LABDSOFT

ISEP – Mestrado em Engenharia Informática

Software Requirements Specification Template

This template is a suggestion to complete the Software Requirements Specification (SRS) assignment as well the process used.

Please send comments and improvement opportunities to ajo@isep.ipp.pt.

Acknowledgements:

Sections of this document are based on the book Software Requirements (Third Edition) by Joy Beatty & Karl Wiegers and in the ISO/IEC/IEEE 29148.

Template revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | António Rocha | Initial version | 2017-10-07 |
|  |  |  |  |
|  |  |  |  |

<PROJECT>

Software Requirements Specification – vERSION 1.0

[Authors]

[Year]

[1 Introduction 6](#_Toc495135090)

[1.1 Purpose 6](#_Toc495135091)

[1.2 Document conventions 6](#_Toc495135092)

[1.3 Project scope 6](#_Toc495135093)

[1.4 References 6](#_Toc495135094)

[2 Overall description 7](#_Toc495135095)

[2.1 Product Prespective 7](#_Toc495135096)

[2.2 User classes and characteristics 7](#_Toc495135097)

[2.3 Operating environment 7](#_Toc495135098)

[2.4 Design and implementation contraints 7](#_Toc495135099)

[2.5 Assumptions and dependencies 8](#_Toc495135100)

[3 Systems features 9](#_Toc495135101)

[3.1 Systema feature X 9](#_Toc495135102)

[3.1.1 Description 9](#_Toc495135103)

[3.1.2 Stimulus/Response Sequences 9](#_Toc495135104)

[3.1.3 Functional Requirements 9](#_Toc495135105)

[3.2 System feature y 10](#_Toc495135106)

[3.3 System feature z 10](#_Toc495135107)

[4 Data requirements 11](#_Toc495135108)

[4.1 Logical data model 11](#_Toc495135109)

[4.2 Data dictionary 11](#_Toc495135110)

[4.3 Reports 11](#_Toc495135111)

[5 External interface requirements 12](#_Toc495135112)

[5.1 User interfaces 12](#_Toc495135113)

[5.2 Software interfaces 12](#_Toc495135114)

[5.3 Hardware interfaces 12](#_Toc495135115)

[5.4 Communication interfaces 12](#_Toc495135116)

[6 Quality attributes 13](#_Toc495135117)

[6.1 Usability 13](#_Toc495135118)

[6.2 Performance 13](#_Toc495135119)

[6.3 Security 13](#_Toc495135120)

[6.4 Safety 13](#_Toc495135121)

[6.5 X [Others] 13](#_Toc495135122)

[7 Internationalization and localization requirements 14](#_Toc495135123)

[8 Other requirements 14](#_Toc495135124)

[9 Process adopted for elicitation 15](#_Toc495135125)

[9.1 Stakeholders 15](#_Toc495135126)

[9.2 Applied techniques 15](#_Toc495135127)

[9.3 Effort involved 15](#_Toc495135128)

[9.4 Constraints and limitations 16](#_Toc495135129)

[10 Product Backlog 17](#_Toc495135130)

[11 Appendix 18](#_Toc495135131)

[11.1 Appendix – Glossary 18](#_Toc495135132)

[11.2 Appendix – Analysis models 18](#_Toc495135133)

[11.3 Elicitation data detail 18](#_Toc495135134)

Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Date | Signature |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | <> | Initial version | <> |
|  |  |  |  |
|  |  |  |  |

# Introduction

# Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. >

## Document conventions

<Describe any standards or typographical conventions that were followed when writing this SRS>

## Project scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## References

<List any other documents or Web addresses to which this SRS refers>

# Overall description

# Product Prespective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product.

It is important one or more diagrams that shows the major components of the overall system and interconnections.>

# User classes and characteristics

<Identify the various user classes that you anticipate will use this product, and describe their pertinent characteristics. User classes represent a subset of the stakeholders. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

# Operating environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions; geographical locations of users, servers, and databases, and any other software components or applications with which it must peacefully coexist.>

# Design and implementation contraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

# Assumptions and dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change.>

# Systems features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

# Systema feature X

State the name of the feature in just a few words, such as “3.1 Spell Check”. Repeat section 3.x for each system feature. You can also group the features by area.

# Description

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

# Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

# Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Requirements should be concise, complete, unambiguous, verifiable, and necessary.

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

# System feature y

# System feature z

# Data requirements

<Optionally, use this section of the template to describe various aspects of the data that the system will consume as inputs, process is some fashion, or create as outputs.>

# Logical data model

<You might include a data model for the business operations being addressed by the system, or a logical representation for the data that the system will manipulate. This is not the same thing as the database design.>

# Data dictionary

<You can define the composition of data structures and the meaning of the data elements. >

# Reports

<If your application will generate any reports, identify them here and describe their characteristics. You can specify an example.>

# External interface requirements

# User interfaces

<Describe the logical characteristics of each user interface the system needs. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Details of the user interface design should be documented in a separate user interface specification.>

# Software interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications.>

# Hardware interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

# Communication interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Quality attributes

This section specifies other nonfunctional requirements. These quality requirements should be specific, quantitative and verifiable.

# Usability

Usability requirements deal with ease of learning, ease of use, efficiency and accessibility.

# Performance

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices.>

# Security

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

# Safety

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

# X [Others]

# Internationalization and localization requirements

<This section ensure that the product will be suitable for use in nations, cultures, and geographic location other than those in which it was created. Ex. Differences in currency, dates, numbers, addresses, language, etc.>

# Other requirements

<Define other requirements that are not covered elsewhere.>

# Process adopted for elicitation

< Describe the methodology used to reach the requirements of this document>

# Stakeholders

< Identify and describe stakeholders >

# Applied techniques

<Describe the techniques used in gathering requirements (detail should be in the appendix)>

# Effort involved

Describe in detail the total effort, and per business analyst, involved in the production of this work.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Brief description | BA name | Hours |
| 1 | <> | <Jorge Lopes> | 2 |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Total hours: xxx

# Constraints and limitations

Describe, if any, constraints, limitations or other problem that occurred in the production of this work.

# Product Backlog

Summary of all features presented in the form of user stories. An initial (IR) requirement may result in several user stories.

Provide an estimate of the development effort for each of the user stories. The effort must take into account all stages of development.

Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| US Nr. | IR  Nr. | User story | Estimative hours | Priority |
| 1 | R001 | <> | 20h | 1 |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

To develop all the functionalities of this document the software house will need a total of X hours.

# Appendix

# Appendix – Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

# Appendix – Analysis models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>*

# Elicitation data detail

<You can include here, data, statements, questions and answers made during the requirements gathering meetings.>