Life Cycle Plan (LCP)

Spherical Modeling Tool

Team 13

Oziel De Oliveira Carneiro, Feasibility Analyst
Lyle Franklin, Project Manager
Sait Ilhaner, IIV & V
Minsuk Heo, Prototyper
Mehmet Sezer, Requirements Engineer
Nikita Vlasenko, OCD
Guoxiong Xie, Life Cycle Planner

12/04/2013

Version History

Date	Author	Version	Changes made	Rationale
09/27/13	Mehmet Sezer	1.0	• Added personnel skills	• To fit CS577 course content
10/12/13	Guoxiong Xie	1.1	• Finished Sections 1 to 5	• To produce an LCP for Draft FC Package
10/21/13	Guoxiong Xie	2.0	• Updated Sections 1 to 5 based on feedback from FCR ARB session	• To produce an LCP for Final FC Package
11/26/13	Guoxiong Xie	2.1	• Updated Sections 1 to 5 for Development Phase	• To produce an LCP for Draft DC Package
11/30/13	Guoxiong Xie	2.2	• Updated Sections 1 to 5 based on feedback from 11/27/13 meeting	• To produce an LCP for Draft DC Package
12/01/13	Guoxiong Xie	3.0	• Added Section 6.1 for iteration planning	• To produce an LCP for Final DC Package
12/03/13	Guoxiong Xie	3.1	• Updated Section 6.1 for iteration planning	• To produce an updated version of LCP for Final DC Package

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the Life Cycle Plan

This Life Cycle Plan (LCP) is organized to answer the most common questions about a project or activity: why?, whereas?, what?, when?, who?, where?, how?, and how much? Specifically, this LCP addresses the objectives to be achieved within the project time frame, the assumptions necessary to meet plans, milestones (dates) and products to be delivered, the responsibilities of each team member, the approach to achieve the goals, and the allocation of resources.

1.2 Status of the Life Cycle Plan

The status of the LCP is currently at the Development Commitment Package version number 3.1. The major changes from Foundations Phase Version are:

- Sections 1 through 5 are finished (remove all TBDs)
- Skills/required skills of each team member are further clarified
- Milestones and products are added
- Responsibilities for each team member are added based on their roles and skills
- Skills sets for new team members are further specified
- Approaches and techniques are identified and stated
- Required resources are addressed
- Estimations are made to elaborate the allocation of resources, such as time and efforts for different phases.
- Re-estimate code size in COCOMO Resource Estimation Section

1.3 Assumptions

1) Requirements stability
2) Architected Agile development pattern is used for the project Spherical Modeling Tool
3) The duration of the project is two semesters: 12 weeks in Fall 2013 and 12 weeks in Spring 2014.
4) Seven members in the development team: six on-campus and one DEN students. There will be new members joining the team in Spring 2014.
5) Weekly team meeting with the client.

2. Milestones and Products

2.1 Overall Strategy

The Spherical Modeling Tool project is following Architected Agile Process Pattern as the ICSM process model. The rationale behind this decision is that compared to Use Single NDI Process, Intensive Process Pattern and Net-Centric Services Intensive Process Patterns, Architected Agile process has the least amount of non-conforming points. For most of the core capabilities of the project, we will build them from scratch. However, some Non-Development Items are used in the project, such as using Highcharts for the generation of the 2D chart.

Exploration Phase

Duration: 09/11/2013 – 09/27/2013
Concept: During Exploration Phase, the team defines the initial scope of the project, identifies success-critical stakeholders, project operational concepts, project requirements and designs the initial software architecture. The team also identifies skills/responsibilities for each team members.
Deliverables: Valuation Commitment Package, Client Interaction Report, Life Cycle Plan, Progress Report, and Project Plan
Milestone: Valuation Commitment Review
Strategy: One Incremental Commitment Cycle

Valuation Phase

Duration: 09/28/2013 – 10/18/2013
Concept: During Valuation Phase, the team sets up win-win negotiation sessions with the client to gather further requirements, evaluates risks, identifies win conditions, and prioritizes win conditions. The team also develops an initial prototype that shows the core capabilities of the system.
Deliverables: Draft Foundations Commitment Package, initial prototype, Progress Reports, System and Software Architecture Description, and Project Plan
Milestone: Architecture Review Board and Foundation Commitment Review.
Strategy: Incremental Commitment Cycles (Architected Agile Process Pattern), Win-win Negotiation, Core-Capability Prototyping

Foundation Phase

Duration: 10/19/2013 – 02/10/2014 (Winter break in between)
Concept: During Foundation Phase, the team will assess project status, including feasibility evidence, life cycle content, operational concept, prototype and components, system architecture. The team will also analyze and prioritize capabilities to prototype, develop the prototype and fix defects. Furthermore, the team will define technology-dependent and –independent architecture and specify architecture styles, patterns and frameworks.

Deliverables: Foundation Commitment Package, Draft Development Commitment Package, Development Commitment Package, Effort Reports, Progress Report, Project Plan, Feasibility Evidence Description, Life Cycle Plan, Operational Concept Description, Project Deliverable (prototype, coding), Release Description, Software Components, System and Software Architecture Description, System and Software Requirements Definition, Test Procedures and Results, and UML Model.
Milestone: Architecture Review Board and Development Commitment Review
Strategy: Incremental Commitment Cycles (Architected Agile Process Pattern), Prototyping, Client and Team Feedbacks and Risk Resolving.

Development Phase — Construction Iteration

Duration: 01/13/2014 – 05/02/2014
Concept: During Development Phase—construction iteration, the team will carry out several construction iterations, in which the team will perform Core Capabilities Drive-Through, assess development iteration, implement the system and perform testing.
Deliverables: Operation Commitment Package, Effort Reports, Progress Reports, Project Plan, Functioning Software System, Life Cycle Plan, NDI Component, Project Deliverable, Release Description, Software Components, Test Plan and Cases, Test Procedures and Results
Milestone: Core Capability Drive-through and Operation Commitment Review
Strategy: Incremental Commitment Cycles (Architected Agile Process Pattern), Implementation Iterations, Client and Team Feedbacks and Bug Resolving.

Development Phase — Transition Iteration

Duration: 05/03/2014 – 05/14/2014
Concept: During Development Phase—transition iteration, the team will carry out a transition iteration in which the team members provide training (develop user manual and training plan) and transition the system to client.
Deliverables: Functioning Software System, Life Cycle Plan, NDI Component, Progress Report, Project Deliverable, Project Plan, Release Description, Software Components, Training Plan, User Manual and WinWin Agreements.
Milestone: Project Transition Readiness ARB Reviews
Strategy: Incremental Commitment Cycles, Implementation Iterations (Architected-Agile), Team and Client Feedbacks and Bug Resolving.

2.2 Project Deliverables

2.2.1 Exploration Phase

Artifacts output during Exploration Phase are listed in Table 1.

