Life Cycle Plan (LCP)

BlackProfessionals.net

Team #6

Tian Xiang Tan	**Project Manager**
Jhih-Sheng Cai	**System/Software Architect**
Aril Alok Jain	**Requirements Engineer**
Pablo Ochoa	**Prototyper**
Jeng-Tsung Tsai	**Operational Concept Engineer**
Sadeem Alsudais	**IIV & V**
Po-Hsuan Yang	**Life Cycle Planner**

12.07.2014

Version History

Date	Author	Version	Changes made	Rationale
09/26/14	PY	1.0	The initial draft	For Valuation Commitment Package; Fit CSCI 577A course content
10/12/14	PY	2.0	Finish section one to five	For draft Foundations Commitment Package; Fit CSCI 577A course content
10/16/14	PY	2.1	Fix Resource section	Fix Resource section after ARB presentation
10/18/14	PY	2.2	Fix Resource section, Monitoring control and tools	For Foundations Commitment Package; Fit CSCI 577A course content
11/11/14	PY	3.0	Add section 6	For draft Development Commitment Package; Fit CSCI 577A course content
11/22/14	PY	3.1	Fix 3.0	Fit CSCI 577A course content
12/07	PY	3.2	Fix 3.1	Fit CSCI 577A course content

Table of Contents

Table of Tables

1. Introduction

1.1 Purpose of the LCP

“The goal is often not to achieve what you said you would do at the beginning of the project, but to achieve the maximum possible within the time and resources available.” Roger Sherman, Microsoft, 1995. In this document, we state every stakeholder’s works, responsibilities and their skills. Also, state the approach of Monitoring and Control.

1.2 Status of the LCP

This is the first version of Life Cycle Plan (LCP). This version is 1.0 and in the Valuation Commitment Package currently.

1.3 Assumptions

- The duration of this project is 24 weeks. The first 12 weeks is for 2014 fall semester. And the remaining 12 weeks is for 2015 spring semester.
- This is a one year project, assume all stakeholders will continue with this project.
- In this document, all stakeholders would know all their responsibilities clearly.

2. Milestones and Products

2.1 Overall Strategy

Exploration phase

- Duration: 09/5/14 - 9/29/14
- Concept: In this phase, all teammates explore the system, requirements and life cycle plan.
- Deliverables: Valuation Commitment Package (LCP, FED, OCD), Project Reports and Plans, Weekly Effort Report.
- Milestone: Valuation Commitment Review
- Strategy: One Incremental Commitment Cycle

Valuation phase

- Duration: 09/30/2014 – 10/20/2014
- Concept: In this phase, the team develops the operational concept, provide feasibility evidence, develop initial prototype.
- Deliverables: Draft Foundations Commitment Package (OCD,PRO,SSAD,LCP,FED), Project Reports and Plans, Weekly Effort Report.
- Milestone: Foundations Commitment Review
- Strategy: One Incremental Commitment Cycle

Foundations phase

- Duration: 10/21/2014 – 12/8/2014
- Concept: Regular stakeholder meetings are to be taken every week, regular project reports and process reports to be submitted every other Wednesday respectively, project plans are to be prepared and released on project web-page, risk resolution, assessing project status, sharing implementation jobs, designed test cases, transited the system, provided training, got feedback from users and client, and fixed bugs.
- Deliverables: Draft Development Commitment Package, Project Reports and Plans, Weekly Effort Report.
- Milestone: Development Commitment Review
- Strategy: One Incremental Commitment Cycle

2.2 Project Deliverables

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
Valuation Commitment Package • OCD Early Section • LCP Early Section • FED Early Section	09/26/2014	.doc, .pdf	Soft copy
Evaluation of Valuation Commitment Package	09/27/2006	.xls	Soft copy
Project Plan	Every Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy
Risk Analysis	Every Wednesday	Text	DART system

2.2.2 Valuation Phase

Table 2: Artifact deliverable in Valuation Phase

Artifact	Due date	Format	Medium
Foundations Commitment Package • OCD • LCP • FED • PRO • SSAD	10/20/2014	.doc, .pdf	Soft copy
Project Plan	Every Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy
Risk Analysis	Every Wednesday	Text	DART system

2.2.3 Foundations Phase

Table 3: Artifact deliverable in Foundations Phase

Artifact	Due date	Format	Medium
Development Commitment Package • OCD • LCP • FED	12/08/2014	.doc, .pdf	Soft copy

