Life Cycle Plan (LCP)

Focus

Team 08

Name	Role
Steven Holland	Requirements Engineer, Prototyper, Feasibility Analyst
Arik Oganesian	Operation Concept Engineer, Software Architect, Prototyper
Marco Alvarez	Feasibility Architect, Software Architect, Requirements Engineer
Pin-Chih (Bill) Lin	Prototyper, Software Architect, Project Manager
Tatsuhiko (Tats) Tomita	Lifecycle Planner, Software Architect, Requirements Engineer
Hamed Sadeghi	Project Manager, Prototyper
Dennis Xiang	IIV&V, Quality Focal

2/20/2017

Version History

Date	Author	Version	Changes made	Rationale
10/12/16	Tats	0.9	Original template for use with Focus v1.0	Initial draft for ARB package
10/17/16	Tats	1.0	Modified some comments from ARB	For Foundation Commitment package
12/1/16	Tats	2.0	For DC Package	Modified for DC and feedbacks of ARB
2/20/17	Tats	3.0	For RDC Package	Modified for RDC and feedbacks of ARB -Figure 1 (Add CodeReview2) -Figure 2 (Change letter's color) -Figure 4 (Change responsibility)

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the LCP

The purpose of the LCP document is to make sure our development plan would fit all stakeholder's demand, especially what we will develop for each milestone. In addition, this clarifies all stakeholder's responsibilities. This makes us our development progress manage easily because we know what each of us should do. Finally, this states our approach to assure our code's quality.

1.2 Status of the LCP

This document is the first version of LCP for Foundations Commitment Package.

1.3 Assumptions

- The duration of the project is 24 weeks, which are 12 weeks in Fall 2016 and 12 weeks in Spring 2017.
    - 5 or 6 of our team members will continue the project for next Spring semester
        - i.e. 1 or 2 of us will discontinue
    - Only prototype will be available at the end of 1st semester

2. Milestones and Products

2.1 Overall Strategy

Our Content Management System, Focus, is following Architected Agile process because there is no Non-Development Item or Web service that would fit to most of the core capabilities.

