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

9/28/14

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.

Table of Contents

Table of Tables

Table of Figures

No table of figures entries found.

1. Introduction

1.1 Purpose of the LCP

<< Discuss the purpose of the LCP>>

1.2 Status of the LCP

<< Discuss the status of the LCP especially key differences from previous version, for example "The status of the LCP is currently at the Operation Commitment Package version number 10.0. This is the version that will be delivered to the client. The major changes from Rebaselined Foundations phase are:

- Two team members, Ritesh Kothari and Jerome Wan, did not continue in Development Phase.
- One core capability, Report and Certificate Generation, is deferred." >>

1.3 Assumptions

<< List all possible assumptions for the project life cycle, such as schedule, personnel resources, standard, guidelines, and etc. For example:

- The duration of the project is 24 weeks, which are 12 weeks in Fall 2006 and 12 weeks in Spring 2007. >>

2. Milestones and Products

2.1 Overall Strategy

<< Identify your overall strategy. Identify the ICSM process you are following and your rationale; Architected Agile or NDI-Intensive or Net-Centric Services. Identify the life cycle phases and its dates, deliverables, milestone and strategy of each phase. For example:

"The Volunteer Tracking System is following Architected Agile process because there is no Non-Development Item or Web service that would fit to most of the core capabilities.

"Exploration phase
   Duration: 08/24/09- 9/21/09
   Concept: They identify project operational concept, system and software requirement, system and software architecture, and life-cycle plan. These phases prioritize the capabilities, conduct investment and feasibility analysis, and implement the software prototype.
   Deliverables: Valuation Commitment Package
   Milestone: Valuation Commitment Review
   Strategy: One Incremental Commitment Cycle"

Note: More information about ICSM process can be found in ICSM EPG> Delivery Process. Schedule of the class and its milestone can be found in the first lecture of the class.
>>

2.2 Project Deliverables

<< Identify project deliverables in each phase and its due date, format, and medium>>

2.2.1   Exploration Phase

<< The following is an example of deliverables in Exploration phase.>>

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	9/17/2006	.doc, .pdf	Soft copy
Valuation Commitment Package • Operational Concept Description (OCD) Early Section • Life Cycle Plan (LCP) Early	09/18/2006	.doc, .pdf	Soft copy

Artifact	Due date	Format	Medium
Section • Feasibility Evidence Description (FED) Early Section			
Evaluation of Valuation Commitment Package	09/27/2006	.xls	Soft copy
Project Effort	Every Monday	Text	ER system
Project Plan	Every Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy

Table 2: Artifact deliverable in Exploration Phase

Artifact	Due date	Format	Medium
<artifact name>	<due data>	<format type: .doc, .pdf>	<Medium type: hard copy, soft copy>
…	…	…	…

2.2.2   Valuation Phase

Table 3: Artifact deliverable in Valuation Phase

Artifact	Due date	Format	Medium
<artifact name>	<due date>	<format type: .doc, .pdf>	<Medium type: hard copy, soft copy>
…	…	…	…

2.2.3   Foundations Phase

Table 4: Artifact deliverable in Foundations Phase

Artifact	Due date	Format	Medium
<artifact name>	<due date>	<format type: .doc, .pdf>	<Medium type: hard copy, soft copy>
…	…	…	…

2.2.4   Development Phase

Table 5: Artifact deliverable in Development Phase

Artifact	Due date	Format	Medium
<artifact name>	<due date>	<format type: .doc, .pdf>	<Medium type: hard copy, soft copy>
…	…	…	…

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities

<< Other than typical stakeholders of CSCI577ab, identified in **ICSM EPG> Task: Identify Responsibilities and Skills,** which are client, user, maintainer, developer and IIV&V, do you have any project-specific stakeholder? If yes, please identify the role and his/her responsibilities.>>

3.2 Responsibilities by Phase

<< Identify responsibilities of each team member including client, user, and maintainer in each phase. Please note that a document name such as OCD, WinWin Agreements or Prototype is not a responsibility. Examples of responsibilities are identify project risk, develop prototype, acquire NDI, and etc.

