Life Cycle Plan (LCP)

SnapValet

TEAM - 03

Name	Role
Brian Vanover	Project Manager, Developer
Brian Bousman	Operational Concept Engineer, Developer
Ditong Ding	System Architect, Developer.
Molly Karcher	IIV & V, Quality Focal Point, Developer.
Patrick Horng	Feasibility Analyst, Developer
Ridhima Manjrekar	Requirements Engineer, Life Cycle Planner, Developer.

04/19/2015

Version History

Date	Author	Version	Changes made	Rationale
09/28/14	Saikarthik	1.0	Updated section 3.3 – Skills of the development team.	To initiate the planning of the project's life cycle by identifying existing and required skills of each member of the development team. This document constitutes an important part of the Valuation Commitment Package.
11/12/14	Saikarthik	2.0	Updated section 3.1,3.2 Updated section 1,2,4 & 5	To document overall strategy, project deliverables, stakeholder responsibilities and approaches for monitoring project progress for the exploration, valuation and foundation phase. To document the estimate of the effort, time and cost of the project calculated using COINCOMO. This document is an initial draft for the Foundations Commitment Package.
11/14/14	Saikarthik	2.1	Updated section 3.2 Updated section 5	To document change in responsibilities in section 3.2, as discussed in the Architecture Review Board. To document updates in the cost drivers and website module in the COINCOMO estimate. This document is part of the Foundations Commitment Package.
12/01/14	Saikarthik	3.0	Updated section 6.1 Updated section 2.1, 2.2 Update section 3.2	To document the overall plan for the iterations in the development phase. This include the capabilities to be implemented, tested and preparation plans for the CCD. To document the overall strategy, responsibilities for the development phase, project deliverables for the rebaselined foundation review.

Date	Author	Version	Changes made	Rationale
02/11/15	Ridhima Manjrekar	4.0	Updated section 2.1 Updated section 2.2 Updated section 3.2 Updated section 3.3 Updated section 4.1 Updated section 4.2	Added the new team members their details and skills. Updated the project plan and tools that are being used. This includes the updated capabilities to be implemented, tested and preparation plans for the CCD.
04/06/15	Ridhima Manjrekar	5.0	Updated section 2.2	Added preparation plans for Transition Readiness Review
04/19/15	Ridhima Manjrekar	5.1	Updates section 2.2	Added preparation plans for the As Built Package and the Project showcase.

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the LCP

The life cycle plan documents the plans of the development team in each phase and iteration throughout the project's life cycle. At each stage, the plan describes stake holder responsibilities, stake holder skills, project deliverables, project milestones, overall team strategies in monitoring, reviewing and developing the project, estimation of resources like effort, cost and time. The LCP thus plots and guides project execution at each stage.

A document with such critical information helps all stakeholders understand their roles and commitment in the project and identify the pace, skills and resources required appropriately at each stage. Moreover the LCP aids re-evaluating risk at each stage, and therefore plans are adjusted suitably to mitigate such risks.

1.2 Status of the LCP

The status of this LCP is currently at the Development Commitment Review and is version 3.0. This version of the LCP will be included as part of the Development Commitment Package.

Changes from the previous version of the Life cycle Plan as part of the Foundation Commitment Package are as follows:

- Updated the overall strategy to be used in the rebaselined foundations phase, the development phase (construction and transition iteration) and the operations phase
- Updated project deliverables for the rebaselined foundations phase.
- Updated responsibilities of each member of the team for the development phase (construction and transition iteration).
- Updated Section 6 which include the iteration plan for the development phase, the capabilities to be implemented and tested and preparation plans for the Core Capability Drivethrough.

1.3 Assumptions

- The duration of the project is 24 weeks, which are 12 weeks in fall 2014 and 12 weeks in spring 2015.
- The development team consist of 6 members, where in 2 members will be off campus students.
- The requirements are relatively stable and only minor changes are expected.

2. Milestones and Products

This section outlines the overall strategy that will be/has been employed in each phase of the project along with project deliverables and their due dates.

2.1 Overall Strategy

SnapValet is a new mobile application that needs to be developed from scratch. The team is following Architected Agile Process because a major portion of the application cannot be implemented using existing Non-development Items or Net-Centric Services. The team follows the Incremental Commitment Spiral Model and the strategic details of different phases of the project are outlined below:

Exploration phase
Duration: 09/12/13- 9/26/13 [September 9th to September 26th]
Concept: The team interacted with the client several times to understand the current operational concept, gather detailed requirements for the desired application, analyze the scope of the project, evaluate the feasibility of any improvement in the valet process, explore alternative solutions, and identify beneficiaries of the product. The team also evaluated current and required skills relevant to this project for each member. The team also assessed potential risks and constraints involved in the current scope of the project.
Deliverables: Client Interaction Report, Valuation Commitment Package (Feasibility Evidence Description V1.0, Life Cycle Plan V1.0)
Milestone: Valuation Commitment Review
Strategy: One Incremental Commitment Cycle

Valuation phase
Duration: 09/27/13- 10/20/13 [September 27th to October 20th]
Concept: The team negotiated win-win conditions with the client to mitigate risks and constraints identified in the exploration phase, to prioritize the requirements and identify the minimum marketable features of the SnapValet app, to develop the new operational concept. The team identified responsibilities for each member, the overall strategy of the project, approaches & tools to monitor and control progress of the project and estimation of required resources. The team greatly reduced risk by demonstrating the new work flow for the valet process using a complete U/I prototype of the new mobile app. The team also demonstrated a working geo-location "check in" prototype.