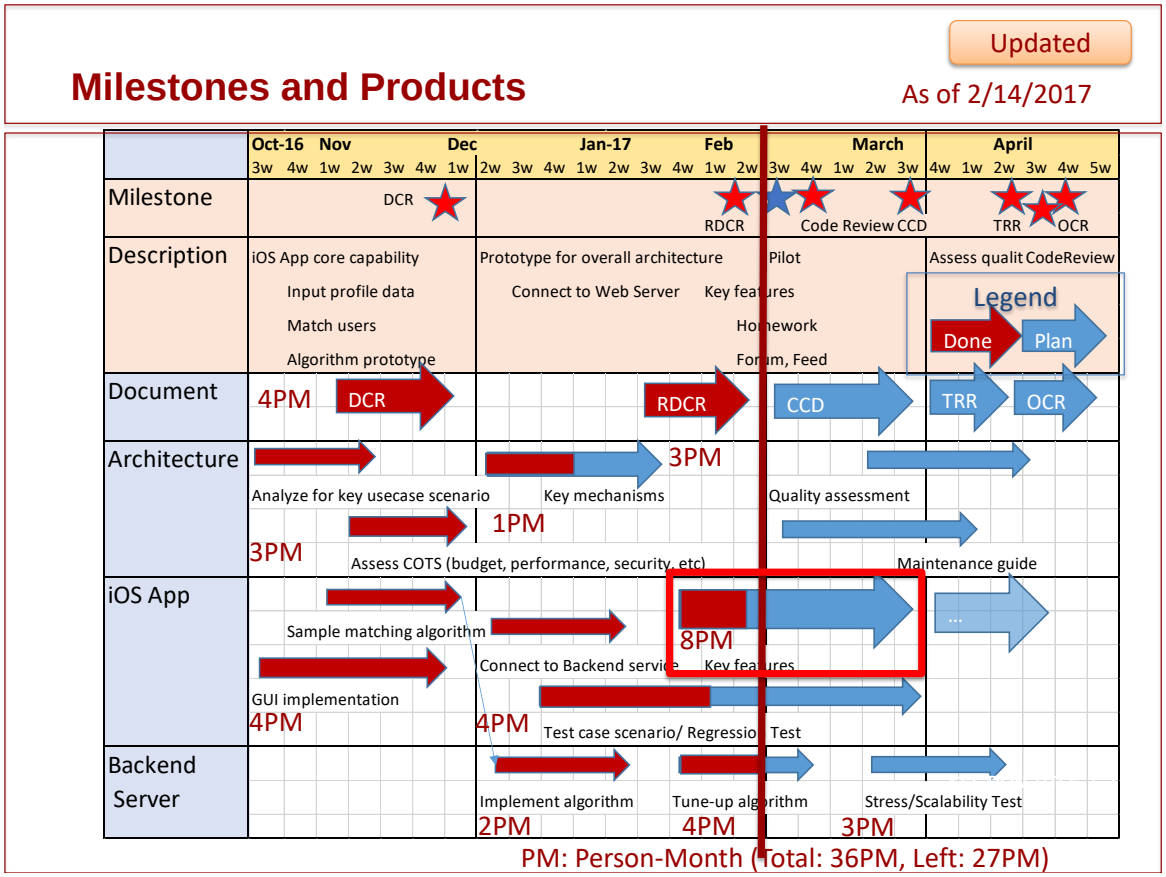


Figure 1 Milestones and Products

Figure 1 shows our development plan for 2 semesters, i.e. 6 more months from as of 10/12/2016. We have five milestones for review. At the same time of each milestone, we are going to incrementally implement our system. Here is the detail:

- Milestone 1: Development Commitment Review
  - We develop sample iOS application which includes the features of input user-profile data, matching users with sample algorithm. It also includes all GUI screen transition. We're going to do this within first semester because one of our members who has a talent of developing iOS app and designing GUI will graduate at this semester.

    - Also we develop architecture in this period. We are going to analyze key use case scenarios and assess COTS in detail such as in terms of budget, performance, security , and so on.
- Milestone 2: Rebaselined Development Commitment Review
    - The Second period is the beginning of new semester. So we don't think we can have a lot of time in this period. So we just prepare backend server and connect it with iOS app. We will start to consider test environment as well. We think this will make it easy to develop all features in next semester.
- Milestone 3: Core Capability Drivethrough
    - The Third period is main period for develop key features for our content management system, such as homework, forum, feed, and so forth. It includes Tune-up algorithm that is one of the main features for our system. And I'd like to emphasize here that it might have a risk that it would take a lot more time to implement all features. In some cases, we might not be able to make it for some features.
- Milestone 4: Transition Readiness Review
- Milestone 5: Operational Commitment Review
    - We create maintenance guide, quality assessment including stress/scalability test.

Furthermore, we estimated effort we need for each item in terms of Person Month. Here, we estimated this Person Month value as tailored for our project, in other words, these values are not related COCOMO value (see in 5). The total of the estimated effort, 36 Person Month, might surpass our capability (here, we estimated we have 7 persons * 5 months = 35 Person Month). Therefore, we still need to keep watch the progress and negotiate with clients what we'll finally have to do.

