Life Cycle Plan (LCP)

**Discovery Tool**

**Team 3**

**Josh Bendig - IIV&V**

**Xizhao Deng - Project manager**

**Jingzhou Hong - Requirement Engineer**

**Guancheng Liu - Tester**

**Michael Russo - Prototyper**

**Shenghao Tang - Feasibility Analyst**

**Fan Zhan - Software Architect**

**Yiming Zhang - System Architect**

**10/21/18**

# Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Author | Version | Changes made | Rationale |
| 10/16/18 | XD | 1.0 | section 1-5, 6.1 | Initial draft for Discovery Tool LCP of DC package |
| 10/21/18 | XD | 1.1 | Small fix in section 5 | Redo COCOMOII estimation |

# Table of Contents

[**Version History**](#_eme2pwqcduy5)1

[**Table of Contents**](#_jocgctwe9ot)2

[**Table of Tables**](#_jst3696b1txa)3

[**Table of Figures**](#_6xw7xp8m8mn1)3

[**1. Introduction**](#_x57abmikxym4)4

[1.1 Purpose of the LCP](#_l7otwyl8bdz0) 4

[1.2 Status of the LCP](#_ak147ev4gou2) 4

[1.3 Assumptions](#_9jwrbitisut1) 4

[**2. Milestones and Products**](#_sndo89utqs5)4

[2.1 Overall Strategy](#_z16trig92oc6) 4

[2.2 Project Deliverables](#_h4vfn6ikqx7b) 6

[2.2.1 Exploration Phase](#_kvhkk3lwgms8) 6

[2.2.2 Valuation Phase](#_2avayn9citbh) 6

[2.2.3 Foundations Phase](#_qgl2ru5d21hs) 7

[2.2.4 Development Phase](#_psu5k15h1e4) 7

[**3. Responsibilities**](#_7b98jiovpz3k)8

[3.1 Project-specific stakeholders’ responsibilities](#_k8ojs39yjpkt) 8

[3.2 Responsibilities by Phase](#_7aw8he3wd46w) 8

[3.3 Skills](#_wv09frrpckga) 11

[**4. Approach**](#_yza9lxh4uy10)12

[4.1 Monitoring and Control](#_ezlcubk8jta4) 12

[4.1.1 Closed Loop Feedback Control](#_7kg2fk8j6es6) 12

[4.1.2 Reviews](#_z25x1uxyacb3) 13

[4.2 Methods, Tools and Facilities](#_3yn1fywv3pc3) 13

[**5. Resources**](#_o4dthbfi1jqx)13

[**6. Iteration Plan**](#_huctc6oy6kiz)22

[6.1 Plan](#_1ibgw2oqexrk) 22

[6.1.1 Capabilities to be implemented](#_xu5y2b914gtt) 23

[6.1.2 Capabilities to be tested](#_vzz40amb2jik) 27

[6.1.3 Capabilities not to be tested](#_7oekvcgn53d3) 32

[6.1.4 CCD Preparation Plans](#_xa2o2nc0jvwj) 32

[6.2 Iteration Assessment << TBD after CCD >>](#_4whh9d9fjkjn) 33

[6.2.1 Capabilities Implemented, Tested, and Results](#_8piqi0prkaws) 33

[6.2.2 Core Capabilities Drive-Through Results](#_fm154xotjk4m) 33

[6.3 Adherence to Plan << TBD after CCD >>](#_tmeusorpc1ef) 33

# Table of Tables

Table 1: Artifacts Deliverables in Exploration Phase ………………..………………...………..…….…. 6

Table 2: Artifact deliverable in Valuation Phase ………………..………………...…………..............…. 6

Table 3: Artifact deliverable in Foundations Phase ………………..………………...……....………..…. 7

Table 4: Artifact deliverable in Development Phase ………………..………………...……………....…. 7

Table 5: Stakeholders' Responsibilities in each phase ………………..………………...………………. 10

Table 6: Team skill table ………………..………………...………………………………………….…. 12

Table 7: Tool table ………………..………………...…………………………………………………... 13

Table 8: COCOMOII Scale Driver ………………..…………………………………....………………. 14

Table 9: COCOMOII Cost Driver - Authentication Module …………………………...………………. 15

Table 10: COCOMOII Cost Driver - Content Module ………..……..…………….…...………………. 16

Table 11: COCOMOII Cost Driver - Goal Module ………………..……..………...……..……………. 17

Table 12: COCOMOII Cost Driver - Team/User Module …………....………..……......………...……. 18

Table 13: Construction iteration capabilities to be implemented ………………..…….…..………...…. 27

Table 14: Construction iteration capabilities to be tested ………………..……...……...………………. 31

Table 15: Capabilities implemented, tested, and results ………………..………..…......………………. 32

# Table of Figures

Figure 1: COCOMOII Scale Factor Screenshot ………………………………………...…………....…. 19

Figure 2: COCOMOII Cost Factor Authentication Module Screenshot ………………..….....…………. 19

Figure 3: COCOMOII Cost Factor Content Module Screenshot ……………………………..…………. 20

Figure 4: COCOMOII Cost Factor Goal Module Screenshot …………………………….…..……....…. 20

Figure 5: COCOMOII Cost Factor Team/User Module Screenshot ……………………….....…………. 21

Figure 6: COCOMOII Result screenshot………………………………………………………………….21

# 1. Introduction

## 1.1 Purpose of the LCP

The purpose of this LCP is to document and manage the lifecycle of the Discovery Tool (an idea management system). The LCP artifact is designed to provide descriptions of deliverables, define workflow, identify individual and team responsibility, and track progress over each iteration of the project.

## 1.2 Status of the LCP

The status of the LCP is currently at the development commitment package version 1.0. This is the initial version that is being developed at the end of DCR-ARB presentation.

