# Product Requirements Document (PRD) Template

* [Product Requirements Document (PRD) Template](#Xd88267b06ebf9fe49de197dfaedc6c112dec1c1)
  + [Document Control](#document-control)
  + [Document Overview](#document-overview)
    - [Document Objectives](#document-objectives)
    - [Benefits of Use](#benefits-of-use)
    - [How to Use](#how-to-use)
      * [Best Practices](#best-practices)
    - [Writing Quality Requirements](#writing-quality-requirements)
  + [Executive Summary](#executive-summary)
  + [Background & Problem Statement](#background--problem-statement)
  + [Product Description](#product-description)
  + [Goals & Success Metrics](#goals--success-metrics)
  + [Features](#features)
  + [Technical Architecture and Design Considerations](#X1227f7c4b5273fc8e33a492f323515b82be00ad)
  + [Product Requirements](#product-requirements)
    - [Functional Requirements](#functional-requirements)

## Document Control

**Document Version:**

**Author(s):**

**Date:**

**Status:** Draft / In Review / Final

## Document Overview

This document is intended to capture the product vision, goals, features, and success metrics of a product.

Who is this document for?

* Technical Leadership
* Product Management
* Engineering Teams

### Document Objectives

* Clearly define the product vision and goals to align all team members on what the product aims to achieve.
* Establish a detailed understanding of the problem statement and the target users or market the product is addressing.
* Specify product features and technical requirements, ensuring every aspect of the product is well-documented and understood.
* Outline success metrics and goals to measure product performance, quality, and compliance.
* Provide a structured plan for technical architecture, design considerations, and risk mitigation to guide development.
* Document project timeline, milestones, and open questions to manage progress and address uncertainties.
* Help manage regulatory and compliance requirements specific to the product or industry.

### Benefits of Use

* Clarity and Alignment: Ensures that everyone—from engineers and designers to stakeholders—is on the same page regarding the product’s purpose, functionality, and requirements.
* Improved Communication: Provides a single source of truth, reducing misunderstandings and enabling better collaboration within the team.
* Efficiency and Productivity: Saves time by avoiding repeated clarifications, enables faster decision-making, and helps prioritize work effectively.
* Risk Management: Identifies potential technical and project risks early, allowing teams to develop mitigation strategies upfront.
* Traceability and Accountability: Tracks the evolution of product requirements and decisions, facilitating accountability and effective project management.

This PRD is a **living document** that should be maintained throughout the product development process. It serves as the single source of truth for the team, capturing both the high-level vision and the detailed requirements.

### How to Use

1. **Initial Setup**
   * Complete all relevant sections before starting development.
   * Align with all stakeholders to ensure agreement on product vision, goals, and requirements.
2. **Ongoing Updates**
   * Update the PRD promptly when requirements change or decisions are made.
   * Track document versions so that changes are transparent and traceable.
3. **Collaboration**
   * Encourage contributions from engineering, product, design, QA, and other stakeholders.
   * Use comments or your version control system’s review process for feedback.
4. **Reference During Development**
   * Use the PRD to guide day-to-day decisions and scope discussions.
   * Validate deliverables against the documented acceptance criteria.
5. **Post-Launch Use**
   * Archive the final version alongside release notes for future reference.
   * Use lessons learned to improve the next PRD iteration.

#### Best Practices

* Keep language clear and concise.
* Use diagrams, tables, and checklists when they improve comprehension.
* Avoid ambiguity—ensure requirements are testable and measurable.
* Agree on one owner responsible for maintaining the PRD.

**Remember:** A PRD is most valuable when it is accurate, accessible, and actively used by the whole team.

### Writing Quality Requirements

**Key Characteristics of Quality Requirements:**

* **Clarity and Unambiguity:** Each requirement should express a single, clear concept that cannot be misinterpreted. Avoid ambiguous terms or subjective words like “easy” or “fast.”
* **Completeness:** Requirements should fully capture the intended functionality or quality without missing critical details.
* **Consistency:** Avoid conflicts or contradictory requirements within the document.
* **Verifiability:** A requirement must be measurable or testable through inspection, analysis, demonstration, or test.
* **Feasibility:** Requirements should be technically and economically feasible within project constraints.
* **Necessary:** Include only essential requirements; avoid “nice to have” features that aren’t critical.
* **Single Requirement per Statement:** Avoid “and” connecting multiple requirements; split them for clarity.
* **Traceability:** Each requirement should be traceable to a source such as a stakeholder need or higher-level system goal.