2.2 Project Deliverables (Iteration plan)

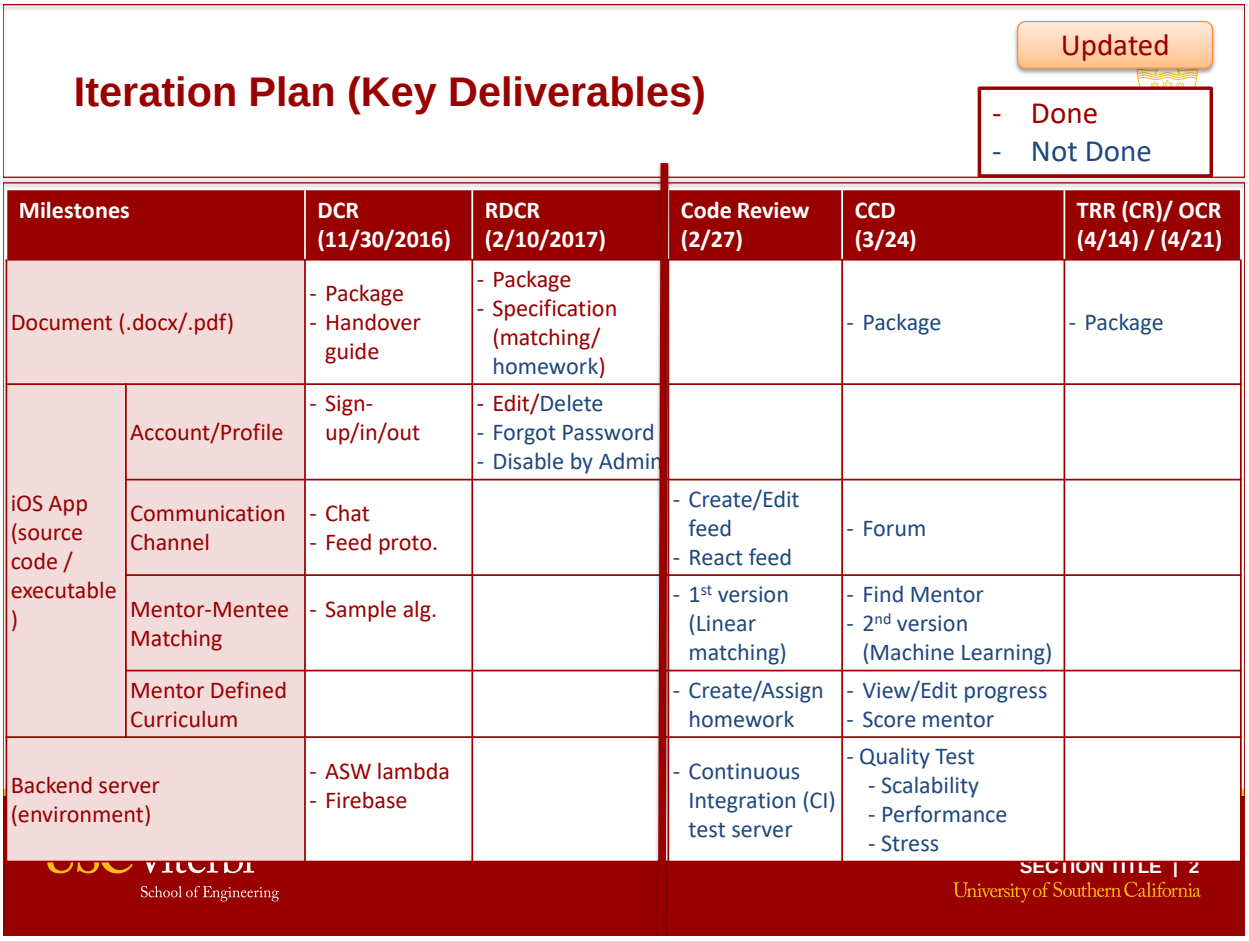


Figure 2 Deliverables overall

Figure 2 shows overall deliverables for this project through all milestones. Since some of us will discontinue after the DCR, we crate handover guide (see ***). By the end of DCR, we get overall GUI transition on iOS application including Sign-up/Sign-in/Sign-out, Chat, and Feed (Prototype). We've also developed Sample Matching Algorithm that uses AWS lambda for computing and Firebase for database as back end servers.

By the end of the RDCR, we create specification document for matching and homework capabilities before we start implementing them because they seem to have complicated specification that is needed to be specified beforehand. We also setup test environment including integration test server by the end of this milestone.

By the end of the CCD, we will finish most of capabilities of this system.

By the end of this project, we finish testing for non-functional requirements such as scalability, performance, and stress test.

2.2.1 Exploration Phase

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	9/16/2016	.doc, .pdf	Soft copy
Project Effort	Every Monday	Web	Jira
Project Plan	Every other Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy
Risk and Defect Report	Every other Wednesday	.xls	Soft copy

2.2.2 Valuation Phase

Table 2: Artifact deliverable in Valuation Phase

Artifact	Due date	Format	Medium
Win Conditions Report	9/26/2016	Web	WinBook
Project Effort	Every Monday	Web	Jira
Project Plan	Every other Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy
Risk and Defect Report	Every other Wednesday	.xls	Soft copy
Prototype - Sample matching algorithm - Backend server with Firebase - iOS app with Firebase	9/30/2016	.txt	Soft copy
Foundation Commitment Package - OCD - PRO - SSAD - LCP (this file) - FED	10/17/2016	.doc, .pdf	Soft copy
Prototype - GUI image - Messaging between users	10/17/2016	.ppt, .pdf	Soft copy