Table 1: Artifacts Deliverables in Exploration Phase

Artifact	Due date	Format	Medium
Client Interaction Report	9/20/2013	.doc, .pdf	Soft copy
Valuation Commitment Package • Operational Concept Description (OCD) Early Section • Life Cycle Plan (LCP) Early Section • Feasibility Evidence Description (FED) Early Section	09/27/2013	.doc, .pdf	Soft copy
Effort Report	Every Monday	Online form	ER system
Project Plan	Every Other Monday	.mpp	Soft copy
Progress Report	Every Other Monday	.xls	Soft copy

2.2.2 Valuation Phase

Artifacts output during Valuation Phase are listed in Table 2.

Table 2: Artifact deliverable in Valuation Phase

Artifact	Due date	Format	Medium
Initial Prototype	10/04/2013	Code, Software	Web Hosting Service
Draft Foundation Commitment Package	10/16/2013	.doc, .pdf	Soft copy
Effort Report	Every Monday	Online form	ER System
Progress Report	Every Other Monday	.xls	Soft Copy
Project Plan	Every Other Monday	.mpp	Soft copy

2.2.3 Foundations Phase

Artifacts output during Foundations Phase are listed in Table 3.

Table 3: Deliverables in Foundations Phase

Artifact	Due date	Format	Medium
Foundations Commitment Package	11/21/2013	.doc, .pdf	Soft copy
System and Software Architecture Description	11/22/2013	.doc, .pdf	Soft copy
UML Model	11/22/2013	.pdf	Soft copy
Draft Development Commitment Package	12/02/2013	.doc, .pdf	Soft copy
Project Deliverable (prototype, coding)	12/02/2013	Code, Software	Web Hosting Service
Software Components	12/02/2013	Code, Software	N/A
Test Procedures and Results	12/02/2013	Code (automated testing); .doc and .pdf	Soft copy
Development Commitment Package	12/09/2013	.doc, .pdf	Soft copy
Effort Report	Every Monday	Online Form	ER System
Progress Report	Every Other Monday	.xls	Soft Copy
Project Plan	Every Other Monday	.mpp	Soft copy

2.2.4 Development Phase—Construction Iteration

Artifacts output during Development Phase—Construction Iteration are listed in Table 4.

Table 4: Artifact deliverable in Development Phase—Construction Iteration

Artifact	Due date	Format	Medium
Draft Rebaselined Development Commitment Package	02/05/2014	.doc, .pdf	Soft copy
Rebaselined Development Commitment Package	02/10/2014	.doc, .pdf	Soft copy
IOC #1	04/01/2014	.doc, .pdf	Soft copy
CCD Report	04/10/2014	.doc, .pdf	Soft copy
Draft TRR Package	04/15/2014	.doc, .pdf	Soft copy
TS Set	04/22/2014	.doc, .pdf	Soft copy
Functioning Software System	04/26/2014	Code, Software	Web Hosting Service
Software Components	04/26/2014	Code, Software	N/A
Test Procedures and Results	04/26/2014	Code (automated testing); .doc and .pdf	Soft copy
IOC #n	05/03/2014	.doc, .pdf	Soft copy
Effort Report	Every Monday	Online form	ER System
Progress Report	Every Other Monday	.xls	Soft Copy
Project Plan	Every Other Monday	.mpp	Soft copy

2.2.5 Development Phase—Transition Iteration

Artifacts output during Development Phase—Transition Iteration are listed in Table 5.

Artifact deliverable in Development Phase—Transition Iteration

Artifact	Due date	Format	Medium
Training Plan	05/08/2014	.doc, .pdf	Soft copy
User Manual	05/08/2014	.doc, .pdf	Soft copy
Close Out Report	05/10/2014	.doc, .pdf	Soft copy
Project Archive	05/10/2014	.doc, .pdf	Soft copy
Effort Report	Every Monday	Online form	ER System
Progress Report	Every Other Monday	.xls	Soft Copy
Project Plan	Every Other Monday	.mpp	Soft copy

3. Responsibilities

3.1 Responsibilities by Phase

Table 6: Stakeholder's responsibilities

Name: Robert Lindstrom	
Role: Client, Trainer (our client claimed that he will take the role as trainer in and after Operation Phase to give tutorial to users and business partners.)	
Exploration	N/A
Valuation	1. Identify objectives, constraints and priorities
Foundations	1. Assess prototype and components
Rebaselined Foundations	Work with teams: 1. Rebaseline prototype, prioritize requirements 2. Plan for CS 577b specifics, including transition strategy, key risk items 3. Participate in RDCR ARB review
Development- Construction Iteration	Scheduled Weekly Meetings with Teams to: 1. Discuss status and plans 2. Provide access to key transition people for strategy and readiness discussions 3. Perform core capability drive-through 4. Participate in Project Transition Readiness ARB Reviews
Development- Transition Iteration	Installation and Transition 1. Install product 2. Execute transition plan 3. Participate in Operational Commitment Review for initial operational capability
Operational	1. Provide training to users

Name: Lyle Franklin	
Role: Project Manager (Primary, 577A&B), Architect (secondary, 577A), Builder (secondary, 577B), Tester (secondary, 577B)	
Exploration (PM & Architect)	1. Come up with detailed project plan 2. Record project individual effort 3. Record project progress
Valuation (PM & Architect)	1. Come up with detailed project plan 2. Record project individual effort 3. Record project progress 4. Create and follow up action items
Foundations (PM &	1. Record project progress 2. Come up with detail project plan

Architect)	3. Access system architecture (secondary) 4. Define technology-independent architecture (secondary) 5. Specify architecture styles, patterns and frameworks (secondary)
Rebaselined Foundations (PM & Builder)	Plan and manage project: 1. Detailed project plan 2. Record project progress Rebaseline project status 1. Construct traceability matrix 2. Assess prototype and Components 3. Assess System Architecture Plan for testing: 1. Identify test plan 2. Identify test cases Start developing the system if time permitted Attend Rebaselined DCR ARB
Development- Construction Iteration **(PM & Builder & Tester)**	Construction Iteration 1 — Core capability: 1. Come up with detailed project plan 2. Record project progress 3. Implement the system: a) Develop components b) Develop glue code c) Tailor components d) Integrate component 4. Perform testing 5. Perform core capabilities drive-through Construction Iteration 2 — Full capability: Ditto except "5. Perform core capabilities drive-through"
Development- Transition Iteration **(PM & Builder & Tester)**	1. Transition the system
Operational (PM & Builder & Tester)	1. Support client, user and maintainer 2. Problem solving

Name: Oziel De Oliveira Carneiro (will not continue to 577B)	
Role: Feasibility Analyst (primary), Prototyper (secondary)	
Exploration	1. Explore alternatives 2. Access and plan to mitigate risks
Valuation	1. Comes up with 2D chart and 3D sphere rendering algorithms 2. Explore alternatives 3. Analyze business case

	4. Assess and evaluate NDI and NCS components candadates 5. Assess and plans to mitigate risks
Foundations	1. Assess feasibility evidence 2. Analyze and prioritize capabilities to prototype (secondary) 3. Develop prototype (secondary) 4. Access prototype and components (secondary) 5. Fix defects (secondary)

