Draft X

September 23rd, 2018

[ Paste Your Organization’s Logo Here ]

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.  *CxPattern\_RevisionHistory* provides details on CxOne’s recommended way to handle document revisions. | 00/00/00 |

**The paragraphs written in the “Comment” style are for the benefit of the person writing the document and should be removed before the document is finalized.**

This template can be used to create Software Requirements Specifications that conform to IEEE Standard 830-1993.

An SRS is a tool for capturing requirements on a project; it is the epitome of “plain language requirements”. Although an SRS is designed to stand on its own, most projects will employ additional tools for capturing requirements. An SRS may be ancillary or unnecessary on many projects, but a partial or lightweight one may be quite useful even if other techniques are being used to capture the bulk of the requirements.

Consider using an SRS when:

* Modeling techniques need to be augmented
* Plain language is the best mechanism for capturing system behavior
* Requirements must be traceable
* Required by regulations

See CxGuide\_CxOneArtifact for details on how to utilize the advanced features of CxOne artifact templates.

Contents

[1 Introduction 1](#_Toc520006015)

[1.1 Purpose 1](#_Toc520006016)

[1.2 Scope 1](#_Toc520006017)

[1.3 Definitions, Acronyms, and Abbreviations 1](#_Toc520006018)

[1.4 References 1](#_Toc520006019)

[1.5 Overview 1](#_Toc520006020)

[2 Overall Description 2](#_Toc520006021)

[2.1 Product Perspective 2](#_Toc520006022)

[2.2 Product Functions 3](#_Toc520006031)

[2.3 User Characteristics 3](#_Toc520006032)

[2.4 Constraints 3](#_Toc520006033)

[2.5 Assumptions and Dependencies 4](#_Toc520006034)

[2.6 Apportioning of Requirements 4](#_Toc520006035)

[3 Specific Requirements 5](#_Toc520006036)

[3.1 over all game description 5](#_Toc520006037)

Provide the game map here

Provide the game description here

[3.1.1 Rooms table 6](#_Toc520006038)

[3.1.2 mostroes table 6](#_Toc520006039)

[3.1.3 Puzzels table 6](#_Toc520006040)

[3.1.4 Artifacts table 6](#_Toc520006041)

*3.1.5.Command menu table………………………………………………………………………..6*

*3.1.6. Fight Management System………………………………………………………………….6*

*3.1.7. Navigation System…………………………………………………………………………….6*

*3.1.8 Anything else related to your game design …………………………………………………6*

[3.2 Software Product Features (List all game features) 6](#_Toc520006042)

[3.2.1 Feature e.g(Rooms, Items, Monster, Puzzle, etc) 6](#_Toc520006043)

[Purpose 6](#_Toc520006044)

Use case diagram……………………………………………………………………………….6

[Associated Functional Requirements 6](#_Toc520006046)

[3.3 Software System Attributes (Non Funcationl Requierments) 7](#_Toc520006049)

[3.5.1 Reliability 7](#_Toc520006050)

[3.5.2 Availability 7](#_Toc520006051)

[3.5.3 Security 7](#_Toc520006052)

[3.5.4 Maintainability 7](#_Toc520006053)

[3.6 Logical Database Requirements (if you are using database) 7](#_Toc520006054)

[3.7 Other Requirements 8](#_Toc520006055)

Introduction

This section should provide an overview of the entire document.

## Purpose

Describe the purpose of this specification and its intended audience.

## Scope

Identify the software product(s) to be produced by name. Explain what the products will and will not do. Describe how the software will be used, and identify relevant benefits, objectives, and goals.

## Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

## References

List all the documents and other materials referenced in this document. This section is like the bibliography in a published book.

## Overview

Describe the content and organization of the rest of this document.

# Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## Product Perspective

This section should place the product in perspective with other related products. If the product is independent and self-contained, state that here. Otherwise, identify interfaces between the product and related systems.

## Product Functions

Provide a summary of the major functions that the software will perform.

## User Characteristics

Describe the general characteristics of the intended users and how would they use the system differently.

## Constraints

Describe any other items that will constrain the design options, including

* reliability requirements
* criticality of the application
* safety and security considerations regulatory policies
* hardware limitations
* interfaces to other applications
* parallel operation
* audit functions
* control functions
* higher-order language requirements
* signal handshake protocols
* etc.

## Assumptions and Dependencies (Non Applicable)

List factors that affect the requirements. These factors are not design constraints, but areas where future changes might drive change in the requirements.

## Apportioning of Requirements (Non Applicable)

Identify any requirements that may be delayed until future versions of the system.

# Specific Requirements

This section should describe all software requirements at a sufficient level of detail for designers to design a system satisfying the requirements and testers to verity that the system satisfies requirements.

The remainder of this sample document is organized according to A.5 Template of SRS Section 3 Organized by Feature shown in the Annex of Std 830-1993. For alternative organizational schemes by system mode, user class, object, stimulus, functional hierarchy, and combinations, see the standard.

## Over All Game Description

Provide a detailed description of your game to the implementing team including:

Provide the game map here

### Rooms

Provide a detailed description to the 30 rooms you have in your game including exists etc.…

### Monsters

Provide a detailed description to the monsters you have in your game etc.…

### Puzzles

Provide a detailed description to the puzzles and their solution

### Artifacts

Provide a detailed description to Artifacts in your game

### User Interfaces

Briefly provide a description to standard menus you have in your game such as navigation menus buttons, directions etc.…

## Software Product Features

### Feature 1

Repeat subsections at this level and below for each feature.

#### Use Case Diagram

#### Purpose

#### Associated Functional Requirements

##### Functional Requirement 1

Repeat subsections at this level and below for each associated functional requirement.

Each functional requirement may be described in natural language using the in four subsections as follows. Functional requirements include:

###### ID: Each functional requirement should have a unique ID

###### DESC: describe the function requirements

###### DEP: Dependency between this functional requirement and any other functional requirements (user the requirement ID to describe dependency)

## Software System Attributes

The following items provide a partial list of system attributes that can serve as requirements that should be objectively verified.

Other possible options include scalability, portability, robustness, recoverability, etc.

### Reliability

### Availability

### Security

### Maintainability

Specify attributes of the software that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

## Logical Database Requirements

Specify the requirements for any information that is to be placed into a database, including

* types of information used by various functions
* frequency of use
* accessing capabilities
* data entities and their relationships
* integrity constraints
* data retention requirements

## Other Requirements