• PRO • SSAD			
Project Plan	Every Wednesday	.mpp, .pdf	Soft copy
Progress Report	Every Wednesday	.xls	Soft copy
Risk Analysis	Every Wednesday	Text	DART system

3. Responsibilities

3.1 Project-specific stakeholder's responsibilities

Table 4: Stakeholder's responsibilities

Prentice E. Parr, JR. Client	Supervise the whole project
Tian Xiang Tan Project Manager	Updates website, Identify risks and feasibility of project, Manage the whole project, Update Bugzilla
Jhih-Sheng Cai System/Software Architect	Build architecture
Aril Alok Jain Requirements Engineer	Gathered requirements, Interact with client
Pablo Ochoa Prototyper	Identify framework, Develop prototype
Jeng-Tsung Tsai Operational Concept Engineer	Conceptualization the system, Interact with clients
Po-Hsuan Yang Life Cycle Planner	Plan the lifecycle and schedule
Sadeem Alsudais IIV & V	Verify work, Update website, Update Bugzilla

3.2 Responsibilities by Phase

Name: Tian Xiang Tan	
Role: Project Manager	
Exploration	Schedule Meetings, Assign Tasks, Plan the project, Analyze Proposed System
Valuation	Plan Project Meeting, Manage Client Interaction, Record Project Progress, Detailed project plan, Manage Bugzilla, Communicate with the clients.
Foundations	Coordinating Meetings with team members and clients, Planning, Monitoring and Control.

Name: Aril Alok Jain	
Role: Requirements Engineer	
Exploration	Analyses of Requirements, Negotiation of requirements.
Valuation	Capture MMF's, Score Win-Win condition, Establish New Operational Concept
Foundations	Capture progress of Win-Win condition.

Name: Jeng-Tsung Tsai	
Role: Operational Concept Engineer	
Exploration	Building the Operational Concept Design Report, System Conceptualization, Assess operational Concept
Valuation	Explore alternatives, Analyze Current System, Identify objectives, constraints, and priorities, Identify organizational and operational
Foundations	System Conceptualization, Implement necessary changes to the OCD and Identify the operational concepts to be developed

Name: Pablo Ochoa	
Role: Prototyper	
Exploration	Project Plan and Progress Report Maintaining
Valuation	Develop the prototype based on top priority requirements & risks
Foundations	Analyze the win conditions to be implemented, Assist in Life Cycle planning

Name: Sadeem Alsudais	
Role: IIV & V	
Exploration	Validation and Verification of COTS Interoperability
Valuation	Analyze Business Cases to Validate the work product, Maintain Bugzilla, Keep track of the win Conditions being the shaper of the project.
Foundations	Assist to maintain FED, Maintain Bugzilla, Evaluating the development, Keep track of the win Conditions being the shaper of the project.

Name: Jhih-Sheng Cai	
Role: System/Software Architect	
Exploration	Identify system modules and functionality,
Valuation	Analyze business case, Develop the proposed system using the Architecture
Foundations	Capturing the business workflow, developing the operational concepts and identifying the system transformation

Name: Po-hsuan Yang	
Role: Life Cycle Planner	
Exploration	Understanding Requirements, Life Cycle Planning
Valuation	Update Life Cycle Plan, Identify Milestones, Identify the features to be implemented
Foundations	Maintaining the Life Cycle Plan and keeping it updated, Maintain Bugzilla, Evaluating the development.

3.3 Skills

Table 5: Teammate's Skills

Team members	Role	Skills
Sadeem Alsudais	IIV & V	Current skills: java, javascript, php, html, css, c#, ajax, jquery, xml, xslt, requirement analysis, testing Required: quality planning, observation and control, understanding of all deliverables for verification and validation
Jhih-Sheng Cai	System/Software Architect	Current skills: Java, html, php, jQuery, javascript Required skill: Software architecture design, life cycle plane
Aril Alok Jain	Requirements Engineer	Current Skills : C/C++, Test Planning, Test Execution, Automation Tool Design Required : Requirement Engineering, Python
Pablo Ochoa	Prototyper	Current Skills: C/C++, Java, Objective-C, Ruby, JavaScript, PHP, HTML, CSS Required: Risk identification and Mitigation, Continuous Integration, Server Administration, Python
Tian Xiang Tan	Project Manager	Current Skills: js, HTML, CSS, PHP, Python, Java, C/C++, Matlab, OpenGL, openCV. Required skill: feasibility analysis skill, project management.
Jeng-Tsung Tsai	Operational Concept Engineer	Current Skills: C/C++, Java, PHP, JavaScript, jQuery, Matlab, OpenCV Required: Operational concept engineering
Po-Hsuan Yan	Life Cycle Planner	Current skill: Java, HTML, PHP, JavaScript, jQuery, ajax,