## 1.3 Assumptions

* The duration of this project is 12 weeks in Fall 2018 semester
* The team will be using 8 individuals, 6 being on-campus students, and 2 being remote DEN students
* The plan is tracked and strictly following MS project
* The project is one-semester only
* The project can repivot once at most
* The client will provide responsive feedback

# 2. Milestones and Products

## 2.1 Overall Strategy

The Discovery Tool idea management system is following the strategy of ICSM NDI/NCS because the system is aimed to be a cloud based web application and it heavily relies on technology such as React, python Flask, JWT, Webpack, Heroku, Swagger UI, Creative TIM, material, mlab and AWS EC2.

**Exploration Phase I**

**Duration**: 9/3/18 - 9/23/18

**Concept**: Identify operational concept, system and software requirements and life cycle plan for the next phase

**Deliverables**: Client Meeting notes, Progress Report, Project plan, use cases, team website

**Milestone**: Valuation Commitment Review

**Strategy**: One Incremental Commitment Cycle, Win-Win negotiation session #1

**Valuation Phase I**

**Duration**: 9/21/18 - 9/23/18

**Concept**: Monitor risks, prioritize requirements, validate crawling based trend identifier, validate API based trend identifier

**Deliverables**: Win-Win condition report, prototype presentation

**Milestone**: Foundation Commitment Review

**Strategy**: One incremental Commitment Cycle, Win-Win negotiation session #2, planning poker, risk analysis assessment

**Foundations Phase I**

**Duration**: 9/24/18 - 9/28/18

**Concept**: Project requirements, Project plan, develop prototype, design preliminary software architecture, investigate API availability

**Deliverables**: None

**Milestone**: None

**Strategy**: One incremental Commitment Cycle

**Exploration Phase II**

**Duration**: 10/2/18 - 10/17/18

**Concept**: Identify operational concept of the new client proposal, system and software requirements and life cycle plan for the next phase, new Win-win conditions

**Deliverables**: Client Meeting notes, Progress Report, Project plan

**Milestone**: Content manager meeting

**Strategy**: One Incremental Commitment Cycle, Content manager meeting

**Valuation Phase II**

**Duration**: 10/10/18 - 10/22/18

**Concept**: Develop frontend UI prototype, develop backend data endpoint prototype, monitor risks, prioritize requirements

**Deliverables**: updated use cases, updated win-win agreement

**Milestone**: None

**Strategy**: One incremental Commitment Cycle, risk analysis assessment

**Foundations Phase II**

**Duration**: 10/10/18 - 10/22/18

**Concept**: Implement ARB deliverables, develop DC package, Project requirements, Project plan, continue to develop prototype, design preliminary software architecture

**Deliverables**: DC package

**Milestone**: DC ARB presentation

**Strategy**: One Incremental Commitment Cycle

**Development Phase**

**Duration**: 10/23/18 - 11/30/18

**Concept**: Develop core capability, implement the full system with agreed features, perform test, implement documentation

**Deliverables**: Project archive, Core capability report

**Milestone**: Core capability presentation, Transition Readiness Review

**Strategy**: One Incremental Commitment Cycle, development, test, deployment

## 2.2 Project Deliverables

### 2.2.1 Exploration Phase

|  |  |  |  |
| --- | --- | --- | --- |
| **Artifact** | **Due date** | **Format** | **Medium** |
| Jira Weekly Survey | Weekly Monday | Jira ticket | Google survey |
| Team website | 9/12/18 | Website | Website |
| Project Plan | Bi-weekly Wednesday | .mpp .pdf | Team site |
| Risk and Defects Report | Bi-weekly Wednesday | .pdf | Team site |
| Progress Report | Bi-weekly Wednesday | .pdf | Team site |

Table 1: Artifacts Deliverables in Exploration Phase

### 2.2.2 Valuation Phase

|  |  |  |  |
| --- | --- | --- | --- |
| **Artifact** | **Due date** | **Format** | **Medium** |
| Jira Weekly Survey | Weekly Monday | Jira ticket | Google survey |
| Project Plan | Bi-weekly Wednesday | .mpp .pdf | Team site |
| Risk and Defects Report | Bi-weekly Wednesday | .pdf | Team site |
| Progress Report | Bi-weekly Wednesday | .pdf | Team site |
| Prototype Presentation | 9/28/18 | .pdf | Team site |

Table 2: Artifact deliverable in Valuation Phase

### 2.2.3 Foundations Phase

|  |  |  |  |
| --- | --- | --- | --- |
| **Artifact** | **Due date** | **Format** | **Medium** |
| Jira Weekly Survey | Weekly Monday | Jira ticket | Google survey |
| Project Plan | Bi-weekly Wednesday | .mpp .pdf | Team site |
| Risk and Defects Report | Bi-weekly Wednesday | .pdf | Team site |
| Progress Report | Bi-weekly Wednesday | .pdf | Team site |
| DC package:   * FED * LCP * SSAD * TPC * PRO * OCD | 10/22/18 | .pdf | Team site |
| Prototype | 10/17/18 | Source code | github |

Table 3: Artifact deliverable in Foundations Phase

### 2.2.4 Development Phase

|  |  |  |  |
| --- | --- | --- | --- |
| **Artifact** | **Due date** | **Format** | **Medium** |
| Jira Weekly Survey | Weekly Monday | Jira ticket | Google survey |
| Project Plan | Bi-weekly Wednesday | .mpp .pdf | Team site |
| Risk and Defects Report | Bi-weekly Wednesday | .pdf | Team site |
| Progress Report | Bi-weekly Wednesday | .pdf | Team site |
| Technical Debt Report | Bi-weekly Friday | .pdf | Team site |
| CCD Report | TBD | .pdf | Team site |
| TRR Report | TBD | .pdf | Team site |
| Functional system | 11/30/18 | Functional web app | AWS, github |
| Documentation | 11/30/18 | Swagger UI rendered doc, .pdf | Github, Team site |

Table 4: Artifact deliverable in Development Phase

# 3. Responsibilities

## 3.1 Project-specific stakeholders’ responsibilities

Other than the stakeholders identified for Discovery Tool so far, which are the client and dev team, we have no any project-specific stakeholders.

