Project Management Plan

for

Theia

Version 3.0 final draft

Prepared by Team Bagel: Arlo Roos, Aaron Howe, Joshua Evans,
Grace Anderson, Nick Vendeland, and Jaeger Nelson

CptS 484 Software Requirements

09/14/25

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Initial Draft	09/14/25	Initial draft	1.0 draft 1
Project Phase I: Final Submission	10/12/25	Revised to focus on deliverables and documentation	2.0 draft 2
Project Phase II: Final Submission	11/25/25	Extended Phase adds work from new documents (KAOS Models, IDEF0, Project Vision)	3.0 final draft

1. Overview

1.1. Project Purpose, Objectives, and Success Criteria

The purpose of Phase I is to conduct thorough requirements elicitation and analysis for Theia, an indoor navigation smartphone application designed to assist blind and visually impaired individuals. This phase emphasizes the requirements engineering process, producing comprehensive documentation that demonstrates deep understanding of stakeholder needs, domain constraints, and system objectives.

The primary objective of this phase is to develop a complete WRS document that identifies and resolves issues in the preliminary project specification. We aim to create three comprehensive AS-IS and TO-BE scenario pairs that demonstrate current problems faced by blind users and how Theia addresses them. Additionally, we will produce interface mockups and a user manual showing how blind users will interact with the system. Throughout this process, our focus remains on producing professional, well-organized documentation suitable for stakeholder review rather than technical implementation.

Success for Phase I means delivering a complete WRS document following the provided template with all sections thoroughly filled out. We will identify at least fifteen distinct issues across domain, functional, and non-functional requirements, with each issue including a clear problem statement, multiple options considered, justified decision, and traceability to requirements. Our three scenario pairs will be complete and detailed, with thorough analysis of the top priority scenario. The mockups will demonstrate all major user interaction flows with the appropriate consideration for accessibility needs. Finally, all team members will contribute meaningfully to the project as documented in our meeting records.

1.2. Project Deliverables

Deliverable	Delivery Date	Delivery Method	Comments
Preliminary Project Plan	Week 3	Canvas	Initial organization and timeline
AS-IS/TO-BE Scenarios	Week 7	Canvas/Zoom presentation	Delivered presentation via Zoom
WRS Document	Week 8	Canvas	To be submitted with mockups, meeting records, and AS-IS/TO-BE scenarios
Revised Project Plan	Week 8	Canvas	Based on phase I of the project plan
Peer Review Forms	Week 8	Canvas	Individual peer review forms
Project Vision and Scope	Week 12	Canvas	Follows template B
Process Specification	Week 14	Canvas	None

1.3. Assumptions, Dependencies, and Constraints

Assumptions:

- **AS-1:** We assume all team members have access to the preliminary project specification and WRS template provided through Canvas.
- **AS-2:** The team can meet virtually at least bi-weekly, and all members have foundational experience with requirements documentation from previous coursework.
- **AS-3:** We expect that example WRS documents from past projects will be available for reference, and that the instructor will be available during office hours for clarification questions.

Dependencies:

- **DE-1:** We need access to example WRS documents showing expected format and depth of analysis.
- **DE-2:** Team member availability will be coordinated through a shared calendar
- **DE-3:** Instructor feedback during the checkup meeting will be crucial for ensuring we are on the right track.
- **DE-4:** Access to WSU accessibility resources would help us better understand the needs of blind users.

Constraints:

- **CO-1:** Phase 1 limited requirements analysis and prototype (no full implementation)
- **CO-2:** Project timeline constrained to academic semester schedule
- **CO-3:** Team resources limited to student availability and domain knowledge/expertise
- **CO-4:** Prototype functionality constrained by available development tools and time
- **CO-5:** We must focus on requirements engineering artifacts and documentation rather than technical implementation details

1.4. References

- https://app.perusall.com/courses/2025-fall-cpt_s-484-onlin-1-01-10381-software-requirements/cpt-s-484-project-specification-i-v1-3?assignmentId=6ievFq7CyWadCrzrr&part=1
- Requirements Review Checklist
- WRS Template
- Sample WRS document

1.5. Definitions and Acronyms

- *AS-IS* - An AS-IS scenario describes the current state or problem before the system exists
- *FR* - Functional Requirements
- *NFR* - Non-functional Requirements
- *TO-BE* - A TO-BE scenario describes the future state or solution after system implementation
- *WRS* - Written Requirements Specification
- *Theia* - Greek goddess of vision; project codename for the indoor navigation app

2. Project Organization

2.1. Process Model

Phase I follows an iterative requirements engineering process model divided into distinct periods. During weeks one and two, we focus on project initiation and team formation. This includes organizing the team and assigning roles, reviewing the project specification and template documents, establishing our meeting schedule and communication channels, and creating the preliminary project plan.

