ISO/IEC/IEEE 29148:2018

**Software Requirements Specification**

# For **<Project Name>**

Version 0.1  
Author: **Dr. Somiya Rani**  
Date: **2025-06-06**

**Note**: *This SRS template is intended for software projects requiring a formal requirements specification. Replace the guidance text in each section with the actual text. Optional sections are 1.3.6 and 4. You may remove these or other sections if not applicable.*

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 0.1 | 2025-06-06 | Dr. Somiya Rani | Initial Draft |
|  |  |  |  |
|  |  |  |  |

# 

# 

# 

# 

# 

# 

# 

# 

# Table of Contents

[**1. Introduction 3**](#_cfk0m09z23k7)

[1.1 Purpose 3](#_nydhar14r1bo)

[1.2 Scope 3](#_j7yxi8nzjqbp)

[1.3 Product overview 3](#_apav1ecxqck2)

[1.3.1 Product perspective 3](#_qqi4v9dmm6dj)

[1.3.1.1 System interfaces 4](#_h8tomzl78s9f)

[1.3.1.2 User interfaces 4](#_k0ca1zxsaf03)

[1.3.1.3 Hardware interfaces 4](#_7i2302w64jql)

[1.3.1.4 Software interfaces 4](#_pcbwfca99jf)

[1.3.1.5 Communications interfaces 4](#_uuezyfapa6dy)

[1.3.1.6 Memory constraints 5](#_8408ot31xafe)

[1.3.1.7 Operations 5](#_b5vmktk6wf7f)

[1.3.1.8 Site adaptation requirements 5](#_nykkt8deftb)

[1.3.1.9 Interfaces with services 5](#_lta901hba74o)

[1.3.2 Product functions 5](#_w8j05bpbeek1)

[1.3.3 User characteristics 6](#_gg63rhw837ii)

[1.3.4 Limitations 6](#_fs63hr7ghhpy)

[1.3.5 Assumptions and dependencies 6](#_ec0jf4a7d0by)

[1.3.6 Apportioning of requirements 6](#_xdk8739009t9)

[1.4 Definitions 7](#_t3swfg8r7qsd)

[1.5 Acronyms and abbreviations 7](#_75905cajjuux)

[**2. Specified requirements 7**](#_l96n17e3zgq6)

[2.1 External interfaces 7](#_ag0v2rwtzq8p)

[2.2 Functions 8](#_3iqxhj1nsna)

[2.3 Usability requirements 8](#_vld604eucipl)

[2.4 Performance requirements 8](#_wlpyk9r9eo90)

[2.5 Logical database requirements 9](#_18343xo5lu6p)

[2.6 Design constraints 9](#_bokv58zfx8e0)

[2.7 Standards compliance 9](#_mge255rqee34)

[2.8 Software system attributes 9](#_sjy4dmnadyqj)

[**3. Verification 10**](#_w9gukg86lb4u)

[**4. Supporting information 10**](#_gidczgcjv0oa)

[**5. References 11**](#_w33n8iyyv691)

# **1. Introduction**

## 1.1 Purpose

Delineate the purpose of the software to be specified.

## 1.2 Scope

Describe the scope of the software under consideration by:

a) Identifying the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.).

b) Explaining what the software product(s) will do.

c) Describing the application of the software being specified, including relevant benefits, objectives, and goals, and

d) Being consistent with similar statements in higher-level specifications (e.g., a system requirements specification), if they exist.

## 1.3 Product overview

### 1.3.1 Product perspective

Define the system's relationship to other related products.

If the product is an element of a larger system, relate the requirements of that larger system to the functionality of the product covered by the SRS.

If the product is an element of a larger system, identify the interfaces between the product covered by the SRS and the larger system of which the product is an element.

Consider a block diagram showing the major elements of the larger system, interconnections, and external interfaces.

Describe how the software operates within the following constraints:

a) System interfaces

b) User interfaces

c) Hardware interfaces

d) Software interfaces

e) Communications interfaces

f) Memory

g) Operations

h) Site adaptation requirements

i) Interfaces with services

#### 1.3.1.1 System interfaces

List each system interface and identify the functionality of the software to accomplish the system requirement, and the interface description to match the system.

#### 1.3.1.2 User interfaces

Specify the logical characteristics of each interface between the software product and its users.

**Note**: *A style guide for the user interface can provide consistent rules for organization, coding, and user interaction with the system.*

#### 1.3.1.3 Hardware interfaces

Specify the logical characteristics of each interface between the software product and the hardware elements of the system. This includes configuration characteristics (number of ports, instruction sets, etc.). It also covers such matters as what devices are to be supported, how they are to be supported, and protocols. For example, terminal support may specify full-screen support as opposed to line-by-line support.