## 3.2 Responsibilities by Phase

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Primary / Secondary Responsibility** | | | | | |
| **Team Member / Role** | **Exploration** | **Valuation** | **Foundations** | **Development-Construction Iteration** | **Development-Transition Iteration** |
| **Name:** Joshua Bendig  **Role:** IIV&V,  Requirement Engineer | **Pri. Res:** Participate in Win-win negotiation  **Sec.Res:** Implement system requirements | **Pri. Res:**  Refine system requirements  **Sec.Res:** Review Jira ticket | **Pri. Res:** Track and update system requirements, maintain diagrams for documentation  **Sec.Res:** Review Jira ticket | **Pri. Res:** Validate all system features against system requirement  **Sec.Res:** Review Jira ticket | **Pri. Res:**  Continue validating all system features against system requirement, review documentation  **Sec.Res:** Review Jira ticket |
| **Name:** Xizhao Deng  **Role:** Project Manager, Life Cycle Planner | **Pri. Res:**Establish recurring comms with client and team, set up recurring team meetings  **Sec.Res:** Assign roles, make project plan | **Pri. Res:**  Facilitate client interaction and follow up after Win-win sessions, create action items, host weekly team meeting  **Sec.Res:** Update project progress, assign Jira ticket to team members | **Pri. Res:**  Create and follow up action items, conduct prototype presentation  **Sec.Res:** Update project progress, plan for risk mitigation, assign Jira ticket to team members | **Pri. Res:**  Create and follow up action items, facilitate client communication, documentation  **Sec.Res:** Assign Jira ticket to team members, plan for risk mitigation and identify technical debt | **Pri. Res:**  Facilitate handoff to the client  **Sec.Res:** Assign Jira ticket to team members, plan for risk mitigation and identify technical debt |
| **Name:** Jingzhou Hong  **Role:** Requirements Engineer, Prototyper | **Pri. Res:**  Implement system requirements, Participate in Win-win negotiation  **Sec.Res:** Research potential technology | **Pri. Res:**  Refine system requirements  **Sec.Res:** Develop prototype | **Pri. Res:**  Develop FED for DC package, refine system requirements  **Sec.Res:** Continue working on prototype | **Pri. Res:**  Verify if system requirement, win-win conditions are fully satisfied  **Sec.Res:** Documentation | **Pri. Res:**  Final Review against system requirement  **Sec.Res:** Review documentation |
| **Name:** Guancheng Liu  **Role:** Tester, Quality Engineer | **Pri. Res:**  Participate in Win-win negotiation  **Sec.Res:** Research potential technology | **Pri. Res:**  Design test plans  **Sec.Res:** Develop prototype | **Pri. Res:**  Implement TCP/QFP for DC package  **Sec.Res:** Implement backend and database | **Pri. Res:** Implement test cases, conduct software testing, fix defects  **Sec.Res:** Documentation | **Pri. Res:**  Continue software testing, conduct quality assurance review  **Sec.Res:** Documentation |
| **Name:** Michael Russo  **Role:** Prototyper, Software Architect | **Pri. Res:**  Participate in Win-win negotiation, implement team website  **Sec.Res:** Research potential technology | **Pri. Res:** Develop prototype  **Sec.Res:** Facilitate designing software architecture | **Pri. Res:** Implement slide deck for ARB presentation, implement frontend UI  **Sec.Res:** Documentation, refine software architect | **Pri. Res:**  Implement frontend UI, integration  **Sec.Res:** Documentation | **Pri. Res:**  Improve UI and UX  **Sec.Res:** Documentation |
| **Name:** Shenghao Tang  **Role:** Feasibility Analyst, System Architect | **Pri. Res:**  Participate in Win-win negotiation, implement system architecture, conduct feasibility analysis  **Sec.Res:** Research potential technology | **Pri. Res:** Analyze and prioritize capabilities to prototype  **Sec.Res:** Facilitate designing system architecture | **Pri. Res:** Develop prototype  **Sec.Res:** Implement FED for DC package | **Pri. Res:** Develop backend, integration  **Sec.Res:** Documentation | **Pri. Res:** Integration, fix defects  **Sec.Res:** Documentation |
| **Name:** Fan Zhang  **Role:** Software Architect, Project Manager | **Pri. Res:** Participate in Win-win negotiation, implement software architecture  **Sec.Res:** Research potential technology | **Pri. Res:** Design software architect  **Sec.Res:** Facilitate project management and planning | **Pri. Res:** Implement TCP/QFP for DC package  **Sec.Res:** making ARB presentation slide deck | **Pri. Res:** Implement backend, integration  **Sec.Res:** Documentation | **Pri. Res:** Integration, fix defects  **Sec.Res:** Documentation |
| **Name:** Yiming Zhang  **Role:** System Architect, Prototyper | **Pri. Res:** Participate in Win-win negotiation, design system architecture  **Sec.Res:** Research potential technology | **Pri. Res:** Design the system architecture  **Sec.Res:** Implement prototype for prototype presentation | **Pri. Res:** Implement PRO for DC package  **Sec.Res:** Implement frontend UI prototype | **Pri. Res:** Implement frontend UI  **Sec.Res:** Documentation | **Pri. Res:** Integration, fix defects  **Sec.Res:** Documentation |
| **Name:** Linda Suen  **Role:** Client | **Pri. Res:** Explain the project deliverables, discuss the Win-win conditions, review use cases, attend client meeting | **Pri. Res:** Define system requirements, review initial prototypes, attend client meeting | **Pri. Res:**  Review ARB deliverables, attend client meeting | **Pri. Res:**  Review progress, attend client meeting | **Pri. Res:**  Review documentation, assist the handoff |