Deliverables: Prototype, Progress Report, Foundations Commitment Package which includes Operations Concept Design, Life Cycle Plan, System & Software Architecture Description, Feasibility Evidence Description.
Milestone: Foundation Commitment Review
Strategy: Win-Win negotiation, Prototyping, Defect-tracking, One Incremental Commitment Cycle

Foundations phase
Duration: 10/21/13- 12/05/13 [October 21st to December 5th]
Concept: The team shall explore and evaluate alternative NDIs and Net-centric services that can be integrated with the app (this augments the foundations required to build the app), assess the status of the project, develop a more detailed system architecture specifying technology dependent and independent designs, construct a traceability matrix, and implement prototypes of high-risk features and present them to the client.
Deliverables: Development Commitment Package which includes Operations Concept Design, Life Cycle Plan, System & Software Architecture Description, Feasibility Evidence Description, Test Plan & cases and a prototype.
Milestone: Development Commitment Review
Strategy: Learn required technology, One Incremental Commitment Cycle

Rebaselined Foundations phase
Duration: 12/15/14- 2/11/15 [December 15th to February 11th]
Concept: Our development team has decided to consider the semester break as part of this phase. In this phase, the team will continue building and implementing user screens, google and brain tree APIs, the queue system on the valet side etc. The team will lose few team members in this phase and will receive new members in this phase. We will re-assess project status, project scope and prepare for the development phase. The team will work on the backend for the application and also start working on the Play framework. Also the team will code for the check-in procedure for the valet side.
Deliverables: Re baselined Development Commitment Package. Re baselined prototype.
Milestone: Re baselined Development Commitment Review.
Strategy: Implement and prototype required technology, One Incremental Commitment Cycle

Development Phase – Construction Iteration
Duration: 2/12/15- 4/8/15 [February 12th to April 8th]

Concept: In this phase, the team will develop the app's core capabilities like using pone gap to deploy both the customer side and the valet side of the app functionality. Also get the website for the valet company registration ready . Each capability is assigned to 2 or more developers. The team will in parallel test (unit test, functional test, and integration test) the capability. The team prepares for the Core capability drive through.
Deliverables: Working mobile application in the CCD, Transition Readiness Package.
Milestone: Core capability drive through, Transition Readiness Review.
Strategy: Develop and test different system modules, One Incremental Commitment Cycle

Development Phase – Transition Iteration
Duration: 4/9/15- 5/4/15 [April 9th to May 4th]
Concept: In this phase, the team will develop the app's additional capabilities like look and feel of the app, any other functionality that comes later on from the client side etc. Each capability is assigned to 2 or more developers. The team will in parallel test (unit test, functional test, and integration test) the capability. In this phase the team will rigorously test the application and deploy the app on the google play store. The team will train users, prepare a user guide and a technical manual. We will also have client evaluation.
Deliverables: Operation Commitment Package, Final product Archive.
Milestone: Operation Commitment Review.
Strategy: Testing, Deployment, Training. One Incremental Commitment Cycle

Operations Phase
Duration: 5/5/15- 5/8/15 [May 5th to May 8th]
Concept: The final phase for the project and the team. The team will provide technical and operational support for the application, this effort will be directed towards the client, users and system maintainers.
Deliverables: NA
Milestone: NA
Strategy: Technical and operational Assistance. One Incremental Commitment Cycle

2.2 Project Deliverables

The following are a list of deliverable artifacts grouped by phase.

2.2.1 Exploration Phase

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	9/19/2014	.doc, .pdf	Soft copy
Program Model and Business Process Model	09/21/2104	.pdf	Soft copy
Valuation Commitment Package • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED)	09/29/2014	.doc, .pdf	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy

2.2.2 Valuation Phase

Table 2: Artifact deliverable in Valuation Phase

Artifact	Due date	Format	Medium
Team Prototype Presentation	10/03/2014	.pdf	Soft copy
Draft Foundation Commitment Package • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype Report (PRO) • Operation Concept Design (OCD) • System and Software Architecture Description (SSAD)	10/13/2014	.pdf, .doc	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy
Foundation Commitment Package • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype Report (PRO) • Operation Concept Design (OCD) • System and Software Architecture Description (SSAD)	10/20/2014	.pdf, .doc	Soft copy

2.2.3   Foundations Phase

Table 3: Artifact deliverable in Foundation Phase

Artifact	Due date	Format	Medium
Draft Development Commitment Package • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype (PRO) • Operation Concept Design (OCD) • System and Software Architecture Description (SSAD) • Test Plan and Cases (TPC)	12/01/2014	.pdf, .doc	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy
Development Commitment Package • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype Report (PRO) • Operation Concept Design (OCD) • System and Software Architecture Description (SSAD) • Test Plan and Cases (TPC)	12/08/2014	.pdf, .doc	Soft copy

2.2.4   Rebaselined Foundation Phase

Table 4: Artifact deliverable in Rebaselined Foundation Phase

Artifact	Due date	Format	Medium
Rebaselined Development Commitment Package • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • Prototype Update • System and Software Architecture Description (SSAD) • Test Plan and Cases (TPC) • Transition plan	2/11/2015	.pdf, .doc	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every alternate	.mpp, .pdf	Soft copy

	Wednesday		
Progress Report	Every alternate Wednesday	.xls	Soft copy

2.2.5 Development Phase – Construction Iteration

Table 5: Artifact deliverable in Core Capability Drive Phase

Artifact	Due date	Format	Medium
Core Capabilities Drivethrough Package • Life Cycle Plan (LCP) • Feasibility Evidence Description (FED) • System and Software Architecture Description (SSAD) • Test Plan and Cases (TPC) • Transition plan	3/31/2015	.pdf, .doc	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every alternate Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every alternate Wednesday	.xls	Soft copy

2.2.6 Development Phase – Transition Iteration

Table 6: Artifact deliverable in Transition Readiness Review

Artifact	Due date	Format	Medium
Transition Readiness Review Package • Life Cycle Plan (LCP) • User Manual • Test Plan and Cases (TPC) • Transition plan	4/20/2015	.pdf, .doc	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every alternate Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every alternate Wednesday	.xls	Soft copy

