Especificación de Requerimientos de Software

11056 - Sistemas de Información I

Universidad Nacional de Luján

Departamento de Ciencias Básicas
División Computación
Área de Sistemas de Información

IEEE-STD-830-1998: Especificación de los requisitos del software.

1. Definiciones

En general las definiciones de los términos usados en estas especificaciones están conforme a las definiciones proporcionadas en IEEE Std 610.12-1990.

Las definiciones siguientes son términos claves usados en esta práctica recomendada.

1.1 Contrato:

Documento legal acordado por el cliente y el proveedor. Esto incluye los requisitos técnicos y requerimientos de la organización, costo y tiempo para un producto. Un contrato también puede contener la información informal pero útil tales como los compromisos o expectativas de las partes involucradas.

1.2 Cliente:

La (s) persona (s) que pagan por el producto y normalmente (pero no necesariamente) definen los requisitos. En el contexto de esta práctica recomendada el cliente y el proveedor pueden ser miembros de la misma organización.

1.3 Proveedor:

La (s) persona (s) que producen un producto para un cliente. En el contexto de esta práctica recomendada el cliente y el proveedor pueden ser miembros de la misma organización.

1.4 Usuario:

La (s) persona (s) que operan o interactúan directamente con el producto. El usuario (s) y el cliente (s) no es (son) a menudo las mismas persona(s).

2. Las consideraciones para producir una buena SRS.

Estas cláusulas proporcionan información contextual que deberían ser consideradas al momento de producir una SRS. Esto incluye lo siguiente:

a) la Naturaleza de la SRS;

b) el Ambiente de la SRS;

c) las Características de una buena SRS;

d) la preparación del acuerdo de la SRS;

e) la evolución de la SRS;

f) Prototipos;

g) Incorporando el diseño en la SRS;

h) Incorporando los requisitos del proyecto en la SRS.

2.1 Naturaleza de la SRS

La SRS es una especificación para un producto de software en particular, programa, o conjunto de programas que realizan ciertas funciones en un ambiente específico. La SRS puede escribirse por uno o más representantes del proveedor, uno o más representantes del cliente, o por ambos. La Subclausula 2.4 recomienda ambos.

Los problemas básicos que se presentan al escribir una SRS van dirigidos a lo siguiente:

a) La Funcionalidad.

¿Qué se supone va hacer el software?

b) Las interfaces Externas.

¿Cómo el software actúa recíprocamente con las personas, el hardware de los sistemas, otro hardware, y otro software?

c) Performance.

¿Cuál es la velocidad, la disponibilidad, tiempo de respuesta, tiempo de recuperación de varias funciones del software, etc.?

d) Los Atributos.

¿Qué portabilidad tiene, exactitud, el mantenimiento, la seguridad, las consideraciones etc.?

e) Las restricciones de diseño impuestas sobre una implementación.

¿Hay algún Standard requerido, lenguaje de la implementación, políticas para la integridad de la base de datos, límites de recursos, entornos operativos etc.?

Los autores de la especificación de Requerimientos de Software deberían evitar poner requerimientos de diseño o del proyecto dentro de la SRS. Para ver el contenido recomendado ver cláusula 3

2.2 Ambiente de la SRS

Es importante considerar la parte que la SRS juega en el plan de proyecto total, que se define en IEEE Std 610.12-1990. El software puede contener toda la funcionalidad del proyecto esencialmente o puede ser parte de un sistema más grande. En el último caso habrá una SRS que establecerá las interfaces entre el sistema y sus partes, y se pondrán los requerimientos de performance y funcionalidad de cada parte del software. Por supuesto, la SRS debería estar de acuerdo con los requerimientos del sistema completo.

El estándar IEEE 1074-1997 describe las etapas en el ciclo de vida del software y las entradas aplicables a cada una de ellas. Otros estándares están relacionados con otras partes del ciclo de vida de software y así los requerimientos de Software pueden ser complementados.

Dado que la SRS tiene un papel específico en el proceso de desarrollo de software, los autores de la SRS deberían cuidar no ir más allá de los límites de ese rol.

Esto significa que la SRS:

a) debe definir todos los requisitos del software correctamente. Un requisito del Software puede existir debido a la naturaleza de la tarea a ser resuelta o debido a una característica especial del proyecto.

b) no debe describir ningún detalle de diseño o implementación. Éstos deben describirse en la fase del diseño del proyecto.

c) no debe imponer restricciones adicionales sobre el software. Éstas son especificadas apropiadamente en otros documentos, tales como los planes de Aseguramiento de la Calidad del Software.

De esta forma, una SRS apropiada limita el rango de diseños válidos pero no especifica un diseño particular.

2.3 Características de una buena SRS.

Una SRS debe ser:

a) Correcta;

b) No ambigua;

c) Completa;

d) Consistente;

e) Clasificar por importancia y/o estabilidad;

f) Verificable;

g) Modificable;

h) Rastreable.

2.3.1 Correcta

Una SRS es correcta si, y sólo si, cada requisito declarado es un requerimiento que el software debe alcanzar.

No hay ninguna herramienta o procedimiento que aseguren la correctitud. La SRS debería ser comparada con alguna especificación superior aplicable, tal como una especificación de los requerimientos del Sistema, con otros documentos del proyecto y con otros estándares aplicables para asegurar que esté de acuerdo. Alternativamente el cliente o el usuario pueden determinar si la SRS refleja las necesidades reales correctamente. La rastreabilidad hace este procedimiento más fácil y menos propenso al error.

2.3.2 No ambigua

Una SRS es no ambigua si, y sólo si, cada requisito declarado tiene sólo una interpretación. Esto requiere que, como mínimo, cada característica del producto final sea descripta usando un único término.

En casos dónde un término en un contexto particular tenga significados múltiples, el término debe ser incluido en un glosario dónde su significado es más específico.

Una SRS es una parte importante del proceso de requerimientos del ciclo de vida de software y se usa en el diseño, implementación, monitoreo del proyecto, validación y verificación y en entrenamiento como se describe en IEEE Std 1074-1997.

La SRS debe ser no ambigua tanto para aquellos que la crean como para aquellos que la usan. Sin embargo, estos grupos no tienen a menudo la misma formación y por consiguiente no tienden a describir los requisitos del software de la misma manera.

Las representaciones que proveen la SRS para los desarrolladores puede ser contraproducente en el entendimiento de los usuarios y viceversa.

Las Subclausulas 2.3.2.1 hasta de 2.3.2.3 recomiendan cómo evitar la ambigüedad.

2.3.2.1 Dificultad del lenguaje natural.

Los requisitos son a menudo escritos en lenguaje natural (por ejemplo, castellano) el lenguaje natural es inherentemente ambiguo. Una SRS en lenguaje natural debería ser revisada por un equipo independiente para identificar el uso ambiguo del lenguaje para que pueda corregirse.

2.3.2.2 Lenguajes de especificación de requisitos

Una manera de evitar la ambigüedad inherente al lenguaje natural es escribir la SRS en un lenguaje de especificación de requisitos particular. Sus procesadores de lenguaje detectan errores léxicos, sintácticos, y semánticos automáticamente.

Una desventaja en el uso de tales lenguajes es la cantidad de tiempo requerido para aprenderlos. Además, muchos usuarios no-técnicos los encuentran ininteligibles. Es más, estos lenguajes tienden a ser mejores para expresar ciertos tipos de requisitos y dirigirse a ciertos tipos de sistemas. Así, ellos pueden influenciar los requerimientos de una manera sutil.

2.3.2.3 Herramientas de Representación

En general, los métodos de requisitos y lenguajes y las herramientas que los apoyan entran en tres categorías generales - objeto, proceso y comportamiento.

Los enfoques orientados a objetos organizan los requisitos en términos de objetos del mundo real, sus atributos, y los servicios realizados por esos objetos.