Table 5: Stakeholders' Responsibilities in each phase

## 3.3 Skills

|  |  |  |
| --- | --- | --- |
| **Team members** | **Role** | **Skills** |
| Joshua Bendig | IIV&V, Requirements Engineer | **Current Skill:** Java, Python, embedded development, software testing, technical writing, UML diagram  **Required Skill:** Web development, React |
| Xizhao Deng | Project Manager, Life Cycle Planner | **Current Skill:** Python, C, C++, embedded development, technical writing, MS Project  **Required Skill:** Web development, React, high level program management |
| Jingzhou Hong | Requirements Engineer, Prototyper | **Current Skill:** Python, HTML, web development, database, Javascript  **Required Skill:** technical writing |
| Guancheng Liu | Tester, Quality Engineer | **Current Skill:** Python, database, MongoDB, Java, React  **Required Skill:** Software Testing, Crucible |
| Michael Russo | Prototyper, Software Architect | **Current Skill:** Flask, React, Node.JS, AWS, MongoDB, UML diagram, HTML, CSS, UI design, mLab, Swagger UI  **Required Skill:** Crucible |
| Shenghao Tang | Feasibility Analyst, System Architect | **Current Skill:** Python, MongoDB, UML diagram, React  **Required Skill:** Technical writing |
| Fan Zhang | Software Architect, Project Manager | **Current Skill:** Flask, MongoDB, AWS, React  **Required Skill:** High level development |
| Yiming Zhang | System Architect, Prototyper | **Current Skill:** Flask, React, Node.JS, AWS, MongoDB, UML diagram, HTML, CSS, UI design, Material, AWS  **Required Skill:** Technical writing |

Table 6: Team skill table

# 4. Approach

## 4.1 Monitoring and Control

We monitor and control our project by the following approach:

* Project Plan: constantly update project plan (for the next 2 weeks or further ahead) in MS project and make sure the team stick to the plans.
* Bi-weekly Progress Report: Track the progress of the project, list third-party components, and SLOC
* Bi-weekly Risk and Defect Report: Analyze top risks, defects, and come up with mitigation plan
* Bi-weekly Technical Debt Report: Identify technical debt, their lifespan, and mitigation plan
* Jira: track issue, log work, assign tasks
* Win book: catch requirements and priority, maintain design consistency
* Weekly Team Meeting: update the team with plans, report progress, issues, concerns, and come up with mitigation plan
* Client Meeting: maintain responsive communication with the client for progress, issues, and concerns

### 4.1.1 Closed Loop Feedback Control

* Slack: The dev team created group chat in Slack to discuss project, check progress, and talk to the client.
* Google Drive: Share files and documents among the team and the client
* Github: For version control repo hosting, and issue tracking
* Jira: Assign tasks to team member, check progress, and log work.
* Skype: Host team meetings

### 4.1.2 Reviews

* Peer Review: cross review code increments, and documentations
* Client Review: frequent client review over use cases, win-win conditions, and progress
* ARB: scheduled on 10/17 to go over architecture review and DC package
* CCD and TRR: TBD

## 4.2 Methods, Tools and Facilities

|  |  |  |
| --- | --- | --- |
| **Tools** | **Usage** | **Provider** |
| Github | Source Code Version Control | Github |
| MagicDraw | UML diagram / workflow making | No Magic |
| Skype | Team meeting hosting | Microsoft |
| Slack | Team group chat | Slack Tech |
| Winbook | Win condition capture and priority identification | CS577A |
| COCOMO II | Software cost estimation | CS577A |
| MS Project | Project plan and life cycle planning | USC |
| Jira | Log work effort, assign task | USC |

Table 7: Tool table

# 5. **Resources**

Below is the required information in order to estimate the software cost:

* Estimated CSCI577a Effort : 8 team members at 18 hrs/week for 12 weeks
* Total estimated effort: 18 hrs/week x 8 members x 12 weeks = 1728 hours
* Budget information: reasonable spendings will be reimbursed by client
* Project duration: 12 weeks
* Component modules in your development project: User/team Module, Content Module, Goal Module, Authentication Module
* Programming language used: Javascript, Python, HTML, CSS

|  |  |  |
| --- | --- | --- |
| **Scale Driver** | **Value** | **Rationale** |
| Precedentedness | Nominal | No existing system available but the new system will mimic current workflow being practiced. Certain level of reference does exist |
| Development Flexibility | Nominal | Client expects the system to mimic current workflow with small tweaks and improvement |
| Architecture / risk resolution | Nominal | Most risks can be mitigated by more client interaction and dev team time investment |
| Team cohesion | Very High | Team communication and interaction occur in a responsive and timely manner |
| Process maturity | Nominal | Team is comfortable and consistent with ICSM guidelines |

Table 8: COCOMOII Scale Driver

|  |  |  |
| --- | --- | --- |
| **Cost Driver** | **Value** | **Rationale** |
| RELY | High | Authentication is required for the system to be successful |
| DATA | Nominal | The user information data size is fairly low, but it may grow based on team size scaling |
| DOCU | Nominal | Documentation will be required for user guidance and maintenance purpose |
| CPLX | Nominal | Authentication can be implemented by utilizing off-the-shelf framework and tools |
| RUSE | Low | Authentication will be purposely build for this system, no plan for reuse at the moment |
| TIME | Nominal | Authentication will be executed at the beginning and the end of the user session. It does not consume significant execution time resource |
| STOR | Nominal | User information does not utilize significant amount of storage |
| PVOL | Low | The cloud service which the web app utilizes is expected to be stable for majority of the time |
| ACAP | Nominal | The analyst personnel has nominal experience, both academically and industrially, for the task of requirement implementation and design implementation |
| PCAP | High | The developers have related academic and work experience and are determined to be highly capable |
| PCON | Very High | The staff turnover rate of the dev team is low within this semester |
| APEX | High | Developers have moderate to rich experience with web application development |
| LTEX | High | Developers are comfortable with the technologies chosen for this project |
| PLEX | Nominal | Developers have moderate experience with the cloud service for hosting the web app and database |
| TOOL | Nominal | Moderate integration may be necessary for the chosen technologies |
| SITE | Extra High | The team has very responsive and interactive communication/workflow between on-campus and remote members |

