Project Specification Document

for

Theia

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

CptS 484 Software Requirements

12/1/25

1. Introduction

This Process Specification Document describes the Requirements Engineering (RE) process used by Team Bagel in the development of Theia, an indoor navigation application designed to support blind and visually impaired users. The document explains how the team identified, analyzed, modeled, validated, and refined requirements across Phase I and Phase II. It also outlines how the RE activities were structured, how artifacts evolved through multiple iterations, and how formal modeling methods were incorporated into the process.

The process includes scenario development, issue analysis, KAOS modeling, non-functional reasoning, mockup creation, and prototype validation. This document complements the WRS and Vision documents by focusing specifically on the workflow and decision-making process that guided development.

2. Overview of the Requirements Engineering Process

The team followed a model-driven, iterative process that gradually increased in formality. The earliest stages emphasized understanding the domain, stakeholders, and preliminary problem space using scenario analysis and document review. As the project progressed, the team shifted toward more rigorous modeling techniques, including KAOS goal hierarchies, NFR softgoal refinement, obstacle modeling, and traceability mapping.

Throughout both phases, the process encouraged continual refinement: each iteration produced new insights that fed back into the WRS, mockups, and eventually the prototype.

3. Iterative Development Process

The team completed five major iterations, each progressively more refined. The first iteration focused on establishing the team structure, choosing tools, and reviewing the preliminary specification. The second iteration was dedicated to identifying issues within domain descriptions, functional requirements, and non-functional requirements, while simultaneously producing AS-IS and TO-BE scenarios.

The third iteration introduced formal modeling, including the KAOS goal hierarchy, responsibility models, and early obstacle analysis. This work formed the foundation of the first major WRS version. The fourth iteration centered on creating high-fidelity mockups, developing the initial user manual, and preparing presentation materials for Phase I.

The fifth iteration corresponded to Phase II and expanded previous artifacts to incorporate newly required considerations such as safety mechanisms, sensor integration, and HIPAA compliance. This iteration also included prototype development and the creation of formal process diagrams.

4. Roles and Responsibilities in the RE Process

Throughout the project, each team member contributed distinct responsibilities that shaped the Requirements Engineering (RE) workflow. Although the entire team collaborated on analysis, review, and decision-making, individual members assumed lead roles in key RE activities:

Grace Anderson, Project Manager & Documentation Lead
Grace coordinated meeting schedules, organized deliverables, and ensured consistency across all RE artifacts. She maintained the WRS and Process Specification documents, integrated feedback, and ensured alignment between modeling work, scenarios, and the evolving requirements.

Josh Evans, Domain Analyst & Accessibility Mockup Lead
Josh led the domain issue analysis by reviewing the preliminary specification for ambiguities, incompleteness, and conceptual gaps. He developed resolution options and rationales for domain-level issues and contributed significantly to the improved domain description in the WRS. In Phase II, he created accessibility-focused interface mockups that reflected realistic user interaction patterns for blind users, supporting scenario refinement and requirements validation.

Nick Vendeland, Functional Requirements Lead
Nick led the examination and refinement of functional requirements throughout both phases. He identified issues with preliminary FRs, developed resolution alternatives, and produced improved dysfunctional requirements and detailed specifications. Nick maintained traceability among objectives, KAOS functional goals, FRs, and prototype behaviors, ensuring that all functional aspects of Theia were grounded in formal RE methods.

Jaeger Nelson, Non-Functional Requirements Lead
Jaeger focused on identifying issues in the preliminary non-functional requirements and translating them into measurable, operationalized criteria. He contributed to softgoal refinement, obstacle analysis, and integration of NFR constraints into the KAOS models. His work ensured that critical qualities such as accessibility, responsiveness, and safety were consistently addressed across both documentation and prototype development.

Arlo Roos, Scenario Developer & Problem-Space Analyst
Arlo developed the AS-IS scenarios that established the foundational understanding of the challenges faced by blind users. His work informed early goal discovery and issue identification. He collaborated closely with Aaron to ensure alignment between AS-IS and TO-BE scenario pairs, creating a coherent narrative that supported both modeling and requirement refinement.

Aaron Howe, TO-BE Scenario Developer & Modeling Lead
Aaron authored the TO-BE scenarios that illustrated how Theia resolves the problems identified in the AS-IS scenarios. He produced the preliminary user manual and performed a detailed analysis of the high-priority scenario for Phase I. In Phase II, Aaron created the KAOS goal

models, obstacle models, responsibility models, and the complete set of IDEF0 process diagrams. His modeling work ensured that requirements, scenarios, and prototype behavior remained traceable and formally grounded.