Los enfoques basados en procesos organizan los requisitos dentro de jerarquías de funciones que se comunican vía el flujo de datos.

Los enfoques de comportamiento describen el comportamiento externo del sistema en términos de algunas nociones abstractas, (tales como cálculo de predicados), funciones matemáticas o máquinas de estado.

El grado en el cual estas herramientas y métodos pueden ser útiles para preparar una SRS dependen del tamaño y complejidad de los programas. No intentamos aquí describir o recomendar ninguna herramienta particular.

Usar cualquiera de estos enfoques es mejor que emplear descripciones en lenguaje natural. De esta manera los clientes que no están familiarizados con las notaciones pueden entender igualmente la SRS.

2.3.3 Completa

Una SRS está completa si, y sólo si, incluye los siguientes elementos:

a) Todos los requerimientos importantes ya sean relacionados con la funcionalidad, performance, restricciones del diseño, atributos o interfaces externas. En particular cualquier requerimiento externo impuesto por una especificación del sistema debería ser acordado y tratado.

b) La definición de las respuestas del software a todos los posibles datos de entrada del sistema en todas las situaciones posibles. Note que es importante especificar tanto las respuestas válidas e inválidas a los valores de entrada.

c) Tener todas las etiquetas llenas y referencias a todas las figuras, tablas, diagramas en la SRS y definición de todos los términos y unidades de medida.

2.3.3.1 Uso de ASDs

Cualquier SRS que usa la frase "A ser definido" (ASD) no es una SRS completa.

El ASD es, sin embargo, ocasionalmente necesario y debe acompañarse por:

a) Una descripción de las condiciones que causan el ASD (por ejemplo, por qué una respuesta no es conocida) para que la situación pueda resolverse;

b) Una descripción de lo que debe hacerse para eliminar el ASD que es responsable para su eliminación y por como debe eliminarse.

2.3.4 Consistente

La consistencia se refiere a la consistencia interna. Si una SRS no está de acuerdo con algún documento de alto nivel, tal como la especificación de requerimientos de sistema, entonces no es correcta.

2.3.4.1 Consistencia interna

Una SRS es internamente consistente si, y sólo si, no hay ningún subconjunto individual de la SRS que esté en conflicto.

Los tres tipos de conflictos en una SRS pueden ser los siguientes :

a) Las características especificadas para un objeto del mundo real pueden estar en conflicto. Por ejemplo

1) Los formatos de un reporte de salida pueden ser descriptos en un requerimiento como una tabla en otro como un texto.

2) Un requerimiento puede establecer que todas las luces deberían ser verdes mientras otro puede establecer que todas las luces deberían ser azules.

b) Podría haber conflictos lógicos o temporales entre dos acciones especificadas. Por ejemplo,

1) Un requerimiento puede especificar que el programa debe sumar dos entradas y otro puede especificar que el programa debería multiplicarlas.

2) Un requerimiento que establece que "A" debe seguir a "B", mientras otro puede requerir que "A"y B" ocurran simultáneamente.

c) Dos o más requerimientos pueden describir el mismo objeto del mundo real pero usando diferentes términos para este objeto. Por ejemplo, las solicitudes de un programa pueden ser llamadas un prompt (petición de orden) en un requerimiento y un que (mostrar) en otro. El uso de definiciones y terminología estándar promueve la consistencia.

2.3.5 Clasificación por importancia y/o estabilidad

Una SRS está ordenada por importancia y/o estabilidad si cada requerimiento en ella ha sido identificado para indicar tanto la importancia como la estabilidad del requerimiento particular.

Habitualmente, todos los requerimientos que se relacionan con un producto de software no tienen la misma importancia. Algunos requerimientos pueden ser esenciales, mientras que otros pueden ser deseables.

Cada requerimiento en la SRS debería estar identificado para hacer explícita y clara esta diferencia. Identificar los requerimientos de la siguiente manera puede ayudar:

a) A hacer que los clientes hagan consideraciones más cuidadosas a cada requerimiento, lo cual a menudo clarifica alguna presunción oculta que ellos pueden tener.

b) Tener desarrolladores que hacen decisiones de diseño más correctas y tengan niveles de dedicación apropiados del esfuerzo para las diferentes partes del producto de software.

2.3.5.1 Grado de estabilidad

Un método de identificar requerimientos es usar la dimensión de estabilidad. La estabilidad puede ser expresada en términos del número de cambios esperados para algún requerimiento basado en la experiencia o el conocimiento de la ocurrencia de algún evento que efectúe la organización, funciones, o la gente que es ayudada por el sistema de software.

2.3.5.2 Grado de necesidad

Otra manera de ordenar requerimientos es distinguir las clases de requisitos que hay: el esencial, el condicional y opcional.

a) Esenciales.
Implica que el software no debería ser aceptado hasta que éstos requerimientos sean provistos en la manera acordada.

b) Condicionales.

Implica que hay requerimientos que podrían hacer más relevante al producto de software, pero no lo harían inaceptable si están ausentes.

c) Opcionales.

Implica una clase de funciones que podrían o no ser importantes. Esto le da al proveedor la oportunidad de proponer algo que exceda la SRS .

2.3.6 Verificable

Una SRS es verificable si, y sólo si, cada requisito establecido en ella es verificable. Un requerimiento es verificable si, y sólo si, existe algún proceso finito y efectivo en costos con el cual una persona o la máquina pueda chequear que el producto de software reúne tal requerimiento. En general los requerimientos ambiguos son no verificables.

Los requerimientos No-verificables incluyen expresiones tales como "Que trabaje bien", "Buena interface humana" y "podría usualmente pasar". Estos requerimientos no pueden ser verificados porque no es posible definir los términos "bueno," "bien" o "usualmente". La expresión "el programa nunca entrará en un loop infinito" no es verificable porque las pruebas de esta cualidad son teóricamente imposibles.

Un ejemplo de una expresión verificable es:

Las salidas del programa deberían producirse dentro de 20 seg el 60% de las veces; y deberían ser producidas dentro de los 30 seg el 100% de las veces.

Estas expresiones son verificables porque usan términos concretos y medidas cuantitativas.

Si no se puede seguir un método para determinar que el software reúne un particular requerimiento, entonces el requerimiento debería ser removido o revisado.

2.3.7 Modificable

Una SRS es modificable si, y sólo si, su estructura y estilo son tales que los cambios a los requerimientos pueden ser hechos fácil, completamente y consistentemente mientras se mantiene la estructura y estilo. La modificabilidad generalmente requiere que la SRS :

a) Tenga una organización coherente y fácil de usar con una tabla de contenidos, un índice y una tabla de referencias cruzadas explícitas;

b) no ser redundante (por ejemplo, el mismo requisito no debería aparecer en más de un lugar en la SRS);

c) Expresar cada requisito separadamente, en lugar de entremezclarlos con otros requerimientos.

La redundancia en si misma no es un error, pero puede fácilmente conducirnos a un error. La redundancia puede ayudar a hacer una SRS más legible de vez en cuando, pero el problema que puede ocurrir cuando un documento redundante es actualizado. Por ejemplo, un requisito puede alterarse en un solo lugar dónde aparece. La SRS entonces se convierte en

inconsistente. Cuando la redundancia es necesaria, la SRS debería incluir referencias cruzadas explícitas para hacerla modificable.

2.3.8 Rastreable

Una SRS es rastreable si el origen de cada uno de sus requisitos está claro y si ella facilita referenciar a cada requerimiento en un futuro desarrollo o documentación. Se recomienda seguir dos tipos de rastreabilidad:

a) Rastreabilidad hacia atrás (por ejemplo estados previos de desarrollo).

Esto cuenta con que cada requerimiento referencie explícitamente a sus fuentes en documentos anteriores.

b) Rastreabilidad hacia adelante (Por ejemplo hacia todos los documentos creados posteriormente a la SRS).