2.2.7 Transition Phase – As built in package

Table 7: Artifact deliverable in As Built Package

Artifact	Due date	Format	Medium
As Built Package • Life Cycle Plan (LCP) • Operational Concept Description • System and Software Architecture Description • Feasibility Evidence Description • User Manual • Test Plan and Cases (TPC) • Test Procedures and Results • Transition plan • Support Plan • Regression Test Package • Source code Baseline • Associated compile time files • Release Description	4/20/2015	.pdf, .doc	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every alternate Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every alternate Wednesday	.xls	Soft copy

2.2.8 Project Showcase –

Table 8: Artifact deliverable for Project Showcase

Artifact	Due date	Format	Medium
Transition Readiness Review Package • Project Presentation • User Manual	4/29/2015	.pdf, .doc	Soft copy
Bugzilla Report	Every Monday	Text	Bugzilla Website
Project Plan	Every alternate Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every alternate Wednesday	.xls	Soft copy

3. Responsibilities

This section outlines the responsibilities of the different stake holders of the project and also describes the current and required skills of each member of the development team.

3.1 Project-specific stakeholder's responsibilities

Our client is Ms. Mona A, an MBA student at USC's Marshall School of Business. SnapValet is a startup. Since SnapValet is a new mobile application that needs to be developed from scratch, Ms. Mona is providing us specific requirements for the first version of this app. Ms. Mona is also responsible for interacting with her investors, SnapValet stake holders, SnapValet clients and valet operators in providing us with their requirements if any. SnapValet users are Valet companies, Valet operators and customers at establishments, and they are responsible for providing requirements and feedback to our client. Maintainers of the app will be SnapValet appointed individuals. The developer is team#03.

3.2 Responsibilities by Phase

The responsibilities of each member of the development team in the different phases of the project is given below in the table.

Table 9: Stakeholder's Responsibilities in each phase

| Team Member / Role | Primary / Secondary Responsibility | | | | |
	Exploration	Valuation	Foundations	Development- Construction Iteration	Development- Transition Iteration
Name: Brian Vanover **Role:** Project Manager, Developer.	**Primary Responsibility:** Plan the Project, Identify Risks, Schedule weekly Meetings **Secondary Responsibility:** Update Bugzilla	**Primary Responsibility:** Manage Client Interaction, Create and follow up action items **Secondary Responsibility:** Update Bugzilla, Develop	**Primary Responsibility:** Detail Project Plan, Record Project Progress, Create and follow up action items **Secondary Responsibility:** Update	**Primary Responsibility:** Detail Project Plan, Record Project Progress, Create and follow up action items **Secondary Responsibility:** Update	**Primary Responsibility:** Detail Project Plan, Record Project Progress, Create and follow up action items **Secondary Responsibility:** Update

		Prototype	Bugzilla, Assess feasibility evidence, Develop prototype	Bugzilla, Develop geo-location module	Bugzilla
Name: Brian Bousman **Role:** Operations Concept Engineer, QFP.	N/A	N/A	N/A	**Primary Responsibility:** Manage project quality, prepare test plans for unit and integration testing	**Primary Responsibility:** Test money transaction module, queue management module.
Name: Patrick Horng **Role:** Feasibility Analyst, Developer.	N/A	N/A	N/A	**Primary Responsibility:** Assess feasibility evidence, Mitigate risks. **Secondary Responsibility:** Develop geo-location module and valet side website	**Primary Responsibility:** Prepare technical manual and user guide. Train new users. Mitigate risk. **Secondary Responsibility:** Test geo-location module and valet side website
Name: Molly Karcher **Role:** IIV & V, Quality Focal Point, Prototyper/Builder.	**Primary Responsibility:** Interact with client ,Analyze current system, **Secondary Responsibility:** Capture requirements, Setup Bugzilla.	**Primary Responsibility:** Capture and score MMFs and win-win conditions. **Secondary Responsibility:** Construct traceability matrix, Identify quality management strategy.	**Primary Responsibility:** Manage project quality, Verify and validate work products using Bugzilla. **Secondary Responsibility:** Assess prototype and components, Prepare list of test cases.	**Primary Responsibility:** Manage project quality, Verify and validate work products using Bugzilla. **Secondary Responsibility:** Develop money transaction module, queue management module.	**Primary Responsibility:** Manage project quality, Verify and validate work products using Bugzilla. **Secondary Responsibility:** Test money transaction module, queue management module.
Name: Ditong Ding **Role:** System Architect, Prototyper/Builder.	**Primary Responsibility:** Interact with client ,Analyze current system, **Secondary**	**Primary Responsibility:** Assess and evaluate NDI/NCS candidates **Secondary**	**Primary Responsibility:** Assess system architecture, Define technology dependent/	**Primary Responsibility:** Assess system architecture. **Secondary Responsibility:**	**Primary Responsibility:** Assess deployment plan and transition environment.

	Responsibility: Setup and update website	**Responsibility:** Analyze and prioritize capabilities of prototype	independent architecture, Specify architecture styles, patterns and frameworks.	Develop profile management module queue management module.	Train new users. **Secondary Responsibility:** Test profile management and queue management system.
Name: Ridhima Manjrekar **Role:** Requirements Engineer, Prototyper/Builder.	**Primary Responsibility:** Analyze current system, Capture requirements **Secondary Responsibility:** Setup winbook	**Primary Responsibility:** Capture and score MMFs and win-win conditions, Capture progress of win-win negotiation. **Secondary Responsibility:** Identify objectives, constraints and priorities.	**Primary Responsibility:** Define the system and software requirements. **Secondary Responsibility:** Analyze priorities and capabilities to prototype, Assess prototype and components, Develop Prototype. Prepare list of test cases.	**Primary Responsibility:** Re-assess requirements. Update winbook. Assess life cycle content, Detail Project Plan. Prepare transition plan **Secondary Responsibility:** Develop profile management module, queue module and valet side website.	**Primary Responsibility:** Prepare support plan, technical manual and user guide. Train new users. Assess life cycle content, Detail Project Plan. Prepare support plan. **Secondary Responsibility:** Test profile management module, queue module and valet side website.