2.2.3 Foundations Phase

Table 3: Artifact deliverable in Foundations Phase

Artifact	Due date	Format	Medium
Project Effort	Every Monday	Web	Jira
Project Plan	Every other Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy
Risk and Defect Report	Every other Wednesday	.xls	Soft copy
Technical Debt Report	Every other Friday	.doc, .pdf	Soft copy
iOS app Prototype - Input user profile - Sample matching feature - Sing-up/in/out - Chat - Feed (prototype)	12/05/2016	Executable (using testify)	Soft copy
Backend Server - AWS lambda for computing matching algorithm - Firebase for data of mentor	12/05/2016	System	System
Development Commitment Package - OCD - PRO - SSAD - LCP (this file) - FED	12/05/2016	.doc, .pdf	Soft copy

2.2.4 Development Phase

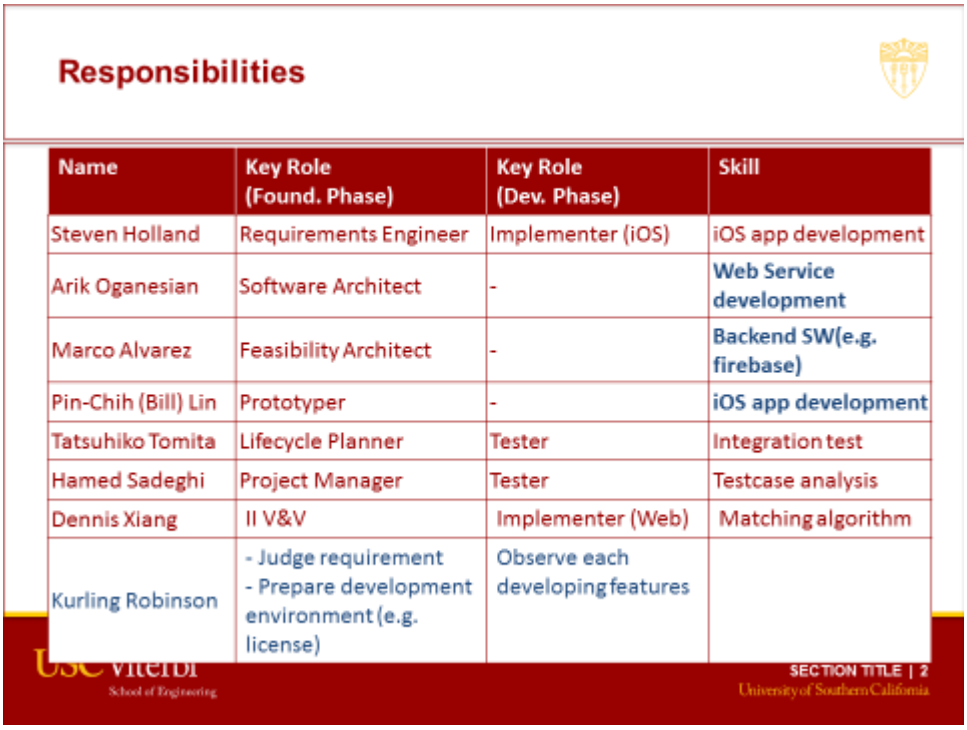
Table 4: Artifact deliverable in Development Phase

Artifact	Due date	Format	Medium
Project Effort	Every Monday	Web	Jira
Project Plan	Every other Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every other Wednesday	.xls	Soft copy
Risk and Defect Report	Every other Wednesday	.xls	Soft copy
Technical Debt Report	Every other Friday	.doc, .pdf	Soft copy
Specification Document - Matching - Curriculum	02/13/2017	.doc, .pdf	Soft copy
Integration Test Server	02/13/2017	System	System

- Jenkins -			
Features - Account/Profile - Homework for Entrepreneurs - Forum - Feed - Matching algorithm	03/13/2017	Executable (using testify)	Soft copy
RDCR Commitment Package - OCD - PRO - SSAD - LCP (this file) - FED	02/15/2017	.doc, .pdf	Soft copy
Operational Commitment Package - OCD - PRO - SSAD - LCP (this file) - FED	03/31/2017	.doc, .pdf	Soft copy

