Product Requirements

**Product Name**

**List Group Member Names**

**Revision Number**

**Revision Date**

Table of Contents

[Introduction 3](#_Toc219798923)

[Product Overview 3](#_Toc219798924)

[Product Functional Requirements 4](#_Toc219798925)

[Product Non-Functional Requirements 4](#_Toc219798926)

[Usability Requirements 5](#_Toc219798927)

[Reliability Requirements 5](#_Toc219798928)

[Portability Requirements 5](#_Toc219798929)

[Interoperability Requirements 5](#_Toc219798930)

[Delivery Requirements 6](#_Toc219798931)

[Implementation Requirements 6](#_Toc219798932)

[External Requirements 6](#_Toc219798933)

[Standards Requirements 7](#_Toc219798934)

[Legislative Requirements 7](#_Toc219798935)

[Privacy Requirements 7](#_Toc219798936)

[Safety Requirements 7](#_Toc219798937)

[Ethical Requirements 7](#_Toc219798938)

[System Architecture Requirements 8](#_Toc219798939)

[System Model Requirements 8](#_Toc219798940)

[System Evolution Requirements 8](#_Toc219798941)

# Introduction

This is often just copied from the Project Plan.

# Product Overview

This is a prose one or two paragraph description of the product.

# Product Functional Requirements

This should be a detailed list of all features that will be incorporated in the product. It is critical for this section be complete. It establishes the doneness criteria for the product development project. You know when the product development is complete when the product successfully has every feature listed. The product should not have any feature no listed here.

This section’s completeness is also critical for the following reasons.

1. In some organizations, it serves as a legal agreement between the vendor and customer.
2. It drives the outline of the Design Document. For every feature listed there will be a section in the design document that describes the design and implementation of that feature.
3. It drives the outline of the Test Plan. For every feature listed there will be an explicit test to ensure the feature works correctly.

# Product Non-Functional Requirements

This is similar to the previous section. Is a list of features that will not be in the product. This does not list ever feature the product will not have. It should list features that a similar type product might have but were deliberately omitted and specific (typical) error conditions that cannot happen. Example: a calculator application should check for and prevent an actual divide by 0 from succeeding.

There should also be tests in the Test Plan to ensure these features cannot happen.

# Usability Requirements

Who should be able to install and use this product? Are there any age expectations? Does the user need to poses any skills to use this product?

# Reliability Requirements

How often would you expect this product fail when used by a typical user in a supported working configuration? This is usually based on failure rate analysis.

# Portability Requirements

Is this product expected to work on more than one platform? Example: if writing the code in C++ should the source files be expected to compile on different platforms so the application can run on different platforms?

# Interoperability Requirements

This is more oriented to network applications where cooperating parts of the application need to work on different or multiple platforms.

# Delivery Requirements

How is the product to be delivered to the end user?

# Implementation Requirements

What hardware and software will the customer need to successfully install and run this application?

# External Requirements

Are there any external requirements beyond the capacity of the product needed to use this application? Example, not just access to the Internet but access to a specific site on the Internet. The user’s system may be working, the Internet is working, but since the specific sever or service is down the application will not work.

# Standards Requirements

Are there any government, industry or organizational (usually published and well known) standards that the design and implementation of this application must comply?

# Legislative Requirements

This section addresses the same issues as the previous section except these requirements are legal in nature. As an example are HIPAA regulations.

# Privacy Requirements

Although not legal in nature, is this product expected to keep user information private?

# Safety Requirements

Are there any safety requirements needed to ensure the product cannot hurt someone?

# Ethical Requirements

This is open ended because it often addresses moral issues as well. AS an example, if this is a web server, should it prevent the distribution of pornography, assuming you can define pornography?

# System Architecture Requirements

Is this product expected to work on new platforms or architectures not created yet? If the application is portable and the new platform implements a working C++ compiler, should the product be expected to work?

# System Model Requirements

This section addresses the same issues as the last section for new models in existing architecture/platform families.

# System Evolution Requirements

This section addresses the same issues as the previous sections but covers when a system under goes upgrades in hardware or software including the operating system.