Aline Koftikian

Project Deliverable 1

Section 7.2 – Artifacts

1.2 Project Description

The project description is an introduction to the entire project. It’s a brief that allows the reader and the user of the system to understand the goal and purpose of the project. This document must be presented as an opening to the rest of the manual, and must include the idea, objective, background, approach, and result of the entire project.

* + 1. Functional Requirements

Requirements are the tasks that are necessary for the completion of the system. They are the components of the system that must be documented, measured and tested in order to ensure the ideal conclusion to the project in question. The requirements that need to be specified for this system are functional and non-functional requirements. The functional requirements are the tasks that define what the system is trying to accomplish. These are the elementary actions that constitute the main goal of the system. They include the user viewing weekly schedule, adding classes, dropping classes, swapping, etc.

Each individual requirement must be presented as a use case. A use case is a description of the behavior of a system that explains the process and results obtained through interactions of the user with the system. The functional requirements must be known before writing use cases, because they are the details that compose the main actions and tasks of the system. The use cases are present in both list form and diagram form.

* List Use Cases: The list-based use cases are full formal documents that list all use cases of the system in detail, with each property of the use case explained in full detail. These include the name of the use case, the general description, the actors, preconditions, postconditions, etc.
* Diagram Use Cases: Use case diagrams are used to depict a visual representation of the use cases listed. The actors, which typically include the user or an external entity, are depicted using stick figures. The use cases themselves are drawn as circles with a name that describes the title of the action that can be initiated by the actors. There are lines connected from users to the use cases called connections, which describe the actor’s interactions with those use cases.

1.3.2. Domain Model

The Domain Model is the model that describes all core concepts and an overview of the system at hand. Its main purpose is to list the objects of interest that form the system, and presents their attributes, constraints, and relationships between them. The model is designed in UML format, with arrows describing relationships from one object to another, and ensuring that there are multiplicity values at both ends of the arrow denoting multiplicity factors of the relationship. The finalized domain model will come with a description that will provide a more detailed explanation of the model as well as each object present in the model.

1.3.3. Constraints and Qualities

The constraints and qualities segment describes the main standards of quality and non-functional requirements that are expected to be met by the system. The non-functional requirements are the general properties of the system. They are the constraints and qualities that will make up the system when in use. These include response times, maintenance times, security, etc. It’s important that the constraints and qualities are as specific as possible, which means they must have concrete metric specifications wherever possible.

1.4.1 Human Resources

The human resources of a project is the group of individuals who compose the main workforce of the project. They are the source of labor of a project, and in the case of large projects, are typically divided into smaller groups for more organized work.

1.4.2. Technical Resources

The technical resources of a project are the computer resources that will be used to complete the project. This includes the computer programs, software, libraries, and websites that are available for use.

1.5. Scoping

The scoping section describes all elements, features, qualities, and goals that have been scoped out from the end result. Each scoped element must come with a description of what it is, in addition to a paragraph that describes why it was chosen to be scoped out. These scoped elements have been typically scoped out in the early phases of a project, where the requirements are being planned and discussed in between team members.

1.6.1. Architecture

The architecture of a project is the high-level system architecture that is planned for the complete project. It must be presented with a UML class diagram that describes the details and relationships between different components of the system, along with a detailed description of the mindset behind the architecture (also known as a design rationale). In addition, each module of the architecture must be accompanied with its own design rationale and responsibilities.

1.6.2. Technologies in Use

This segment must provide a technical description of all the technologies that will be used to build the project. These include programming languages, libraries, servers, databases, IDE and compilers, etc. For each technology listed, there must be a short description that explains the rational for the choice and use of the specific tool.

1.7.1 Activities

The activities section is a section that will list a step-by-step process of how the separate artifacts of the project will be produced in order to reach the completion of the project. Each activity must be presented with a clear purpose, description, and must produce at least one artifact in consequence.

1.7.2. Artifacts

An artifact is a by-product that is produced as a consequence of the development of a software project. The artifacts segment will list each document that will be produced in the project. Each artifact listed must come with a description that provides details about the role of that specific document in the completion of the project.

1.7.3. Project Estimates

1.7.4. Activities Assignments

1.7.5. Schedule

1.7.6. Risk

1.8. Prototyping

2.2. Introduction to Part 2

2.3.1. Architecture Diagram

2.3.2. Subsystem Interfaces Specification

2.4.1. Detailed Design Diagram

2.4.2. Unit Description

2.5. Dynamic Design Scenario

2.6. Estimation

2.7. Rapid Prototyping and Risk

3.2. Introduction

3.3.1.1. Tested Items

3.3.1.2. Untested Items

3.3.2.1. Unit Testing

3.3.2.2. Requirements Testing

3.3.2.3. Stress Testing

3.3.2.4. Security Testing

3.4.1. Installation Manual

3.4.2. User’s Manual

3.5. Final Cost Estimate

The architectural design is a high-level description of the architecture of the system.