Shared Responsibilities
All team members contributed to RE activities, including issue identification, scenario refinement, model review, WRS updates, and prototype evaluation. The team collectively engaged in meetings, provided peer feedback, rehearsed presentations, and maintained accurate meeting documentation. This shared participation ensured that the RE process remained iterative, collaborative, and consistently validated.

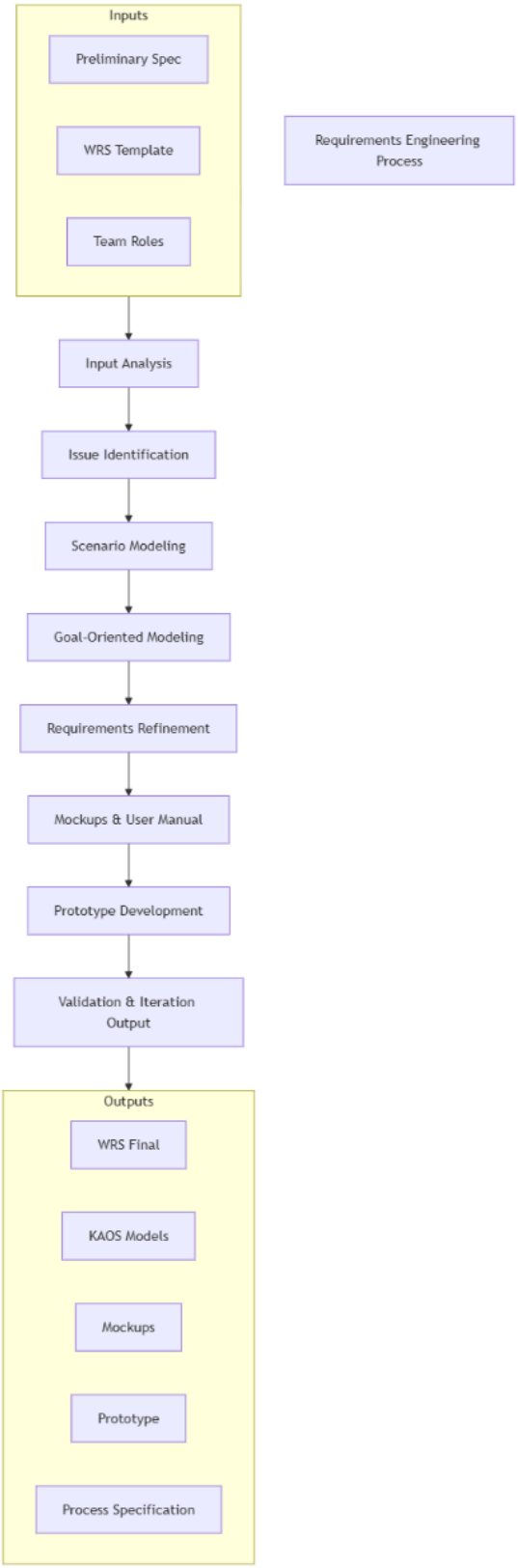
5. Relationship Between Phase I and Phase II

Phase 1 established the foundational artifacts, being the AS-IS and TO-BE scenarios, the issues list, early mockups, and the initial WRS. These deliverables were then used as direct inputs for Phase II, where they were expanded and formalized. The scenarios set guided prototype interaction design, while the issues list helped guide the refinement of functional and non-functional requirements.