Name: New Team Member #1 (substitution for Oziel De Oliveira Carneiro)	
Role: Builder (primary), Tester (secondary), Feasibility Analyst (secondary)	
Rebaselined Foundations	Rebaseline project status: 1. Assess feasibility evidence 2. Construct traceability matrix 3. Assess prototype and Components Plan for testing: 1. Identify test plan 2. Identify test cases Start developing the system if time permitted Attend Rebaselined DCR ARB
Development- Construction Iteration	Construction Iteration 1 — Core capability: 1. Implement the system: e) Develop components f) Develop glue code g) Tailor components h) Integrate component 2. Perform testing 3. Perform core capabilities drive-through Construction Iteration 2 — Full capability: Ditto except "3. Perform core capabilities drive-through"
Development- Transition Iteration	1. Transition the system
Operational	1. Support client, user and maintainer 2. Problem solving

Name: Sait IIhaner	
Role: IIV & V (primary), Tester (primary), Quality Focal Point (secondary)	
Exploration	N/A
Valuation	1. Verify and validate work products using issue (Defect) tracking system 2. Identify quality management strategy (secondary) 3. Identify configuration management strategy 4. Construct traceability matrix (secondary)
Foundations	1. Construct traceability matrix

Rebaselined Foundations	Rebaseline project status: 1. Construct traceability matrix 2. Assess prototype and Components Plan for testing: 1. Identify test plan 2. Identify test cases Start testing the system if time permitted Attend Rebaselined DCR ARB
Development- Construction Iteration	Construction Iteration 1 — Core capability: 1. Perform testing 2. Perform core capabilities drive-through Construction Iteration 2 — Full capability: Ditto except "2. Perform core capabilities drive-through"
Development- Transition Iteration	1. Transition the system
Operational	1. Support client, user and maintainer 2. Problem solving

Name: Minsuk Heo (will not continue to 577B)	
Role: Prototyper (primary), Tester (secondary)	
Exploration	N/A
Valuation	1. Analyze and prioritize capabilities to prototype 2. Prepare development/product environment 3. Develop prototype
Foundations	1. Analyze and prioritize capabilities to prototype 2. Develop prototype 3. Access prototype and components 4. Fix defects

Name: New Team Member #2 (substitution for Minsuk Heo)	
Role: Builder (primary), Tester (secondary)	
Rebaselined Foundations	Rebaseline project status: 1. Construct traceability matrix 2. Assess prototype and Components Plan for testing: 1. Identify test plan 2. Identify test cases Start developing the system if time permitted Attend Rebaselined DCR ARB
Development- Construction	Construction Iteration 1 — Core capability: 1. Implement the system:

Iteration	a) Develop components b) Develop glue code c) Tailor components d) Integrate component 2. Perform testing 3. Perform core capabilities drive-through Construction Iteration 2 — Full capability: Ditto except "3. Perform core capabilities drive-through"
Development- Transition Iteration	1. Transition the system
Operational	1. Support client, user and maintainer 2. Problem solving

Name: Nikita Vlasenko (will not continue to 577B)	
Role: Operational Concept Engineer (primary), Tester (secondary)	
Exploration	1. Analyze current system
Valuation	1. Identify objectives, constraints and priorities 2. Analyze current system 3. Identify shared vision 4. Establish new operational concept 5. Identify organizational and operational transformation
Foundations	1. Assess operational concept

Name: New Team Member #3 (substitution for Nikita Vlasenko)	
Role: Builder (primary), Tester (secondary)	
Rebaselined Foundations	Rebaseline project status: 3. Construct traceability matrix 4. Assess prototype and Components Plan for testing: 3. Identify test plan 4. Identify test cases Start developing the system if time permitted Attend Rebaselined DCR ARB
Development- Construction Iteration	Construction Iteration 1 — Core capability: 4. Implement the system: e) Develop components f) Develop glue code g) Tailor components h) Integrate component 5. Perform testing 6. Perform core capabilities drive-through

	Construction Iteration 2 — Full capability: Ditto except "3. Perform core capabilities drive-through"
Development- Transition Iteration	1. Transition the system
Operational	1. Support client, user and maintainer 2. Problem solving

Name: Mehmet Sezer	
Role: Builder (primary: 577B), Requirements Engineer (primary: 577A), Life Cycle Planner (secondary: 577A & B), Trainer (secondary: 577B)	
Exploration (Requirements Engineer & Life Cycle Planner)	1. Identify responsibilities and skills (secondary)
Valuation (Requirements Engineer & Life Cycle Planner)	1. Capture and score MMF and win-conditions 2. Capture progress of win-win negotiation 3. Identify Responsibilities and skills (secondary)
Foundations (Requirements Engineer & Life Cycle Planner)	1. Assess life cycle content (secondary)
Rebaselined Foundations (Builder & Life Cycle Planner & Trainer)	Rebaseline project status: 1. Assess life cycle content 2. Construct traceability matrix 3. Assess prototype and Components Prepare for Development Phase: 1. Develop transition plan 2. Identify development iteration Start developing the system if time permitted Attend Rebaselined DCR ARB
Development- Construction Iteration **(Builder & Life Cycle Planner & Trainer)**	Construction Iteration 1 — Core capability: 1. Come up with a detail project plan 2. Assess development iteration 3. Implement the system: i) Develop components j) Develop glue code k) Tailor components l) Integrate component 4. Perform core capabilities drive-through Construction Iteration 2 — Full capability: Ditto except "4. Perform core capabilities drive-through"

Development- Transition Iteration **(Builder & Life Cycle Planner & Trainer)**	1. Transition the system
Operational (Builder & Life Cycle Planner & Trainer)	1. Support client, user and maintainer 2. Problem solving 3. Provide training

Name: Guoxiong Xie	
Role: Life Cycle Planner (primary: 577A & B), Builder (secondary: 577B), Tester (secondary: 577B), Prototyper (secondary: 577A)	
Exploration (Life cycle planner & prototype)	1. Identify responsibilities and skills
Valuation (Life cycle planner & prototype)	1. Identify Responsibilities and skills 2. Analyze and prioritize capabilities to prototype (secondary) 3. Prepare development/product environment (secondary) 4. Develop prototype (secondary)
Foundations (Life cycle planner & prototype)	1. Assess life cycle content 2. Analyze and prioritize capabilities to prototype (secondary) 3. Develop prototype (secondary) 4. Access prototype and components (secondary) 5. Fix defects (secondary)
Rebaselined Foundations (Builder & Life Cycle Planner & Tester)	Rebaseline project status: 1. Assess life cycle content 2. Construct traceability matrix 3. Assess prototype and Components Prepare for Development Phase: 1. Develop transition plan 2. Identify development iteration Plan for testing: 1. Identify test plan 2. Identify test cases Start developing the system if time permitted Attend Rebaselined DCR ARB
Development- Construction Iteration **(Builder & Life Cycle Planner & Tester)**	Construction Iteration 1 — Core capability: 1. Come up with a detail project plan 2. Assess development iteration 3. Implement the system: m) Develop components n) Develop glue code

	o) Tailor components p) Integrate component 4. Perform testing 5. Perform core capabilities drive-through Construction Iteration 2 — Full capability: Ditto except "5. Perform core capabilities drive-through"
Development- Transition Iteration **(Builder & Life Cycle Planner & Tester)**	1. Transition the system
Operational (Builder & Life Cycle Planner & Tester)	1. Support client, user and maintainer 2. Problem solving