#### 1.3.1.4 Software interfaces

Specify the use of other required software products (e.g., a data management system, an operating system, or a mathematical package), and interfaces with other application systems (e.g., the linkage between an accounts receivable system and a general ledger system).

For each required software product, specify:

a) Name

b) Mnemonic

c) Specification number

d) Version number

e) Source

**Note**: *It is acceptable to specify required platforms or operating systems, but rarely feasible to require a specific version. Typically, a version number most recent version or any currently maintained version can be specified for software.*

For each interface, specify:

a) Discussion of the purpose of the interfacing software as related to this software product.

b) Definition of the interface in terms of message content and format. It is not necessary to detail any well-documented interface, but a reference to the document defining the interface is required.

#### 1.3.1.5 Communications interfaces

Specify the various interfaces to communications, such as local network protocols.

#### 1.3.1.6 Memory constraints

Specify any applicable characteristics and limits on primary and secondary memory.

#### 1.3.1.7 Operations

Specify the normal and special operations required by the user, such as:

a) the various modes of operations in the user organization (e.g., user-initiated operations)

b) periods of interactive operations and periods of unattended operations

c) data processing support functions

d) backup and recovery operations

**Note**: *This is sometimes specified as part of the User Interfaces section.*

#### 1.3.1.8 Site adaptation requirements

The site adaptation requirements include:

a) Definition of the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode (e.g., grid values, safety limits, etc.).

b) Specification of the site or mission-related features that should be modified to adapt the software to a particular installation.

#### 1.3.1.9 Interfaces with services

Specify interactions with services, e.g., Software as a Service (SaaS) or cloud services.

API services (e.g., REST/GraphQL), cloud providers (e.g., AWS S3, Azure Functions), and authentication services (e.g., OAuth, OpenID).

### 1.3.2 Product functions

Provide a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statements, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.

Sometimes the function summary necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product.

Use cases, user stories, and scenarios are also used to describe product functions.

**Note**: *For the sake of clarity:*

*a) The product functions should be organized to make the list of functions understandable to the acquirer or anyone else reading the document for the first time.*

*b) Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.*

### 1.3.3 User characteristics

Describe those general characteristics of the intended groups of users of the product, including characteristics that may influence usability, such as educational level, experience, disabilities, and technical expertise. This description should not state specific requirements, but rather should state the reasons why certain specific requirements are later specified in specific requirements in 2.0.

**Note**: *For additional information on the context of use and user needs, see ISO/IEC 25063 and ISO/IEC 25064.*

### 1.3.4 Limitations

Provide a general description of any other items that will limit the supplier's options, including:

a) Regulatory requirements and policies

b) Hardware limitations (e.g., signal timing requirements)

c) Interfaces to other applications

d) Parallel operation

e) Audit functions

f) Control functions

g) Higher-order language requirements

h) Signal handshake protocols (e.g., XON-XOFF, ACK-NACK)

i) Quality requirements (e.g., reliability)

j) Criticality of the application

k) Safety and security considerations

l) Physical/mental considerations

m) Limitations that are sourced from other systems, including real-time requirements from the controlled system through interfaces

### 1.3.5 Assumptions and dependencies

List each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software, but any changes to these factors can affect the requirements in the SRS. For example, an assumption may be that a specific operating system will be available on the hardware designated for the software product. If the operating system is not available, the SRS would have to change accordingly.

### 1.3.6 Apportioning of requirements

Apportion the software requirements to software elements. For requirements that will require implementation over multiple software elements, or when allocation to a software element is initially undefined, this should be so stated. A cross-reference table by function and software element should be used to summarize the apportionments. Identify requirements that may be delayed until future versions of the system (e.g., blocks and/or increments).

## 1.4 Definitions

Provide definitions for any words or phrases that have special meaning beyond normal dictionary definitions.

## 1.5 Acronyms and abbreviations

Spell out or define all acronyms and abbreviations used in the documents.

# 2. Specified requirements

Specify the software system requirements to a level of detail sufficient for software design, development, and verification of the software increment or release in process.

The requirements should:

a) be stated in conformance with all the characteristics described in ISO/IEC/IEEE 29148:2018 Section 5.2.

b) Be cross-referenced to earlier versions or related documents.

c) Be uniquely identifiable (e.g., REQ-001, REQ-002).

d) Describe every input (stimulus) into the software system, every output (response) from the software system, and all functions performed by the software system in response to an input or in support of an output.