3.3 Skills

Given below are the current and required skills relevant to this project, for each member of the development team.

Table 10: **Development team's skills**

Team members	Role	Skills
Brian Vanover	Project Manager, Developer.	Current skills: Java, C, SQL, various Apache projects, SQL, Microsoft Project, Rally, COINCOMO 2.0, Software Architecture, HTML, CSS, JavaScript, DOM, XML, A.I., Algorithms, Project Management, Data Mining, Machine Learning, Regression

		Analysis, Unit Testing, Android, Server Architecture, Git, Google Places API,. Required skills: iOS, UML, Client-, ICSM, Play framework, PhoneGap.
Brian Bousman	Operations Concept Engineer, Developer.	Current skills: C/C++, MySQL, COINCOMO 2.0, Git, HTML, Javascript, PHP, Android, iOS, Google Places API. Required skills:, Yelp API, PayPal API, Play framework, PhoneGap.
Patrick Horng	Feasibility Analyst, Developer.	Current skills: Java, MySQL, COINCOMO, HTML Required skills: Risk Analysis, , Android API, Google Map API, Paypal API,JavaScript, Play framework, PhoneGap
Molly Karcher	IIV & V, Quality Focal Point, Dveloper.	Current skills: Java, C/C++, Objective-C, Python, Perl, Erlang, Elixir, HTML/CSS, Javascript/Node.js, SQL/MySQL, MongoDB, Cassandra, Redis, ElasticSearch, Git, Jenkins, Bugzilla/JIRA, UML, XML Required skills: Android, Google Places API, , Play framework, PhoneGap , transaction management tools
Ditong Ding	System Architect, Developer.	Current skills: JSP, Java, SSH, MySQL, Oracle, Tomcat, Weblogic, UML, Android API. Required skills: Google Map API, , Play framework, PhoneGap.
Ridhima Manjrekar	Requirements Engineer, Life	Current skills: Bugzilla,

	Cycle Planner, Developer.	Winbook,Java, C/C++,MySQL, XML, HTML,CSS,JavaScript, PHP, Ajax, Bootstrap Required skills:GitHub, Android API, Google Map API, Playframewok, Phone Gap.

4. Approach

This section outline the approach/tools the team uses to better track and control the progress of the project.

4.1 Monitoring and Control

The following are documentation and tools used for tracking and controlling the progress of the project:

- Bi-weekly progress report: To track current risks, COTS and activities planned for the next 2 weeks.
- Bi-weekly project plan: To plan future events and assign task to project members.
- Weekly meeting: Twice a week to discuss/show our individual work done as our 'role', integrate our work and plan for the next week.
- Client Meeting: Once a week, to show our work to the client and document feedback / requirement changes.
- Bugzilla: To track bugs, defects in tasks/documentation among team members.
- Winbook: During the requirements capture and prioritization, the client and the team used winbook to post issues, comments on different win conditions.
- GitHub: version control

4.1.1 Closed Loop Feedback Control

Internally within the team, we use the following as means of communication and feedback/ review of each other's work:

- Google Groups: The team has created a google group, and we use the group to post/discuss issues, share hyperlinks, media and documentation amongst ourselves. We review posts/documents uploaded by other team members.
- Facebook Messenger: The team has created a group chat and this is our primary means of scheduling team meetings, team events.
- Bugzilla: Team uses this to track bugs or defects in tasks and documentation assigned to specific team members.

4.1.2 Reviews

The team uses the following review strategies as and when it is possible:

- IIV&V review: Every document uploaded is reviewed by this team member and bugs/tickets are issued incase defects are detected.
- Peer review: During team meeting we review each other's progress and work. Also, since we share documents on the google group. Any team member is allowed to post issues/comment on any documentation.
- Review by teaching staff: Team can also get its work reviewed by the teaching staff of the course.

4.2 Methods, Tools and Facilities

The following table summarizes the tools the team uses in the project:

Table 11: **Tools used for the project**

Tools	Usage	Provider
Winbook	Capture win conditions and MMFs and prioritize them.	USC
Bugzilla	Track and report defects / bugs	USC
Google Group	Share and review documents	Google
Facebook Messenger	Organize/schedule team meetings,	Facebook
Project Website	Main repository of finished documents	USC
Microsoft Project	Project management and planning	Microsoft
Microsoft Visio	Drawing flow charts, swim lane activity diagrams	Microsoft
Microsoft Office	Documentation such as Client Interaction report, LCP, FED, OCD, SSAD, SID.	Microsoft'
Balsamiq	Developing UI prototypes and demonstrating the new work flow.	Balsamiq
Visual Paradigm	UML case tool for modelling UML diagrams	USC
COINCOMOII	Estimation of effort, schedule and cost of the project	USC
Java SDK	Development of project prototype	Oracle
Android SDK	Development of mobile application	Android
Eclipse	Integrated Development Environment	Eclipse Foundation
Google places API	App performs a check in at location.	Google
Brain tree API	App to perform the payment transaction	Brain Tree
Phone Gap	App to deploy the project on iOS and Android platform	Adobe
Play frameworks	Open source web application framework	Play

5. Resources

In this section, we calculate the effort and schedule for our project using COINCOMOII.
The following conditions were used to calculate the cost of our system:
- Budget: $10,000
- Project Duration – 24 weeks (12 weeks in the development phase)
- Programming language used – Java
- Number of modules – 5
- Number of developers - 6

The following are the key modules of development for the SnapValet application along with their estimated source lines of code (SLOC).