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities



Responsibilities

Name	Key Role (Found. Phase)	Key Role (Dev. Phase)	Skill
Steven Holland	Requirements Engineer	Implementer (iOS)	iOS app development
Arik Oganesian	Software Architect	-	Web Service development
Marco Alvarez	Feasibility Architect	-	Backend SW(e.g. firebase)
Pin-Chih (Bill) Lin	Prototyper	-	iOS app development
Tatsuhiko Tomita	Lifecycle Planner	Tester	Integration test
Hamed Sadeghi	Project Manager	Tester	Testcase analysis
Dennis Xiang	II V&V	Implementer (Web)	Matching algorithm
Kurling Robinson	- Judge requirement - Prepare development environment (e.g. license)	Observe each developing features	

Figure 3 Responsibilities

Figure 3 shows our project's responsibilities. In Figure 3, The red color means our team members and blue one is our client. Here, three of us, Arik, Marco, Pin-Chih, will discontinue the development phase. Therefore, in next semester, we need more students who have skills of Web service development, Backend Software knowledge, and iOS app development experience.

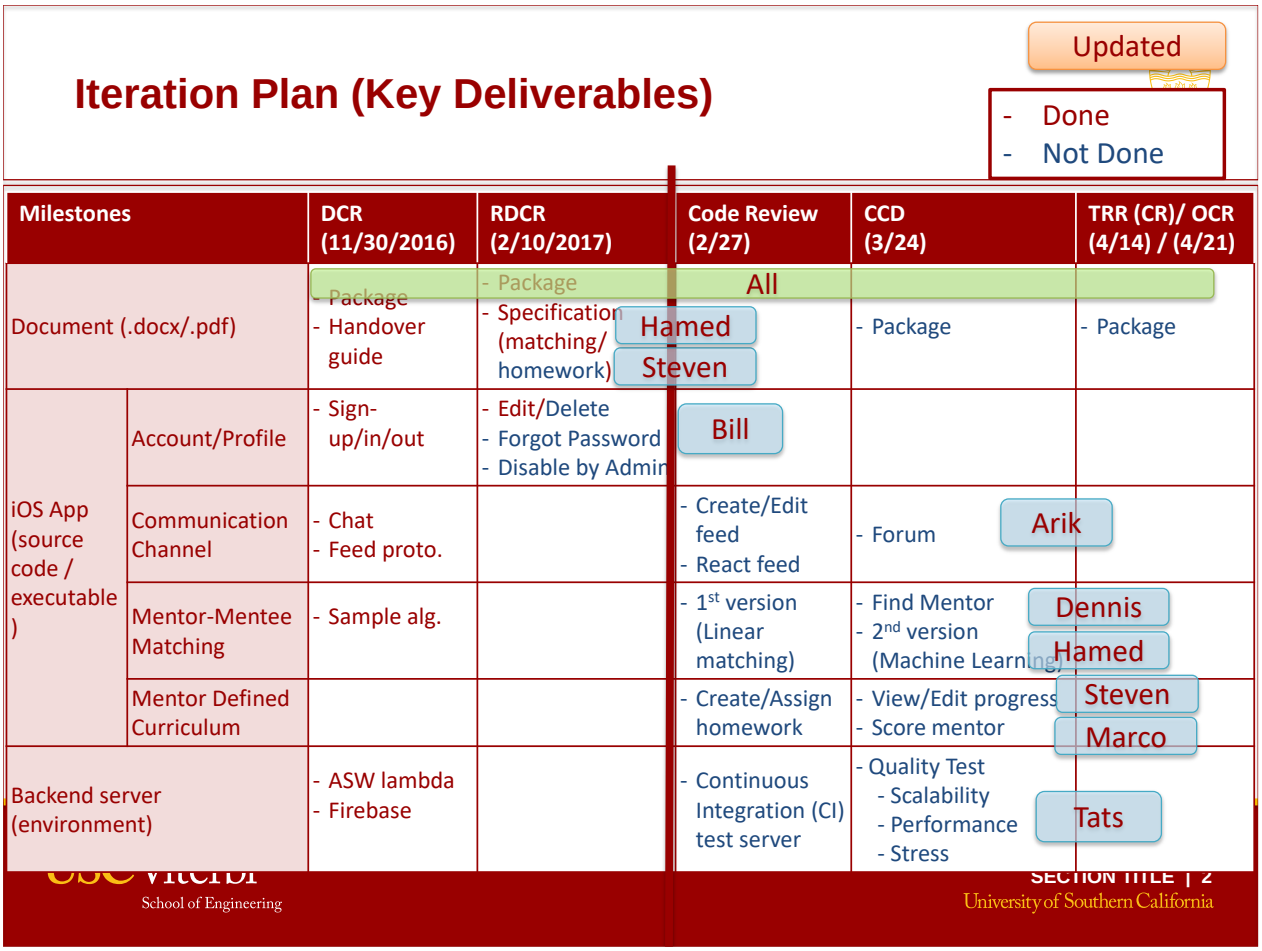


Figure 4 Responsibilities for features

Figure 4 shows which of our members will be in charge of which features in terms of development. See also Figure 2 and compare it if it's hard to see this figure.

3.2 Responsibilities by Phase

Figure 3 and Figure 4 already showed our responsibilities by phase.

3.3 Skills

Figure 3 already showed our skills.