Table 9: COCOMOII Cost Driver - Authentication Module

|  |  |  |
| --- | --- | --- |
| **Cost Driver** | **Value** | **Rationale** |
| RELY | High | Content is required for the system to be successful |
| DATA | High | The content data size is high considering amount of content generated in a unit period of time |
| DOCU | Nominal | Documentation will be required for user guidance and maintenance purpose |
| CPLX | High | The subsystem needs to be built from scratch |
| RUSE | Low | Content will be purposely build for this system, no plan for reuse at the moment |
| TIME | Very High | Content operations take significant execution time in client’s workflow |
| STOR | High | Content utilizes significant amount of storage |
| PVOL | Low | The cloud service which the web app utilizes is expected to be stable for majority of the time |
| ACAP | Nominal | The analyst personnel has nominal experience, both academically and industrially, for the task of requirement implementation and design implementation |
| PCAP | High | The developers have related academic and work experience and are determined to be highly capable |
| PCON | Very High | The staff turnover rate of the dev team is low within this semester |
| APEX | High | Developers have moderate to rich experience with web application development |
| LTEX | High | Developers are comfortable with the technologies chosen for this project |
| PLEX | Nominal | Developers have moderate experience with the cloud service for hosting the web app and database |
| TOOL | Nominal | Moderate integration may be necessary for the chosen technologies |
| SITE | Extra High | The team has very responsive and interactive communication/workflow between on-campus and remote members |

Table 10: COCOMOII Cost Driver - Content Module

|  |  |  |
| --- | --- | --- |
| **Cost Driver** | **Value** | **Rationale** |
| RELY | High | Goal is required for the system to be successful |
| DATA | Nominal | The goal data size is moderate considering amount of goal generated in a unit period of time |
| DOCU | Nominal | Documentation will be required for user guidance and maintenance purpose |
| CPLX | Nominal | The subsystem needs to be built from scratch with moderate complexity |
| RUSE | Low | Goal will be purposely build for this system, no plan for reuse at the moment |
| TIME | High | Goal operations take significant execution time in client’s workflow |
| STOR | High | Goal utilizes significant amount of storage |
| PVOL | Low | The cloud service which the web app utilizes is expected to be stable for majority of the time |
| ACAP | Nominal | The analyst personnel has nominal experience, both academically and industrially, for the task of requirement implementation and design implementation |
| PCAP | High | The developers have related academic and work experience and are determined to be highly capable |
| PCON | Very High | The staff turnover rate of the dev team is low within this semester |
| APEX | High | Developers have moderate to rich experience with web application development |
| LTEX | High | Developers are comfortable with the technologies chosen for this project |
| PLEX | Nominal | Developers have moderate experience with the cloud service for hosting the web app and database |
| TOOL | Nominal | Moderate integration may be necessary for the chosen technologies |
| SITE | Extra High | The team has very responsive and interactive communication/workflow between on-campus and remote members |

Table 11: COCOMOII Cost Driver - Goal Module

|  |  |  |
| --- | --- | --- |
| **Cost Driver** | **Value** | **Rationale** |
| RELY | High | Team/User management is required for the system to be successful |
| DATA | Nominal | The team/user data size is moderate considering the current team size and potential growth in team size in the future |
| DOCU | Nominal | Documentation will be required for user guidance and maintenance purpose |
| CPLX | Nominal | The subsystem needs to be built from scratch |
| RUSE | Low | Team/user management will be purposely build for this system, no plan for reuse at the moment |
| TIME | High | Team/user management operations take significant execution time in client’s workflow |
| STOR | High | Team/user data utilizes significant amount of storage |
| PVOL | Low | The cloud service which the web app utilizes is expected to be stable for majority of the time |
| ACAP | Nominal | The analyst personnel has nominal experience, both academically and industrially, for the task of requirement implementation and design implementation |
| PCAP | High | The developers have related academic and work experience and are determined to be highly capable |
| PCON | Very High | The staff turnover rate of the dev team is low within this semester |
| APEX | High | Developers have moderate to rich experience with web application development |
| LTEX | High | Developers are comfortable with the technologies chosen for this project |
| PLEX | Nominal | Developers have moderate experience with the cloud service for hosting the web app and database |
| TOOL | Nominal | Moderate integration may be necessary for the chosen technologies |
| SITE | Extra High | The team has very responsive and interactive communication/workflow between on-campus and remote members |