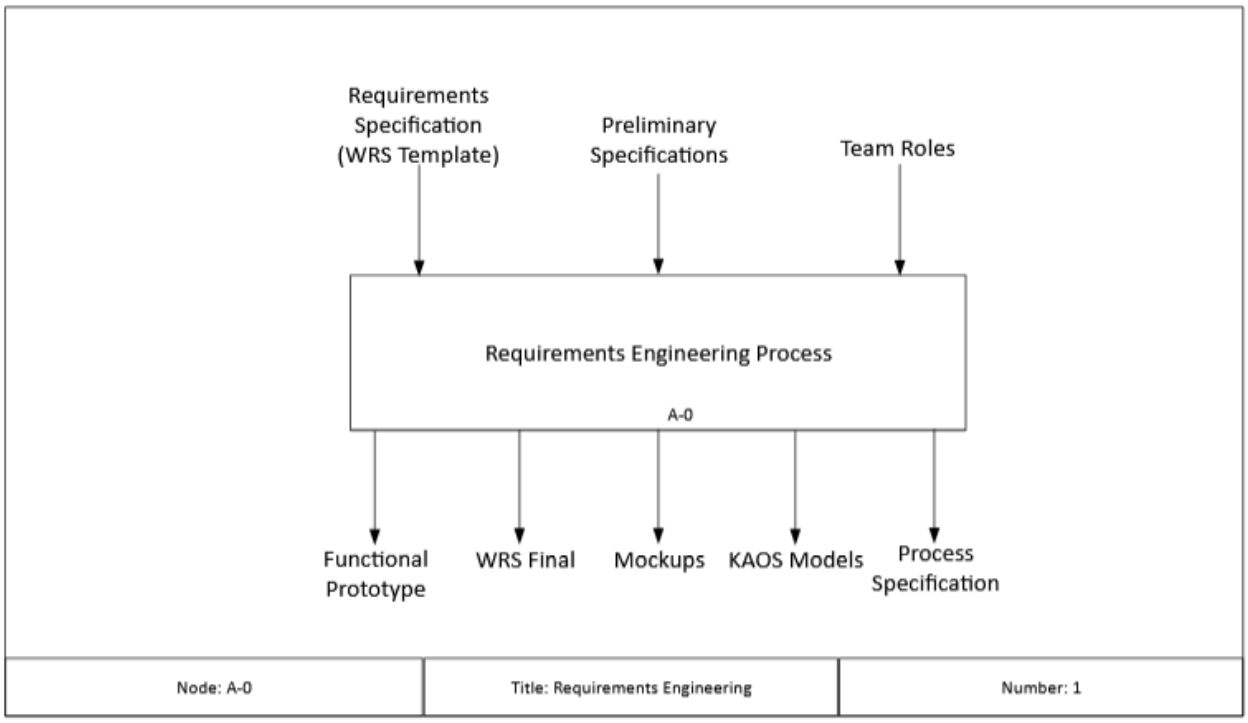
KAOS models were created after the end of Phase I and served as the structural basis for Phase II additions, allowing the team to plan for safety requirements, sensory-based functions, and privacy standards. Our mockups then evolved into interactive flows that could be used to evaluate the Phase II prototype.

Looking back, Phase II acted as more of an extension than a reset, primarily transforming Phase I artifacts into more formal, testable, and implementable specifications.

