**Software Requirement Specification**

# Introduction

*<The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the SRS.>*

## Purpose

*< Specify the purpose of this SRS document along with the intended audience >*

## Scope

*< Describe the scope of the Software Requirements Document as it relates to the project >*

## Definitions, Acronyms, Abbreviations

*< This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.>*

|  |  |  |
| --- | --- | --- |
| ID No | Acronyms | Definition |
|  |  |  |
|  |  |  |
|  |  |  |

## References

*<List any other documents or Web addresses to which this SRS refers to for additional information. These may include standards, guidelines, benchmark data, system requirements specifications, Statement of work etc. Provide enough information so that the reader could access a copy of each reference, including title, version number, source or location>*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID No | Reference | Version Number | Document Name/Source & Location reference | Brief Description | Remarks |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Overall Description

*< This section of the SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements>*

## Product Perspective

*<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. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces needs to be provided>*

## Technical Standards

*<This section should provide a list of International/Certifications standards such as MPEG4 and Toshiba Internal standards such as OAF that the product/components must adhere to with brief description of mandatory parameters that must be met and options/exceptions if any>*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID No** | **Standard** | **Mandatory Specifications/ Parameters** | **Options considered** | **Options not in scope** | **Remarks** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## User characteristics/Operational Scenarios

*<Identify the various user classes or the expected operational scenario of product. 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. Similarly, operational scenarios refers to the actual condition in which the product or component is likely to be subjected in terms of data transfer rates, expected response times, peak loads, unintended usage that a product or component is expected to encounter etc. Diagrammatically explain the intended usage/operational profiles of the software by each user/operational scenario>*

## Operating Environment

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must interface/peacefully coexist. In case of porting or customization projects specify the target environment. Specify all the compiler defined options applicable for the software. Ex: Big endian, little endian data formats 32-bit, 64-bit data support etc.>*

*<Please utilize this section to clarify all things related to your testing environment, then confirm clearly with customer. This is a lesion learn from a MI project which has good result in reduce problem of different test environment between TSDV & customer side. Key points is to:   
+ list up all necessary things for testing both hardware & software  
+ make clear not only name but also detail version*

*+ be clear about scope of testing environment. Some teams just write down environment like: Win 7 & Win 8 or IE8 & IE11, but please be clear that in this case you need to run test 4 times in 4 combination of testing environment. You need to take care this in your estimation phase. >*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID No** | **Environment** | **Description** | Remark |
|  |  |  |  |
|  |  |  |  |

# Dependencies, Constraints & Limitations

*<This section of SRS should provide a general description of items that will limit the developer’s options for designing the system. These can include:*

*1. Regulatory policies*

*2. Hardware limitations; for example, signal timing requirements*

*3. Assumptions and risk if any associated with the requirements that could have an impact on requirements*

# Functional Requirements

*<This subsection should specify the inputs to the software product, transformation processes and outputs. It describes the fundamental actions that must take place in the software.*

*In this subsection, describe the fundamental actions of the system as functional requirements. For each class of function or sometimes for each individual function, it may be necessary to specify requirements on inputs, processing and outputs.*

*Each requirement may be organized with the Requirement ID (e.g. requirement name\_sr.no, followed by information containing brief description, Inputs, Processing, and Outputs preferably in the tabular format as per guidelines)*

* **Brief Description**

*<This subsection should provide a description of the purpose of the function, the approaches and techniques employed. It should contain any introductory or background information, which might clarify the intent of the function>*

* **Inputs**

*<This subparagraph should contain:*

*A detailed description of all data inputs to this function, it may include:*

*1. The sources of the inputs*

*2. Packet/Data Formats*

*3. Packet Size*

*4. Timing*

*5. The ranges of the valid inputs to include accuracy and tolerances*

*- The details of operator control requirements should include names and descriptions of operator actions>*

* **Processing**

*<This subparagraph should define the operations to be performed on the input data and intermediate parameters to obtain the output. It may include:*

*1. Validity checks on the input data*

*2. The exact sequence of operations to include timings of events*

*3. Responses to abnormal situations, for example:*

*4. Overflow /Underflow/Degraded Operations*

*5. Communication failure/Error handling*

*6. Parameters affected by the operations*

*7. Any methods (for example, equations, mathematical algorithms, and logical operations) which must be used to transform the system inputs into corresponding outputs.*

*8. Validity checks in the output data>*

* **Output**

*< This subparagraph should contain:*

*A detailed description of all data output from this function, and it may include:*

*1. Destinations of the outputs*

*2. Formats*

*3. Size*

*4. Timing/Sequence*

*5. The range of valid outputs to include accuracy’s and tolerances*

*6. Error messages>*

|  |  |  |
| --- | --- | --- |
| *<requirement name>* | *<No.>* | *<Brief Description>* |
|  |  | **Input** |
|  |  | <inputs description> |
|  |  | **Proceeding** |
|  |  | <proceeding description> |
|  |  | **Output** |
|  |  | <output description> |
| *<requirement name>* | *<No.>* | *<Brief Description>* |
|  |  | **Input** |
|  |  | <inputs description> |
|  |  | **Proceeding** |
|  |  | <proceeding description> |
|  |  | **Output** |
|  |  | <output description> |

# External Interface Requirements

*<This should specify external interface requirements explaining the use by other software products/components/applications/ systems>*

# Non-functional Requirements

*<This subsection should specify attributes of quality that can place specific requirements on the software. >*

## Performance Requirements

*< The system’s performance characteristics should be outlined in this section*

*1. Resource utilization, such as program/data memory (RAM/ROM)*

*2. Throughput, for example, frames/ bitrate per second*

*3. Peak load conditions*

*4. Operating frequency /clock speed*

*5. MCPS/MIPS etc.>*

## Memory Requirements

*< 1. Memory overlay requirements*

*2. Memory map>*

## Error Handling

*<Specify how error situations/scenarios are handled by the software including list of error codes that needs to be addressed in line with higher layer programs/interfaces>*

## Other Requirements & Attributes

*<Define any other requirements not covered elsewhere in the SRS. This might include portability, reliability, maintainability, safety and security considerations, notification sequence/state transition requirements etc.>*

# Acceptance criteria

*<Specify in detail the acceptance criteria for the software developed with sufficient details to enable validated by Customer or an independent team.*

*These must include following parameters as applicable to the product/domain:*

*1. Quality of audio/video outputs in quantifiable/Measurable terms such as RMS value, Bit exactness etc.*

*2. Quantitative Performance goals to be met such as peak MIPS/MCPS*

*3. Compliance to standards/certifications*

*4. Code Coverage in internal / product validation testing*

*5. Acceptance Test Criteria to be met etc.>*

# Appendix

*< Enclose detailed documents used in developing requirements such as usecase, state transition diagrams, Initialization tables etc.>*

*<Guidelines – the text follows>*

*<Guidelines – the text follows>*