Software Requirements Specification

Instructions

Version 1.2 ● 14 JAN 2008

TEXAS PROJECT DELIVERY FRAMEWORK

DIR

Version History

This and other Framework Extension tools are available on the Framework Web site.

Release Date	Description
14-Jan-2008	Version 1.2 released. Modified “Using this Template” section of the Template and italicized all section instructions to align with the Framework and Change Request (CR) #34. CR #34 was recommended by the Framework Change Advisory Board (CAB) and approved by DIR.
13-Mar-2007	Version 1.1 released. Made minor modifications to indicate Framework Extension.
24-May-2006	Version 1.0 Instructions and Template Released.

Contents

Introduction

The Software Requirements Specification (SRS) Template and Template Instructions are included within the within the System Development Life Cycle (SDLC) Extension of the Texas Project Delivery Framework (Framework) to establish a consistent method for documenting software requirements for technology projects. Software requirements describe what the software will do, the software's expected environment, the software's usage profile, its performance parameters, and its expected quality and effectiveness. The SRS is a structured collection of information that embodies the requirements of the software. The SRS completely describes all inputs, outputs, and required relationships between inputs and outputs.

Documenting software requirements can reduce project risk by reducing uncertainty in implementation of the software. Documentation of detailed and accurate software requirements contributes to the success of information technology systems by establishing and communicating expectations for all aspects of the software's features and performance. In addition, the documented software requirements provide the basis for ensuring that requirements are addressed during software design and testing.

Use of the Software Requirements Specification

Within the Framework, the Software Requirements Specification (SRS) is completed, reviewed, and approved in the Project Planning Review Gate. The SRS documents and communicates the software requirements to the technical community who will specify and build the software. The collection of requirements in the SRS must be understandable by stakeholder entities and the technical community.

Approval of the SRS constitutes agreement that the software satisfying the specifications within will be accepted. Once approved, changes can be made to the specifications in the SRS only through the change management process.

The Software Requirements Specification Template should be used to develop a SRS for each project. The SRS should be developed in coordination with and be accessible by appropriate project team and stakeholder entities. In addition, all information in the SRS should be consistent with all project documentation, including the Project Plan and related system development life cycle documents.

NOTE: The Software Requirements Specification should contain descriptive labels for and references to every figure, table, and diagram included within the document.

Section 1. Overview

In the following subsections, provide an overview of the entire Software Requirements Specification (SRS). This section should stand alone as an executive summary.

1.1 Purpose

Specify the purpose the SRS and its intended audience.

1.2 Business Context

Provide an overview of the business organization sponsoring the development of the software application, including the mission statement and organizational objectives of the business unit.

1.3 Scope

Describe the scope of the software application to be produced. Within the description, include:

- The identity of the software product(s)
- A brief description of the software's functionality
- An explanation what the software will and will not do
- A description of the application of the software
- A description of the relevant benefits, objectives, and goals of the software

This description should be consistent with similar statements in preceding project documents.

1.4 User Characteristics

Identify each type of user of the software by function, location, and type of device. Specify the number of users in each group and the nature of their use of the software. Characteristics of the users of the software product will affect the specific requirements. Many people interact with software during the operation and maintenance phases of the software life cycle. Some of these people are users, operators, and maintenance and systems personnel. Certain characteristics of these people, such as educational level, experience, and technical expertise may impose important constraints on the software's operating environment.

Section 2. Assumptions, Dependencies, Constraints

2.1 Assumptions

Describe each assumption that can affect the requirements specified in the SRS. Assumptions are factors that are believed to be true during the life cycle of the project that, if changed, may affect the project outcomes negatively including, but not limited to, end-user characteristics, known technology infrastructure, resource availability, and funding availability.

2.2 Dependencies

Describe each dependency that can affect the requirements specified in the SRS. Dependencies are outside of project scope and control and must remain true for the project to succeed. For example, a dependency might be that an application relies on a different application to provide specific data or that an interface to a third-party application requires the purchase of an application programming interface (API).

2.3 Constraints

Provide a description of the factors that limit the scope and functionality of the software including, but not limited to, regulatory policies, infrastructure limitations, resources, and licensing. Constraints are requirements that are imposed on the solution by circumstance, force, or compulsion. Constraints limit absolutely the options available to a designer of a solution by imposing immovable boundaries and limits.