3.2 Skills

Table 7: Roles and Skills of Team Members

Team Members	Role	Skills
Lyle Franklin (Will continue to CSCI577B)	Project Manager (Primary, 577A&B), Architect (secondary, 577A), Builder (secondary, 577B), Tester (secondary, 577B)	**Current Skills:** - Languages: Java, C#, JavaScript (incl. Backbone JS), HTML, Responsive CSS, SQL. - Familiarity with agile methods, including SCRUM and TDD. - Familiarity with software architecture principles, including UML, architectural styles, and design patterns. - Strong communication skills. - Ability to Create and follow up action items. **Required Skills:** - Understanding of software architectural patterns - Strong communication skills. - Team leader skills.
Minsuk Heo (Will not continue to CSCI577B, see Table 8 for required skills for new team member in CS 577B)	Prototyper (primary)	**Current Skills:** - Languages: Java, C, JavaScript (incl. Backbone JS), HTML5, Responsive CSS, MongoDB. - Familiarity with agile methods, including SCRUM, plan driven project. **Required Skills:** - Knowledge of User Interface design - Familiarity with software architecture principles, including UML, architectural styles, and design patterns. - Communication skills. - Knowledge of Jasmine for Javascript testing.

Team Members	Role	Skills
Oziel De Oliveira Carneiro (Will not continue to CSCI577B, see Table 8 for required skills for new team member in CS 577B)	Feasibility Analyst (primary) Prototyper (secondary)	**Current Skills:** - Ability to acquire NDI or NCS components and to analyze their interoperability for NDI/NCS project. - Ability to analyze the Business Case. - Ability to assess and evaluate NDI and NCS components. - Ability to assess and plan to mitigate risks. - Ability to assess feasibility evidence. - Ability to explore alternatives. - Languages: Java and C++, algorithm design, artificial intelligence. **Required Skills:** - Familiarity of various NDI/NCS and pros and cons of them. - Ability to analyze the Business Case. - Ability to assess feasibility evidence. - Familiarity with User Interface Design
Sait Ilhaner (Will continue to CSCI577B)	IIV & V (primary), Tester (primary), Quality Focal Point (secondary)	**Current Skills:** - Ability to verify and validate work products. - Software Engineering. - Web Development, Testing, Automation, Scripting. **Required Skills**: - Knowledge of Jasmine for Javascript testing.

Team Members	Role	Skills
Guoxiong Xie (Will continue to CSCI577B)	Life Cycle Planner (primary: 577A & B), Builder (secondary: 577B), Tester (secondary: 577B), Prototyper (secondary: 577A)	**Current Skills:** - Previous experience as a project manager. - Technical writing skills, oral communication skills. - Ability to assess development iteration, access life cycle content, identify development iteration. - Knowledge of Git version control and relational database. - Heroku deployment skills **Required Skills:** - COINCOMO Cost Estimation skills. - Ability to know life cycle management approach. - Ability to Provide Process Feasibility Evidence. - Ability to Identify Responsibilities and Skills. - Identify Milestones and Products. - Estimate Project Effort and Schedule using COTIPMO for NDI/NCS Project. - Ability to Develop Transition Plan and Support Plan, and Detail Project Plan.
Nikita Vlasenko (Will not continue to CSCI577B)	Operational Concept Engineer (primary)	**Current Skills:** - Ability to manage documentation. - Ability to analyze the current system. - Ability to establish/assess the operational concept. - Ability to Explore Alternatives. **Required Skills:** - Ability to Identify Objectives, Constraints and Priorities. - Ability to Identify Organizational and Operational Transformation. - Ability to Identify Shared Vision. - Ability to verify and validate

Team Members	Role	Skills
		work products. - Knowledge of Jasmine for Javascript testing.

Team Members	Role	Skills
Mehmet Sezer (Will continue to CSCI577B)	Builder (primary: 577B), Requirements Engineer (primary: 577A), Life Cycle Planner (secondary: 577A & B), Trainer (secondary: 577B)	**Current Skills:** - Ability to analyze, discover, review, articulate, understand, and document the requirements of the proposed system. - Ability to establish a new operational concept. - Ability of verbal and graphic statement of an organization's assumptions/intent in regard to a system or a related set of systems. - Ability to describe of an imagined sequence of events that includes the interaction of the product or service with its environment and users. **Required Skills**: - Ability to ensure that the requirements are identified, documented, maintained, communicated and traced throughout the life cycle of a system, product, or service - Technical writing skills

Table 8: Roles and Desired Skills of New Team Members in CSCI 577B

Team Members	Role	Skills
New Team Member #1	Builder (primary), Tester (secondary), Feasibility Analyst (secondary)	**Required Skills:** - Ability to analyze the Business Case. - Ability to assess feasibility evidence. - Ability to analyze the Business Case. - Languages: JavaScript (incl. Backbone JS), HTML5, Responsive CSS, MongoDB. - Knowledge of Jasmine for Javascript testing. - Familiarity with agile development process (SCRUM)
New Team Member #2	Builder (primary) Tester (secondary)	Required Skills: - Languages: JavaScript (incl. Backbone JS), HTML5, Responsive CSS, MongoDB. - Knowledge of Jasmine for Javascript testing.
New Team Member #3	Builder (primary) Tester (secondary)	Required Skills: - Languages: JavaScript (incl. Backbone JS), HTML5, Responsive CSS, MongoDB. - Knowledge of Jasmine for Javascript testing.

4. Approach

4.1 Monitoring and Control

For the project Spherical Modeling Tool, the approach used in monitoring and controlling the project include:

1. Progress Reports
2. Project Plan
3. Effort Reports
4. Commitment Reviews at Milestones
5. WinWin Negotiations
6. Bugzilla Defect Tracking and Resolving
7. Meetings with Clients

4.1.1 Closed Loop Feedback Control

The team's main approach to get and provide feedback internally within the team is utilizing Bugzilla Defect Tracking System. When a team member is assigned a task, a bug item is created, and that particular member is responsible for finishing the task and report to the others via Bugzilla regularly. Then other team members are encouraged to double check the finished or the in-progress tasks and provide feedbacks as comments to that bug item.

Winbook is another way to get feedback internally within the team. Team members are encouraged to discuss the winwin conditions and the feature prioritization. Winwin conditions are in the form of user stories, serving as a reference to check if the team is developing the right thing.

4.1.2 Reviews

The team has regular meeting to discuss the artifacts and the process and provide feedbacks to each other's work. In addition to team meetings, IIV & V and Quality Focal Point constantly evaluate the deliverables and the process to keep up the quality of the artifacts and assess how well the team is performing.