		js. Required skill: Human resource planning skill Life cycle planning skill

4. Approach

4.1 Monitoring and Control

In order to keep our progress of our project on track, we are relying heavily on Bugzilla and regular weekly meetings. The planning was being done internally via Facebook, email, phone calls. Those are updated on Bugzilla as well. We are also using Progess Report (PR) to keep track of the individual contribution.

The major way we communicate with the client is through Winbook and email. Occasional phone call and in-person meetings are conducted upon if needed.

4.1.1 Closed Loop Feedback Control

In our project, we use Progress and Project Plan to monitor and control our project. Project manager and IIV&V are responsible for bugs tracking. We use Bugzilla for bug tracking. It's provided by USC 577A course website. Every teammate should report and resolve all bugs/defects to Bugzilla.

In order to keep everything on track, we use Google drive to share documents, Github to share our code and we also have Facebook group to discuss schedule, share information, etc.

4.1.2 Reviews

Weekly group review: We have a weekly meeting on Monday before class. In this meeting, all teammates should report their progress, what problems they faced, how to solved,etc for the past week.

WinWin Negotiations: We have WinWin session/negotiation to review and list all values from the SCS. Also, it helps to estimate and prioritize requirements to be done.

4.2 Methods, Tools and Facilities

Table 6: Tools and usage and their providers

Tools	Usage	Provider
Bugzilla	Track activities and report bugs	Open Source
Microsoft Office	Used for documenting deliverables	Microsoft
Google Doc	Share every document	Google

Facebook Group	Discuss everything	Facebook
Github	Share code	Open Source
WinBook	Keep track of the information resulting from negotiations with client, win conditions and issues raised	577A course
COCOMO	Estimate the software developing cost	USC CSSE
ICSM EPG	Better understanding of our roles as software engineers; help with documentation and other submissions	577A course
Heroku	A server program testing service provider	Heroku

5. Resources

For Architected Agile, use COCOMOII for your calculation, for NDI-Intensive project, use COCOTS for your calculation

Identify the following information in order to estimate the software cost:

- Estimated CSCI577a effort: 7 team members at 20 hrs/week for 24 weeks
- Total estimated effort: 3360 hrs.
- Project duration: 24 Weeks
- Programming language used: Ruby on Rails

Table 7: Module lists and SLOC of each module - estimation

No.	Module Name	Brief Description	SLOC	REVL
1	Profile Management	Users can build their profile	1000	10%
2	Message	Users can send message to each others	700	10%
3	Recommendation Engine	Recommend users to users	1200	15%
4	Search and Connect	Users can search friends and connect with others who are not friends	650	15%
5	Social Feed	Users can post or share information	600	10%
6	Invitation	Users can invite their friends to our website	300	5%
7	Advertising	Company can put advertisements on our website	200	3%

Table 8: COCOMOII Scale Drivers

Scale Driver	Value	Rationale
PREC	High	This is similar to the projects that our teammates had developed before
FLEX	High	The client briefly defines how the system would be; however, they are open to discussions with the development team
RESL	High	This is similar to the projects that our teammates had developed before and Rails is quite stable

TEAM	Very High	All stakeholders have strong commitments to achieve the goals of the project. Also, we have an easy way to synchronize the project
PMAT	Normal	The whole team is quite agile, CMMI level 2

Table 9: COCOMOII Cost Drivers of Module 1: Profile Management

Driver	Value	Rationale
RELY	High	Basically, everything is based on profile
DATA	High	100 ≤D/P <1000
DOCU	Normal	Since our project follows agile development pattern and all the documents that can support further maintenance are enough, there are no stringent requirements for micro-level.
CPLX	Normal	It's not complex. Just moderately complex SQL queries and graphic interface management are required, complexity of this module should be low.
RUSE	Low	It's not reusable in other projects
TIME	Normal	≤ 50% use of available execution time because it's simple edit and save into database
STOR	High	Less than 50% because this module doesn't generate data or retrieve data from memory
PVOL	Low	Ruby on Rails is widely used and it's stable
ACAP	High	All teammates have great experiences in this capability
PCAP	Low	We have only one expert Rails programmer in our team. But, rest teammates are doing their best to learn Rails
PCON	Low	Half of teammate will not continue this project
APEX	High	Most teammates have already done similar part before
LTEX	Low	We have an expert Rails programmer in our team. Rest teammates are doing their best to learn Rails. But for front-end, all teammates have more than two years experience
PLEX	Low	As mention before, we have an expert programmer. Rests are learning right now.
TOOL	Normal	Simple front-end, backend, little integration
SITE	Very High	All the team members are all on-campus students and can arrange meetings easily. Additionally, we use wideband electronic communication and occasional videoconference.