Section 3. Requirements

In the following subsections, specify all the requirements to a level of detail sufficient to enable developers to specify and build the software application. Every stated requirement should be understandable by users, developers, operators, and external systems staff.

Describe at a minimum the transformation of inputs of the software into outputs of the software and all functions performed by the software in response to an input or in support of an output. This description may consist of a model of the requirements. For example, the model may contain data flow diagrams, entity relationship diagrams, a Product Network Diagram (PND), and a data dictionary.

Requirements should be:

- Correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable
- Cross-referenced to earlier documents that relate
- Uniquely identifiable
- Organized for maximum readability

NOTE: Each requirement documented in this section must have a unique identifier for requirements traceability and should be ranked for importance and/or stability.

3.1 Business Requirements

Describe all requirements from a business perspective. Business requirements are the parts of the fully defined business process that will be automated by the software application. Business requirements may be defined as use cases.

3.2 Functional Requirements

Customize this subsection to contain the subsections necessary to comprehensively define the fundamental actions that must take place within the software to accept and process the inputs and to process and generate the outputs.

Fundamental actions in functional requirements include:

- Validity checks on the inputs
- Exact sequence of operations
- Responses to abnormal situations, including overflow and communication facilities
- Effect of parameters
- Relationship of outputs to inputs, including
- Input/output sequences
- Formulas for input to output conversion
- Definitions of the responses of the software to all realizable classes of input data in all realizable classes of situations

Functional requirements should include specific requirements for business rules. Business rules describe and document the steps in a business process.

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

The two common means of specifying functional requirements are functional decomposition and use cases.

Subsection templates for each of the means of specifying functional requirements are provided below.

3.2.*XF* Function *X*

When functional decomposition is used as the means of specifying the functional requirements provide a 3.2.*xf* subsection for each function. Each 3.2.*xf* subsection should be labeled and titled appropriately for a specific function, where *xf* is the appropriate sequential subsection number and *X* is the name of the specific function.

3.2.*XF*.1 Function *X* Purpose

Describe the intent of the function.

3.2.*XF*.2 Function *X* Inputs

Describe the inputs to the function, including sources, valid ranges of values, timing considerations, operator requirements, and special interfaces.

3.2.*XF*.3 Function *X* Operations

Describe the operations to be performed within the function, including validity checks, responses to abnormal conditions, and types of processing required.

3.2.*XF*.4 Function *X* Outputs

Describe the outputs from the function, including output destinations, valid ranges of values, timing considerations, and considerations for handling of illegal values, error messages, and interfaces required.

3.2.*XU* Use Case *Y*

When use cases are used as the means of specifying the functional requirements provide a 3.2.*xu* subsection for each use case. Each 3.2.*xu* subsection should be labeled and titled appropriately for a specific use case, where *xu* is the appropriate sequential subsection number and *Y* is the name of the specific use case.

Within each use case subsection, specify the use case information, including the actor, pre-conditions, post-conditions, scenarios, and alternate scenarios.

3.3 Logical Data Requirements

Describe all logical data requirements for the software. Logical data requirements may include:

- Types of information used by various functions
- Frequency of use
- Accessing capabilities
- Data entities and their relationships
- Integrity constraints
- Data retention requirements

3.4 User Requirements

Describe user requirements for the software application. User requirements capture the intended behavior of the human interface for the application. For example, if the user operates through a display terminal, specify the required screen content, content of any reports or menus, and relative timing of inputs and outputs. User requirements may include example screen or report formats as prototypes to illustrate requirements.

3.5 Information Management Requirements

Specify requirements for managing the creation, capture, organization, maintenance, use, protection, and disposition of information in accordance with applicable laws, regulations, policies, and standards.

3.6 Systems Requirements

3.6.1 Performance Requirements

Describe the performance conditions and their associated capabilities. Include such considerations as:

- Dynamic actions or changes that occur (e.g., rates, velocities, movements, and noise levels)
- Quantitative criteria covering endurance capabilities of the equipment required to meet the user needs under stipulated environmental and other conditions, including minimum total life expectancy. Indicate required operational session duration and planned utilization rate.
- Performance requirements for the operational phases and modes
- The number of terminals to be supported
- The number of simultaneous users to be supported
- The numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions
- Acceptable performance under atypical stress

State these requirements in measurable terms. For example, 95 percent of the transactions shall be processed in less than one second, rather than, operator shall not have to wait for the transaction to complete.