Table 12: **List of modules for SnapValet**

No	Module Name	Brief Description	SLOC	REVL
1	Login and profile management	Users (customers/valets) can register on the app and login to their profile. This requires an authentication check.	900	15%
2	Geolocation Check-in	The user can check-in at a location. This links the valet, the customer and the company.	300	5%
3	Communication Module	This sets up the communication channel between the customer and the valet once they check-in at the same location, to communicate with each other.	200	10%
4	Money Transaction Module	The user can pay money via the app to the valet company.	800	10%
5	Queue Management	The queue is a list of requests for car retrieval, this is the main interface for the valet head.	400	10%
6	Website	A simple website where only valet companies can login and manage their employee information.	800	5%

Table 13: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	High (2.48)	The team has a thorough understanding of product objectives. There is scope for concurrent development and there is minimal need of innovative data processing etc.
FLEX	High (2.03)	There is a basic need for software conformance with pre-established requirements and a considerable need for conformance with external interfaces.
RESL	Low (5.65)	The team needs to identify a time to resolve the riskiest module. There is significant amount of uncertainty in some key architecture drivers
TEAM	High (2.19)	There is considerable consistency in stakeholder's objectives and basic experience in working with a team.
PMAT	Low (6.24)	Since this is the first time, most of the team members are working in such an environment the process maturity level is low.

Table 14: COCOMOII Cost Driver, Module 1: Login and profile management

Cost Driver	Value	Rationale
RELY	Nominal (1.00)	A failed login is a moderately recoverable error and loss
DATA	High (1.14)	Login and sign up does require a lot of test data, mostly to handle invalid inputs.
DOCU	Nominal (1.00)	Just about enough documentation to fit the life cycle needs is sufficient.
CPLX	Nominal (1.00)	This will be reused code and has low complexity.
RUSE	Very High (1.15)	The profile management's and login code can be used across multiple projects.
TIME	Nominal (1.00)	Login does not require too much execution time.
STOR	Nominal (1.00)	Profile data does not consume lots of memory.
PVOL	Low (0.87)	Database platform updates less than once a month.
ACAP	High (0.85)	Graduates with good analytical ability
PCAP	High(0.88)	Graduates with good programming ability
PCON	High (0.90)	3 member left the team in spring 2015
APEX	Low (1.10)	Team's average relevant experience is low.
LTEX	Low (1.09)	Team's average relevant experience is low.
PLEX	Very Low (1.19)	Most team members have not worked on android platform before.
TOOL	Very High (0.78)	Main tool is Eclipse with Android SDK plugged in. Mature, Well Integrated tool.
SITE	Extra High	Team collaborates and works on campus.

	(0.80)	
SCED	Nominal(1.00)	Currently there is no expected acceleration or expansion

Table 15: COCOMOII Cost Driver, Module 2: Geolocation Check-in

Cost Driver	Value	Rationale
RELY	Nominal (1.00)	A failed check-in is a moderately recoverable error and loss
DATA	Low (0.90)	There are very few test cases that really challenge the google places API for check-in.
DOCU	Very Low (0.81)	The google api is already well documented code.
CPLX	Nominal (1.00)	This is straightforward code and has low complexity.
RUSE	Very High (1.15)	The geo-location code can be used across product line.
TIME	Nominal (1.00)	Check-in does not require too much execution time.
STOR	Nominal (1.00)	Does not consume lots of memory.
PVOL	Low (0.87)	The related platform being called seldom changes.
ACAP	High (0.85)	Graduates with good analytical ability
PCAP	High(0.88)	Graduates with good programming ability
PCON	High (0.90)	3 member left the team in spring 2015
APEX	Low (1.10)	Team's average relevant experience is low.
LTEX	Low (1.09)	Team's average relevant experience is low.
PLEX	Very Low (1.19)	Most team members have not worked on android platform before.
TOOL	Very High (0.78)	Main tool is Eclipse with Android SDK plugged in. Mature, Well Integrated tool.
SITE	Extra High (0.80)	Team collaborates and works on campus.
SCED	Nominal(1.00)	Currently there is no expected acceleration or expansion

Table 16: COCOMOII Cost Driver, Module 3: Communication Module

Cost Driver	Value	Rationale
RELY	Low (0.92)	A failed communication between valet and customer is a recoverable error and has low loss
DATA	Nominal (1.00)	Moderate amount of code to verify proper communication between valet and customer.
DOCU	Nominal (1.00)	This model can be re-implemented very easily

Cost Driver	Value	Rationale
		even without any comments.
CPLX	Nominal (1.00)	This is straightforward code and has low complexity.
RUSE	Very High (1.15)	Once implemented, this module can be used on other projects.
TIME	Nominal (1.00)	It does not require too much execution time.
STOR	Nominal (1.00)	Does not consume lots of memory.
PVOL	Low (0.87)	Platform does not change often.
ACAP	High (0.85)	Graduates with good analytical ability
PCAP	High(0.88)	Graduates with good programming ability
PCON	High (0.90)	3 member left the team in spring 2015
APEX	Low (1.10)	Team's average relevant experience is low.
LTEX	Low (1.09)	Team's average relevant experience is low.
PLEX	Very Low (1.19)	Most team members have not worked on android platform before.
TOOL	Very High (0.78)	Main tool is Eclipse with Android SDK plugged in. Mature, Well Integrated tool.
SITE	Extra High (0.80)	Team collaborates and works on campus.
SCED	Nominal(1.00)	Currently there is no expected acceleration or expansion

Table 17: COCOMOII Cost Driver, Module 4: Money Transaction Module

Cost Driver	Value	Rationale
RELY	High (1.10)	A failed or faulty transaction may risk high financial loss.
DATA	Nominal (1.00)	There is not much 'code' that can verify money transactions.
DOCU	Low (0.91)	Third party transaction management have their own existing documentation.
CPLX	Nominal (1.00)	This is straightforward code and has low complexity.
RUSE	Very High (1.15)	Once implemented, this module can be used on other projects.
TIME	Nominal (1.00)	It does not require too much execution time.
STOR	Nominal (1.00)	Does not consume lots of memory.
PVOL	Low (0.87)	3^{Rd} party money transaction's platform seldom changes.
ACAP	High (0.85)	Graduates with good analytical ability
PCAP	High(0.88)	Graduates with good programming ability
PCON	High (0.90)	3 member left the team in spring 2015