Table 12: COCOMOII Cost Driver - Team/User Module

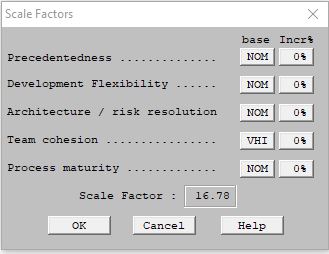


Figure 1: COCOMOII Scale Factor screenshot

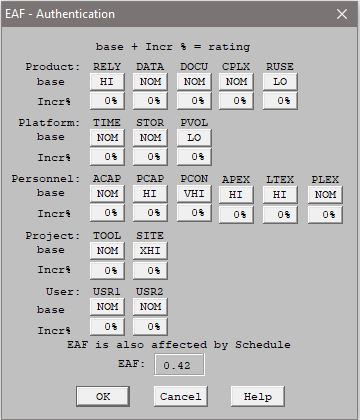


Figure 2: COCOMOII Cost Driver Authentication Module screenshot

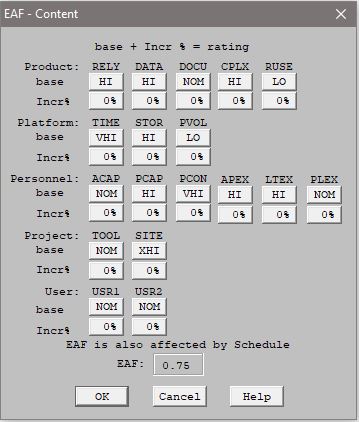


Figure 3: COCOMOII Cost Driver Content Module screenshot

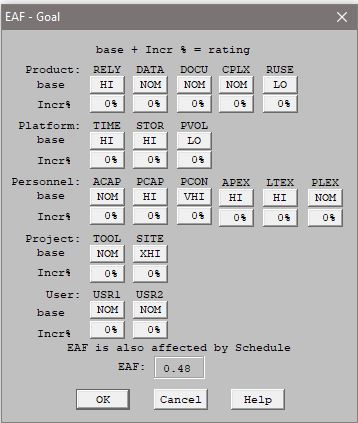


Figure 4: COCOMOII Cost Driver Goal Module screenshot

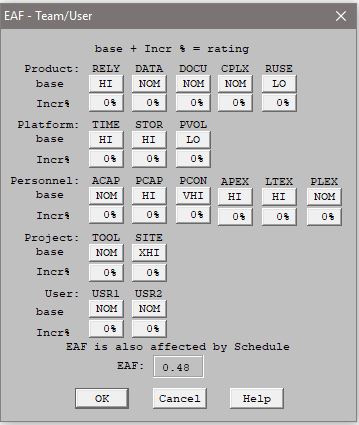


Figure 5: COCOMOII Cost Driver Team/User Module screenshot

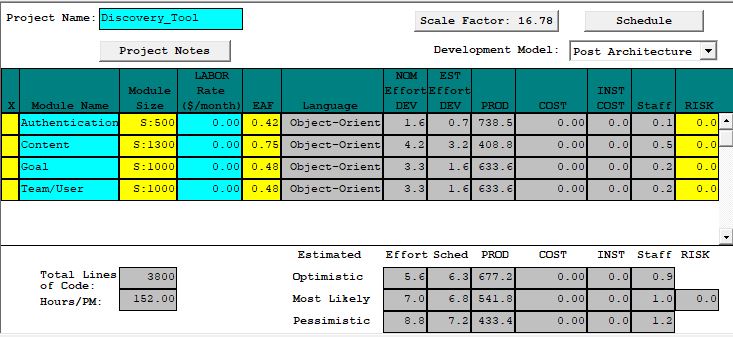


Figure 6: COCOMOII Result screenshot

Result Analysis:

Total estimated line of code: 3800

Estimated Most-likely effort by COCOMII: 7.0 person-month

Effort per person: 18 hours/week

Total time available:

Monthly team efforts: 18 hr/(person-week) \* 8 members \* 4 weeks = 576 hr/month

Time required as per COCOMOII:

Total time: 7.0 person-month \* 152 hr/person-month / 576 hr/month = 1.8 months

Interpretation: According to the above calculation, the project development phase is estimated by COCOMOII to take 1.8 months starting from 10/23/18 to complete, which is inline with our estimation based on the assumption of 8 members working 18 hours per week for 12 weeks from 9/3/18.

# 6. Iteration Plan

## 6.1 Plan

This section aims to accomplish features in the system requirement in two project iterations, and to describe the detailed plan for each project iteration. The team will prioritize implementing the mission critical features. If time allows, the team will also implement lower prioritized features. The first iteration is to prototype sub systems, then complete and integrate the entire backend and frontend system simultaneously. However, this first iteration will only focus on implementing and testing the high priority features.. The second iteration is to implement and test low priority features based on time allowance, then conduct more thorough software regression testing, deployment of the system, gain feedback from the client, implement documentation, and fix any defects.