## 2.1 External interfaces

Define all inputs into and outputs from the software system. The description should complement the interface descriptions in 1.3.1.1 through 1.3.1.5, and should not repeat information there.

Each interface defined should include the following content:

a) Name of item

b) Description of purpose

c) Source of input or destination of output

d) Valid range, accuracy, and/or tolerance

e) Units of measure

f) Timing

g) Relationships to other inputs/outputs

h) Data formats

i) Command formats

j) Data items or information included in the input and output

## 2.2 Functions

Define the fundamental unique actions that have to take place in the software in accepting and processing the inputs and in processing and generating the outputs, including:

a) Validity checks on the inputs.

b) Exact sequence of operations.

c) Responses to abnormal situations, including:

1) Overflow

2) Communication facilities

3) Hardware faults and failures

4) Error handling and recovery

d) Effect of parameters

e) Relationship of outputs to inputs, including:

1) Input/output sequences

2) Formulas for input-to-output conversion

It may be appropriate to partition the functional requirements into sub-functions or sub-processes. This does not imply that the software design will also be partitioned that way.

## 2.3 Usability requirements

Define usability and quality in use requirements and objectives for the software system that can include measurable effectiveness, efficiency, satisfaction criteria, and avoidance of harm that could arise from use in specific contexts of use.

**Note**: *Additional guidance on usability requirements can be found in ISO/IEC TR 25060.*

## 2.4 Performance requirements

Specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole.

Static numerical requirements may include the following:

a) The number of terminals to be supported

b) The number of simultaneous users to be supported

c) The amount and type of information to be handled

Static numerical requirements are sometimes identified under a separate section entitled Capacity.

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

The performance requirements should be stated in measurable terms.

For example,

95 % of the transactions shall be processed in less than 1s.

Rather than that, an operator shall not have to wait for the transaction to complete.

**Note**: *Numerical limits applied to one specific function are normally specified as part of the processing subparagraph description.*

## 2.5 Logical database requirements

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

a) Types of information used by various functions

b) Frequency of use

c) Accessing capabilities

d) Data entities and their relationships

e) Integrity constraints

f) Security

g) Data retention requirements

## 2.6 Design constraints

Specify constraints on the system design imposed by external standards, regulatory requirements, or project limitations.

## 2.7 Standards compliance

Specify the requirements derived from existing standards or regulations, including:

a) Report format

b) Data naming

c) Accounting procedures

d) Audit tracing

For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database shall be recorded in a trace file with before and after values.

## 2.8 Software system attributes

Specify the required attributes of the software product. The following is a partial list of examples: a) Reliability - specify the factors required to establish the required reliability of the software system at the time of delivery.

b) Availability - specify the factors required to guarantee a defined availability level for the entire system, such as checkpoint, recovery, and restart.

c) Security - specify the requirements to protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Specific requirements in this area could include the need to:

1) Utilize certain cryptographic techniques

2) Keep specific logs or history data sets

3) Assign certain functions to different modules

4) Restrict communications between some areas of the program

5) Check data integrity for critical variables

6) Ensure data privacy

d) Maintainability - specify attributes of software that relate to the ease of maintenance of the software itself. These may include requirements for certain modularity, interfaces, or complexity limitations. Requirements should not be placed here just because they are thought to be good design practices.

e) Portability - specify attributes of software that relate to the ease of porting the software to other host machines and/or operating systems, including:

1) Percentage of elements with host-dependent code

2) Percentage of code that is host-dependent

3) Use of a proven portable language

4) Use of a particular compiler or language subset

5) Use of a particular operating system

# 3. Verification

Provide the verification approaches and methods planned to qualify the software. The information items for verification are recommended to be given in parallel with the information items in the specified requirements. Consider listing:

a) Verification methods: Inspection, analysis, demonstration, testing

b) Mapping to requirement IDs (traceability matrix)

c) Reference to test plans, if available

# 4. Supporting information

Additional supporting information to be considered includes:

a) Sample input/output formats, descriptions of cost analysis studies, or results of user surveys.

b) Supporting or background information that can help the readers of the SRS.

c) A description of the problems to be solved by the software, and

d) Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

The SRS should explicitly state whether or not these information items are to be considered part of the requirements.

# 5. References

Include the following information regarding references:

a) Provide a complete list of all documents referenced elsewhere

b) Identify each document by title, report number (if applicable), date, and publishing organization

c) Specify the sources from which the references can be obtained