**TABLE OF CONTENTS**

**1** **INTRODUCTION 3**

***1.1*** ***Document overview 3***

***1.2*** ***Abbreviations and Glossary 3***

1.2.1 Abbreviations 3

1.2.2 Glossary 3

***1.3*** ***References 3***

1.3.1 Project References 3

1.3.2 Standard and regulatory References 3

***1.4*** ***Conventions 3***

**2** **REQUIREMENTS 5**

***2.1*** ***States 5***

***2.2*** ***Functionalities and Performance 5***

***2.3*** ***Safety, security, and privacy protection 6***

***2.4*** ***User maintenance 6***

***2.5*** ***Usability and human-factors engineering 6***

2.5.1 Man machine interface layout 6

2.5.2 Help 7

***2.6*** ***Regulatory requirements 7***

***2.7*** ***System environment 7***

***2.8*** ***External interfaces 7***

2.8.1 Hardware interfaces 8

2.8.2 Network interfaces 8

2.8.3 Data exchange 8

***2.9*** ***Resources 8***

2.9.1 Hardware resources 8

2.9.2 Software resources 8

***2.10*** ***Internal data 8***

***2.11*** ***Adaptation 8***

***2.12*** ***Verification 8***

***2.13*** ***Personnel and training 9***

***2.14*** ***Packaging and installation 9***

**3** **VERIFICATION METHODS 10**

**4** **REQUIREMENTS TRACEABILITY 12**

**5** **CRITICAL REQUIREMENTS 13**

1. **INTRODUCTION**
   1. ***Document overview***

This document presents the software requirements specifications of XXX software development project.

It describes:

* Requirements of functionalities, performances, interfaces, environment …
* Tests principles and definitions of validation methods of requirements,
* The compliance of requirements to customer needs,
* The relative importance and precedence of requirements
  1. ***Abbreviations and Glossary***
     1. **Abbreviations**

Add here abbreviations

* + 1. **Glossary**

Add here words definitions

* 1. ***References***
     1. **Project References**

| # | Document Identifier | Document Title |
| --- | --- | --- |
| [R1] | ID | Add your documents references.  One line per document  For example a statement of work, a user interface mock-up … |

* + 1. **Standard and regulatory References**

| # | Document Identifier | Document Title |
| --- | --- | --- |
| [STD1] |  | Add your documents references.  One line per document  For example: ISO 13485, ISO 14971, IEC 62304, and so on |

* 1. ***Conventions***

Requirements listed in this document are constructed according to the following structure:

| **Requirement ID** | **SRS-XXX-000** |
| --- | --- |
| Title | *Title of XXX-000 requirement* |
| Description | Description of XXX-000 requirement |
| Version | Version of XXX-000 requirement |

Example:

| **Requirement ID** | **SRS-GUI-010** |
| --- | --- |
| Title | *Main Window Background Color* |
| Description | The background color of the main window is grey RGB(192,192,192) |
| Version | V1.0 |

1. **REQUIREMENTS**

Note : have a look at <http://en.wikipedia.org/wiki/Requirement>, article in wikipedia. It’s well written and the links at the bottom are useful.

* 1. ***States***

FOO software works in three states:

* Starting: the software loads its components;
* In use: all the functionalities of the software are available to the users;
* Stopping: the software is being stopped.
* Maintenance: the software is in maintenance mode
* And so on …

Add a diagram with states and transitions if necessary

* 1. ***Functionalities and Performance***

This is the core of your SRS. It contains the purpose of your software expressed in technical requirements

You may organize this part in sub-sections like:

* Module 1
  + Function 1.1
  + Function 1.2
  + …
* Module 2
  + …
* Module 3
  + …
* (and so on)

Or sub-sections like:

* Function 1
  + Sub-Function 1.1
  + Sub-Function 1.2
  + …
* Function 2
  + …
* Function 3
  + …
* (and so on)

Choose your own structure which best fits your needs.

Requirements shall not be vague. They shall be understandable and testable.

Ask yourself “How am I going to test this?” when you write a requirement

Examples of requirement:

| **Requirement ID** | **SRS-XXX-010 SAMPLE** |
| --- | --- |
| Title | *Sample requirement about a function* |
| Description | FOO software shall compute the zzz parameters with the a, , c and d input parameter, with the use of the XXX algorithm. |
| Version | V1.0 |

| **Requirement ID** | **SRS-XXX-020 SAMPLE** |
| --- | --- |
| Title | *Sample requirement about a function* |
| Description | FOO software shall save the result of computations in boo-bar format. |
| Version | V1.0 |

* 1. ***Safety, security, and privacy protection***

This section is about software features like confidentiality, integrity control, reliability, and availability. You can add subsections on:

