A Template for A Final Project in Software Engineering Application Requirements Document[1]

November 17, 2014

[1]Derived from a previous documents authored by *Dr. Eliezer Kaplansky* and *Dr. Mayer Goldberg.*

Abstract

This document describes the requirements of the software project *from the standpoint of the client.*

Contents

Guidelines

This document formalizes your understanding what your customer has requested. In writing this document, you should maintain a global view of the proposed software, its purpose, use, and relation to the rest of the world. Avoid low-level details; Those will go in the application desing document (ADD).

You should analyze the project description and the requirements that the customer expressed and come up with your own suggestions for additional and/or improved and/or more general functionality. Do not worry that you will need to implement all these extensions. After the application design document (ARD) is submitted and accepted, you will define what functionality needs to be implemented in each iteration and what functionality will be reserved for future versions. The ARD should be updated at each iteration of the project, as your understanding of the client requirements evolves.

Chapter 1

Introduction

This chapter provides an overview of the entire requirements document. The goal is to provide the "big picture" into which your software projects should fit. The following sections document the context of your software system and how it should interact with the rest of the world.

1.1 Vision

Describe the overall goals and objectives of your software project.

1.2 The Problem Domain

Describe in detail the problem domain in which your system operates. Include a block diagram showing the major components of the larger system, how they inter-connect, and what external interfaces are maintained.

1.3 Stakeholders

Who are the people that are relevant to the product? Who are your customers? Who has a say on the design?

1.4 Software Context

Provide a high-level description of the software you are planning. Describe the major inputs, functionality, processing, and outputs. Omit implementation and low-level details. Describe several use-cases of common usage scenarios that illustrate how users, the environment, and other systems will interact with your software.

Chapter 2

Functional Requirements

- Describe each of the major functions to be handled by the software.
- By *functions* we mean the capabilities and functionality of the software, itemized as individual operations. We do not mean functions in the programming languages sense of program units.
- The description of the functional requirements should be understandable to your customer, and indeed, to anyone else reading the document for the first time.
- It will be helpful to organize the functions into some hierarchical taxonomy.
- You may use charts of various kinds (e.g., use-case and sequence diagrams) to help the reader grasp the logical relationship between actors, functions, and data.
- Make sure you quantify your requirements as much as possible (see example bellow).

Chapter 3

Non-functional requirements

- Describe any constraints that effects the way your software is to be specified, designed, implemented, tested, etc.
- These constraints may be related to the nature of the business of your client, the nature of the product line, etc.
- Make sure you quantify these requirements so that they are measurable, demonstrable and verifiable.
- Here are three examples of how to state a requirement, the first acceptable, the second unacceptable, and the third even worse:

  Acceptable: 95% of all transactions will complete in under one second.

  Unacceptable: Transactions should complete quickly.

  Worse: The user should not have to wait for a transaction to complete.
- See the next section for constraint domains; Feel free to delete some and/or add others.

3.1 Constraints Domains

Speed, Capacity & Throughput: Quantify the performance requirements for your software.

Reliability: Describe and quantify the factors that effect the reliability of the software.

Safety & Security: Is confidentiality a constraint in your system? Does your data need to be encrypted? Are different views of your data accessible to users with different security/authority levels? Etc.

Portability: Should your system be deployable from different operating systems? If your system is a web service, does your system depend on a specific browser? Should your system support text in different languages and character sets (e.g., using Unicode)? Etc.

Reliability: Is your system required to withstand certain hardware, software, network failures? Is your system required to support data recovery, error-correction, etc?

Usability: What is the level of expertise that the users of your systems are expected to have? What is the level of training required to begin using the application?

Availability: Describe and quantify the level of availability required of the system. Describe what factors bear on the availability of the system.

Platform: Describe any constraints that limit/restrict your choice of development tools, software packages, platforms, etc. E.g., some projects may require you to use Visual Basic and ASP, restrict you to .NET, etc.

Demo: You will need to give a demo of your system. Is the system interactive? Where will its inputs come from? If its inputs arrive naturally from some device, will you have access to this device? Will you have to simulate the device? Will you use random data or samples of actual data in your simulation?

Environment: If the software requires special hardware, special operating conditions, special access rights, access to proprietary or secret data, etc, how and where will you develop the software? How will you test it? How will you present the final system?

Other: Describe any additional constraints not covered by previous sections. For example, users with specific disabilities, conformance to various standards, etc.

Chapter 4

Usage Scenarios

In this chapter you will describe the main usage scenarios of your software. You are to distill these scenarios from the information collected during the requirements elicitation process.[1]

4.1 User Profiles — The Actors

Describe the profiles of the main user categories. Describe the relevant characteristics of the intended users of the system. Two things to keep in mind about actors are that

- Actors are external to the system
- Actors exchange data with the system

Humans that use your system are certainly good candidates for actors. Actors can be non-human, but keep in mind that actors must be external to your system, and another way of saying this is that *actors begin where the system ends.* A reasonable candidate for a non-human actor could be, e.g., a radioactivity sensor that triggers some emergency system.

4.2 Use-cases

Describe the main use-cases for the software.

4.3 Special usage considerations

Describe any special requirements associated with the use of this software.

[1] **Hint:** Think about the relationship between the usage scenarios and the test cases you will use to test your system.

Chapter 5

Plan for Software Iteration One & Risk assessment

In this chapter you will plan the next step of the project in which you will develop in the first software iteration.

The main objective of iteration One is to build a partial version of the solution to analyze risks, get feedback from the customer, determine the solution to technical requirements and improve the requirements definition. Some questions you may deal with are: how two systems might be integrated; can a certain throughput be achieved with a given configuration etc.

The first iteration can be a partial solution that involves a relatively small number of functionality and scope to establish whether the system satisfies some aspect of the requirements and to test various implementation options. It is also meant for you to get acquainted with the technologies and form an opinion towards using them in the next iterations. Specifically, when describing the planned iteration (and when presenting it later), make sure to convince the reader how it is expected to give:

- Better understanding of requirements.
- Understanding of the capabilities and limitations of new technologies.
- Ability to assess design decisions early in the process.
- Ability for the customer to visualize the look-and-feel of the solution.
- And, most impotently: **reduction in the risk of project failure.**

In this chapter you will also assess the main risk factors in your project. You should supply a table identify the risk items and for each such item, the way you are going to mitigate it.

Appendices

Include any information that is relevant to the project, and that supplements the requirements specification. Such information includes:

- I/O format information, such as file formats, etc.
- Reports of cost analysis studies, user surveys, descriptions and/or surveys of similar products, etc.
- Supporting and/or background information to help readers of this ARD.
- Glossary: List of all the domain- and technical terms with a brief definition.

Your client is a likely source for such information.