Review Report

Alexandros Gonatas
alexthecipher@gmail.com

Alexander Spetz
spetzal@student.chalmers.se

Haoning Ma
mahaoning21@outlook.com

Xingrong Zong
xingrong@student.chalmers.se

Contents

1 Review Planning Report

1.1 Introduction

This report describes the review process our group will follow to evaluate Group 1's Requirement Document on requirements specification. The goal of the review is to assess the document's quality in terms of completeness, consistency, structure, and clarity, and to identify potential issues that could hinder later stages of development.

1.2 Review Objectives

The main objectives of the review are:

- Detect inconsistencies, ambiguities, and missing requirements.
- Evaluate the structure and organization of the requirements document.
- Ensure traceability between goals, functional, and non-functional requirements.
- Assess clarity, testability, and feasibility of the stated requirements.

1.3 Review Method

Each group member will independently review the assigned requirements document following a predefined checklist. After the individual reviews, we will hold a group meeting to consolidate and prioritize findings. We use a **checklist-based review** approach.

1.4 Review Schedule

Planned timeline:

- Individual reviews: 08/10/2025
- Group discussion: 09/10/2025
- Consolidation and submission: 09/10/2025

1.5 Review Checklist

The following checklist will be used by all members during their individual review sessions. It is provided in Lausen's book, "Software Requirements; Styles and Techniques", in Chapter 9.

Each category is analysed in different subsections, which will be used to thoroughly check whether they exist in the reviewed document.

Contents check	Observations found and missing	Problem?
Customer and sponsor	Missing, Monetary sponsor	Unclear (e.g. add VGR)
Background	Found, Complete	
Supplier type	Missing	Mention explicitly
Business goals	Found, Complete, clearly linked to user needs	
Trace goals ->requirements	Found, Complete	
Data Requirements:		
Database contents	Found	Difficulty understanding diagram and data
Input/output formats	Found	
Communication states	Found, OK	
Initial data and states		
Functional requirements:		
Limits and interfaces	Found,	
Domain events per Interface	Found, mentioned indirectly, but not explicit	Recommend explicit mapping
Domain-level requirements, e.g. tasks	Found, but are written in a system-centric way	These read as system requirements rather than domain needs, blurring the boundary between the problem and solution space
Product-level events and functions	Found, some overlap with domain requirements	Overlap between domain-level and product-level descriptions reduces traceability between business needs and implementation
Design-level requirements, e.g. prototypes or comm. Protocols	Missing, OK	
Spec of non-trivial functions	Found, OK	
Special cases:		
Stress cases	Missing, OK	
Power failure, HW failure, configuration	Missing	Not being able to handle real-time failure and response time
Use cases dealing with special events or errors	Found, OK	
Quality requirements:		
Performance	Found	
Capacity, accuracy	Missing, OK	
Usability	Found	
Safety and security	Found	Could mention data encryption
Reliability	Found	
Availability	Found	
Fault tolerance	Found, security section	
Maintainability	Found	
Portability	Missing	
Reusability	Missing, OK	
Interoperability	Found	
Installability	Missing, OK	
Other qualities		
Other deliverables:		
Documentation	Found	
Training	Missing, OK	
Installation	Missing, OK	
Data conversion	Missing, OK	
System operation	Missing, OK	
Support and maintenance	Found	
Glossary:		
Domain and product terms	Missing, Explanations of technical terms	Misinterpreting requirements

Structure check	Observations found and missing	Problem?
ID for each requirement	Found, Complete	
Verifiable requirements	Found, Complete	
Purpose of each requirement	Found, Complete	
Examples of ways to meet requirement	Found, Complete	
Text explanation of diagrams, etc	Missing, Context diagram and Use Case diagram	Unregistered relations between entities
Importance and stability	Found, Complete	
No duplicate information	Found, OK	
Index	Found, Complete	
An electronic version	Found, Complete	
Consistency checks	**Observations found and missing**	**Problem?**
CRUD check: Create, read, update, delete all data?	Found, Complete	
All events handled by task or function?	Found, Complete	
Task and message data in database?	Found, Complete	
Virtual window data in database?	Missing, OK	
Virtual windows support all tasks?	Missing, OK	
Reviews	**Observations found and missing**	**Problem?**
Reviewed by developer and customer?	Missing	
Goals and critical issues covered?	Found, Complete	
Requirements justified?	Found, Complete	
Risk assessment	Missing, OK	
High-risk areas improved	Missing, OK	

2 Prioritized List of Issues

During the examination of the document, we identified primarily minor issues concerning the content, as well as several structural and editorial aspects. Below, we present a prioritized list of issues, ordered from the most serious content-related errors to the least critical structural and stylistic points. In other words, we prioritize fixing the content of the document before addressing editorial and presentation details.

1. In your data requirements and Business Goal A3 ("Integrate a review system for rides"), the concept of a "review" appears multiple times. However, it is not clear whether the review refers to a specific route, a vehicle, or a station. Could you please clarify the intended scope and purpose of this feature?
   Example: How will the collected review data be aggregated and presented? If users take different routes, how will you handle potential data redundancy or scalability issues?
   Suggestion: Applying reviews to bus/tram lines or stations.

2. Changing FR-06 in the functional requirements.
   Example: The sub-tasks and pre-condition of the requirement are conflicting. Either remove the first two points of the sub-tasks or change the pre-condition.

3. In your Traceability Matrix, FR10 has no corresponding system requirement (SR), and some other functional requirements (e.g., FR3, FR5, FR6) all link to the same SR3.1. Have you considered refining your system-level requirements so that each functional requirement can be individually traced and validated?
   Example: Should FR10 ("recommend routes") explicitly include system-level details such as the machine learning model or required data sources?

4. Fix Use Case diagram.
   Example: The use cases do not have ID's, that makes them untraceable. If the Use Cases are not referenced anywhere, maybe the diagram is not needed.

5. Obtaining the UUID of the vehicle.
   Example: In the data requirements, it is mentioned that a user will submit a report in case

they notice a fault, and the report may contain the UUID of the vehicle. It is not mentioned *how* the user will obtain the UUID.

6. Changing the person who sets the severity level for a report.
   Example: According to the data dictionary, a user sets the severity level for a report. This is a job that should be allocated to the person handling the report, as the users can mess the queue system by doing that.

7. Add more information about safety and security.
   Example: In the quality grid provided, Security is graded as 'Critical'.

8. Fix context diagram.
   Example: There is no relationship description between Västtrafik App and Payment Service Provider entities.

9. Your non-functional requirements mention GDPR compliance and secure login mechanisms, but the document does not clearly explain how these will be implemented, measured, or verified.
   Example: Will data transmission and storage be encrypted? How the system's security will be tested or validated? How user privacy—especially regarding location data—will be protected or anonymized? to assess the team's understanding and practical approach to fulfilling security and privacy requirements.

10. Remove appendices.
    Example: The appendices provided in the end of the document are useful, but they do not belong in the Requirements Document, they belong in the Experience Report as they are describing how you used certain techniques.

11. Change or remove User Story 9 - user stories are supposed to describe the needs of the stakeholders, not the system's.

12. Remove project descriptions from the document.

13. Edit and review the document as a whole, there are missing semantics in some areas.
    Example: "Pain points"- this has no meaning, but if it has meaning for you, you should add it to a glossary.