4. Approach

4.1 Monitoring and Control

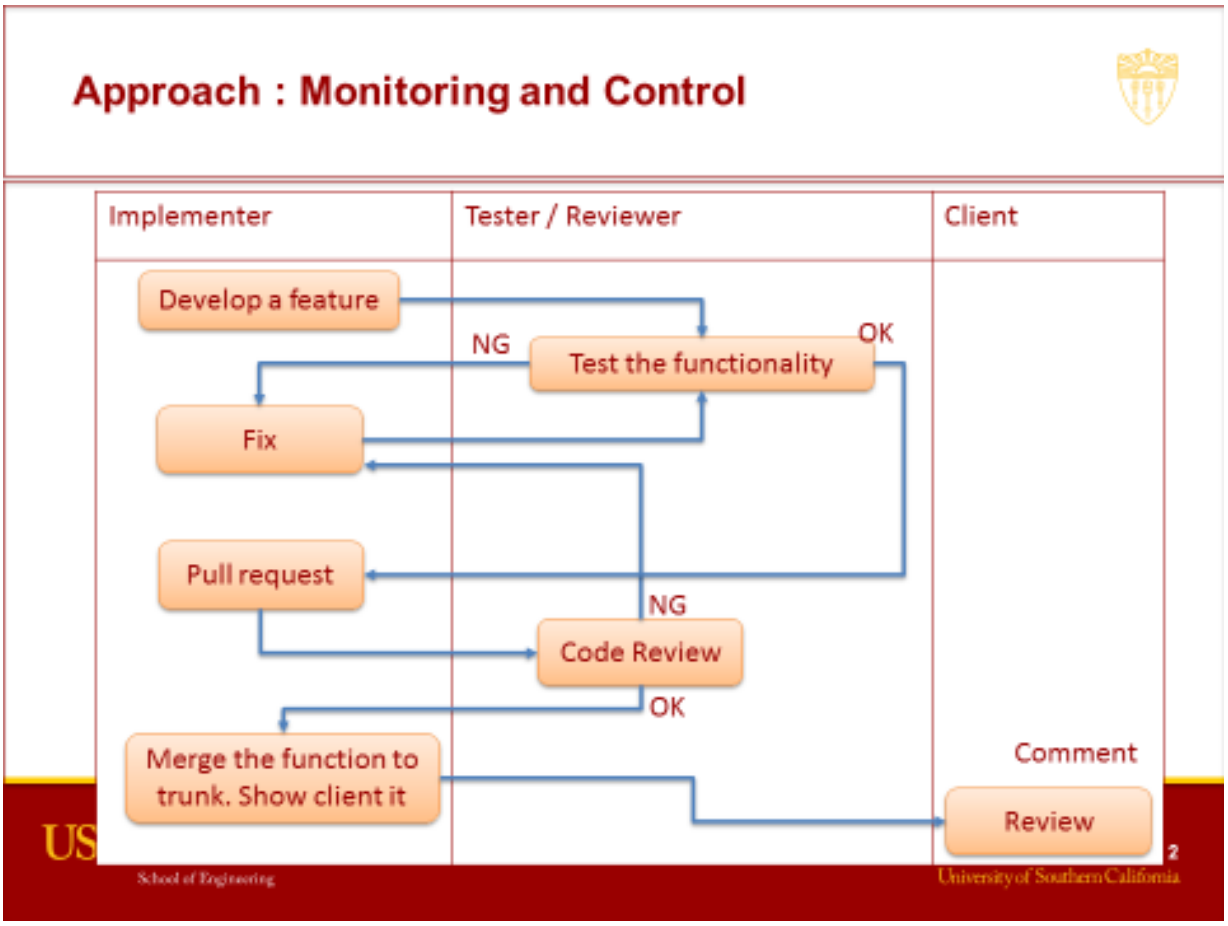


Figure 5 Monitoring and Control

This is our approach to assure the quality of our code. When an implementer finish developing a feature function, it will be tested. And they fix some bugs. When it passes the test, he can pull request. And other members have to review their code. In other words, the code cannot go ahead unless someone reviews it. If it's OK, they merge the function to main branch. And explain it to the client. And client will give us some feedback. By doing this approach, we think we can maintain the quality of our code before it goes to clients.

4.1.1 Closed Loop Feedback Control

Figure 5 already showed our feedback control strategy.

4.1.2 Reviews

Figure 5 already showed our concept for reviews.