### 6.1.1 Capabilities to be implemented

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Capability** | **Description** | **Priority** | **Iteration** |
| UC-1 | Register for account | User can create a new account profile | high | 1 |
| UC-2 | Login | User logs into the system with a username and password | high | 1 |
| UC-3 | View current month target and progress | Content writer can view the total number of titles needed for this month. The desired verticals and number of titles needed by each vertical, titles approved so far. Assignee of each vertical. Vertical assigned to this writer. Titles assigned/worked/approved to this writer. | high | 1 |
| UC-4 | View approved titles | A content writer can view all the title created by him/her that were approved for content production | high | 1 |
| UC-5 | View past month targets | Content writer can view targets made in past months | high | 1 |
| UC-6 | View future month targets | Content writer can view targets set for future months | high | 1 |
| UC-7 | Receive notifications | A content writer can receive notifications as a piece of content moves through the various stages of development (approved, rejected, parked, unassigned, assigned and untitled, titled, approved for publishing, parked for publishing) | low | 2 |
| UC-8 | Send message | A content writer can send a message to a specific user | low | 2 |
| UC-9 | Receive message | A content writer can receive message from other users | low | 2 |
| UC-10 | View content success rates by draft to title conversion ratio | A content writer can track his/her own success rate by the draft to approved title conversion ratio | low | 2 |
| UC-11 | Logout | Content writer logs out of the system and ends their current session | high | 1 |
| UC-12 | Create a draft | content writer can create a new draft and add details, notes, and tags | high | 1 |
| UC-13 | Edit a draft | content writer can modify a draft | high | 1 |
| UC-14 | Delete a draft | content writer can remove a draft | high | 1 |
| UC-15 | View a draft | content writer can view a draft’s details: draft ID, creator, type, vertical, description, time created, last time modified, URL, buy-side network, tags | high | 1 |
| UC-16 | Pitch a draft | When a content writer thinks this draft is ready, he/she can click the “pitch it” button to publish this draft to the global pitch list. This draft is then removed from this draft board. | high | 1 |
| UC-17 | View a pitch | Content writer can click on a pitch in the list to view its details: pitch ID, creator, type, vertical, description,time created,last time modified, URL, buy-side network, tags, status(new, approved, rejected, parked), upvote number | high | 1 |
| UC-18 | Upvote a pitch | Content writer can click on the upvote button to upvote a pitch they like | low | 2 |
| UC-19 | Filter pitches by criteria | A content writer can filter pitches by criteria such as status, creator etc | low | 2 |
| UC-20 | Sort pitches by upvote | A content writer can change the view of the board locally by sorting pitches in order of upvotes to identify promising pitches via crowdsourcing | low | 2 |
| UC-21 | Export global pitch board | A content writer can export the master pitch list into a spreadsheet | high | 1 |
| UC-22 | Edit a title | Content writer can make an eye-catching title for this title instance, the status of the title will change from “assigned and untitled” to “titled” | high | 1 |
| UC-23 | View a title | Content writer can click on a title in the list to view its details: title ID, creator, assignee, type, vertical, description, time created, last time modified, URL, buy-side network, tags, status | high | 1 |
| UC-24 | Filter titles by criteria | A content writer can filter titles by criteria such as status, creator etc | low | 2 |
| UC-25 | Export local title board | A content writer can export the local title board list into a spreadsheet | high | 1 |
| UC-26 | View current month target and progress | Admin can view the target number of titles needed for this month and progress(number of approved / still needed). The desired verticals, number of titles needed by each vertical, assignee of each vertical, progress of each vertical (titles approved / needed) | high | 1 |
| UC-27 | View past month targets | Admin can view targets made in past months | high | 1 |
| UC-28 | View future month targets | Admin can view targets set for future months | high | 1 |
| UC-29 | Receive notifications | Admin can receive notifications as a piece of content moves through the various stages of development (approved, rejected, parked, unassigned, assigned and untitled, titled, approved for publishing, parked for publishing) | low | 2 |
| UC-30 | Send message | Admin can send a message to a specific user | low | 2 |
| UC-31 | Receive message | Admin writer can receive message from other users | low | 2 |
| UC-32 | Logout | Admin logs out of the system and ends their current session | high | 1 |
| UC-33 | View a pitch | Admin can click on a pitch in the list to view its details: pitch ID, creator, type, vertical, description,time created,last time modified, URL, buy-side network, tags, status(new, approved, rejected, parked), upvote number | high | 1 |
| UC-34 | Approve/reject a pitch | Admin can click on a pitch, change its status to approved or rejected. If two admin has different opinions, the status of the pitch is changed to “parked” | high | 1 |
| UC-35 | Filter pitches by criteria | Admin can filter pitches by criteria such as status, creator etc | low | 2 |
| UC-36 | Sort pitches by upvote | Admin can change the view of the board locally by sorting pitches in order of upvotes to identify promising pitches via crowdsourcing | low | 2 |
| UC-37 | Export global pitch board | Admin can export the master pitch list into a spreadsheet | high | 1 |
| UC-38 | View a title | Admin can click on a title in the list to view its details: title ID, creator, assignee, type, vertical, description, time created, last time modified, URL, buy-side network, tags, status | high | 1 |
| UC-39 | Assign a title to content writer | Admin can change the assignee of a title to a specific content writer. By default, the assignee is blank and status of the title is “unassigned and untitled”. Once assigned a writer, the status changes to “assigned” | high | 1 |
| UC-40 | Approve a title | One admin needs to approve a title for production. Status changes from “titled” to “approved”. If a title is reviewed but not approved, the status changes to “parked” | high | 1 |
| UC-41 | Filter titles by criteria | Admin can filter titles by criteria such as status, creator etc | low | 2 |
| UC-42 | Export global title list | Admin can export the global title board list into a spreadsheet | high | 1 |
| UC-43 | View a team member details | Admin can click on a team member and view his/her information: member name, account type(writer, admin), number of draft pitched, number of titles assigned, number of titles approved, assigned vertical, number of title needed by that vertical, success rate, contact info | high | 1 |
| UC-44 | Change a team member’s type | Admin can change a team member’s type between admin and writer | high | 1 |
| UC-45 | Set target verticals for current month | Admin can set a list of verticals desired for current month | high | 1 |
| UC-46 | Set target verticals for future months | Admin can set a list of verticals desired for future months | high | 1 |
| UC-47 | Set target number of titles needed for each vertical | Admin can set the number of titles needed for each vertical, then the numbers will be summed up to be the total number of titles needed by that month | high | 1 |
| UC-48 | Assign vertical to content writers | Admin can assign a content writer to a vertical | high | 1 |