Esto cuenta con que cada requerimiento en la SRS tenga un nombre único o un número de la referencia.

La rastreabilidad hacia adelante de la SRS es esencialmente importante cuando el producto del software entra en la fase de operación y mantenimiento. Cuando los documentos de diseño o el código son modificados, es esencial que estén disponibles el conjunto de requerimientos completos que pueden ser afectados por estas modificaciones.

2.4 Preparación común para la SRS

El proceso de desarrollo de software debería comenzar por el acuerdo entre proveedor y cliente acordando respecto del software completo que debe hacerse. Este acuerdo, en la forma de una SRS, debería ser preparado en forma conjunta. Esto es importante porque habitualmente ni los clientes ni los proveedores están calificados para escribir una buena SRS solos.

a) Los clientes no entienden el proceso de diseño y desarrollo de software tan bien como para escribir una SRS usable.

b) Los Proveedores usualmente no entienden los problemas del cliente y el campo de aplicación tan bien como para especificar requerimientos para un sistema satisfactorio.

Por consiguiente, los clientes y proveedores deberían trabajar juntos para producir una SRS bien escrita y completamente entendible.

Una situación especial existe cuando el sistema y su software son ambos definidos concurrentemente. Entonces la funcionalidad, interfaces, performance y otros atributos, y restricciones del software no están predefinidos, sino que son más bien definidos en conjunto, así como también los cambios y los asuntos de negocios. Esto hace más difícil, pero no menos importante, alcanzar las características establecidas en 2.3. En particular, una SRS que no cumple con la especificación de requerimientos de su sistema padre es incorrecta. Estas prácticas recomendadas no discuten específicamente estilos, lenguajes usados o técnicas de

buena escritura, sin embargo es muy importante que la SRS esté bien escrita. Las técnicas generales para escribir libros pueden ser usadas como guía.

2.5 Evolución de la SRS

La SRS puede necesitar modificarse a medida que progresa el desarrollo del producto de software. Puede ser imposible especificar algunos detalles al momento de iniciar el proyecto (por ejemplo, puede ser imposible definir todos los formatos de la pantalla para un programa interactivo durante la fase de requisitos). Cambios adicionales pueden asegurar poner al descubierto deficiencias, fallas y equivocaciones en la SRS. Las dos principales consideraciones en este proceso son las siguientes:

a) Deben especificarse los requisitos completamente como son conocidos en el momento, aun cuando las revisiones evolutivas puedan preverse como inevitables. El hecho que ellos están incompletos debe ser anotado.

b) Un proceso de cambio formal debe ser inicializado para identificar, controlar, seguir y reportar cambios en el proyecto. Una vez aprobados los cambios en los requerimientos deberían ser incorporados en la SRS para

1. Proporcionar un adecuado y completo camino para auditar los cambios.
2. Permitir la revisión de la porción actual y la reemplazada en la SRS .

2.6 Prototipos.

Los prototipos frecuentemente se usan durante la fase de los requisitos de un proyecto. Muchas herramientas existen para permitir un prototipo que exhiba algunas características de un sistema, creado en forma rápida y fácil. Ver también ASTME 1340-96.

Los prototipos son útiles por las siguientes razones:

a) Los clientes pueden ser más propensos a ver los prototipos y reaccionar a ellos, que leer la SRS y reaccionar a ella.

Así, los prototipos proporcionan una más rápida retroalimentación.

b) El prototipo muestran aspectos imprevistos del comportamiento de los sistemas. Así, produce no sólo respuestas sino también nuevas preguntas. Esto ayuda a alcanzar el final de la SRS.

c) Una SRS basada en un prototipo tiende a sufrir menos cambios durante el desarrollo, así acorta el tiempo de desarrollo.

Un prototipo debe usarse como una manera de elicitar los requisitos del software. Algunas características tales como pantalla o formatos del reporte pueden ser extraídos directamente del prototipo. Otros requisitos pueden ser inferidos de la ejecución de los prototipos.

2.7 Incorporando el diseño en la SRS

Un requisito especifica una función visible externamente o atributo de un sistema. Un diseño describe un subcomponente particular del sistema y/o sus interfaces con otros subcomponentes. La escritura de la SRS debería distinguir claramente entre restricciones identificadas al diseño requerido y proponer un diseño específico. Note que cada requisito en la SRS limita las alternativas de diseño. Esto no significa, sin embargo, que cada requisito sea un diseño.

La SRS debería especificar qué funciones deben ser realizadas, sobre qué datos, para producir qué resultados, en qué situación y para quien. La SRS se debería enfocar en los servicios a ser realizados. La SRS no debería normalmente especificar items de diseño tales como los que siguen:

a) Particionamiento del software en módulos;

b) Asignaciones de funciones a los módulos;

c) Describir el flujo de información o controles entre los módulos;

d) Escogiendo las estructuras de los datos.

2.7.1 Requisitos de diseño necesarios

En casos especiales algunos requisitos pueden restringir el diseño severamente. Por ejemplo, seguridad o requisitos de seguridad pueden reflejarse directamente en el diseño cuando sean necesarios para:

a) Mantener ciertas funciones en módulos separados;

b) Permitir sólo comunicación limitada entre algunas áreas del programa;

c) Verificar la integridad de datos de algunas variables críticas.

Ejemplos de restricciones del diseño válidos son requisitos físicos, requisitos de performance, estándar de desarrollo de software y de aseguramiento de la calidad del software.

Por consiguiente, los requisitos deberían ser establecidos desde un punto de vista completamente externo. Cuando se utilizan modelos para ilustrar los requisitos, recuerde que los modelos sólo indican el comportamiento externo, y no especifican un diseño

2.8 Incorporando los requisitos del proyecto en la SRS.

La SRS debería conducir al producto software, no al proceso de producir el producto del software.

Los requisitos del proyecto representan un acuerdo entre el cliente y el proveedor acerca de la manera contractual adecuada para producir el software y por lo tanto no debería incluirse en la SRS. Esto normalmente incluye ítems como:

a) Costos;

b) Planes de entrega;

c) Procedimientos de reportes;

d) Métodos de desarrollo de Software;

e) Aseguramiento de la Calidad;

f) Criterios de verificación y validación;

g) Procedimientos de aceptación.

Los requisitos del proyecto son especificados en otros documentos, habitualmente en un plan de desarrollo de software, uno de aseguramiento de la calidad del software o una declaración de trabajo.

3. Las partes de una SRS

Esta cláusula discute cada parte esencial de la SRS.

Estas partes están ordenadas en la Figura 1 en un borrador que puede servir como un ejemplo de escrito de una SRS.

Si bien, una SRS no tiene que seguir este borrador o usar los nombres dados aquí para sus partes, una buen SRS debería incluir toda la información que es discutida aquí.

Tabla de Contenidos

1. Introducción

1.1 Propósito

1.2 Alcance

1.3 Definiciones, acrónimos, y abreviaturas

1.4 Referencias

1.5 Apreciación global

2. Descripción global

2.1 Perspectivas del producto

2.2 Funciones del producto

2.3 Características del usuario

2.4 Restricciones

2.5 Presunciones y dependencias

3. Los requisitos específicos (Vea del 3.3.1 al de 3.3.8 para una explicación de los posibles requerimientos específicos) Ver también anexo A para varias maneras de organizar esta sección de la SRS.

Apéndices

Índice

3.1 Introducción (Sección 1 de la SRS)

La introducción de la SRS debería proveer una visión global de la SRS completa.

Debería contener las siguientes subsecciones:

a) Propósito;

b) Alcance;

c) Definiciones, acrónimos, y abreviaturas;

d) Referencias;

e) Apreciación global.

3.1.1 Propósito (1.1 de la SRS)

Esta subsección debería:

a) Delimitar el propósito de la SRS;

b) Especificar la audiencia pretendida para la SRS.