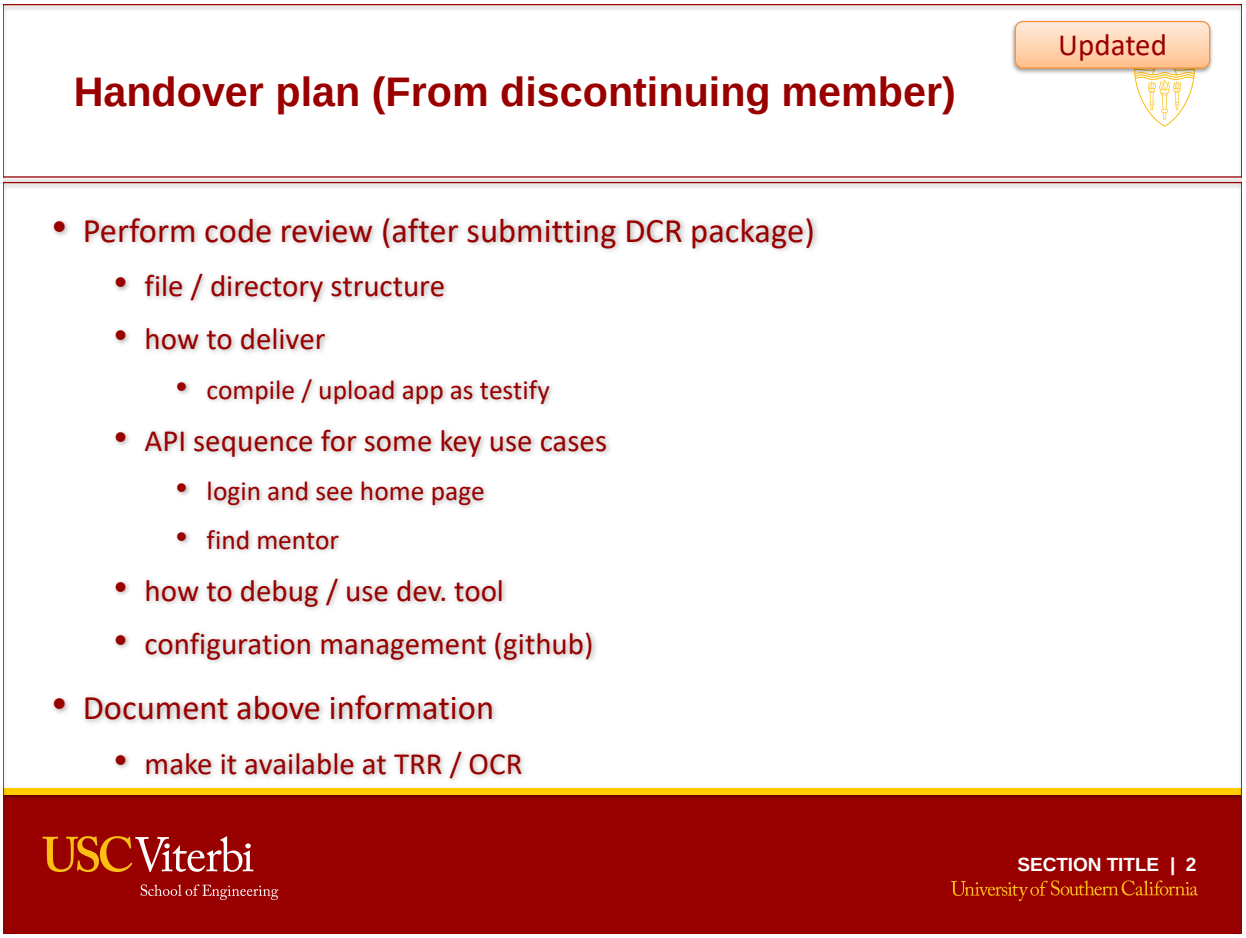


Figure 6 Handover review

Since some of us will quit this project at the end of the 1st semester, we need to perform a review for the code that they've created. Figure 6 contains what we do in that review. We also create documents that include those information and it will be available at the end of this project when we will handover our code to the client.

4.2 Methods, Tools and Facilities

Figure 7 Methods, Tools and Facilities

Figure 7 shows methods or tools to manage our project. In terms of communication, we use Basecamp, that is used to obtain information from our client. We also use Slack to communicate within team members. For development, we use github not only for version control but also for code review as explained in 4.1. We use both Asana and SmartSheet for project management. Asana for detailed communication for each capability and SmartSheet for overall progress / plan management. We're still investigating test tools. But so far, we think we can use Jenkins for integration test. And we are still investigating the coding guideline as well.

5. Resources

Here we use COCOMO II to estimate our effort for this project. Since this project would be a small project, default value for COCOMO is set as High/Low (one level easier than nominal value).
Table 5 and **Error! Reference source not found.** show COCOMO factors that have no-default value.

Table 5: COCOMOII Scale Driver

Cost Estimation

Updated

- Key factors for COCOMO II (Focus on development phase)
  - Default value is High/Low (one level easier than nominal)

Module	KSLOC	Factors	Rate	Rationale
Scale Factors: 15.13				
-		PREC	Low	(unprecedented) No legacy system, no experience
Cost Drivers: 0.38 (iOS), 0.30 (Backend)				
iOS App.	7(3.5 done)	DOCU	High	Not all members will continue the project
