test Results	If fail, why?
< ID >	< Capability >	< TC-XX >	Pass/Fail	< comments >
...				

6.2.2 Core Capabilities Drive-Through Results

<< Briefly summarize the feedback you received from your client(s). You need to be specific enough to cover the critical capabilities or scenarios that were discussed, demoed, or shown. Your descriptions MUST, but not limited to, cover the following areas:

- Positive feedbacks
- Improvements needed/suggested
- Changes to-be considered (Reprioritized capabilities, requirements, GUI, etc.)
- Risks (New risks introduced, risks mitigated, etc.)

Note: Make sure to be specific to the capabilities shown/demonstrated/driven-through. Simply stating that the clients liked the capabilities is not sufficient. >>

6.3 Adherence to Plan

<< Describe how well the iteration ran according to plan. Was it on budget and on time? Is there any uncertainty in the Software Development Status? Provide some insight to avoid mistakes for future iterations. >>