Table 10: COCOMOII Cost Drivers of Module 2: Message

Driver	Value	Rationale
RELY	Low	Low, easily recoverable losses. It's not a big deal, if messaging system is down within 30 minutes.
DATA	High	D/P ≥ 1000. It has to store many messages.
DOCU	Normal	Since our project follows agile development pattern and all the documents that can support further maintenance are enough, there are no stringent requirements for micro-level.
CPLX	Normal	It's not complex. Just moderately complex SQL queries and graphic interface management are required, complexity of this module should be low.
RUSE	Low	It's not reusable in other projects
TIME	Normal	≤ 50% use of available execution time
STOR	Normal	Less than 50% because this module doesn't generate data or retrieve data from memory
PVOL	Low	Ruby on Rails is widely used and it's stable
ACAP	High	All teammates have great experiences in this capability
PCAP	Low	We have only one expert Rails programmer in our team. But, rest teammates are doing their best to learn Rails
PCON	Low	Half of teammate will not continue this project
APEX	High	Most teammates have already done similar part before
LTEX	Low	We have an expert Rails programmer in our team. Rest teammates are doing their best to learn Rails. But for front-end, all teammates have more than two years experience
PLEX	Low	As mention before, we have an expert programmer. Rests are learning right now.
TOOL	Normal	Simple front-end, backend, little integration
SITE	Very High	All the team members are all on-campus students and can arrange meetings easily. Additionally, we use wideband electronic communication and occasional videoconference.

Table 11: COCOMOII Cost Drivers of Module 3: Recommendation Engine

Driver	Value	Rationale
RELY	Low	Low, easily recoverable losses.
DATA	Normal	We just have to store a little data in our storage
DOCU	High	Since our project follows agile development pattern and all the documents that can support further maintenance are enough, there are no stringent requirements for micro-level.
CPLX	High	CPLX is high, because we are using complex algorithms for the recommendation engine.
RUSE	Low	It's not reusable in other projects
TIME	Normal	≤ 50% use of available execution time
STOR	Normal	Less than 50% because this module doesn't generate data or retrieve data from memory
PVOL	Low	Ruby on Rails is widely used and it's stable
ACAP	High	All teammates have great experiences in this capability
PCAP	Low	We have only one expert Rails programmer in our team. But, rest teammates are doing their best to learn Rails
PCON	Low	Half of teammate will not continue this project
APEX	High	Most teammates have already done similar part before
LTEX	Low	We have an expert Rails programmer in our team. Rest teammates are doing their best to learn Rails. But for front-end, all teammates have more than two years experience
PLEX	Low	As mention before, we have an expert programmer. Rests are learning right now.
TOOL	Normal	Simple front-end, backend, little integration
SITE	Very High	All the team members are all on-campus students and can arrange meetings easily. Additionally, we use wideband electronic communication and occasional videoconference.

Table 12: COCOMOII Cost Drivers of Module 4: Search and Connect

Driver	Value	Rationale
RELY	Low	Low, easily recoverable losses. Just cannot search users for a while. It's acceptable.
DATA	Normal	We have to store index for search.
DOCU	Normal	Since our project follows agile development pattern and all the documents that can support further maintenance are enough, there are no stringent requirements for micro-level.
CPLX	Normal	It's not complex.
RUSE	Low	It's not reusable in other projects
TIME	Normal	≤ 50% use of available execution time
STOR	Normal	Less than 50% because this module doesn't generate data or retrieve data from memory
PVOL	Low	Ruby on Rails is widely used and it's stable
ACAP	High	All teammates have great experiences in this capability
PCAP	Low	We have only one expert Rails programmer in our team. But, rest teammates are doing their best to learn Rails
PCON	Low	Half of teammate will not continue this project
APEX	High	Most teammates have already done similar part before
LTEX	Low	We have an expert Rails programmer in our team. Rest teammates are doing their best to learn Rails. But for front-end, all teammates have more than two years experience
PLEX	Low	As mention before, we have an expert programmer. Rests are learning right now.
TOOL	Normal	Simple front-end, backend, little integration
SITE	Very High	All the team members are all on-campus students and can arrange meetings easily. Additionally, we use wideband electronic communication and occasional videoconference.