3.1.2 Alcance (1.2 de la SRS)

Esta subsección debería:

a) Identificar los productos software a ser producidos por su nombre (por ejemplo, Host DBMS, Generador de Reportes, etc.);

b) Explicar que hará el producto software y si es necesario que no hará.

c) Describir la aplicación del software especificada, incluyendo beneficios relevantes, objetivos, y metas;

d) Ser consistente con lo establecido en forma similar en especificaciones de más alto nivel (por ejemplo, las especificaciones de los requisitos del sistema), si existen.

3.1.3 Definiciones, Acrónimos, y abreviaturas (1.3 de la SRS)

Esta subsección debería proporcionar las definiciones de todos los términos, acrónimos, y abreviaturas para interpretar apropiadamente la SRS . Esta información puede ser provista haciendo referencia a uno o más apéndices en la SRS o referenciando a otros documentos.

3.1.4 Referencias (1.4 de la SRS)

Esta subsección debería:

a) Proporcionar una lista completa de todos los documentos referenciados en otra parte en la SRS;

b) Identificar cada documento por título, número reporte (si es aplicable), fecha, y editorial;

c) Especificar las fuentes de donde se obtuvieron las referencias.

Esta información puede proporcionarse por referencias a un apéndice o a otro documento.

3.1.5 Apreciación global (1.5 de la SRS)

Esta subsección debería:

a) Describir que contiene el resto de la SRS;

b) Explicar cómo está organizada la SRS.

3.2 Descripción global (Sección 2 de la SRS)

Esta sección de la SRS debería describir los factores generales que afectan el producto y sus requisitos. Esta sección no establece requisitos específicos. A pesar de ello provee el contexto de aquellos requerimientos que son definidos en detalle en Sección 3 de la SRS, y que los hacen más fáciles de comprender.

Esta sección normalmente consiste en seis subdivisiones, como sigue:

a) Perspectiva del Producto;

b) Funciones del Producto;

c) Características del Usuario;

d) Restricciones;

e) Presunciones y dependencias;

f) División de requerimientos.

3.2.1 Perspectiva del producto (2.1 de la SRS)

Esta subsección de la SRS debería poner al producto en perspectiva con otros productos relacionados. Si este producto es independiente y está totalmente autocontenido, debería también ser establecido aquí. Si la SRS define un producto que es un componente de un sistema mayor, como frecuentemente ocurre, entonces esta subsección debería relacionar los requisitos del sistema mayor a la funcionalidad del software y debería identificar las interfaces entre ese sistema y el software.

Puede ser útil un diagrama de bloques que muestre los componentes principales del sistema mayor, interconexiones, e interfaces externas.

Esta subsección debería también describir cómo el software opera dentro de varias restricciones. Por ejemplo, estas restricciones podrían incluir:

a) Interfaces de Sistema;

b) Interfaces de Usuario;

c) Interfaces de Hardware;

d) Interfaces de Software;

e) Interfaces de Comunicaciones;

f) Memoria;

g) Operación;

h) Requerimientos de adaptación del Site.

3.2.1.1 Interfaces del sistema.

Esto debe listar cada interfaz del sistema y debe identificar la funcionalidad del software para cumplir los requisitos del sistema y la descripción de las interfaces que concuerdan con el sistema.

3.2.1.2 Interfaces del usuario.

Esto debería especificar lo siguiente:

a) *Las características lógicas de cada interfaz entre el producto del software y sus usuarios.* Esto incluye aquellas características de configuración necesarias para cumplir los requisitos del software. (Por ejemplo, formatos de la pantalla requeridos, esquemas de ventanas o página, contenido de algún reporte o menús, funciones de teclas programables o disponibles).

b) *Todos los aspectos de optimización de las interfaces con las personas que deban usar el sistema.* Esto simplemente puede consistir de una lista de cómo el sistema hace para mostrarse al usuario o como no. Un ejemplo puede ser un requisito de la opción de un mensaje de error largo o corto. Además todos estos requisitos deben ser verificables (esto debe también ser especificado en los Atributos de Sistema Software bajo una sección titulada Facilidad de Uso).

3.2.1.3 Interfaces del hardware.

Esto debería especificar las características lógicas de cada interfaz entre el producto del software y los componentes del hardware del sistema. Esto incluye las características de la configuración (el número de puertos, conjunto de instrucciones, etc.), esto también cubre cosas importantes cómo qué dispositivos deben ser soportados, cómo serán soportados y los protocolos. Por ejemplo, se puede especificar soporte para terminales de pantalla completa como opuesto a pantallas de línea a línea.

3.2.1.4 Interfaces con el software.

Esto debería especificar el uso de otros productos del software requeridos (por ejemplo, un sistema de administración de datos, un sistema operativo o un paquete matemático), e interfaces con otros sistemas de aplicación (por ejemplo, la unión entre, un Sistema de Cuentas a Cobrar y Sistema de Contabilidad General). Para cada producto de software requerido se debería proveer lo siguiente:

- Nombre;

- Mnemotécnico;

- Número de especificación;

- Número de versión;

- Fuente.

Para cada interfaz, debería proporcionarse lo siguiente:

- Discusión del propósito de la interfaz del software en relación con el producto del software.

- Definiciones de las interfaces en términos de contenidos del mensaje contenidos y formato. No es necesario detallar interfaz bien documentada, pero si es necesaria una referencia al documento que la define.

3.2.1.5 Interfaces de comunicaciones

Esto debería especificar las varias interfaces para la comunicación tales como protocolos de las redes locales, etc.,

3.2.1.6 Restricciones de memoria

Esto debería especificar cualquier característica aplicable y límites en la memoria primaria y la memoria secundaria.

3.2.1.7 Operaciones

Esto debería especificar las operaciones normales y especiales requeridas por los usuarios tales como:

a) Los varios modos de operación en la organización del usuario.

b) Periodos de operación interactiva y periodos de operación no supervisada;

c) Funciones de soporte al procesamiento de datos

d) Operaciones de Backup y recuperación.

NOTA: Esto es algunas veces especificado como parte de la Sección de interfaces de Usuario.

3.2.1.8 Requisitos de adaptación del Site.

Esto debería:

a) Definir los requisitos para cualquier dato o secuencia de inicialización que son especificadas para un site dado, misión, o modo de operación (por ejemplo, grillas de valores, límites de seguridad, etc.);

b) Especificar el site o características relacionadas con la misión que deberían ser modificadas para adaptar el software a una instalación particular.

3.2.2 Funciones del Producto (2.2 de la SRS)

Esta subsección de la SRS debería proveer un resumen de las funciones principales que el software realizará.

Por ejemplo, una SRS para un programa de cuentas podría usar esta parte para introducir al mantenimiento de Cuentas de Clientes, declaraciones de clientes, preparación de inventarios, sin mencionar el vasto tamaño de los detalles que cada una de estas funciones requiere.

Algunas veces el resumen de funciones que es necesario para esta parte puede ser tomado directamente desde las secciones de especificación de alto nivel (si existen) que asignan funciones particulares al producto software. Note que para ganar claridad:

a) Las funciones deberían estar organizadas en una forma que la lista de funciones sean entendibles para los clientes o para cualquier otro que lea el documento por primera vez.

b) Métodos gráficos o textuales pueden ser utilizados para mostrar las diferentes funciones y sus relaciones.

Tales diagramas no tienen la intención de mostrar un diseño de un producto, sino simplemente mostrar las relaciones lógicas más variables.

3.2.3 Características del usuario (2.3 de la SRS)

Esta subsección de la SRS debería describir aquellas características generales del usuario pretendido del producto incluyendo nivel de educación, experiencia, experticia técnica. No debería ser usado para establecer requerimientos específicos, sino más bien para proveer las razones por las cuales ciertos requerimientos específicos son luego especificados en la sección 3 de la SRS.