At Milestones, commitment reviews are conducted to review the team's work and plan for future phases as well as gather feedbacks from the teaching staff and the client.

4.2 Methods, Tools and Facilities

Table 9: Tools Used in Project

Tools	Usage	Provider
Highcharts	Renders 2D chart based on the data user provides	Open source
Three.js	Renders 3D graph	Open source
MongoDB	Stores user data and model information	Open Source
Node.js	Creates server-side REST API to facilitate the communication between modules	Open Source
Backbone.js	Sets up client-side MV framework	Open Source
Bootstrap	Provides mobile first frontend framework	Open Source
jQuery	Facilitates DOM manipulation	Open Source
Heroku	Hosts web application and database services	Open Source
MS project	Used for creating project plan	Microsoft
Microsoft word	Used for composing LCP, OCD, FED, etc.	Microsoft

5. Resources

In this section, we present the project effort and schedule estimation of the project using COCOMO II.

The following conditions were used to estimate the cost of our system, the Spherical Modeling Tool.

1. Estimated CSCI577a Effort: 7 team members at 14 hrs/week for 12 weeks
2. Estimated CSCI577b Effort: 7 team members at 14 hrs/week for 12 weeks
3. Total estimated effort is 2352 Person-Hours (7*14*24 = 2352).
4. This project has no budget for our development efforts, and the team uses only Open Source Software.
5. The duration of the project is 24 weeks, which are 12 weeks in CSCI577a and 12 weeks in CSCI577b.
6. There are five modules in this system.
    a. 2D Visualization module
    b. 3D Visualization module
    c. Authentication module
    d. Data Entry module
    e. Sphere Editing module
7. All modules are developed with Java technology (mostly JavaScript).
8. Highcharts and Three.js for 2D and 3D rendering modules, respectively.

Table 10 below is module listed in the system and its estimated size with Source Lines of Code (SLOC)

Table 10: Module lists and SLOC of each module

No.	Module Name	Brief Description	SLOC	REVL
1	2D Visualization	Renders 2D chart based on the data input by users	200	10%
2	3D Visualizaion	Renders 3D sphere based on the data input by users	1000	10%
3	Authentication	User authentication and authorization mechanism	400	5%
4	Data Entry	Collects user input for questions and ratings and forwards data to database	300	5%
5	Sphere Editing	Handles the editing of the 2D chart (e.g. dragging the points to modify user input)	1000	10%

Table 11 is COCOMOII Scale Drivers and rationales of choosing the values.

Table 11: COCOMOII Scale Driver

Scale Driver	Value	Rationale
PREC	LOW	The project requires graphical manipulation on a web platform as the core capabilities. None of the team members have experience in 2D and 3D graphic rendering.
FLEX	HI	The client does not mandate any platform or development environment so the flexibility is high.
RESL	HI	Risks are frequently identified before moving on to the next phase and mitigation plans are come up promptly. So, the risk resolution is high.
TEAM	VHI	Team members are willing and frequently commit to the project and help each other.
PMAT	HI	Team members adhere to ICSM guidelines on ICSM-EPG and rigidly practice the development process. So, PMAT is high.

Tables 12 through 16 are COCOMOII Cost Drivers of each module and rationales of choosing the values.

Table 12: COCOMOII Cost Drivers of Module 1 – 2D Visualization module

Cost Driver	Value	Rationale
RELY	NOMINAL	Although this module supports one of the core capabilities, the effect of the software failure is moderate and losses are easily recoverable.
DATA	LOW	Since the database will store only information of the user and his/her sphere modeling info, which are questions and ratings, the ratio of bytes in the testing database to SLOC in the program are approximately less than 10.
DOCU	NOMINAL	Because the development process follows ICSM, the document for life-cycle needs is normal.
CPLX	NOMINAL	It contains simple data passing control, standard math and statistical routines for generating 2D chart using the data input by the user. The NDI Highcharts easily handles the graphical rendering.
RUSE	LOW	It is not intended to be reused for the future project.
TIME	NOMINAL	There is no strict constrains for execution time.

STOR	NOMINAL	The percentage of available storage expected to be used by the system and subsystem is less than 50% because the most data is general text.
PVOL	LOW	Major changes of the platform, i.e. Heroku, MongoDB, and web browsers, are approximately every year.
ACAP	HIGH	The analysts have the ability to analyze, design, communicate, and cooperate well.
PCAP	HIGH	Programmers are capable, efficient and thorough. They are able to communicate and cooperate very well.
PCON	LOW	We have 7 team members in CSCI577a and 2 of us will not continue to CSCI577b. The percentage of leaving personnel is approximately 28%.
APEX	LOW	The average experience of the team members for this graphical intensive online web-based application is about 6 months.
LTEX	NOMINAL	The development team plans to develop this web-based application with mostly Javascript, and uses MongoDB queries for database manipulations. Some of us are familiar with Javascript and two members have used MongoDB before. Therefore, the language and tool experience is nominal because the average experience with these language and tools of team members is one year.
PLEX	LOW	The local testing server platform is Apache Tomcat and the database is MongoDB. The deployment platform is heroku. Only a couple of the team members have experience with MongoDB and heroku.
TOOL	LOW	The software tools development team plan to use is just simple, frontend, backend CASE, and supporting little integration. There is no support for life-cycle.
SITE	HIGH	Six of seven team members are on-campus students. The team members meet in person in study rooms in the library regularly.
SCED	NOMINAL	The schedule is fixed for 12 weeks in Fall semester and 12 weeks in Spring semester.

Table 13: COCOMOII Cost Drivers of Module 2 – 3D Visualization module

Cost Driver	Value	Rationale
RELY	NOMINAL	Although this module supports one of the core capabilities, the effect of the software failure is moderate and losses are easily recoverable.
DATA	LOW	Since the database will store only information of the user and his/her sphere modeling info, which are questions and ratings, the ratio of bytes in the testing database to SLOC in the program are approximately less than 10.
DOCU	NOMINAL	Because the development process follows ICSM, the document for life-cycle needs is normal.
CPLX	VERY HIGH	It contains non-trivial point distribution algorithm to generate a 3D sphere, moderately complex 3D dynamic graphics, and multimedia.
RUSE	LOW	It is not intended to be reused for the future project.
TIME	NOMINAL	There is no strict constrains for execution time.
STOR	NOMINAL	The percentage of available storage expected to be used by the system and subsystem is less than 50% because the most data is general text.
PVOL	LOW	Major changes of the platform, i.e. Heroku, MongoDB, and web browsers, are approximately every year.
ACAP	HIGH	The analysts have the ability to analyze, design, communicate, and cooperate well.
PCAP	HIGH	Programmers are capable, efficient and thorough. They are able to communicate and cooperate very well.
PCON	LOW	We have 7 team members in CSCI577a and 2 of us will not continue to CSCI577b. The percentage of leaving personnel is approximately 28%.
APEX	LOW	The average experience of the team members for this graphical intensive online web-based application is about 6 months.
LTEX	NOMINAL	The development team plans to develop this web-based application with mostly Javascript, and uses MongoDB queries for database manipulations. Some of us are familiar with Javascript and two members have used MongoDB before. Therefore, the language and tool experience is nominal because the average experience with these language and tools of team members is one year.
PLEX	LOW	The local testing server platform is Apache Tomcat and the database is MongoDB. The deployment platform is heroku. Only a couple of the team members have experience with MongoDB and heroku.
TOOL	LOW	The software tools development team plan to use is just simple, frontend, backend CASE, and supporting little integration. There is no support for life cycle.
SITE	HIGH	Six of seven team members are on-campus students. The team

		members meet in person in study rooms in the library regularly.
SCED	NOMINAL	The schedule is fixed for 12 weeks in Fall semester and 12 weeks in Spring semester.