Table 13: COCOMOII Cost Drivers of Module 5: Social Feed

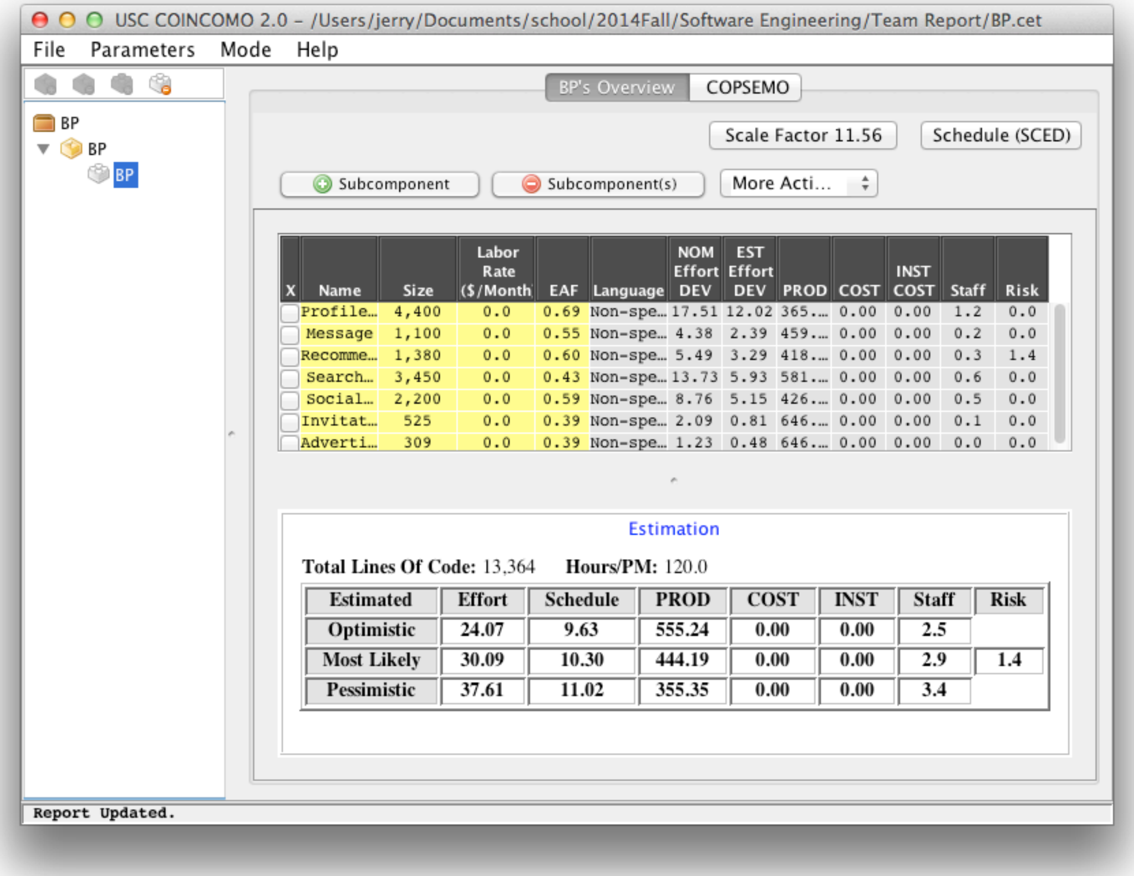
Driver	Value	Rationale
RELY	High	It will be a huge loss if this model doesn't work. Because it helps to show much information on homepage
DATA	High	D/P ≥ 1000. It has to store many messages.
DOCU	Normal	Since our project follows agile development pattern and all the documents that can support further maintenance are enough, there are no stringent requirements for micro-level.
CPLX	Normal	It's not complex. Some simple SQL query to get data from database. And beautify the results
RUSE	Low	It's not reusable in other projects
TIME	Normal	≤ 50% use of available execution time
STOR	Normal	Less than 50% because this module doesn't generate data or retrieve data from memory
PVOL	Low	Ruby on Rails is widely used and it's stable
ACAP	High	All teammates have great experiences in this capability
PCAP	Low	We have only one expert Rails programmer in our team. But, rest teammates are doing their best to learn Rails
PCON	Low	Half of teammate will not continue this project
APEX	High	Most teammates have already done similar part before
LTEX	Low	We have an expert Rails programmer in our team. Rest teammates are doing their best to learn Rails. But for front-end, all teammates have more than two years experience
PLEX	Low	As mention before, we have an expert programmer. Rests are learning right now.
TOOL	Normal	Simple front-end, backend, little integration
SITE	Very High	All the team members are all on-campus students and can arrange meetings easily. Additionally, we use wideband electronic communication and occasional videoconference.