Table 13: Construction iteration capabilities to be implemented

### 6.1.2 Capabilities to be tested

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Capability** | **Description** | **Priority** | **Iteration** |
| UC-1 | Register for account | User can create a new account profile | high | 1 |
| UC-2 | Login | User logs into the system with a username and password | high | 1 |
| UC-3 | View current month target and progress | Content writer can view the total number of titles needed for this month. The desired verticals and number of titles needed by each vertical, titles approved so far. Assignee of each vertical. Vertical assigned to this writer. Titles assigned/worked/approved to this writer. | high | 1 |
| UC-4 | View approved titles | A content writer can view all the title created by him/her that were approved for content production | high | 1 |
| UC-5 | View past month targets | Content writer can view targets made in past months | high | 1 |
| UC-6 | View future month targets | Content writer can view targets set for future months | high | 1 |
| UC-7 | Receive notifications | A content writer can receive notifications as a piece of content moves through the various stages of development (approved, rejected, parked, unassigned, assigned and untitled, titled, approved for publishing, parked for publishing) | low | 2 |
| UC-8 | Send message | A content writer can send a message to a specific user | low | 2 |
| UC-9 | Receive message | A content writer can receive message from other users | low | 2 |
| UC-10 | View content success rates by draft to title conversion ratio | A content writer can track his/her own success rate by the draft to approved title conversion ratio | low | 2 |
| UC-11 | Logout | Content writer logs out of the system and ends their current session | high | 1 |
| UC-12 | Create a draft | content writer can create a new draft and add details, notes, and tags | high | 1 |
| UC-13 | Edit a draft | content writer can modify a draft | high | 1 |
| UC-14 | Delete a draft | content writer can remove a draft | high | 1 |
| UC-15 | View a draft | content writer can view a draft’s details: draft ID, creator, type, vertical, description, time created, last time modified, URL, buy-side network, tags | high | 1 |
| UC-16 | Pitch a draft | When a content writer thinks this draft is ready, he/she can click the “pitch it” button to publish this draft to the global pitch list. This draft is then removed from this draft board. | high | 1 |
| UC-17 | View a pitch | Content writer can click on a pitch in the list to view its details: pitch ID, creator, type, vertical, description,time created,last time modified, URL, buy-side network, tags, status(new, approved, rejected, parked), upvote number | high | 1 |
| UC-18 | Upvote a pitch | Content writer can click on the upvote button to upvote a pitch they like | low | 2 |
| UC-19 | Filter pitches by criteria | A content writer can filter pitches by criteria such as status, creator etc | low | 2 |
| UC-20 | Sort pitches by upvote | A content writer can change the view of the board locally by sorting pitches in order of upvotes to identify promising pitches via crowdsourcing | low | 2 |
| UC-21 | Export global pitch board | A content writer can export the master pitch list into a spreadsheet | high | 1 |
| UC-22 | Edit a title | Content writer can make an eye-catching title for this title instance, the status of the title will change from “assigned and untitled” to “titled” | high | 1 |
| UC-23 | View a title | Content writer can click on a title in the list to view its details: title ID, creator, assignee, type, vertical, description, time created, last time modified, URL, buy-side network, tags, status | high | 1 |
| UC-24 | Filter titles by criteria | A content writer can filter titles by criteria such as status, creator etc | low | 2 |
| UC-25 | Export local title board | A content writer can export the local title board list into a spreadsheet | high | 1 |
| UC-26 | View current month target and progress | Admin can view the target number of titles needed for this month and progress(number of approved / still needed). The desired verticals, number of titles needed by each vertical, assignee of each vertical, progress of each vertical (titles approved / needed) | high | 1 |
| UC-27 | View past month targets | Admin can view targets made in past months | high | 1 |
| UC-28 | View future month targets | Admin can view targets set for future months | high | 1 |
| UC-29 | Receive notifications | Admin can receive notifications as a piece of content moves through the various stages of development (approved, rejected, parked, unassigned, assigned and untitled, titled, approved for publishing, parked for publishing) | low | 2 |
| UC-30 | Send message | Admin can send a message to a specific user | low | 2 |
| UC-31 | Receive message | Admin writer can receive message from other users | low | 2 |
| UC-32 | Logout | Admin logs out of the system and ends their current session | high | 1 |
| UC-33 | View a pitch | Admin can click on a pitch in the list to view its details: pitch ID, creator, type, vertical, description,time created,last time modified, URL, buy-side network, tags, status(new, approved, rejected, parked), upvote number | high | 1 |
| UC-34 | Approve/reject a pitch | Admin can click on a pitch, change its status to approved or rejected. If two admin has different opinions, the status of the pitch is changed to “parked” | high | 1 |
| UC-35 | Filter pitches by criteria | Admin can filter pitches by criteria such as status, creator etc | low | 2 |
| UC-36 | Sort pitches by upvote | Admin can change the view of the board locally by sorting pitches in order of upvotes to identify promising pitches via crowdsourcing | low | 2 |
| UC-37 | Export global pitch board | Admin can export the master pitch list into a spreadsheet | high | 1 |
| UC-38 | View a title | Admin can click on a title in the list to view its details: title ID, creator, assignee, type, vertical, description, time created, last time modified, URL, buy-side network, tags, status | high | 1 |
| UC-39 | Assign a title to content writer | Admin can change the assignee of a title to a specific content writer. By default, the assignee is blank and status of the title is “unassigned and untitled”. Once assigned a writer, the status changes to “assigned” | high | 1 |
| UC-40 | Approve a title | One admin needs to approve a title for production. Status changes from “titled” to “approved”. If a title is reviewed but not approved, the status changes to “parked” | high | 1 |
| UC-41 | Filter titles by criteria | Admin can filter titles by criteria such as status, creator etc | low | 2 |
| UC-42 | Export global title list | Admin can export the global title board list into a spreadsheet | high | 1 |
| UC-43 | View a team member details | Admin can click on a team member and view his/her information: member name, account type(writer, admin), number of draft pitched, number of titles assigned, number of titles approved, assigned vertical, number of title needed by that vertical, success rate, contact info | high | 1 |
| UC-44 | Change a team member’s type | Admin can change a team member’s type between admin and writer | high | 1 |
| UC-45 | Set target verticals for current month | Admin can set a list of verticals desired for current month | high | 1 |
| UC-46 | Set target verticals for future months | Admin can set a list of verticals desired for future months | high | 1 |
| UC-47 | Set target number of titles needed for each vertical | Admin can set the number of titles needed for each vertical, then the numbers will be summed up to be the total number of titles needed by that month | high | 1 |
| UC-48 | Assign vertical to content writers | Admin can assign a content writer to a vertical | high | 1 |

Table 14: Construction iteration capabilities to be tested

### 6.1.3 Capabilities not to be tested

Every capability listed above will be fully tested by the end of the second iteration.

### 6.1.4 CCD Preparation Plans

The clients and users that will involve in the CCD are:

* Linda Suen: client side representative
* Content writer: client side user
* Content admin: client side user
* Maintainer: client side developer to maintain the system after handoff

The action items for the CCD preparation are:

* Develop all the core capabilities
* Ensure regression test is completed and all defects are fixed
* Draft user manual
* Develop use cases and scenarios to provide user context-setting
* Prepare stakeholder interaction via dev team dry runs
* Develop feedback survey for stakeholders who will be attending CCD
* Develop CCD risk management plan

## 6.2 Iteration Assessment << TBD after CCD >>

### 6.2.1 Capabilities Implemented, Tested, and Results

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

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

Table 15: Capabilities implemented, tested, and results

### 6.2.2 Core Capabilities Drive-Through Results

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

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

Note: Make sure to be specific to the capabilities shown/demonstrated/driven-through.

Simply stating that the clients liked the capabilities is not sufficient. >>

## 6.3 Adherence to Plan << TBD after CCD >>

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