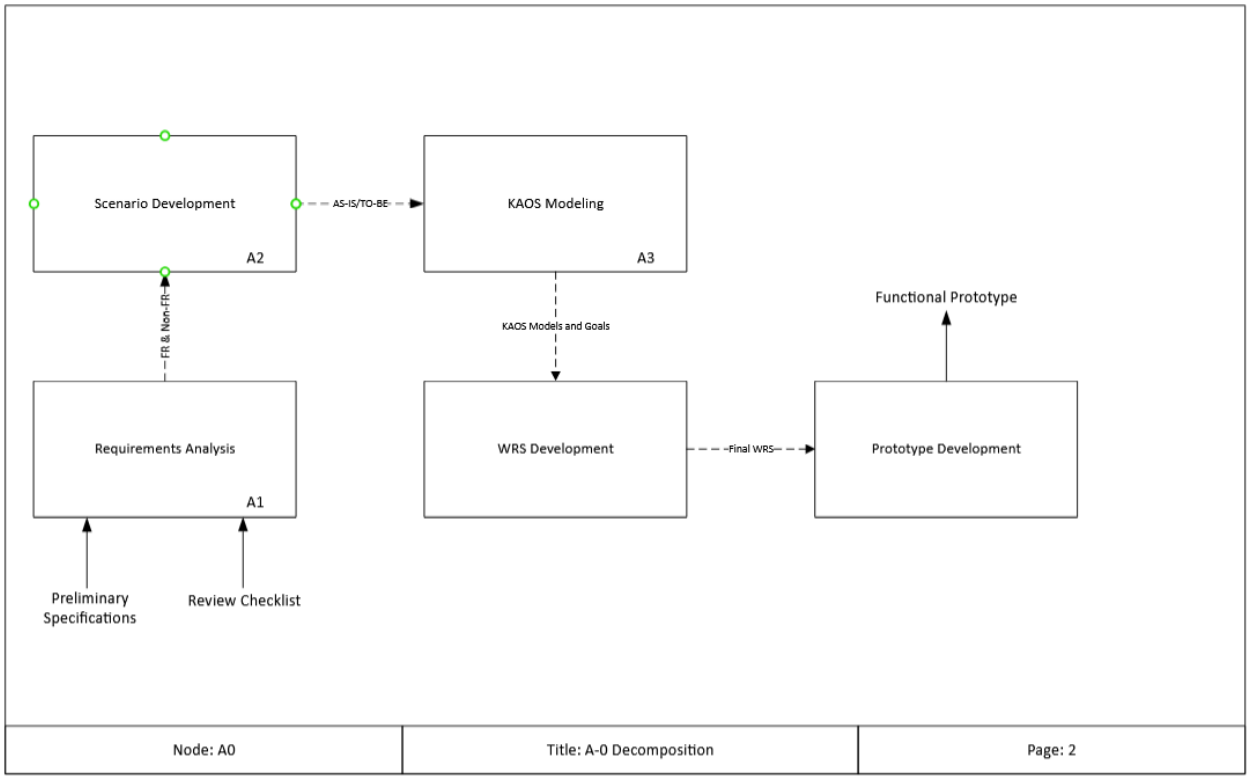
6. Functional Process Modeling



6.1 A0 Context Diagram

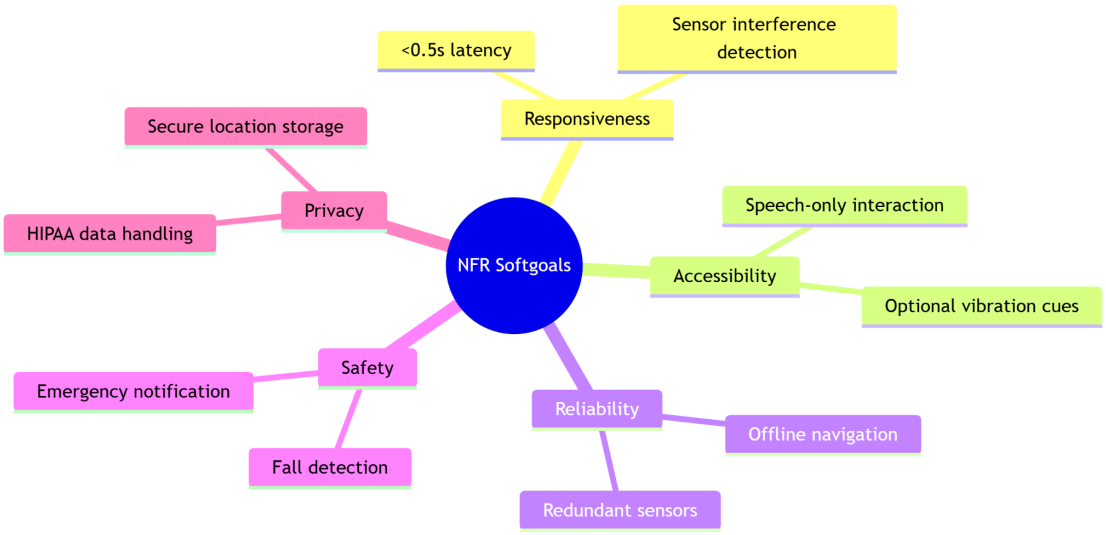


6.2 A0 Decomposition Diagram

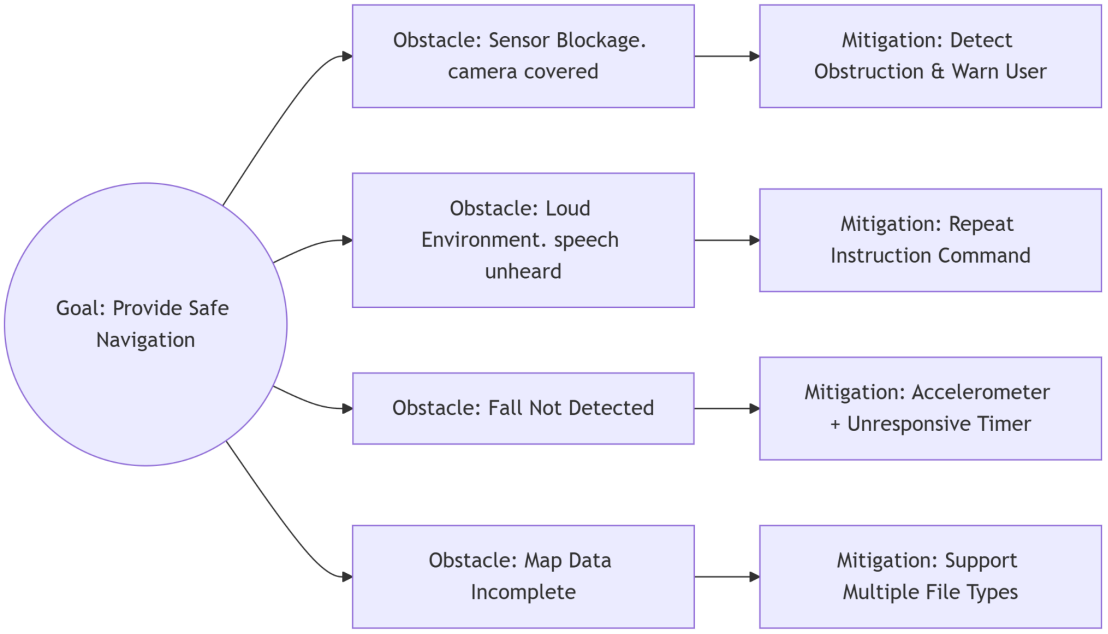


7. Non-Functional Process Modeling

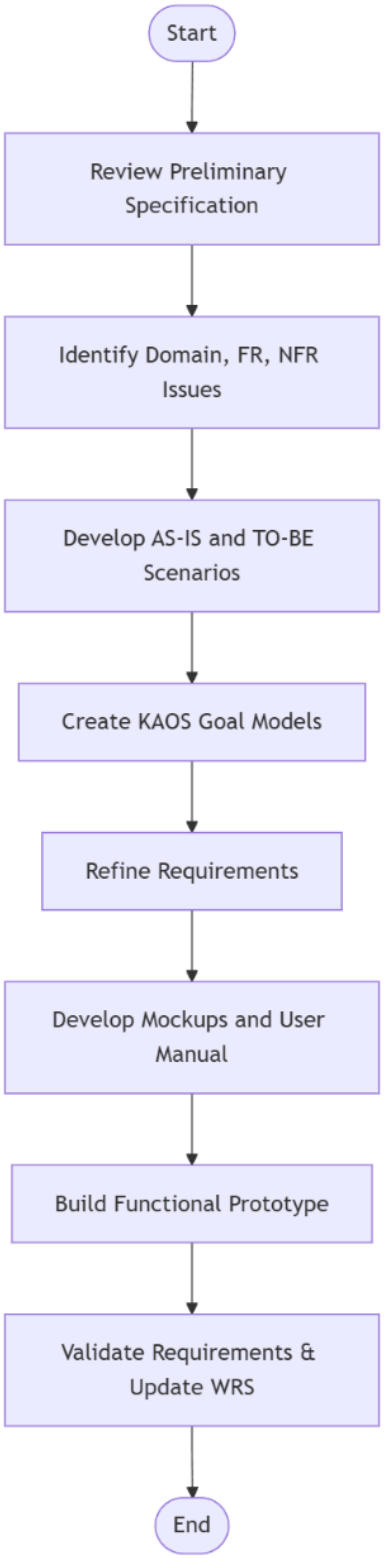
7.1. NFR Softgoal Model



7.2. KAOS obstacle diagram



7.3. UML activity diagram for Entire RE Process



8. Tools and techniques used

Team Bagel employed a consistent set of tools throughout the RE process:

Documentation & Collaboration Tools

- GitHub: version control, branching strategy, document tracking, and release management
- Google Docs: creation of WRS, Vision, and Process Specification documents
- Discord: coordination, RE discussion, meeting scheduling, and asynchronous communication

Modeling & Diagramming Tools

- Objectiver: reference for KAOS modeling conventions and structure

Prototype Development Tools

- Flutter: framework used to build the functional mobile prototype for both Android and iOS
- Dart: programming language used within Flutter
- VS Code: development environment, debugging tools, emulator support

Flutter was chosen because it offered many things, including fast iteration, strong accessibility support, cross-platform consistency, and simple integration with device sensors.

Requirements Engineering Techniques

- Ambiguity, incompleteness, and inconsistency analysis
- Scenario-based requirement elicitation
- Issue analysis using options and reasoning
- KAOS goal refinement, obstacle modeling, and traceability mapping
- NFR softgoal refinement and trade-off analysis
- Prototype-driven requirement validation in Phase II

These tools and techniques collectively supported a model-driven, iterative RE process anchored in stakeholder needs, accessibility considerations, and formal modeling methods.

9. Conclusion

This Process Specification Document captures the full workflow and reasoning used by Team Bagel in developing Theia. Through structured iterations, formal modeling techniques, and the integration of feedback, the team produced a comprehensive and validated set of requirements. This process ensured a clear connection between user needs, system goals, design decisions, and prototype implementation.