Table 14: COCOMOII Cost Drivers of Module 3 – Authentication module

Cost Driver	Value	Rationale
RELY	NOMINAL	Although this module supports one of the core capabilities, the effect of the software failure is moderate and losses are easily recoverable.
DATA	LOW	Since the database will store only information of the user and his/her sphere modeling info, which are questions and ratings, the ratio of bytes in the testing database to SLOC in the program are approximately less than 10.
DOCU	NOMINAL	Because the development process follows ICSM, the document for life-cycle needs is normal.
CPLX	LOW	User authentications are common routines and require basic database manipulations.
RUSE	LOW	It is not intended to be reused for the future project.
TIME	NOMINAL	There is no strict constrains for execution time.
STOR	NOMINAL	The percentage of available storage expected to be used by the system and subsystem is less than 50% because the most data is general text.
PVOL	LOW	Major changes of the platform, i.e. Heroku, MongoDB, and web browsers, are approximately every year.
ACAP	HIGH	The analysts have the ability to analyze, design, communicate, and cooperate well.
PCAP	HIGH	Programmers are capable, efficient and thorough. They are able to communicate and cooperate very well.
PCON	LOW	We have 7 team members in CSCI577a and 2 of us will not continue to CSCI577b. The percentage of leaving personnel is approximately 28%.
APEX	LOW	The average experience of the team members for this graphical intensive online web-based application is about 6 months.
LTEX	NOMINAL	The development team plans to develop this web-based application with mostly Javascript, and uses MongoDB queries for database manipulations. Some of us are familiar with Javascript and two members have used MongoDB before. Therefore, the language and tool experience is nominal because the average experience with these language and tools of team members is one year.

PLEX	LOW	The local testing server platform is Apache Tomcat and the database is MongoDB. The deployment platform is heroku. Only a couple of the team members have experience with MongoDB and heroku.
TOOL	LOW	The software tools development team plan to use is just simple, frontend, backend CASE, and supporting little integration. There is no support for life cycle.
SITE	HIGH	Six of seven team members are on-campus students. The team members meet in person in study rooms in the library regularly.
SCED	NOMINAL	The schedule is fixed for 12 weeks in Fall semester and 12 weeks in Spring semester.

Table 15: COCOMOII Cost Drivers of Module 4 – Data Entry module

Cost Driver	Value	Rationale
RELY	NOMINAL	Although this module supports one of the core capabilities, the effect of the software failure is moderate and losses are easily recoverable.
DATA	LOW	Since the database will store only information of the user and his/her sphere modeling info, which are questions and ratings, the ratio of bytes in the testing database to SLOC in the program are approximately less than 10.
DOCU	NOMINAL	Because the development process follows ICSM, the document for life-cycle needs is normal.
CPLX	LOW	Data entry is a common routine and requires basic database I/O manipulations.
RUSE	LOW	It is not intended to be reused for the future project.
TIME	NOMINAL	There is no strict constrains for execution time.
STOR	NOMINAL	The percentage of available storage expected to be used by the system and subsystem is less than 50% because the most data is general text.
PVOL	LOW	Major changes of the platform, i.e. Heroku, MongoDB, and web browsers, are approximately every year.
ACAP	HIGH	The analysts have the ability to analyze, design, communicate, and cooperate well.
PCAP	HIGH	Programmers are capable, efficient and thorough. They are able to communicate and cooperate very well.
PCON	LOW	We have 7 team members in CSCI577a and 2 of us will not continue to CSCI577b. The percentage of leaving personnel is approximately 28%.
APEX	LOW	The average experience of the team members for this graphical intensive online web-based application is about 6 months.
LTEX	NOMINAL	The development team plans to develop this web-based application with mostly Javascript, and uses MongoDB queries

		for database manipulations. Some of us are familiar with Javascript and two members have used MongoDB before. Therefore, the language and tool experience is nominal because the average experience with these language and tools of team members is one year.
PLEX	LOW	The local testing server platform is Apache Tomcat and the database is MongoDB. The deployment platform is heroku. Only a couple of the team members have experience with MongoDB and heroku.
TOOL	LOW	The software tools development team plan to use is just simple, frontend, backend CASE, and supporting little integration. There is no support for life cycle.
SITE	HIGH	Six of seven team members are on-campus students. The team members meet in person in study rooms in the library regularly.
SCED	NOMINAL	The schedule is fixed for 12 weeks in Fall semester and 12 weeks in Spring semester.