Performance characteristics unique to a specific function (see Functional Requirements subsection) and outside the general performance characteristics of the software should be specified as part of the processing description of that function.

3.6.2 Quality Requirements

Describe requirements for the quality characteristics of the software. Specify the requirements in measurable and verifiable terms. Describe any trade-offs between the characteristics (e.g., security versus portability). Definitions of the quality characteristics include:

- Correctness – extent to which program satisfies specifications and fulfills user's mission objectives
- Efficiency – amount of computing resources and code required to perform function
- Flexibility – effort needed to modify operational program
- Integrity/Security – extent to which access to software or data by unauthorized people can be controlled. Security requirements relate to both the facility that houses the system and operational security requirements. Examples of security requirements might be to specify the security and privacy requirements, including access limitations to the software, such as existence of log-on procedures and passwords, and of data protection and recovery methods. This could include the factors that would protect the software from accidental or malicious access, use, modification, destruction, or disclosure.

- Interoperability – effort needed to couple one software application to another
- Maintainability – effort required to locate and correct an error during operation. Maintainability requirements apply to maintenance in the planned maintenance and support environments. Examples of quantitative maintainability requirements are time (e.g., mean and maximum downtime, reaction time, turnaround time, mean and maximum times to repair, mean time between maintenance actions), rate (e.g., maintenance staff hours per specific maintenance action, operational ready rate, maintenance time per operating hour, frequency of preventative maintenance), maintenance complexity (e.g., number of people and skill levels, variety of support equipment), and maintenance action indices (e.g., maintenance costs per operating hour, staff hours per overhaul).
- Portability – effort needed to transfer from one hardware or software environment to another
- Reliability – extent to which the software performs with required precision
- Reusability – extent to which software and associated artifacts can be reused in another application
- Testability – effort needed to test to ensure that the software performs as intended
- Usability – effort required to learn, operate, prepare input, and interpret output. Usability requirements include any special or unique requirements that identify and define human-factor considerations and constraints (e.g., design space limits, climatic limits, eye movement, reach, ergonomics, cognitive limits, and usability) that affect operation of the software. Examples include those specific areas, stations, or equipment that require concentrated human engineering attention due to the sensitivity of the operation or criticality of the task (e.g., those areas where the effects of human error would be particularly serious).

3.7 Interfaces

Describe the logical characteristics of each interface between the application and other hardware, software, and communication protocols. Include within the description for each interface the:

- Purpose of the interface
- System the application interfaces to, including whether external or internal
- Interchange mechanism

3.8 Other Requirements

Identify any other requirements that do not fit appropriately into the preceding requirement sections.

Section 4. Requirements Traceability Matrix

In this section, provide reference to the location of the Requirements Traceability Matrix (RTM) that indicates traceabilty from the system requirements documented in the System Requirements Specification (SyRS) to the design elements documented in the System Design Description (SyDD) to the software requirements documented in the Software Requirements Specification (SRS).

The RTM is initiated in the SyRS and is updated appropriately during the life of the project to indicate traceabilty to the design elements documented in the SyDD, the software requirements documented in the SRS, and the design elements documented in the SDD. The completed RTM assures that every requirement has been addressed in the design and that every design element addresses a requirement. The RTM also provides the necessary traceability for integration, acceptance, regression, and performance testing.

The Requirements Traceability Matrix in the SRS should:

- Contain the columns used to illustrate traceability of the system requirements to the system design elements, software requirements, and software design elements
- Contain the columns necessary to illustrate traceability for integration, acceptance, regression, and performance testing
- Be populated with all requirements documented in the SyRS
- Be populated with all design elements documented in the SyDD
- Be populated with all requirements documented in the SRS
- Indicate traceabilty from the system requirements documented in the SyRS to the design elements documented in the SyDD
- Indicate traceability from design elements documented in the SyDD to the software requirements documented in the SRS
- Indicate the source or origin of each requirement

A sample RTM template is provided as an additional tool in the Appendix of the System Requirements Specification Template Instructions.

Section 5. References

Identify the information sources referenced in the Software Requirements Specification and utilized in developing the Software Requirements Specification. Include for each the document number, title, date, and author.

Section 6: Glossary

Define of all terms and acronyms required to interpret the Software Requirements Specification properly.

Section 7: Revision History

Identify changes to the Software Requirements Specification.

Section 8: Appendices

Include any relevant appendices.