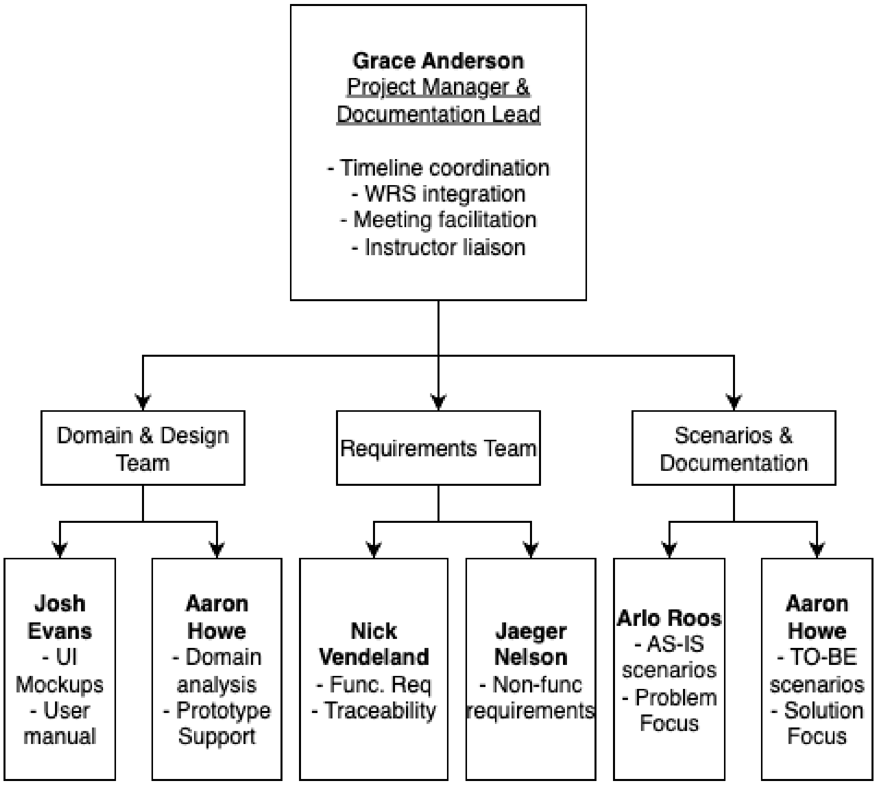
Weeks three and four are dedicated to requirements analysis and issue identification. We will conduct detailed analysis of the preliminary specification and systematically identify issues with the domain description, functional requirements, and non-functional requirements. Each issue will be documented with multiple resolution options and clear rationales for our decisions. During this period, we will also begin developing the AS-IS scenarios that describe current problems.

During weeks five and six, we focus on WRS development and the optional check-up meeting with the instructor. We will develop an improved understanding of the domain, stakeholders, and objectives, creating complete requirements traceability from problems through goals, objectives, and requirements. The TO-BE scenarios will be drafted during this period. The check-up meeting provides an opportunity to receive feedback and ensure we are meeting expectations. Presentation slide preparation also begins during this week.

Week seven is devoted to prototype mockups and user manual development. We will create interface mockups showing key user interactions, develop the preliminary user manual, and complete the specification section of the WRS.

Week eight focuses on finalization and delivery. We will incorporate any feedback received from the check-up meeting, finalize all documentation, complete presentation preparation, submit all deliverables, and conduct our team presentation.

2.2. Organizational Structure



2.3. Roles and Responsibilities

Grace Anderson, as Project Manager and Documentation Lead, coordinates overall project activities and manages the timeline. She integrates the WRS document components and checks for consistency throughout. Grace facilitates meetings, prepares agendas, and submits deliverables to Canvas. She serves as the primary liaison with the instructor.

Josh Evans leads the domain issues analysis, carefully examining the domain description in the preliminary specification to identify ambiguities and incompleteness. He develops options and rationales for resolving domain issues and contributes to the improved domain description in the WRS. Josh supports the overall requirements analysis effort across the team. He creates interface mockups for the prototype, ensuring the mockups address accessibility needs of blind users.

Nick Vendeland leads functional requirements analysis, examining each preliminary functional requirement to identify issues. He develops options and rationales for functional requirements issues and creates the improved functional requirements and specifications sections. Nick maintains traceability from objectives through requirements to specifications.

Jaeger Nelson focuses on non-functional requirements, analyzing each preliminary NFR to identify issues. He develops options and rationales for NFR issues and creates improved non-functional requirements. Where possible, Jaeger works to operationalize the non-functional requirements into measurable criteria.

Arlo Roos develops three AS-IS scenarios showing current problems faced by blind users attempting indoor navigation. Arlo coordinates with Aaron to ensure consistency between AS-IS and TO-BE scenarios.

Aaron Howe develops three TO-BE scenarios showing how Theia solves the problems illustrated in the AS-IS scenarios. He writes the preliminary user manual, ensuring it is appropriate for blind users who will rely on screen readers or other assistive technologies. Aaron conducts detailed analysis of the top priority TO-BE scenario for the presentation.