Confidentiality, security, integrity,

Virus, malware protection

Operational security

See CyberSecurity requirements of FDA

See also 2016/679 GPDR in Europe if necessary of data privacy

| **Requirement ID** | **SRS-XXX-030 SAMPLE** |
| --- | --- |
| Title | *Patient data* |
| Description | XXX stores patient data with a checksum to ensure their integrity. |
| Version | V1.0 |

| **Requirement ID** | **SRS-XXX-030 SAMPLE** |
| --- | --- |
| Title | *Password protected functions* |
| Description | The following functions are protected by password:   * Configuration * Firmware update |
| Version | V1.0 |

| **Requirement ID** | **SRS-XXX-030 SAMPLE** |
| --- | --- |
| Title | *Antivirus – antimalware* |
| Description | XXX software shall run with an antivirus / antimalware present on the target PC. |
| Version | V1.0 |

* 1. ***Personal Data***

This section is about management of personal data to be compliant with regulations.

See HIPAA requirements

See also 2016/679 GPDR in Europe, especially articles 15 to 22

| **Requirement ID** | **SRS-XXX-130 SAMPLE** |
| --- | --- |
| Title | *Right to forget* |
| Description | XXX software has a function to delete all patient data. The deletion is permanent and not reversible |
| Version | V1.0 |

* 1. ***User maintenance***

Maintenance functions accessible by users or by administrators. Do sub-sections if there are different types of users.

| **Requirement ID** | **SRS-XXX-040 SAMPLE** |
| --- | --- |
| Title | *Application logs* |
| Description | XXX generates a log file containing:   * The state of the application and the steps performed to reach that state, * The possible error logs, if any. |
| Version | V1.0 |

* 1. ***Usability and human-factors engineering***

The requirements here may have traceability with the results of 62366 standard implementation

* + 1. **Man machine interface layout**

The layout of XXX is ….

Instead of a dozen of text requirements, a mock-up of the software GUI is very appreciated

Add only requirements for which a description of layout/behaviour is necessary and/or requested by a user.

| **Requirement ID** | **SRS-XXX-050 SAMPLE** |
| --- | --- |
| Title | *Menu items and other widgets* |
| Description | XXX software has the following items:   * Menu file ... * Widgets in the main window (slider, button, radiobutton, textfield). |
| Version | V1.0 |

* + 1. **Help**

The user guide is always very important for medical devices. It may be online, in this case add requirements here about the online help ….

| **Requirement ID** | **SRS-XXX-060 SAMPLE** |
| --- | --- |
| Title | *Online user guide* |
| Description | XXX contains an online user guide |
| Version | V1.0 |

* 1. ***Regulatory requirements***

Regulations can have an impact on software design. For example, this is the case with the future Unique Device Identification of FDA.

An about window is a good way to identify software version and provide a UDI….

| **Requirement ID** | **SRS-XXX-070 SAMPLE** |
| --- | --- |
| Title | *About XXX* |
| Description | XXX shall display an “About…” window. This window displays the current version of the application. |
| Version | V1.0 |

In Europe the CE Mark may be somewhere in the GUI:

| **Requirement ID** | **SRS-XXX-075 SAMPLE** |
| --- | --- |
| Title | *CE Mark* |
| Description | XXX shall display the CE Mark in the “About…” window.  The CE Mark is displayed with the 4-digits number of the notified body |
| Version | V1.0 |

* 1. ***System environment***

If software is integrated in a specific system, describe briefly the system and add specific requirements for the integration of your software in this system

Warning : for PEMS/Electro-medical Devices with a big system architecture, a system architecture document is necessary to describe the system/software architecture.

* 1. ***External interfaces***

This section describes hardware and software interfaces of the software in the system

* + 1. **Hardware interfaces**

For PEMS/Electro-medical Devices, add requirements about integration of software and hardware.

* + 1. **Network interfaces**

Also add here communication and networks stuff, like IP, wireless, Bluetooth …

* + 1. **Data exchange**

If XXX software is in interface with other software, describe here the requirements on data exchanges.

* 1. ***Resources***
     1. **Hardware resources**

| **Requirement ID** | **SRS-XXX-080 SAMPLE** |
| --- | --- |
| Title | *Hardware configuration* |
| Description | XXX shall run with the expected response times on a PC with the following minimal configuration:   * 2 Go RAM * ... |
| Version | V1.0 |

* + 1. **Software resources**

| **Requirement ID** | **SRS-XXX-090 SAMPLE** |
| --- | --- |
| Title | *Software configuration* |
| Description | XXX runs in the following software environment:   * (describe OS version), |
| Version | V1.0 |

* 1. ***Internal data***

If specific requirements for internal data, like databases, binary files, xml …

It can be necessary to specify internal data if their design mitigates a risk

* 1. ***Configuration or Adaptation***

If specific requirements adaptability or configuration of software

* 1. ***Verification***

Special functions to test the software, if necessary. For example a hidden function to activate a log file during beta tests. But not a backdoor or a security hole!!!