The [Easy Approach to Requirements Syntax (EARS)](https://alistairmavin.com/ears/) website provides great guidance on a structured requirement syntax that helps requirements have a consistent structure.

**Reminders:**

* The goal of writing requirements isn’t to define *how* the product works, but *what* it needs to do.
* Don’t fall into the trap of writing requirements with too much precision.
  + The primary goal is to align the team around building the same product.
* Expect your requirements will change frequently as you refine your understanding of the customer needs and your technology stack.

## Executive Summary

**Product Name:**

**Project Overview:**

(Brief description of the product and its core purpose)

**Key Objectives:**

(Main goals of the product)

**Target Market / Users:**

(Primary users and market segments)

## Background & Problem Statement

**Context:**

(Industry background, technical environment)

**Problem Statement:**

(Key problem or need the product addresses)

## Product Description

**Overview:**

(Basic description of product—hardware/software components)

**Key Differentiators:**

(Main unique selling points)

## Goals & Success Metrics

**Goals:**

(Core, measurable objectives)

**Success Metrics:**

(KPIs like performance, cost, compliance)

## Features

This table lists the main features needed for the product.

| Feature Name | Description | Acceptance Criteria |
| --- | --- | --- |
|  |  |  |

## Technical Architecture and Design Considerations

**High-level Architecture:**

(Summary of system design and key components)

**Key Risks & Mitigations:**

(Primary technical risks and how to address them)

## Product Requirements

Requirements may be organized into subsections under these categories (or others) to improve clarity, traceability, and ease of review. Teams can adapt or combine categories as needed for the project’s size, complexity, and domain.

* **Functional:** Defines what the system must do, its specific behaviors or services.
* **Non-Functional:** Specifies system qualities like performance, usability, reliability, processes, or maintainability.
* **Safety:** Ensures the system operates without causing unacceptable risks to people or the environment.
* **Environmental:** Addresses operational conditions like temperature, humidity, vibration, or interference.
* **Cybersecurity:** Protects against unauthorized access and ensures data confidentiality, integrity, and availability.
* **Communication:** Defines protocols, data formats, and connectivity for system interactions.
* **Regulatory/Compliance:** Ensures adherence to relevant laws, standards, and certifications.
* **Design Requirements:** Specifications for a product’s shape, aesthetics, ergonomics, materials, and user experience.
* **Manufacturing Requirements:** Specifications for production processes, assembly, quality, materials, and compliance.

The following fields are used when writing requirements:

* **ID** – Unique identifier for the requirement (e.g., REQ-001), ensuring traceability and easy referencing.
* **Title** – Short, descriptive name summarizing the requirement.
* **Description** – A clear, concise, and unambiguous statement detailing what the requirement entails.
* **Acceptance Criteria** – Measurable conditions or tests that must be satisfied for the requirement to be considered complete.
* **Source** – The origin of the requirement, such as the stakeholder, meeting, document, or regulation from which it came.

Example Requirements Table:

| ID | Title | Description | Acceptance Criteria | Source |
| --- | --- | --- | --- | --- |
| REQ-001 | User Login | The system shall allow users to log in using email and password | Tested with valid and invalid credentials | Stakeholder Meeting |
| REQ-002 | Password Reset | Users shall be able to reset their password via email | Email with reset link sent on request | Product Manager |
| REQ-003 | Dashboard View | The dashboard shall display user statistics in real time | Statistics update within 5 seconds | Project Charter |

### Functional Requirements

| ID | Title | Description | Acceptance Criteria | Source |
| --- | --- | --- | --- | --- |
| REQ-FUNC-001 |  | <Clear, concise, unambiguous statement> |  |  |
| REQ-FUNC-002 |  |  |  |  |