All team members share responsibility for attending scheduled meetings, reviewing and providing feedback on others' work, contributing to presentations and rehearsals, maintaining meeting records when assigned, and completing peer review forms honestly and constructively.

3. Managerial Process Plans

3.1. Management objectives and priorities

Our approach when it comes to the management of this project will focus on breaking the project down into small, well-defined tasks. With well-defined tasks, we can expect clear deliverables that will be made through steady progress. This strategy should keep us away from rushing tasks at the last minute. Our team will be focused on quality communication, early identification of challenges or blocks, and collaborative problem-solving. As the project evolves, we will also focus on adjusting the requirements or scope of the project to ensure a complete deliverable at the end. Because this is both an academic and technical project, the team emphasizes adaptability and continuous learning to ensure all members gain experience while still delivering documentation.

We will use GitHub and Kanban boards to track the progress of each task. Progress will also be discussed in weekly team meetings. These tools will help keep team members accountable for the tasks assigned to them.

3.2. Assumptions, dependencies, and constraints

Our application will assume all team members have access to tools such as GitHub for version control, Kanban boards for task tracking, Discord for regular meetings and general communication, and Canvas for submitting deliverables. In addition, development will rely on tools such as Figma, React Native, and Node.js documentation. Constraints include conflicting schedules and a fixed academic semester timeline.

3.3. Risk management

No.	Risk	Type	Likelihood	Description
1	Integration Difficulties	Technical	Likely	New features may have issues integrating with the system.
2	Scheduling Conflicts	Managerial	Possible	Differing schedules can limit opportunities for collaboration.
3	Learning curve delays	Technical	Possible	Technical topics could require more study than anticipated.
4	External Service Failure	Technical	Unlikely	API's hosting services, or libraries, could be prone to failure or unreliability.
5	Scope Creep	Managerial	Possible	Last-minute or unnecessary features could push the project beyond the expected deadline.
6	Uneven Workload Distribution	Managerial	Possible	Members could become overloaded if tasks are not balanced.

Table 1: Potential risks and their descriptions

3.4. Monitoring and Controlling Mechanisms

No.	Risk	Monitor	Control
1	Integration Difficulties	Perform regular code reviews and incremental integration tests.	Require small, frequent commits. Enforce testing before merges.
2	Scheduling Conflicts	Track member availability through a shared calendar.	Rotate meeting times or redistribute work to balance schedules.
3	Learning curve delays	Monitor the time spent by each member on new tools and technologies during the early development phase.	Allocate time specifically for researching during the early stages. Pair less experienced team members with those more familiar.
4	External Service Failure	Beware of service updates. Monitor downtime reports.	Identify alternate tools or frameworks during planning as backups.
5	Scope Creep	Log all new feature requests in documentation.	Evaluate each new feature request against the timeline and scope before approval.
6	Uneven Workload Distribution	Review task assignments during weekly meetings	Rebalance tasks if a member has a consistently heavier workload.

Table 2: Mechanisms for controlling and monitoring the risks predicted in Table 1.

4. Technical Process Plans

4.1. Methods, tools, and techniques

Our requirements elicitation employs several complementary methods. Document analysis forms the foundation, involving systematic review of the preliminary specification using the requirements review checklist. Scenario development helps us understand both current problems and proposed solutions through concrete examples. We will practice self-ethnography by placing ourselves in the position of blind users to develop empathy and understanding of their needs. Research into accessibility guidelines, existing navigation applications, and indoor positioning technologies will inform our analysis.

For analysis, we follow a structured issue analysis framework. For each preliminary requirement, we systematically check for ambiguity by asking whether it can be interpreted in multiple ways. We examine incompleteness by identifying critical missing information. We look for inconsistency by checking whether requirements conflict with each other. We assess feasibility by considering whether the requirement can realistically be implemented. This framework ensures thorough and consistent analysis across all team members.

When we identify an issue, we employ options analysis by generating two to three resolution options with pros and cons for each. We then document clear decision rationale explaining our reasoning for the chosen option. This demonstrates critical thinking rather than arbitrary choices. Throughout the process, we maintain traceability analysis to verify that every requirement traces back to a problem or goal and forward to a specification.

Our documentation tools include Microsoft Word or Google Docs for the WRS document itself, Draw.io and Visio or Lucidchart for mockups and diagrams, and GitHub for document versioning and collaboration. GitHub Issues provides project management and task tracking. Discord facilitates team communication. Figma will be used for creating interface mockups. Presentation slides will be developed in PowerPoint or Google Slides.

4.2. Software documentation

Document	Template or Standard	Created By	Reviewed By	Target Date	Distribution
Requirements Specification (WRS)	Course WRS Template	Team Bagel	Bob		Canvas, GitHub
Revised Preliminary Plan	Course Project Plan Template	Team Bagel	Bob		Canvas, GitHub
Project Vision and Scope	Template B	Team Bagel	Bob		Canvas, Github
Process Specification	Standard	Team Bagel	Bob		Canvas, Github