Table 14: COCOMOII Cost Drivers of Module 6: Invitation

Driver	Value	Rationale
RELY	Low	Low, easily recoverable losses. It's acceptable that users can import their contact list and invite their friends later
DATA	Low	We just have to store a little data into our database.
DOCU	Normal	Since our project follows agile development pattern and all the documents that can support further maintenance are enough, there are no stringent requirements for micro-level.
CPLX	Normal	It's not complex. We have to parse information from user's social network and get contact list.
RUSE	Low	It's not reusable in other projects
TIME	Normal	≤ 50% use of available execution time
STOR	Normal	Less than 50% because this module doesn't generate data or retrieve data from memory
PVOL	Low	Ruby on Rails is widely used and it's stable
ACAP	High	All teammates have great experiences in this capability
PCAP	Low	We have only one expert Rails programmer in our team. But, rest teammates are doing their best to learn Rails
PCON	Low	Half of teammate will not continue this project
APEX	High	Most teammates have already done similar part before
LTEX	Low	We have an expert Rails programmer in our team. Rest teammates are doing their best to learn Rails. But for front-end, all teammates have more than two years experience
PLEX	Low	As mention before, we have an expert programmer. Rests are learning right now.
TOOL	Normal	Simple front-end, backend, little integration
SITE	Very High	All the team members are all on-campus students and can arrange meetings easily. Additionally, we use wideband electronic communication and occasional videoconference.

Table 15: COCOMOII Cost Drivers of Module 7: Advertising

Driver	Value	Rationale
RELY	Low	The website can still working if this model doesn't work. Also, it can be fix shortly.
DATA	Low	We don't have to store anything in our storage. Because advertisements are from external websites.
DOCU	Normal	Since our project follows agile development pattern and all the documents that can support further maintenance are enough, there are no stringent requirements for micro-level.
CPLX	Normal	It's not complex because advertisements are function call from external websites.
RUSE	Low	It's not reusable in other projects
TIME	Normal	≤ 50% use of available execution time
STOR	Normal	Less than 50% because this module doesn't generate data or retrieve data from memory
PVOL	Low	Ruby on Rails is widely used and it's stable
ACAP	High	All teammates have great experiences in this capability
PCAP	Low	We have only one expert Rails programmer in our team. But, rest teammates are doing their best to learn Rails
PCON	Low	Half of teammate will not continue this project
APEX	High	Most teammates have already done similar part before
LTEX	Low	We have an expert Rails programmer in our team. Rest teammates are doing their best to learn Rails. But for front-end, all teammates have more than two years experience
PLEX	Low	As mention before, we have an expert programmer. Rests are learning right now.
TOOL	Normal	Simple front-end, backend, little integration
SITE	Very High	All the team members are all on-campus students and can arrange meetings easily. Additionally, we use wideband electronic communication and occasional videoconference.



X	Name	Size	Labor Rate ($/Month	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	Risk
□	Profile...	4,400	0.0	0.69	Non-spe...	17.51	12.02	365....	0.00	0.00	1.2	0.0
□	Message	1,100	0.0	0.55	Non-spe...	4.38	2.39	459....	0.00	0.00	0.2	0.0
□	Recomme...	1,380	0.0	0.60	Non-spe...	5.49	3.29	418....	0.00	0.00	0.3	1.4
□	Search...	3,450	0.0	0.43	Non-spe...	13.73	5.93	581....	0.00	0.00	0.6	0.0
□	Social...	2,200	0.0	0.59	Non-spe...	8.76	5.15	426....	0.00	0.00	0.5	0.0
□	Invitat...	525	0.0	0.39	Non-spe...	2.09	0.81	646....	0.00	0.00	0.1	0.0
□	Adverti...	309	0.0	0.39	Non-spe...	1.23	0.48	646....	0.00	0.00	0.0	0.0

Total Lines Of Code: 13,364 **Hours/PM:** 120.0

Estimated	Effort	Schedule	PROD	COST	INST	Staff	Risk
Optimistic	24.07	9.63	555.24	0.00	0.00	2.5	
Most Likely	30.09	10.30	444.19	0.00	0.00	2.9	1.4
Pessimistic	37.61	11.02	355.35	0.00	0.00	3.4	

Report Updated.