Cost Driver	Value	Rationale
APEX	Low (1.10)	Team's average relevant experience is low.
LTEX	Low (1.09)	Team's average relevant experience is low.
PLEX	Very Low (1.19)	Most team members have not worked on android platform before.
TOOL	Very High (0.78)	Main tool is Eclipse with Android SDK plugged in. Mature, Well Integrated tool.
SITE	Extra High (0.80)	Team collaborates and works on campus.
SCED	Nominal(1.00)	Currently there is no expected acceleration or expansion

Table 18: COCOMOII Cost Driver, Module 5: Queue Management

Cost Driver	Value	Rationale
RELY	Nominal (1.00)	A failed queue management is a moderately recoverable error and loss
DATA	High (1.14)	There can be a fair amount of code to test the validity of the queue.
DOCU	Nominal (1.00)	This is a simple FIFO queue and does not require detailed documentation.
CPLX	Nominal (1.00)	This is straightforward code and has low complexity.
RUSE	Very High(1.15)	Once implemented, this module can be used on other projects.
TIME	High (1.11)	The queue remains continuously displayed/ updated for the valet for extended periods of time.
STOR	Nominal (1.00)	Does not consume lots of memory.
PVOL	Low (0.87)	This is java running on Android. Android seldom changes version. (in a span of one year)
ACAP	High (0.85)	Graduates with good analytical ability
PCAP	High(0.88)	Graduates with good programming ability
PCON	High (0.90)	3 member left the team in spring 2015
APEX	Low (1.10)	Team's average relevant experience is low.
LTEX	Low (1.09)	Team's average relevant experience is low.
PLEX	Low (1.09)	Most team members have not worked on android platform before.
TOOL	Very High (0.78)	Main tool is Eclipse with Android SDK plugged in. Mature, Well Integrated tool.
SITE	Extra High (0.80)	Team collaborates and works on campus.
SCED	Nominal(1.00)	Currently there is no expected acceleration or expansion

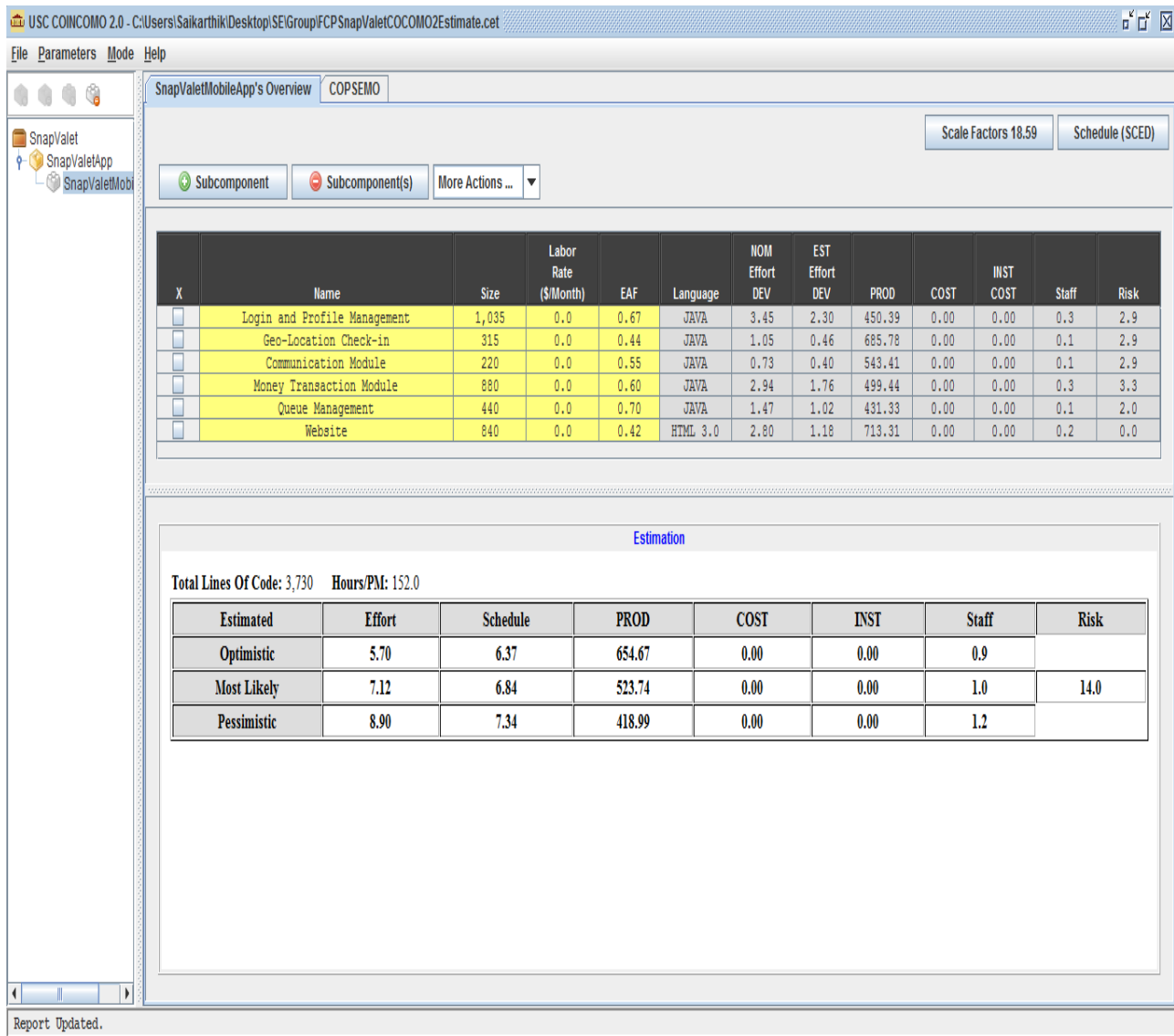**Table 19: COCOMOII Cost Driver, Module 6: Website**