* 1. ***Personnel and training***

Requirements about the capabilities/knowledge of users, the training they shall have before using software

| **Requirement ID** | **SRS-XXX-USR-010 SAMPLE** |
| --- | --- |
| Title | *E-learning* |
| Description | XXX is delivered with e-learning module. |
| Version | V1.0 |

* 1. ***Packaging and installation***

| **Requirement ID** | **SRS-XXX-PAK-010 SAMPLE** |
| --- | --- |
| Title | *Packaging* |
| Description | XXX shall be delivered on zzz media. |
| Version | V1.0 |

| **Requirement ID** | **SRS-XXX-PAK-010 SAMPLE** |
| --- | --- |
| Title | *Install-shield* |
| Description | XXX shall be installed with the use of an install shield. |
| Version | V1.0 |

1. **VERIFICATION METHODS**

Discard this section if you don’t want to have verification methods attached to your requirements.

The verification methods of the requirements are defined below:

* Inspection (I): control or visual verification
  + Control of the physical implementation or the installation of a component. The control verifies that the implementation or the installation of a component is compliant with the requirements of diagrams.
  + Control of the documentation describing a component. The control verifies that the documentation is compliant with the requirements.
* Analysis (A): verification based upon analytical evidences
  + Verification of a functionality, performance or technical solution of a component by analyzing the data collected by tests in real conditions, by simulation of real conditions or by a analysis report.
  + Analysis of test data or of design data is used as appropriate to verify requirements.
  + The verification is based upon analytical evidences obtained by calculations, like modeling, simulation and forecasting.
  + Analysis is used when an acceptable level of confidence cannot be established by other methods or if analysis is the most cost-effective solution.
* Demonstration (D): verification of operational characteristics, without quantitative measurement
  + Verifying a requirement by demonstration implies that the required functionality specified by a requirement is complete.
  + Demonstration is used when quantitative measurement is not required for verification of the requirements
  + Demonstration includes the control of the technical solutions specified by the non-functional requirements.
* Test (T): verification of quantitative characteristics with quantitative measurement
  + Verifying a functionality, performance or technical solution of a component by executing testing scenarios in predefined, controlled and traceable testing conditions.
  + Tests require the use of special equipment, instrumentation, simulation techniques, or the application of established principles and procedures,
  + Data produced during tests is used to evaluate quantitative results and compare them with requirements.

For each requirement of the SRS, a verification method is defined. Method is abbreviated I, A, D or T.

| Requirement ID | Requirement Title | Method |
| --- | --- | --- |
| REQ-001 | Verify that the speed is displayed in rpm | D |
| REQ-001 | Verify that the color of background is blue | I |

Note: do not mistake the two meanings of the word “test” in this document:

* The method of verification, named Test and abbreviated (T), as defined above.
* A test, or test case, is a sequence of actions to verify a requirement. Tests are defined in the software test plan.

Examples of tests methods:

Inspection:

* Verify that the color of background is blue,
* Verify that the user manual has the CE mark on its cover
* Verify that the PC has 4Gb memory
* Verify that firmware version on electronic card is 1.0.1

Demonstration

* Verify that when the user closes the window, a confirmation message appears
* Verify that the file is saved in the output directory
* Verify that the result is shown
* Verify that if a value is out of range, a warning is displayed

Analysis:

* Verify that the statistical distribution of results of xxx algorithm is a Gaussian with mean=x and stdev=y, when input data are blah blah
* Verify that the linear regression of results of xxx algorithm is a line which value is 1 on the y-axis, at zero on the x-axis,

Test:

* Verify that a file of 1Gb is processed in less than 3s
* Verify that the response time of the server is 15ms with 20 simultaneous requests

Rule of thumb for software, 80% of requirements are verified by demonstration, 15% by inspection and 5% by analysis or test methods.

1. **REQUIREMENTS TRACEABILITY**

Add a table with traceability of software requirements of this document with user or system requirements.

Example

| SRS Req. | Req Title | Functional Req. | Req. Title |
| --- | --- | --- | --- |
| SRS-REQ-001 | Reading ECG values | FUN-REQ-00A | ECG post treatment |
| SRS-REQ-002 | Writing results | FUN-REQ-00A | ECG post treatment |

1. **CRITICAL REQUIREMENTS**

If necessary, add a list of critical requirements, or a list of reference to requirements in previous sections.

This list may be the result of risk analysis (ISO 14971).

Examples

| Requirement ID | Requirement Title | Origin |
| --- | --- | --- |
| REQ-001 | Alarm when value out of range | Risk Analysis |
| REQ-002 | Do not open file if no patient name | Risk Analysis |
| REQ-003 | Display negative values in red color | Human factor engineering |