		PCON	Low	
Matching Alg. on Backend	1	PLEX	Low	Unfamiliar with Database/Web Service
Both	-	SITE	Very High	Can communicate with clients; DEN student lives in the same area

USC Viterbi School of Engineering

SECTION TITLE | 2
University of Southern California

In scale driver, we set PREC as low since there is no legacy system and we don't have experience that much.

Regarding to Cost drivers, Document and Personal continuity for iOS application would be high and low because not all members will continue this project.
PLEX for Backend system would be low because we are unfamiliar with Database/WebService. However, SITE is very high, since our client comes to USC once a week. So we can communicate with him easily. Also we can meet DEN student as well.

Here is the result of COCOMO.
- Estimated SLOC : 4.0K (Mobile: 3.5K (remaining), Web: 1.0K)
- Effort : 5.2 (Person Month) (4.2 for iOS and 1.0 for Backend)

- Tailored Effort for our project : 5.2 PM * 152/100 = 7.9 PM
    - 1PM (COCOMO) = 152 hours, 1PM(CS577) = 100 hours
- Total hours we need : 7.9 PM * 100 hrs/PM = 790 hrs
- Hours each of us has : 12 hrs/week/person * 10 weeks = 120 hrs/person
- Estimated number of members : 790 hrs / 120 hrs/person = **6.58 persons**

Since we are student, we only spend 100 hours / person month. But COCOMO estimated the effort (Person Month) as 152 hours / PM, therefore we need to recalculate the effort, 5.2 PM, to our number, 7.9 PM. This means we use 790 hours for this project in total.
We will spend 12 hours per week per person and it will continue for 10 weeks for developing phase, that means each member has 120 hours.
The total amount time divided each person's time is equal to the number of persons that we need. This value is within range of the number of our team members.