Draft X

July 17, 2001

[ 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)

[2.1.1 System Interfaces 2](#_Toc520006023)

[2.1.2 User Interfaces 2](#_Toc520006024)

[2.1.3 Hardware Interfaces 2](#_Toc520006025)

[2.1.4 Software Interfaces 2](#_Toc520006026)

[2.1.5 Communications Interfaces 2](#_Toc520006027)

[2.1.6 Memory Constraints 3](#_Toc520006028)

[2.1.7 Operations 3](#_Toc520006029)

[2.1.8 Site Adaptation Requirements 3](#_Toc520006030)

[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 External Interface Requirements 5](#_Toc520006037)

[3.1.1 User Interfaces 6](#_Toc520006038)

[3.1.2 Hardware Interfaces 6](#_Toc520006039)

[3.1.3 Software Interfaces 6](#_Toc520006040)

[3.1.4 Communications Interfaces 6](#_Toc520006041)

[3.2 Software Product Features 6](#_Toc520006042)

[3.2.1 Feature 1 6](#_Toc520006043)

[Purpose 6](#_Toc520006044)

[Stimulus/Response Sequence 6](#_Toc520006045)

[Associated Functional Requirements 6](#_Toc520006046)

[3.3 Performance Requirements 6](#_Toc520006047)

[3.4 Design Constraints 7](#_Toc520006048)

[3.5 Software System Attributes 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 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

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements.

## 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.

### System Interfaces

List each system interface and identify the related functionality of the product.

### User Interfaces

Specify the logical characteristics of each interface between the software product and its users (e.g., required screen formats, report layouts, menu structures, or function keys).

Specify all the aspects of optimizing the interface with the person who must use the system (e.g., required functionality to provide long or short error messages). This could be a list of do’s and don’ts describing how the system will appear to the user.

### Hardware Interfaces

Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols.

### Software Interfaces

Specify the use of other required software products (e.g., a DBMS or operating system), and interfaces with other application systems.

For each required software product, provide identification information including at least name, version number, and source.

For each interface, discuss the purpose of the interfacing software, and define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

### Communications Interfaces

Specify any interfaces to communications such as local area networks, etc.

### Memory Constraints

Specify any applicable characteristics and limits on RAM, disk space, etc.

### Operations

Specify any normal and special operations required by the user, including:

* periods of interactive operations and periods of unattended operations
* data processing support functions
* backup and recovery operations
* etc.

### Site Adaptation Requirements

Define requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode. Specify features that should be modified to adapt the software to a particular installation.

## Product Functions

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

## User Characteristics

Describe the general characteristics of the intended users, including

* educational level
* experience
* technical expertise

## 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

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

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.

## External Interface Requirements

Provide a detailed description of all inputs into and outputs from the software. This section should complement the interface descriptions under section 2.1 and should not repeat information there. Include both content and format as follows:

* name of item
* description of purpose
* source of input or destination of output
* valid range, accuracy, and/or tolerance
* units of measure
* timing
* relationships to other inputs/outputs
* screen formats/organization
* window formats/organization
* data formats
* command formats
* end messages

These requirements may be organized in the following subsections.

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communications Interfaces

## Software Product Features

### Feature 1

Repeat subsections at this level and below for each feature.

#### Purpose

#### Stimulus/Response Sequence

#### 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, pseudocode, or in four subsections as follows. Functional requirements include:

* validity checks on inputs
* exact sequencing of operations
* responses to abnormal situations, including error handling and recovery
* effects of parameters
* relationships of inputs to outputs, including input/output sequences and formulas for input to output conversion

###### Introduction

###### Inputs

###### Processing

###### Outputs

## Performance Requirements

Specify static and dynamic numerical requirements placed on the software or on human interaction with the software.

Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.

Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

## Design Constraints

Specify requirements imposed by standards, hardware limitations, etc.

## 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

Specify the factors that will protect the software from accidental or malicious access, misuse, or modification. These factors may include:

* cryptography
* activity logging
* restrictions on intermodule communications
* data integrity checks

### 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