Figure 1: COCOMOII Cost estimation

The form of schedule our project uses is the Independent Variable (SAIV) strategy, 24–week schedule drives development of a set of top priority core capabilities. Therefore, the estimates show the effort required for the project.

In COCOMOII, Hours/PM is 120. Our Hours/PM minimum is 60. So, we can time all staff for 2. We can see pessimistic part is 6.8. Since, we have 7 people, from this effort estimation; we would be able to finish the project in time.

6. Iteration Plan

6.1 Iteration Plan

The first iteration of the development process concentrates on getting the basic website structure up. Running on team member's local machines and Heroku server. This structure will later be modified according to the feedback from the client or some minor changes on design.

During this iteration, team members are assigned different functions to implement during this phase. Top three prioritized functions are assigned to all members individually or in pairs depending on difficulty of functions. This iteration covers the DC Package and the Development – Construction Iteration milestones.

In the second iteration, we plan to get the top three modules complete and go back to the client for feedbacks. Also, we are going to do the rest functions. After finishing all functions, we will do testing. Along with the above, any bugs and improvements on the functionalities from the first iteration were implemented in this iteration. This iteration spanned the Development – Transition Iteration milestone.

6.1.1 Capabilities to be implemented

Below are the capabilities that our team plans to develop in the development phase. The priorities of each capability are different and implemented according to it.

Table 16: Construction iteration 1: High (Profile Management)

ID	Capability	Description
UC-3	Create profile	User can have present on BlackProfessionals.net
UC-4	Create short summary/headline for profile	User can generate customized short summary/headline.
UC-5	Upload profile picture	User can upload profile picture

Table 17: Construction iteration 1: High (Message)

ID	Capability	Description
UC-6	Send message	User can send message to others
UC-7	View sent messages	User views sent messages to others.

Table 18: Construction iteration 1: High (Recommendation Engine)

ID	Capability	Description
UC-8	Get recommendation	Generate recommendations of users to connect with`

Table 19: Construction iteration 2: Medium (Search and connect)

ID	Capability	Description
UC-9	General Search functionality	User can search other users based on certain parameters
UC-10	Advanced Search functionality	User can search other users based on certain parameters
UC-11	User send connection request	User can search other users based on certain parameters
UC-12	User accept/reject connection	User can connect to other users.
UC-13	User cancel connection	User cancel connection
UC-14	Follow/Unfollow a company	User can follow a company in the website.

Table 20: Construction iteration 2: Medium (Social feed)

ID	Capability	Description
UC-24	Post an update	User can post an update
UC-25	View social feed	User can view his connection latest activities
UC-26	Comment on a post	User can post a social feed

Table 21: Construction iteration 2: Medium (Invitation)

ID	Capability	Description
UC-21	Invite friends through email	Invite friends through email
UC-22	Invite friends from LinkedIn	Invite friends from LinkedIn
UC-23	Invite friends to group	Invite friends in our platform to one group

Table 22: Construction iteration 3: Low (Advertising)

ID	Capability	Description
UC-15	Manage Ads	Administration can create advertisement for application.

Table 23: Construction iteration 3: Low (Group formation)

ID	Capability	Description
UC-18	Create group	User can create groups
UC-19	Group management	Admin can manage group status
UC-20	Join/Leave group	User can join or leave group anytime

Table 24: Construction iteration 3: Low (Recommendation letter)

ID	Capability	Description
UC-16	Write recommendation letter	User can write recommendation letter to others.
UC-17	Accept/Reject recommendation letter	User can accept or reject recommendation letter from others.

6.1.2 Capabilities to be tested

All modules and use cases are tested in the development phase. The testing will be initially done by the developers, followed by testing by all the team members and lastly the system needs to be tested by the clients and users. The priority is based on the above order.

6.1.3 Capabilities not to be tested

In the all iterations, all the capabilities will be tested at the end of each iteration.