3.2.4 Restricciones (2.4 de la SRS)

Esta subsección de la SRS debería proporcionar una descripción general de algunos otros items que limitarán las opciones de desarrollo. Ésto incluye:

a) Políticas regulatorias;

b) Limitaciones de Hardware; (por ejemplo requerimientos de tiempo de señal)

c) Interfaces con otras aplicaciones;

d) Operación en Paralelo;

e) Funciones de Auditoría;

f) Funciones de Control;

g) Requerimientos de lenguaje de orden mayor;

h) Protocolos para establecimiento de comunicaciones (por ejemplo, XON-XOFF, ACK-NACK);

i) Requerimientos de Fiabilidad;

j) Criticidad de la aplicación;

k) Seguridad y consideraciones de seguridad y fiabilidad.

3.2.5 Presunciones y dependencias (2.5 de la SRS)

Esta subsección de la SRS debería listar cada uno de los factores que afectan los requerimientos establecidos en la SRS.

Estos factores no son restricciones de diseño sobre el software, sino, más bien, algunos cambios de estos que pueden afectar los requerimientos en la SRS. Por ejemplo, una presunción podría ser que un sistema operativo específico esté disponible para un hardware diseñado para el producto software. Si, de hecho, el sistema operativo no está disponible, la SRS deberá tener que cambiar acorde a esto.

3.2.6 Particionamiento de los requerimientos (2.6 de la SRS)

Esta subsección de la SRS debería identificar los requerimientos que serán demorados hasta futuras versiones del sistema.

3.3 Requerimientos específicos (Sección 3 de la SRS)

Esta sección de la SRS debería contener todos los requerimientos del software a un nivel de detalle suficiente para asegurar que los diseñadores diseñen un sistema que satisfaga estos requerimientos, y hacer que los "testers" (probadores) prueben que el sistema satisface estos requerimientos. A través de esta sección, cada requisito establecido debería ser percibido externamente por un usuario, operador u otro sistema externo. Estos requerimientos deberían ser incluidos con una mínima descripción de cada entrada (el estímulo) dentro del sistema, cada salida (respuesta) del sistema, y todas las funciones realizadas por el sistema en respuesta a la entrada o como soporte de una salida. Esta es a menudo la sección más larga y más importante de la SRS, los siguientes principios deberían aplicarse:

a) Los requisitos específicos deberían ser establecidos en conformidad con todas las características descritas en 2.3.

b) los requisitos específicos deberían tener referencias cruzadas a documentos anteriores que están relacionados.

c) Todos los requerimientos deberían ser identificados unívocamente.

d) Se debería prestar atención a la organización de los requerimientos para maximizar la legibilidad.

Antes de examinar la manera específica de la organización de los requerimientos es útil entender los varios ítems que comprenden los requisitos como son descriptos desde 3.3.1 hasta 3.3.7.

3.3.1 Interfaces externas

Aquí se deberían detallar todas las entradas y salidas del sistema del software. Debería completarse la descripción de las interfaces en 3.2 y debería no repetirse a información aquí.

Debería incluirse tanto contenidos y formatos como sigue:

a) Nombre del ítem;

b) Descripción de propósito;

c) Fuente de entrada o destino de la salida;

d) Rango válido, adecuación, y/o tolerancia;

e) Unidades de medida;

f) Tiempos;

g) Relaciones con otras entradas/salidas;

h) Formato de pantallas u organización;

i) Formato de ventanas u organización;

j) Formato de datos;

k) Formato de comandos;

l) Mensajes finales.

3.3.2 Funciones

Los requerimientos funcionales deberían definir la acción fundamental que debe realizar el software, en aceptación y procesamiento de la entrada, y en el procesamiento y generación de las salidas. Esto es generalmente listado con la sentencia "deberá" que empieza con "El sistema deberá…."

Esto incluye:

a) Prueba de validación de la entrada

b) Secuencia exacta de operaciones

c) Respuesta de situaciones anormales, incluyendo

1) overflow (sobredimensionamiento)

2) facilidad de comunicación

3) manejo de errores y recuperación

d) Efecto de los parámetros

e) Relaciones de las salidas y las entradas, incluyendo

1) Secuencia de entrada/salidas

2) Fórmulas de entrada para conversión de las salidas

Puede ser apropiado particionar los requerimientos funcionales en subfunciones o subprocesos.

Esto no implica que el diseño del software deba ser particionado de esta manera.

3.3.3 Requerimientos de performance

Esta subsección debería especificar los requerimientos numéricos estáticos como los dinámicos puestos sobre el software o sobre la interacción humana con el software como un todo. Los requerimientos numéricos estáticos pueden incluir lo siguiente:

a) Cantidad de terminales a ser soportadas;

b) El número de usuarios simultáneos a ser soportados;

c) El tamaño y tipo de información manejada.

Algunas veces los requerimientos numéricos estáticos son identificados bajo una sección separada titulada Capacidades. Los requerimientos numéricos dinámicos pueden incluir por ejemplo la cantidad de transacciones, tareas y un tamaño de datos a ser procesado dentro de ciertos periodos de tiempo tanto para una operación normal como para condiciones de carga de trabajo profunda.

Todos estos requisitos deberían ser establecidos en términos medibles. Por ejemplo,

95% de las transacciones deberían ser procesadas en menos de 1 seg., en lugar de

Un operador debería no tener que esperar para que una transacción se complete.

NOTA: Los límites numéricos aplicados a una función específica son habitualmente especificados como parte de la descripción del proceso de tal función.

3.3.4 Requerimientos de las bases de datos lógicas

Esto debería especificar los requerimientos lógicos para alguna información que es puesta dentro de la Base de datos. Esto puede incluir lo siguiente:

a) Tipo de información usada en varias funciones;

b) Frecuencia de uso;

c) Capacidades de accesibilidad;

d) Entidades de datos y sus relaciones;

e) Restricciones de integridad;

f) Requerimientos de retención de datos.(Tiempo que hay que guardar los datos).

3.3.5 Restricciones de diseño.

Esto debería especificar las restricciones del diseño que pueden ser impuestas por otros estándares, las limitaciones del hardware, etc.,

3.3.5.1 Cumplimiento con estándares

Esta subsección debería especificar los requisitos derivados de los estándares existentes o regulaciones. Ellos pueden incluir lo siguiente:

a) Formato del reporte;

b) Nombres de los datos;

c) Procedimientos de contabilidad;

d) Seguimientos de Auditoría.

Por ejemplo, aquí se podrían especificar los requerimientos para el seguimiento de las actividades del proceso del software. Tales seguimientos son necesarios para algunas aplicaciones que alcance regulación mínima o estándares financieros. Un requerimiento de seguimiento de auditoría puede por ejemplo establecer que todos los cambios a la base de

datos de nómina de pagos deben ser registrados en un archivo de seguimiento con el valor anterior y el nuevo.

3.3.6 Atributos del sistema software.

Hay varios atributos del software que pueden servir como requisitos. Es importante que los atributos requeridos sean especificados para que su logro sea objetivamente verificado.

Subclaúsulas 3.3.6.1 hasta 3.3.6.5 proporcionan una lista parcial de ejemplos.

3.3.6.1 Fiabilidad

Esto debería especificar los factores requeridos para establecer la fiabilidad requerida del sistema software al momento de la entrega.

3.3.6.2 Disponibilidad

Esto debería especificar que los factores son requeridos para garantizar un nivel de disponibilidad definido para el sistema entero tales como puntos de chequeo, recuperación y reinicio.

3.3.6.3 Seguridad

Esto debería especificar los factores que protegen al software de accesos accidentales o maliciosos, usos, modificaciones, destrucción o improvisación.. Requerimientos específicos en esta área podrían incluir la necesidad para:

a) Utilizar ciertas técnicas criptográficas;

b) Mantener un Log específicos o un conjunto de datos históricos;

c) Asignar ciertas funciones a módulos diferentes;

d) Restricciones de comunicaciones entre algunas áreas del programa;

e) Datos de chequeo de integridad de variables críticas.