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

Table 6: Stakeholder's Responsibilities in each phase

Team Member / Role	Primary / Secondary Responsibility				
	Exploration	**Valuation**	**Foundations**	**Development-** Construction Iteration	**Development-** Transition Iteration
Name: Role	**Primary Responsibility** Responsibility 1 Responsibility 2 **Secondary Responsibility** Responsibility 3 Responsibility 4	**Primary Responsibility** Responsibility 1 Responsibility 2 **Secondary Responsibility** Responsibility 3 Responsibility 4	**Primary Responsibility** Responsibility 1 Responsibility 2 **Secondary Responsibility** Responsibility 3 Responsibility 4	**Primary Responsibility** Responsibility 1 Responsibility 2 **Secondary Responsibility** Responsibility 3 Responsibility 4	**Primary Responsibility** Responsibility 1 Responsibility 2 **Secondary Responsibility** Responsibility 3 Responsibility 4
Name: Role	**Primary Responsibility** Responsibility 1 Responsibility 2 **Secondary Responsibility** Responsibility 3 Responsibility 4				

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	**Current skills:** Database, Analytics, General Scripting **Rusted skills:** HTML, CSS, PHP, Java **Required skills:** Organization, Tracking, Winbook, Loud Voice, and Process & Deliverable Knowledge
Suleyman Erten	Operational Concept Engineer, Requirements Engineer	**Current Skills:** HTML, JavaScript, Java **Required Skills:** Goal Decomposition, Use Case Scenarios, UML Modelling, Winbook, Knowledge Contributor
Kamonphop Srisopha	Prototyper, UML Modeler	**Current Skills:** HTML, Java, PHP, MySQL, Rust, C++ **Required Skills:** Winbook, Advanced UNIX/ Linux Command line, R (statistical programming), NDI/ NCS/COTS integration knowledge

4. Approach

4.1 Monitoring and Control

<< Identify the approach you are using in monitoring and controlling your project. Examples are Progress Report, Project plan, and etc. >>

4.1.1 Closed Loop Feedback Control

<< Explain how your team gets and provides feedback internally within the team. >>

4.1.2 Reviews

<<Describe various kinds of review that your team is using to control your project. >>

4.2 Methods, Tools and Facilities

<< Describe methods, tools, facilities and their usage and provider that you used in your project>>

Tools	Usage	Provider
Red Ridge 3.0	Provides examples for user interface and system functionality, is helpful in the development of prototype	CSC
PEAR	Creates a framework and distribution system for reusable PHP components	Open source
<Tool>	<Usage>	<Tool Provider>

5. **Resources**

<< For Architected Agile, use COTIPMO for Architected Agile project for your calculation, for NDI/NCS project, use COTIPMO for NDI/NCS project for your calculation

Identify the following information in order to estimate the software cost:
- Estimated CSCI577a Effort : X team members at X hrs/week for 12 weeks
- Estimated CSCI577b Effort : X team members at X hrs/week for 12 weeks
- Total estimated effort
- Budget information
- Project duration
- Component modules in your development project.
- Programming language used

For Architected Agile, the example of how to rate scale factors and cost drivers can be found at **ICSM EPG> Task: Estimate Project Effort and Schedule using COCOMO II**

For the most common mistakes in cost estimation for Architected Agile, please go to **ICSM EPG> Concept: Common Mistakes in COCOMOII Calculation**

You should provide rationale for every cost driver and scale factor of each module.

Note: Refer to Barry W. Boehm, et al, Software Cost Estimation With COCOMO II, Prentice all PTR, New Jersey, 2000 on how to estimate software cost . >>

Table 7: COCOMOII Scale Driver

Scale Driver	Value	Rationale
<Driver name>	<value>	<comments>
…	…	…

Table 8: COCOMOII Cost Driver

Cost Driver	Value	Rationale
<Driver name>	<value>	<comments>
…		…

<< Provide screenshot of your COCOMO II analysis result and interpret what does that mean to your project. Please note the number of COCOMOII Cost Driver tables (Refer to Table 8) is equal to the number of software modules in your project. More information can be found at ICSM EPG> Task: Estimate Project Effort and Schedule using COTIPMO for Architected Agile Project. >>

<< For NDI/NCS project, COTIPMO for NDI/NCS project is based on COCOMO II Application Point, so please use Table 9 - 12 instead of using Table 7 and 8. More information can be found at ICSM EPG> Task: Estimate Project Effort and Schedule using COTIPMO for for NDI/NCS Project. >>

Table 9: Application Count: Screens

Screen	Number of views	Number of source of data tables	Complexity level	Rationale
< Screen name >	<value>	<value>	<value>	<comments>
…				

Table 10: Application Count: Reports

Report	Number of sections	Number of source of data tables	Complexity level	Rationale
< Report name >	<value>	<value>	<value>	<comments>
…				

Table 11: Application Count: 3GL components

Component	Rationale
< Component name >	<comments>
…	

Table 12: Application Point Parameters

Parameter	Value	Rationale
Developer's Experience and Capability	<value>	<comments>
ICASE Maturity and Capability	<value>	<comments>

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
< ID >	< Capability >	< comments >	<value>	<value>

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
< ID >	< Capability >	< comments >	<value>	<value>

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. >>