

Requirements Specification Template

Use this specification to define what a product or system must do and how each requirement will be checked. Keep requirement identifiers stable so design decisions, verification evidence, and deferred work can be traced to the same statement.

Complete the bracketed fields with approved information for the product, organization, or program being documented. Repeat the named forms where several tasks or subjects need separate records. Remove sections that do not apply and retain links to the material used to complete the document.

Document control	Entry
Document identifier	[Identifier used by your team]
Scope and applicable version	[Product, program, locations, and version covered]
Prepared and maintained by	[Responsible role or team]
Revision and review	[Revision, review date, and approval reference]

Introduction and Product Overview

Define the product boundary and the purpose of the specification. Identify the people and source material that establish its requirements.

Field	Complete with
Purpose and scope	[Product, intended use, covered release, system boundary, and exclusions]
Context and functions	[Users, neighboring systems, major functions, and context diagram]
Stakeholders and assumptions	[User classes, stakeholder concerns, dependencies, and assumptions]
Definitions and sources	[Terms, abbreviations, referenced documents, and source revisions]

External Interface Requirements

Repeat the requirement record for each interface class or major interface. Keep obligations separate from lookup information such as message schemas or protocol field tables.

Field	Complete with
Requirement identifier	[Stable identifier, interface, requirement status, and source]
Requirement statement	[One clear obligation identifying the actor, condition, and required behavior]
Interface contract	[Inputs, outputs, formats, timing or units, and reference to the contract]
Verification and priority	[Priority, rationale, measurable acceptance condition, and verification method]

Functional Requirements

Group requirements by feature or use case and give each statement its own identifier. Replace ambiguous terms with an observable result while preserving the requirement's original intent.

Field	Complete with
Requirement identifier	[Identifier, feature group, applicable user or mode, and source reference]
Trigger and prerequisites	[Event, starting state, permissions, data, and dependencies]
Required behavior	[One testable requirement and its expected result or state change]
Alternative and failure behavior	[Exceptions, invalid inputs, failure response, and acceptance criteria]

Non Functional Requirements

Repeat this record for each supported quality category. Record the environment and units of measurement so a target can be evaluated consistently.

Field	Complete with
Quality requirement	[Identifier, quality characteristic, applicability, and rationale]
Scenario	[Stimulus, operating conditions, affected item, and expected response]
Measure	[Metric, target, unit, tolerance, and measurement period or load when specified]
Verification	[Method, test environment, evidence, and acceptance decision reference]

Data Requirements

Describe the data the system owns or exchanges and the rules that protect its meaning. Keep field definitions and lifecycle rules traceable to the requirements that use them.

Field	Complete with
Entity and identity	[Entity name, purpose, identifier, relationships, and source]
Data dictionary	[Field name, type, format, units, allowed values, and required or nullable status]
Integrity and validation	[Validation rules, uniqueness, relationships, and error handling]
Lifecycle	[Creation, access, retention, archive, deletion, and applicable policy reference]

Design Constraints and Compliance

Record mandated constraints without presenting an implementation preference as an external obligation. Identify the authoritative source and the part of the system it affects.

Field	Complete with
Constraint identifier	[Identifier, requirement owner, status, and authoritative source]
Mandated condition	[Technology, platform, environment, standard, or other required constraint]
Applicability	[Affected component, product variant, location, or user group]
Acceptance	[Evidence, review method, approved exception, and decision reference]

Verification and Acceptance

Map every in-scope requirement to a verification method and an observable acceptance criterion. Keep a missing or blocked check visible rather than treating it as a pass.

Field	Complete with
Coverage	[Requirement groups and versions included in this acceptance review]
Method	[Inspection, analysis, demonstration, or test as defined for the requirement]
Evidence	[Test or review identifier, observed result, date, and reviewer role]
Decision	[Accepted, rejected, blocked, or deferred status and the approval reference]

Traceability Matrix

Requirement ID	Verification method	Acceptance criterion	Evidence	Status
[ID]	[Method]	[Observable criterion]	[Record reference]	[Disposition]

Open Issues and Deferred Requirements

Keep unresolved questions and deferred requirements separate from accepted scope. Record the decision needed, the consequence of delay, and the release or review point at which the item will be reconsidered.

Field	Complete with
Issue identifier	[Identifier, affected requirement, description, and source]
Impact	[Affected user, dependency, risk, or verification gap]
Disposition	[Decision needed, responsible role, proposed treatment, and approval reference]
Follow up	[Target release or review point if defined, status, and related records]