Cost Driver	Value	Rationale
RELY	Nominal (1.00)	A failed website is a moderately recoverable error and loss. The website is seldom used by valet companies.
DATA	Nominal(1.00)	Website testing does not require a lot of test data, mostly required to handle invalid inputs.
DOCU	Nominal (1.00)	Just about enough documentation to fit the life cycle needs is sufficient.
CPLX	Nominal (1.00)	This will be reused code and has low complexity.
RUSE	Very High (1.15)	The website can be used across multiple projects, as a code base for more website functionalities.
TIME	Nominal (1.00)	Website does not require too much execution time.
STOR	Nominal (1.00)	Website data does not consume lots of memory.
PVOL	Low (0.87)	Browsers platform updates less than once a month.
ACAP	High (0.85)	Graduates with good analytical ability
PCAP	High(0.88)	Graduates with good programming ability
PCON	High (0.90)	3 member left the team in spring 2015
APEX	Nominal (1.00)	Team's average relevant experience is about 1 yr.
LTEX	Nominal (1.00)	Team's average relevant experience is about 1yr.
PLEX	Nominal (1.00)	Most team members have worked on developing websites before.
TOOL	Very High (0.78)	Main tool is Adobe Dreamweaver. It is Mature, and well integrated tool.
SITE	Extra High (0.80)	Team collaborates and works on campus.
SCED	Nominal(1.00)	Currently there is no expected acceleration or expansion

The following is the COINCOMOII estimate of effort based on the scale drivers and cost drivers mentioned above.

Figure 1: COINCOMO II Estimate

USC COINCOMO 2.0 - C:\Users\Saikarthik\Desktop\SE\Group\FCPSnapValetCOCOMO2Estimate.cet

File Parameters Mode Help

SnapValetMobileApp's Overview | COPSEMO

SnapValet
SnapValetApp
SnapValetMobi

Scale Factors 18.59 | 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
☐	Login and Profile Management	1,035	0.0	0.67	JAVA	3.45	2.30	450.39	0.00	0.00	0.3	2.9
☐	Geo-Location Check-in	315	0.0	0.44	JAVA	1.05	0.46	685.78	0.00	0.00	0.1	2.9
☐	Communication Module	220	0.0	0.55	JAVA	0.73	0.40	543.41	0.00	0.00	0.1	2.9
☐	Money Transaction Module	880	0.0	0.60	JAVA	2.94	1.76	499.44	0.00	0.00	0.3	3.3
☐	Queue Management	440	0.0	0.70	JAVA	1.47	1.02	431.33	0.00	0.00	0.1	2.0
☐	Website	840	0.0	0.42	HTML 3.0	2.80	1.18	713.31	0.00	0.00	0.2	0.0

Estimation

Total Lines Of Code: 3,730   Hours/PM: 152.0

Estimated	Effort	Schedule	PROD	COST	INST	Staff	Risk
Optimistic	5.70	6.37	654.67	0.00	0.00	0.9	
Most Likely	7.12	6.84	523.74	0.00	0.00	1.0	14.0
Pessimistic	8.90	7.34	418.99	0.00	0.00	1.2	

Report Updated.

From the above estimate: In the worst case, it requires about 152 hours per month for 7.34 months' worth of effort. Since the development team has 6 members, each working for at least 18 hours a week during the development phase. That's about 6*18*4 = 432 hours/per month. At this rate the project can be completed in 152*7.34/432 = 2.6 months.

As stated above the team has 12 weeks in the development phase and the estimate comes up to about 11 weeks. Therefore this project is feasible in this time frame.

6. Iteration Plan

This section outlines the overall plan for each development iteration. It describes the capabilities to be implemented, tested and those that will not be tested.

6.1 Plan

The first iteration of development process focuses on getting the basic SnapValet mobile application setup with its core capabilities implemented. There will be a low emphasis on user interface design in this iteration. The goal of the first iteration is to have the basic sequence of screens on the customer and on the valet side of the app to work. The capabilities to be implemented are described in the table below. Different team members are assigned in groups to work on one or more capability. The team will show the client the basic functionalities and work accordingly on the feedback. This iteration covers half the development phase - construction iteration.

The second iteration of development process focuses on improving the user interface of the mobile application and adding any new additional capabilities that may prop up. The team will also setup the client side website in this iteration and integrate it with the mobile application. Client will be shown the new user interface and the team will work on the feedback. This iteration along with the first iteration will cover the core capability drive through review and the development phase – construction iteration.

The third and final iteration involves rigorous testing and a transition of the mobile application over to the system maintainers and system users. The primary test plan and various test cases have been identified and elaborated in the artifact "Test Plan & Cases". Different test cases are assigned to team members and are tested and fixed in parallel. This iteration covers the operations commitment review and the development phase – transition iteration.

6.1.1 Capabilities to be implemented

The following capabilities have been noted in order of similarity / implementation dependence with one another. Also, the dependency of each capability is documented with either its use case and/or the operational capability as documented in the 2 artifacts "System and Software architecture Description" and the "Operational Concept Description".