3.3.6.4 Mantenibilidad

Aquí se debería especificar atributos de software que relacionan la facilidad de mantenimiento del software en si misma. Puede haber algún requisito para cierta modularidad, interfaces, la complejidad, etc. Estos requerimientos no deberían ser puestos aquí, porque son conceptos de buenas prácticas de diseño.

3.3.6.5 Portabilidad

Aquí se deberían especificar atributos de software relacionados con lo fácil de portar del software a otra máquina anfitrión y/o sistema operativo. Esto puede incluir lo siguiente:

a) Porcentaje de componentes con código dependiente del host ;

b) Porcentaje de código que es dependiente del Host;

c) Uso de un lenguaje portable;

d) Uso de un particular compilador o subconjunto de lenguajes;

e) Uso de un sistema operativo particular.

3.3.7 Organización de los requerimientos específicos.

Para cualquier sistema trivial, los requerimientos detallados tienden a ser extensos. Por esta razón, es recomendable que se considere cuidadosamente la organización dada, para que se haga de una manera óptima para el entendimiento. No hay una organización óptima para todos los sistemas, diferentes clases de sistemas se prestan para diferentes organizaciones de requisitos en la sección 3 de la SRS. Algunas de estas organizaciones se describen desde 3.3.7.1 hasta 3.3.7.7.

3.3.7.1 Modo sistema

Algunos sistemas se comportan muy diferente dependiendo del modo de operación. Por ejemplo, un sistema de control puede tener diferentes conjuntos de funciones dependiendo de su modo: entrenamiento, normal o emergencia. Al organizar esta sección por el modo, se debería utilizar el borrador en A.1 o A.2. Esta elección depende de si las interfaces y la performance son dependientes del modo.

3.3.7.2 Clases de usuario

Algunos sistemas proveen diferentes conjuntos de funciones para diferentes clases de usuarios. Por ejemplo, un sistema de control de ascensor presenta las capacidades diferentes a los pasajeros, obreros de mantenimiento y bomberos. Al organizar esta sección por clase del usuario, debería usarse el borrador en A.3.

3.3.7.3 Objetos

Los objetos son entidades del mundo real que tienen una contraparte dentro del sistema.

Por ejemplo, en un sistema que supervisa pacientes, los objetos incluyen a los pacientes, los sensores, enfermeras, los cuartos, médicos, las medicinas, etc. Asociado con cada objeto hay un conjunto de atributos (de estos objetos) y funciones (realizadas por ese objeto). Estas funciones también se llaman servicios, métodos o procesos. Al organizar esta sección por objetos, el borrador en A.4 debería usarse. Nótese que conjuntos de objetos pueden compartir atributos y servicios. Éstos se agrupan como las clases.

3.3.7.4 Características

Una característica es un servicio deseado externamente de un sistema que puede requerir una secuencia de entradas para efectuar un resultado deseado. Por ejemplo, en un sistema de teléfonos, las características incluyen llamada local, transferencia de llamada y llamada en conferencia. Cada característica se describe generalmente en una secuencia de estímulo – respuesta. Cuando se organiza esta sección por característica se debería utilizar el borrador en A.6.

3.3.7.5 Estímulo

Algunos sistemas pueden ser mejor organizados describiendo sus funciones en término de estímulos. Por ejemplo, las funciones de un sistema de aterrizaje automático de un avión, puede ser organizado dentro de secciones : pérdida de potencia, el cambio súbito en el destino, la velocidad vertical excesiva, etc. Al organizar esta sección por estímulo, el borrador en A.6 debería usarse.

3.3.7.6 Respuestas

Algunos sistemas pueden organizarse mejor describiendo todas las funciones que ayudan para la generación de una respuesta. Por ejemplo, pueden organizarse las funciones de un sistema del personal en secciones que corresponden a todas las funciones asociadas con la generación del pagos, todas las funciones asociadas con la generación de la lista actual de empleados, etc. Se podría usar el borrador en A.6 (con todas las ocurrencias de estímulo reemplazadas por respuesta).

3.3.7.7 Jerarquía Funcional

Cuando ninguno de los esquemas organizacionales anteriores provee ayuda, la funcionalidad global puede organizarse dentro de una jerarquía de funciones organizadas por cualquier entrada común, salida común o acceso a datos internos comunes. Los diagramas de Flujo de Datos y diccionarios de datos pueden ser usados para mostrar las relaciones entre las funciones y los datos. Cuando se organiza esta sección por jerarquía funcional, se podría utilizar el borrador en A.7.

3.3.8 Comentarios adicionales

Cuando se piensa en una nueva SRS, más de una técnica organizacional dada en 3.3.7.7 puede parecer apropiada. En tales casos, organice los requisitos específicos por jerarquías múltiples haciéndolo a la medida de las necesidades específicas del sistema bajo especificación.

Por ejemplo, en A.8 se ha combinado la organización clases de usuarios y características. Algunos requerimientos adicionales pueden ser puestos en una sección separada al final de la SRS.

Hay muchas notaciones, métodos y herramientas de soporte automatizadas disponibles para ayudar en la documentación de requerimientos. La mayor parte, de su utilidad es una función de organización. Por ejemplo, máquinas de estado finitas o diagramas de estado pueden ser utiles cuando se organiza por modo; en análisis orientado a objetos puede proveer ayuda si se organiza por objetos, las secuencias de estímulo-respuesta pueden proveer ayuda cuando se organiza por característica y al organizar por la jerarquía funcional, los diagramas de flujo de datos y los diccionarios de datos pueden ser de utilidad.

En alguno de los borradores dados desde A.1 hasta de A.8, aquellas secciones llamadas "Requerimientos Funcionales i " pueden ser descriptos en lenguaje nativo (por ejemplo, castellano), en pseudo código, en un lenguaje de definición de sistema, o en cuatro subdivisiones tituladas: Introducción, Entradas, Procesamiento, y Salidas.

3.4 Información de soporte

La información de soporte hace a la SRS fácil de usar. Incluye lo siguiente:

a) Tabla de contenidos;

b) Índices;

c) Apéndice.

3.4.1 Tabla de contenidos e índices

La tabla de contenidos e índices son muy importantes y debería seguir las prácticas de composición general.

3.4.2 Apéndices

Los apéndices no siempre son considerados parte de la SRS actual y no siempre son necesarios. Ellos pueden incluir:

a) Ejemplos de formatos de las entradas/salidas, descripciones del análisis del costo, resultados de intervenciones de los usuarios;

b) Información de soporte o contextual que puede ayudar a los lectores de la SRS;

c) Una descripción de los problemas a ser resuelto por el software;

d) las instrucciones de empaquetamiento especiales para el código y los medios de seguridad a ser alcanzados, exportar, carga inicial u otros requisitos.

Cuando los apéndices son incluidos, la SRS debería específicamente establecer cuando los apéndices son considerados parte de los requerimientos y cuando no.

Anexo A

Plantillas de la SRS

A.1 Plantilla de la SRS Sección 3 organizada por el modo: Versión 1

3. Requisitos específicos
    - 3.1 requisitos de interfaces externas
        - 3.1.1 interfaz con el usuario
        - 3.1.2 interfaces de hardware
        - 3.1.3 interfaces del software
        - 3.1.4 interfaces de comunicaciones
    - 3.2 requisitos funcionales
        - 3.2.1 modo 1
            - 3.2.1.1 requisito funcional 1.1
            - .
            - .
            - .
            - 3.2.1.n requisito Funcional 1.n
        - 3.2.2 modo 2
        - .
        - .
        - .
        - 3.2.m Modo m
            - 3.2.m.1 requisito Funcional m.1
            - .
            - .