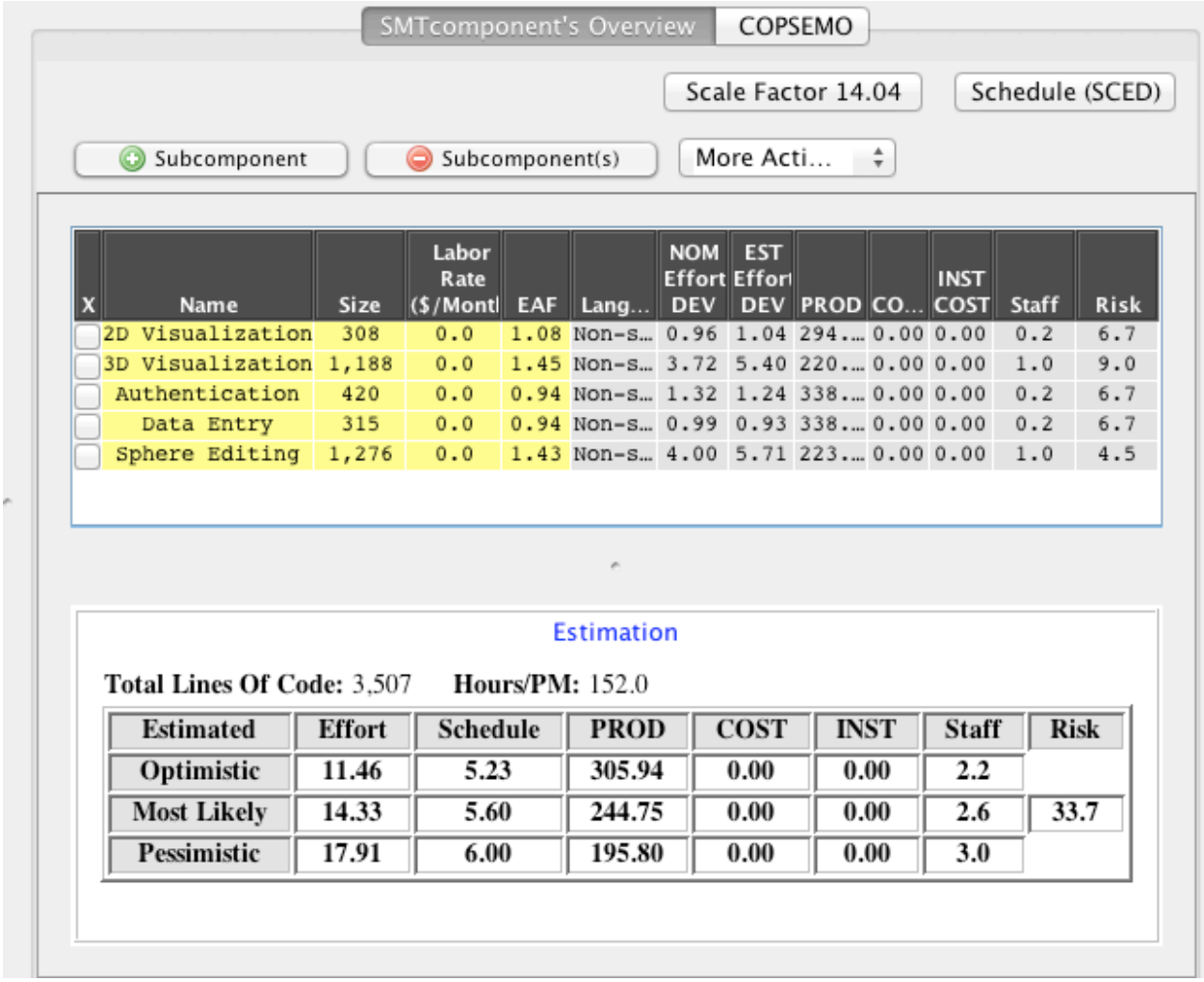
Table 16: COCOMOII Cost Drivers of Module 5 – Sphere Editing module

Cost Driver	Value	Rationale
RELY	NOMINAL	Although this module supports one of the core capabilities, the effect of the software failure is moderate and losses are easily recoverable.
DATA	LOW	Since the database will store only information of the user and his/her sphere modeling info, which are questions and ratings, the ratio of bytes in the testing database to SLOC in the program are approximately less than 10.
DOCU	NOMINAL	Because the development process follows ICSM, the document for life-cycle needs is normal.
CPLX	HIGH	Data is retrieved from 2D Highcharts. The NDI Highcharts doesn't support the dragging of the points and the team needs to develop an algorithm to interpret the point position.
RUSE	LOW	It is not intended to be reused for the future project.
TIME	NOMINAL	There is no strict constrains for execution time.
STOR	NOMINAL	The percentage of available storage expected to be used by the system and subsystem is less than 50% because the most data is general text.
PVOL	LOW	Major changes of the platform, i.e. Heroku, MongoDB, and web browsers, are approximately every year.
ACAP	HIGH	The analysts have the ability to analyze, design, communicate, and cooperate well.
PCAP	HIGH	Programmers are capable, efficient and thorough. They are able to communicate and cooperate very well.
PCON	LOW	We have 7 team members in CSCI577a and 2 of us will not

		continue to CSCI577b. The percentage of leaving personnel is approximately 28%.
APEX	LOW	The average experience of the team members for this graphical intensive online web-based application is about 6 months.
LTEX	NOMINAL	The development team plans to develop this web-based application with mostly Javascript, and uses MongoDB queries for database manipulations. Some of us are familiar with Javascript and two members have used MongoDB before. Therefore, the language and tool experience is nominal because the average experience with these language and tools of team members is one year.
PLEX	LOW	The local testing server platform is Apache Tomcat and the database is MongoDB. The deployment platform is heroku. Only a couple of the team members have experience with MongoDB and heroku.
TOOL	LOW	The software tools development team plan to use is just simple, frontend, backend CASE, and supporting little integration. There is no support for life-cycle.
SITE	HIGH	Six of seven team members are on-campus students. The team members meet in person in study rooms in the library regularly.
SCED	NOMINAL	The schedule is fixed for 12 weeks in Fall semester and 12 weeks in Spring semester.

The following is the result from COCOMOII estimation based on Scale Drivers and Cost Drivers discussed above (COCOMO II does not support Mac OS, and the online version of COCOMO II disables form submission. Thus, COINCOMO is used here instead.).

Figure 1: COCOMO Estimation Result



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

According to the estimation in Figure 1 above, pessimistically it will require a total effort of 17.91 person-months to finish the project. With the setting 152.0 Hours/PM, 17.91 person-months is roughly equivalent to 2356.0 person-hours, which is slightly greater than our team estimated total effort 2352 person-hours. Based on this updated estimation, the project is doable within 24-week timeframe and with a team size of seven.

6. Iteration Plan

6.1 Plan

The Construction Iteration consists of two iterations — Core Capability iteration and Full Capability iteration. Within these two iterations, we will adopt Scrum development process. Each iteration consists of multiple two-week sprints. Specifically, there are 3 sprints in Core Capability iteration and 1 in Full Capability iteration.

The capabilities have story points (the ease of realization in Winbook). We used the story points to decide the number of features to implement in each sprint. Based on our prior experience, we can achieve 50 points per sprint. When we actually start implementing the system, the points we can achieve will be measured, and the product backlog for each sprint will be updated accordingly.

The following iterations are planned for Core Capability Drive Through Milestone. During the Development Phase—Construction Iteration, project team will be implementing the core capabilities specified in Winbook, and perform test for the core capabilities.

6.1.1 Capabilities to be implemented

Before the Core Capability Drive Through Milestone, the capabilities to be implemented are listed in Table 17:

Table 17: Construction iteration capabilities to be implemented

ID	Capability	Description	Priority	Iteration	Sprint	Story points
1	Toggle between 2D and 3D visualizations	As a user I can toggle between 2D and 3D visualizations. This task only concerns switching between views, not implementing those views.	1	1	1	3
2	Title a sphere	As a user I can title a sphere. No restrictions on duplicate names. Spheres require a unique id.	1	1	1	3
3	Edit data in table format (editing node values).	As a user I can edit data in table format (editing node values).	2	1	1	5
4	Generate questions for each node	As a user I can generate questions for each node.	2	1	1	5
5	Delete sphere	As a sphere owner I can delete my	2	1	1	3