Table 20: Construction iteration capabilities to be implemented

ID	Relation	Capability	Description	Priority	Iteration
1	UC-1,5	Geo-Location check in	A customer and a valet should be able to check-in at the establishment (location) they are at.	High	1
2	UC-1,5 OC-7	Auto generate suggestions for establishments	A customer or a valet see a list of suggestions for nearby snapValet registered clients or establishments during check-in.	Medium	2
3	UC-6, OC-1	Mobile transaction	A customer should be able to pay for valet service using his credit card on the application.	High	1
4	UC-6, OC-1	Change credit card	A customer is allowed to add/ change his credit card of use, when he is at the payment screen.	Medium	2
5	UC-6, OC-4	Ticket number entry	The customer must be able to enter his valet ticket number into the application.	High	1
6	UC-6, OC-2	Request Vehicle	A customer should be able to request for his vehicle via the app.	High	1
7	UC-4, OC-5	Retrieval Notification	A customer should receive a notification on his device when his vehicle is being retrieved.	High	1
8	UC-4, OC-3	Queue : Retrieve	The valet is able to visually validate the ticket number and then notify the customer of car retrieval.	High	1
9	UC-4, OC-3	Queue : Report invalid ticket number	The valet is able to notify a customer that he/she entered a wrong ticket number	High	1
10	UC-4, OC-3	Queue : Close out request	A valet is able to close out a served request and remove it from the queue	High	1
11	UC-4, OC-3	Queue : Color coded requests	The valet is able to see color coded requests on the queue that indicate the current status of the request	Medium	2
12	UC-2	Start and close out a shift	A valet should be able to start a shift for other valets to add on to and be able to close out a shift.	High	1
13	UC-2,7	Profile management	A customer and a valet are able to register and create a profile on	High	1

			the app.		
14	UC-2	Remain signed in	A customer should be able to remain signed in after a complete car request and payment process.	Low	2
15	UC-8,9,11,12,13 OC-6	Client Website	A snapValet client is able to add/remove his locations (establishments) and employees on a website.	Medium	2

6.1.2 Capabilities to be tested

The following is a list of capabilities that will be tested along with their corresponding iteration. Also documented is the corresponding test case from the artifact "Test Plan & Cases". The test case number mentioned below have already been mapped/ traced to their corresponding win conditions (requirements) in the above mentioned artifact.

Table 21: Construction iteration capabilities to be tested

ID	Relation	Capability	Description	Priority	Iteration
1	TC-02-01, TC-03-01	Geo-Location check in	Valet and customer check-in through the app. Their location needs to be correctly logged in the back end.	High	1
2	TC-03-01	Auto generate suggestions for establishments	A customer or a valet see a list of suggestions for nearby snapValet registered clients or establishments during check-in.	Medium	2
3	TC-05-01	Mobile transaction	A customer should be able to pay for valet service using his credit card on the application.	High	1
4	TC-05-01	Change credit card	A customer is allowed to add/ change his credit card of use, when he is at the payment screen.	Medium	2
5	TC-04-01	Ticket number entry	The customer must be able to enter his valet ticket number into the application.	High	1
6	TC-04-01	Request Vehicle	A customer should be able to request for his vehicle via the app.	High	1
7	TC-02-04	Retrieval Notification	A customer should receive a notification on his device when	High	1

			his vehicle is being retrieved.		
8	TC-02-04	Queue : Retrieve	The valet is able to visually validate the ticket number and then notify the customer of car retrieval.	High	1
9	TC-02-04	Queue : Report invalid ticket number	The valet is able to notify a customer that he/she entered a wrong ticket number	High	1
10	TC-02-04	Queue : Close out request	A valet is able to close out a served request and remove it from the queue	High	1
11	TC-02-04	Queue : Color coded requests	The valet is able to see color coded requests on the queue that indicate the current status of the request	Medium	2
12	TC-02-02, TC-02-06	Start and close out a shift	A valet should be able to start a shift for other valets to add on to and be able to close out a shift.	High	1
13	TC-01-01, TC-01-02, TC-01-03	Profile management	Profile creation for both customers and valet are tested. In addition, both must be able to edit their information on the profile.	High	1
14	TC-02-06	Remain signed in	A customer should be able to remain signed in after a complete car request and payment process.	Low	2
15	TC-06-01, TC-06-02, TC-06-03, TC-06-04, TC-06-05	Client Website	A snapValet client is able to create a valet administrative account and also able to add/remove his locations (establishments) and employees on a website.	Medium	2

6.1.3 Capabilities not to be tested

We will be testing all the capabilities over the three iterations.

6.1.4 CCD Preparation Plans

The Core Capability Drive through (CCD) is scheduled to be on March 25th, 2015. The following stakeholders will be involved in the CCD:
  ➢ Client

> ➢ A valet
> ➢ Development team

Dry runs:
The development team will carry out dry runs at establishments around Los Angeles to test different capabilities of the app before the CCD date.

Sample Usage Scenario at CCD:
Our plan is to have our client assume the role of a snapValet 'customer' and a current valet operator to be a 'valet'. The 2 users will then test the application on site. The following should be the sequence of events for both users:
> ➢ Customer – register, log in, check-in, enter a ticket number, request a vehicle, pay for service, receive car retrieval request and log out.
> ➢ Valet– register, login, check-in, start shift, validate ticket number, send car retrieval notification, end shift, log out.

This will be the core set of capabilities to be used and demonstrated in the CCD.

Risk Management:
The following are some measures the development team needs to carry out to ensure a smooth CCD.
> ➢ Continue prototyping through the semester holidays to identify and mitigate technical risk.
> ➢ Develop and test the mobile application well in advance to the day of the CCD.
> ➢ Ensure the server is up and running by pinging the server.
> ➢ The two mobile devices have access to internet.
> ➢ Double-check whether every requirement has been implemented.

Feedback Form:
The team decided to use a feedback form to properly document issues, problems, comments and feedback in a printed form. The feedback form will be filled in by the client, the valet, CS577a instructors and other students present at the CCD.

Figure 2: CCD feedback form.

Response number	Email for contact (optional)	Valet side experience rating (1-10)	Customer side experience rating(1-10)	What did you like?	What needs to be improved?