.
3.2.m.n requisito Funcional m.n
3.3 Requisitos de performance
3.4 Restricciones de diseño
3.5 Atributos de sistema software
3.6 Otros requisitos

A.2Plantilla de la SRS Sección 3 organizada por modo: Versión 2

3. Requisitos específicos
3.1. Requisitos funcionales
3.1.1 modo 1
3.1.1.1 interfaces externas
3.1.1.1.1 interfaz con el usuario
3.1.1.1.2 interfaces de hardware
3.1.1.1.3 interfaces con el software
3.1.1.1.4 interfaces de comunicaciones
3.1.1.2 requisitos funcionales
3.1.1.2.1 requisito funcional 1
.
.
3.1.1.2.n requisito Funcional n
3.1.1.3 Performance
3.1.2 Modo 2
.
.
.
3.1.m Modo m
3.2 Restricciones de diseño
3.3 Atributos de sistema software
3.4 Otros requisitos

A.3 Plantilla de la SRS Sección 3 organizada por clase de usuario

3. Requisitos específicos
3.1 requisitos de interfaces externas
3.1.1 interfaz con el usuario
3.1.2 interfaces de hardware
3.1.3 interfaces del software
3.1.4 interfaces de comunicaciones
3.2 requisitos funcionales
3.2.1 Clase de usuario 1
3.2.1.1 requisito funcional 1.1
.
.
.
3.2.1.n requisito Funcional 1.n
3.2.2 Clase de usuario 2
.
.
.

3.2.m Clase de usuario m
3.2.m.1 requisito Funcional m.1
.
.
.
3.2.m.n requisito Funcional m.n
3.3 Requisitos de performance
3.4 Restricciones de diseño
3.5 Atributos de sistema software
3.6 Otros requisitos

A.4 Plantilla de la SRS Sección 3 organizada por objeto

3. Requisitos específicos
3.1 requisitos de interfaces externas
3.1.1 interfaz con el usuario
3.1.2 interfaces de hardware
3.1.3 interfaces del software
3.1.4 interfaces de comunicaciones
3.2 Clases/Objetos
3.2.1 Clase/Objeto 1
3.2.1.1 atributos (directo o heredado)
3.2.1.1.1 atributo 1
.
.
.
3.2.1.1.n Atributo n
3.2.1.2 funciones (los servicios, los métodos, directos o heredados)
3.2.1.2.1 requisito funcional 1.1
.
.
.
3.2.1.2.m requisito Funcional 1.m
3.2.1.3 Mensajes (comunicaciones recibidas o enviadas)
3.2.2 Clase/Objeto 2
.
.
.
3.2.p Clase/Objeto p
3.3 Requisitos de performance
3.4 Restricciones de diseño
3.5 Atributos del sistema software
3.6 Otros requisitos

A.5 Plantilla de la SRS Sección 3 organizada por características

3. Requisitos específicos
3.1 requisitos de interfaces externas
3.1.1 interfaz con el usuario
3.1.2 interfaces de hardware
3.1.3 interfaces del software

3.1.4 interfaces de comunicaciones
3.2 Características del sistema
3.2.1 Característica 1
3.2.1.1 Introducción / Propósito de la característica
3.2.1.2 Secuencia de estímulo / Respuesta
3.2.1.3 Requisitos funcionales asociados
3.2.1.3.1 Requisito funcional 1
.
.
.
3.2.1.3.n Requisito funcional n
3.2.2 Característica del sistema 2
.
.
.
3.2.m Característica del Sistema m
.
.
.
3.3 Requisitos de performance
3.4 Restricciones de diseño
3.5 Atributos del sistema software
3.6 Otros requisitos

A.6 Plantilla de la SRS Sección 3 organizada por estímulo

3. Requisitos específicos
3.1 requisitos de interfaces externas
3.1.1 interfaz con el usuario
3.1.2 interfaces de hardware
3.1.3 interfaces del software
3.1.4 interfaces de comunicaciones
3.2 Requisitos funcionales
3.2.1 Estímulo 1
3.2.1.1 Requisito funcional 1.1
.
.
.
3.2.1.n Requisito funcional 1.n
3.2.2 Estímulo 2
.
.
.
3.2.m Estímulo m
3.2.m.1 Requisito funcional m.1
.
.
.
3.2.m.n Requisito funcional m.n
3.3 Requisitos de performance

3.4 Restricciones de diseño
3.5 Atributos del sistema software
3.6 Otros requisitos

A.7 Plantilla de la SRS Sección 3 organizada por jerarquía funcional

3. Requisitos específicos
- 3.1 requisitos de interfaces externas
  - 3.1.1 interfaz con el usuario
  - 3.1.2 interfaces de hardware
  - 3.1.3 interfaces del software
  - 3.1.4 interfaces de comunicaciones
- 3.2 Requisitos funcionales
  - 3.2.1 Flujo de información
    - 3.2.1.1 Diagrama Flujo de datos 1
      - 3.2.1.1.1 Entidades de datos
      - 3.2.1.1.2 Procesos pertinentes
      - 3.2.1.1.3 Topología
    - 3.2.1.2 Diagrama Flujo de datos 2
      - 3.2.1.2.1 Entidades de datos
      - 3.2.1.2.2 Procesos pertinentes
      - 3.2.1.2.3 Topología
    - .
    - .
    - .
    - 3.2.1.n Diagrama Flujo de datos n
      - 3.2.1.n.1 Entidades de Datos
      - 3.2.1.n.2 Procesos Pertinentes
      - 3.2.1.n.3 Topología
  - 3.2.2 Descripciones de procesos
    - 3.2.2.1 Proceso 1
      - 3.2.2.1.1 Entidades de datos de entrada
      - 3.2.2.1.2 Algoritmo o fórmula del proceso
      - 3.2.2.1.3 Entidades de datos afectados
    - 3.2.2.2 Proceso 2
      - 3.2.2.2.1 Entidades de datos de entrada
      - 3.2.2.2.2 Algoritmo o fórmula del proceso
      - 3.2.2.2.3 Entidades de datos afectados
  - .
  - .
  - .
  - 3.2.2.m Proceso m
    - 3.2.2.m.1 Entidades de datos de Entrada
    - 3.2.2.m.2 Algoritmo o fórmula del proceso
    - 3.2.2.m.3 Entidades de datos Afectados
  - 3.2.3 Especificaciones de las construcciones de los datos
    - 3.2.3.1 Construcción 1
      - 3.2.3.1.1 Tipo de registro
      - 3.2.3.1.2 Campos constitutivos
    - 3.2.3.2 Construcción 2

3.2.3.2.1 Tipo de registro
3.2.3.2.2 Campos constitutivos.
.
.
3.2.3.p Construcción p
3.2.3.p.1 Tipo de Registro
3.2.3.p.2 Campos constitutivos
3.2.4 Diccionario de datos
3.2.4.1 Elemento Dato 1
3.2.4.1.1 Nombre
3.2.4.1.2 Representación
3.2.4.1.3 Unidad/Formato
3.2.4.1.4 Precisión/Validación
3.2.4.1.5 Rango
3.2.4.2 Elemento Dato 2
3.2.4.2.1 Nombre
3.2.4.2.2 Representación
3.2.4.2.3 Unidad/Formato
3.2.4.2.4 Precisión/Validación
3.2.4.2.5 Rango
.
.
3.2.4.q Elemento Datos q
3.2.4.q.1 Nombre
3.2.4.q.2 Representación
3.2.4.q.3 Unidad/Formato
3.2.4.q.4 Precisión/Validación
3.2.4.q.5 Rango
3.3 Requisitos de desarrollo
3.4 Restricciones de diseño
3.5 Atributos del sistema software
3.6 Otros requisitos

A.8 Plantilla de la Sección de SRS 3 mostrando múltiples organizaciones

3. Requisitos específicos
3.1 requisitos de interfaces externas
3.1.1 interfaz con el usuario
3.1.2 interfaces de hardware
3.1.3 interfaces del software
3.1.4 interfaces de comunicaciones
3.2 Requisitos funcionales
3.2.1 Clase de Usuario 1
3.2.1.1 Característica 1.1
3.2.1.1.1 Introducción / Propósito de la característica
3.2.1.1.2 secuencia de estímulo /Respuesta
3.2.1.1.3 requisitos funcionales asociados
3.2.1.2 Característica 1.2
3.2.1.2.1 Introducción / Propósito de la característica
3.2.1.2.2 Secuencia de estímulo/ Respuesta

3.2.1.2.3 Requisitos funcionales asociados

.
.
.