		spheres.				
6	Enter sectors (categories) for each node	As a user I can enter sectors (categories) for each node. This includes different colored nodes for each sector. This doesn't include rendering effort.	2	1	1	5
7	User Registration	As a user I can register so that I can create a sphere. Requires email, name, and password.	2	1	1	5
8	Reset password	As a user I can reset my password.	2	1	1	3
9	Enter node labels	As a user I can enter node labels. Weight -> Is my weight where I want it to be? -> Physical	2	1	1	5
10	Open an existing sphere project	As a user I can open an existing sphere project. This includes loading a sphere from the database. This includes a page listing all spheres the user has access to.	2	1	1	5
11	Update display name	As a user I can update my display name.	2	1	1	3
12	Overlay data in 2D chart	As a user I can overlay data from different users so that I can compare different shapes (w/different colors). This only includes the 2D rendering.	2	1	1	5
13	Edit/add/remove nodes (questions), sectors (categories), and sphere title (with admin permission).	As a user I can edit/add/remove nodes (questions), sectors (categories), and sphere title (with admin permission). Assume sphere creation functions already exist.	2	1	2	8
14	Assign permission to other users	As a user I can assign permissions to other registered users. This includes defining (hardcoded) user roles. This doesn't include how those permissions are used/enforced. Permissions are Sphere Owner, Admin, Collaborator, and Viewer.	2	1	2	8
15	Save a sphere as a template	As a user I can save a sphere as a template. All questions and nodes labels exist, but no values.	3	1	2	8
16	Search sphere by	As a user I can search my spheres	3	1	2	8

	name and owner	by name and owner.				
17	View the average of all (or subset) of selected user data in a single system	As a user I can view the average of all (or subset) of selected user data in a single system. This does not include collaborative editing features. This does not include rendering the overlay.	3	1	2	8
18	Visualize the data in a 2D chart	As a user I can visualize the data in a 2D chart. This includes grouping the nodes into categories.	3	1	2	13
19	Prompt other users via email	As a user I can prompt other users to enter data (notification through app and email).	3	1	3	13
20	Create a sphere from template	As a user I can create a sphere from a template.	4	1	3	13
21	Select the set of data to display in 2D/3D visualization	As a user I can select which sets of data show up in the 2D/3D visualizations.	4	1	3	13

6.1.2 Capabilities to be tested

Before the Core Capability Drive Through Milestone, the software features to be tested are listed in Table 18:

Table 18: Construction iteration Software Features to be tested

ID	Capability	Description	Priority	Iteration	Sprint	Story points
2	Title a sphere	As a user I can title a sphere. No restrictions on duplicate names. Spheres require a unique id.	1	1	1	3
3	Edit data in table format (editing node values).	As a user I can edit data in table format (editing node values).	2	1	1	5
4	Generate questions for each node	As a user I can generate questions for each node.	2	1	1	5
5	Delete sphere	As a sphere owner I can delete my spheres.	2	1	1	3
6	Enter sectors (categories) for	As a user I can enter sectors (categories) for each node. This	2	1	1	5

	each node	includes different colored nodes for each sector. This doesn't include rendering effort.				
7	User Registration	As a user I can register so that I can create a sphere. Requires email, name, and password.	2	1	1	5
8	Reset password	As a user I can reset my password.	2	1	1	3
9	Enter node labels	As a user I can enter node labels. Weight -> Is my weight where I want it to be? -> Physical	2	1	1	5
10	Open an existing sphere project	As a user I can open an existing sphere project. This includes loading a sphere from the database. This includes a page listing all spheres the user has access to.	2	1	1	5
11	Update display name	As a user I can update my display name.	2	1	1	3
13	Edit/add/remove nodes (questions), sectors (categories), and sphere title (with admin permission).	As a user I can edit/add/remove nodes (questions), sectors (categories), and sphere title (with admin permission). Assume sphere creation functions already exist.	2	1	2	8
14	Assign permission to other users	As a user I can assign permissions to other registered users. This includes defining (hardcoded) user roles. This doesn't include how those permissions are used/enforced. Permissions are Sphere Owner, Admin, Collaborator, and Viewer.	2	1	2	8
15	Save a sphere as a template	As a user I can save a sphere as a template. All questions and nodes labels exist, but no values.	3	1	2	8
16	Search sphere by name and owner	As a user I can search my spheres by name and owner.	3	1	2	8
18	Visualize the data in a 2D chart	As a user I can visualize the data in a 2D chart. This includes grouping the nodes into categories.	3	1	2	13
20	Create a sphere from template	As a user I can create a sphere from a template.	4	1	3	13

Non-functional requirements that will be tested before Core Capability Drive Through Milestone are listed in Table 19:

Table 19: Construction iteration non-functional requirements to be tested

ID	Capability	Description	Priority	Iteration	Sprint
1	Fast rendering of 2D chart	With HighChart API, the rendering time of 2D chart should be no more than 5 seconds.	1	1	1

6.1.3 Capabilities not to be tested

Background information: Before the Core Capability Drive Through Milestone, the software features that will NOT be implemented are listed in Table 20 (will be implemented in Full Capability Iteration):

Table 20: Software Features NOT to be implemented before Core Capability Drive Through Milestone

ID	Capability	Description	Priority	Iteration	Sprint	Story points
22	Share a sphere with another user	As a user I can share a sphere with another user. This includes allowing another user to view your sphere and edit data (based on permissions).	5	2	1	20
23	View 3D sphere data visualization	As a user I see the data visualize in a 3D sphere. This includes grouping the nodes by category.	5	2	1	40

Before the Core Capability Drive Through Milestone, the software features that will not be tested are listed in Table 21:

Table 21: Construction iteration Software Features not to be tested

ID	Capability	Rationale	Priority	Iteration	Sprint	Story points
1	Toggle between 2D and 3D visualizations	This feature can only be tested after the implementation of "View 3D sphere data visualization", which is in Full Capability Iteration.	1	1	1	3
12	Overlay data in 2D chart	This feature can only be tested after the implementation of "Share a sphere with another user", which is in Full Capability Iteration.	2	1	1	5
17	View the average of all (or subset) of selected user data in a single system	This feature can only be tested after the implementation of "Share a sphere with another user", which is in Full Capability Iteration. Only after an sphere/project is shared with other users, other users can then enter data. This feature also depends on the feature "View 3D sphere data visualization", which is also planned in Full Capability Iteration.	3	1	2	8
19	Prompt other users via email	This feature is subsequent to "Share a sphere with another user". It can only be tested after the implementation of "Share a sphere with another user", which is in Full Capability Iteration.	3	1	3	13
21	Select the set of data to display in 3D visualization	This feature can only be tested after the implementation of "View 3D sphere data visualization", which is in Full Capability Iteration.	4	1	3	13

6.1.4 CCD Preparation Plans

<< Identify the clients and other users who will be involved in the Core Capability Drive-through, the usage scenarios that it will support, and the specific CCD preparation plans and milestones. These may include

- user context-setting
- site preparation dry runs,
- feedback forms, and
- CCD risk management plans. >>

6.2 Iteration Assessment

6.2.1 Capabilities Implemented, Tested, and Results

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

Table 2: Capabilities implemented, tested, and results

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

6.2.2 Core Capabilities Drive-Through Results

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

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

Note: Make sure to be specific to the capabilities shown/demonstrated/driven-through. Simply stating that the clients liked the capabilities is not sufficient. >>

6.3 Adherence to Plan

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