3.2.1.m Característica 1.m

3.2.1.m.1 Introducción / Propósito de la característica
3.2.1.m.2 Secuencia de estímulo /Respuesta
3.2.1.m.3 Requisitos funcionales Asociados

3.2.2 Clase de usuario 2

.
.
.

3.2.n Clase de usuario n

.
.
.

3.3 Requisitos de desarrollo
3.4 Restricciones de diseño
3.5 Atributos del sistema software
3.6 Otros requisitos

Bibliografía

IEEE, “An American National Standard. IEEE Guide to Software Requirements Specifications. ANSI/IEEE Std 830-1984”, en Dorfman, M., Thayer, R.H., *Standards, Guidelines, and Examples on Systems and Software Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, 1990

Trabajos Prácticos Obligatorios:

1	**Tema**	Especificación de Requerimientos
	Título	Distribuidora XX
	Consigna	Dada la siguiente nota de relevamiento, especifique los Requerimientos del sistema.

La siguiente nota de relevamiento es el producto de una serie de entrevistas realizadas en la empresa de distribución "XX" para la cual tenemos que diseñar un sistema de ventas.

La empresa se dedica a la compra y venta de productos. Tiene una cartera de aproximadamente 2000 clientes. Los clientes son captados por sus representantes de ventas quienes concurren al local del posible cliente y le ofrecen comercializar los productos que distribuye la empresa. Una vez que el cliente efectúa el primer pedido, se lo incorpora como cliente ante lo cual el representante de ventas debe entregar en la administración una ficha en la que completo todos los datos comerciales del nuevo cliente. La administración controla que el nuevo cliente no haya tenido relaciones comerciales con la distribuidora con anterioridad, de ser así, se debe consultar en los archivos que no haya quedado con saldo deudor en aquel momento pues entonces se requerirá que cancele esa deuda antes de volverlo a habilitar como cliente. Si no tuvo relación comercial anterior se le asigna un código de cliente en forma correlativa secuencial, sino mantiene su código anterior.

Los pedidos de productos son tomados diariamente por los representantes de venta que visitan a los clientes en su local comercial. Cada representante de venta tiene asignada una determinada zona de venta y visita con diferente periodicidad a los clientes de cada zona dependiendo de la densidad de población que existe en cada una. Cuando el cliente efectúa el pedido se registra en el formulario de pedidos que es una única planilla en la cual, a renglón seguido, se asienta lo que solicita cada cliente.

Terminado su recorrido de ventas diario, cada representante de ventas entrega en administración la planilla de pedidos y las fichas de los nuevos clientes que hizo en ese día. La administración separa las fichas de los nuevos clientes para que el encargado administrativo verifique si tuvieron o no cuenta anteriormente. Las planillas de pedidos son procesadas por un empleado que las ingresa en un sistema debiendo para ello codificar cada uno de los productos que solicitaron los clientes. El empleado ingresa el código del cliente, el código del artículo y la cantidad pedida el sistema le informa si el código de cliente corresponde a un cliente existente, si está habilitado para efectuar pedidos, si el código del producto corresponde a uno existente y si la cantidad solicitada puede ser satisfecha con el stock actual. En el caso de que la cantidad pedida supere la existencia de ese producto, el sistema, informa la cantidad disponible para que el empleado decida si rechaza el pedido o le envía la cantidad que se tiene disponible. El empleado, por expresas instrucciones de la gerencia, debe rechazar el pedido si la diferencia entre lo solicitado y la existencia es superior al 50% de la cantidad requerida.

Una vez que se termina la carga de la planilla de pedidos, el sistema le informa la cantidad total de productos por cada tipo y el empleado debe verificar la coincidencia de esos totales con los que efectuó el representante de ventas al cerrar la planilla. Si no hay coincidencia, verifica cada uno de los pedidos ingresados hasta encontrar el error.

Una vez que se procesaron la totalidad de las planillas de venta que entregaron los representantes en ese día, se procede a la emisión de las facturas. El sistema de facturación, verifica que el monto que totaliza la venta por cliente, sumado al saldo que el mismo posea en su cuenta corriente, no supere el límite de crédito que tiene asignado por la gerencia. De superarse dicho límite, el sistema no emite la factura generando un informe de pedidos rechazados por falta de crédito. En función de la condición ante el IVA del cliente el sistema emitirá comprobantes A o B de facturación.

El monto de la factura es agregado, automáticamente, a la cuenta corriente del cliente como un débito. La factura se confecciona con los precios de venta que tiene fijado cada producto los que son determinados y decididos por la Gerencia.

Diariamente, se efectúan los repartos de los productos entregándose al cliente el original de la factura y debiendo el mismo firmar con la leyenda de "recibido" el duplicado que deberá entregar el repartidor en administración al final de su recorrido.

El repartidor, conjuntamente con las facturas correspondientes a la mercadería que debe entregar, recibe un listado de cobranzas que emite la administración donde consta cada uno de los clientes que debe visitar y el saldo en cuenta corriente que poseen a la fecha. En el momento que le entrega a cada cliente lo solicitado, le debe reclamar el pago del saldo adeudado. Si el cliente paga la totalidad de lo adeudado, tacha en el listado de cobranza el importe de la deuda, si paga solo una parte de la misma anota al lado del importe de deuda el monto que pago.

Cuando el repartidor termina su recorrido entrega en la administración los duplicados de las facturas con la firma de recepción por parte del cliente y el listado de cobranza. La administración totaliza el listado de cobranza y recibe del repartidor la recaudación la cual deberá coincidir con la sumatoria del listado. De no coincidir el repartidor deberá completar el faltante.

La administración archiva los duplicados de las facturas y procesa todos los listados de cobranzas asentando en la cuenta corriente de cada cliente el crédito correspondiente. Al final de la jornada, deberá emitir un listado para Gerencia con las deudas de clientes no canceladas.

Mensualmente, la administración emite el libro de IVA ventas en el cual constan todas las ventas realizadas en el período el que es remitido al estudio contable que se responsabiliza por la gestión impositiva de la empresa. Por cada comprobante que se asienta en el libro de IVA consta la fecha de emisión, el número, el tipo (A o B), su número, la razón social del cliente, su número de CUIT, el total del comprobante, el neto gravado y el IVA (que en todos los casos es del 21%).

De los formularios recolectados durante el relevamiento, se desprende el siguiente listado de datos que maneja la empresa en el circuito anteriormente descripto, debiendo considerar al mismo tiempo los que ya se han detallado:

Código de cliente
Razón social
Domicilio del cliente
Código de zona del cliente
Descripción de la zona
Condición ante el IVA
Número de Cuit
Límite de crédito
Saldo del cliente
Teléfonos del cliente (*)
Fecha del pedido
Renglones(*)
    Código de cliente
    Código de producto
    Descripción del producto
    Cantidad
Fecha de factura
Tipo de factura (A/B)
Número de comprobante
Código del cliente
Razón social del cliente
Domicilio
Condición ante el IVA
Número de CUIT
Renglones(*)
    Código de producto
    Descripción del producto
    Cantidad
    Precio unitario
    Total del renglón
Neto gravado
IVA
Total de la factura