

Índice

INTRODUCCIÓN.

1. ANÁLISIS DE VIABILIDAD TÉCNICA Y ECONÓMICA.

- 1.1. Estudio de viabilidad del sistema en métrica 3.
- 1.2. Actividad EVS 1: Establecimiento del alcance del sistema.
- 1.3. Actividad EVS 2: Estudio de la situación actual.
- 1.4. Actividad EVS 3: Definición de requisitos del sistema.
- 1.5. Actividad EVS 4: Estudio de alternativas de solución.
- 1.6. Actividad EVS 5: Valoración de las alternativas.
- 1.7. Actividad EVS 6: Selección de la solución.

2. ANÁLISIS DE SISTEMAS: ESPECIFICACIÓN FUNCIONAL DEL SISTEMA.

- 2.1. Análisis del sistema de información en métrica 3.

3. BÚSQUEDA Y DESCRIPCIÓN DE REQUISITOS FUNCIONALES.

3.1. Búsqueda de requisitos funcionales.

- 3.1.1. Entrevistas.
- 3.1.2. Reuniones.
- 3.1.3. Desarrollo conjunto de aplicaciones (JAD).
- 3.1.4. Planificación conjunta de requisitos (JRP).
- 3.1.5. Prototipado.
- 3.1.6. Observación.
- 3.1.7. Estudio de documentación.
- 3.1.8. Cuestionarios.
- 3.1.9. Tormenta de ideas (brainstorming).

3.2. Descripción de requisitos funcionales.

- 3.2.1. La especificación de requisitos del software (ERS).
- 3.2.2. Características de una buena ERS (IEEE STD. 830).
- 3.2.3. Estructura de la ERS según el IEEE (STD. 830).
- 3.2.4. Evolución de la ERS (IEEE STD. 830).

4. ESPECIFICACIÓN DE SOLUCIONES TÉCNICAS.

CONCLUSIÓN.

INTRODUCCIÓN.

A lo largo del desarrollo de un sistema software desde el estudio de requisitos hasta la implementación y puesta en marcha del mismo, se contemplan una multitud de tareas complejas. El software ha evolucionado gracias a la ingeniería del software que pretende aplicar los principios de la ingeniería al desarrollo de software de forma que se sistematice la producción del mismo.

El ciclo de vida de desarrollo de software que propone Métrica V3, se divide en los siguientes procesos:

PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN.

DESARROLLO DE SISTEMAS DE INFORMACIÓN.

MANTENIMIENTO DE SISTEMAS DE INFORMACIÓN.

En cuanto al Proceso de Desarrollo de Sistemas de Información, para facilitar la comprensión y dada su amplitud y complejidad se ha subdividido en cinco procesos:

ESTUDIO DE VIABILIDAD DEL SISTEMA (EVS).

ANÁLISIS DEL SISTEMA DE INFORMACIÓN (ASI).

DISEÑO DEL SISTEMA DE INFORMACIÓN (DSI).

CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI).

IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA (IAS).

Una vez visto que el desarrollo es viable, dentro del proceso de análisis la especificación funcional del sistema es fundamental para que tanto el cliente como el desarrollador acuerden el producto que se desea. De lo contrario puede haber discrepancias que, detectadas sólo al final, pueden ser desastrosas.

1. ANÁLISIS DE VIABILIDAD TÉCNICA Y ECONÓMICA.

1.1. Estudio de viabilidad del sistema en métrica 3.

El estudio de viabilidad es previo al Análisis del Sistema de Información.

El objetivo del Estudio de Viabilidad del Sistema (EVS) es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas.



A partir del estado inicial, la situación actual y los requisitos planteados, se estudian las alternativas de solución. Se describe cada una de las alternativas, indicando los requisitos que cubre. Una vez descritas, se valora su impacto en la organización la inversión a realizar en cada caso y los riesgos asociados. Esta información se analiza con el fin de evaluar las distintas alternativas y seleccionar la más adecuada, definiendo y estableciendo su planificación.

1.2. Actividad EVS 1: Establecimiento del alcance del sistema.

En esta actividad se estudia el alcance de la necesidad planteada por el cliente o usuario realizando una descripción general de la misma. Se determinan los objetivos, se inicia el estudio de los requisitos y se identifican las unidades organizativas afectadas estableciendo su estructura.

1.3. Actividad EVS 2: Estudio de la situación actual.

Como resultado de esta actividad se genera un diagnóstico, estimando la eficiencia de los sistemas de información existentes e identificando los posibles problemas y las mejoras.

1.4. Actividad EVS 3: Definición de requisitos del sistema.

Determinación de los requisitos generales, mediante una serie de sesiones de trabajo con los usuarios participantes. los requisitos y sus prioridades, que se añaden al catálogo de requisitos que servirá para el estudio y valoración de las distintas alternativas de solución que se propongan.

1.5. Actividad EVS 4: Estudio de alternativas de solución.

Propone diversas alternativas que respondan satisfactoriamente a los requisitos planteados.

1.6. Actividad EVS 5: Valoración de las alternativas.

La valoración de las alternativas, considera el impacto en la organización, tanto desde el punto de vista tecnológico y organizativo como de operación, y los posibles beneficios que se esperan contrastados con los costes asociados.

Tareas asociadas a la actividad EVS5:

Tarea EVS 5.1: Estudio de la inversión.

Tarea EVS 5.2: Estudio de los riesgos.

Tarea EVS 5.3: Planificación de alternativas.

1.7. Actividad EVS 6: Selección de la solución.

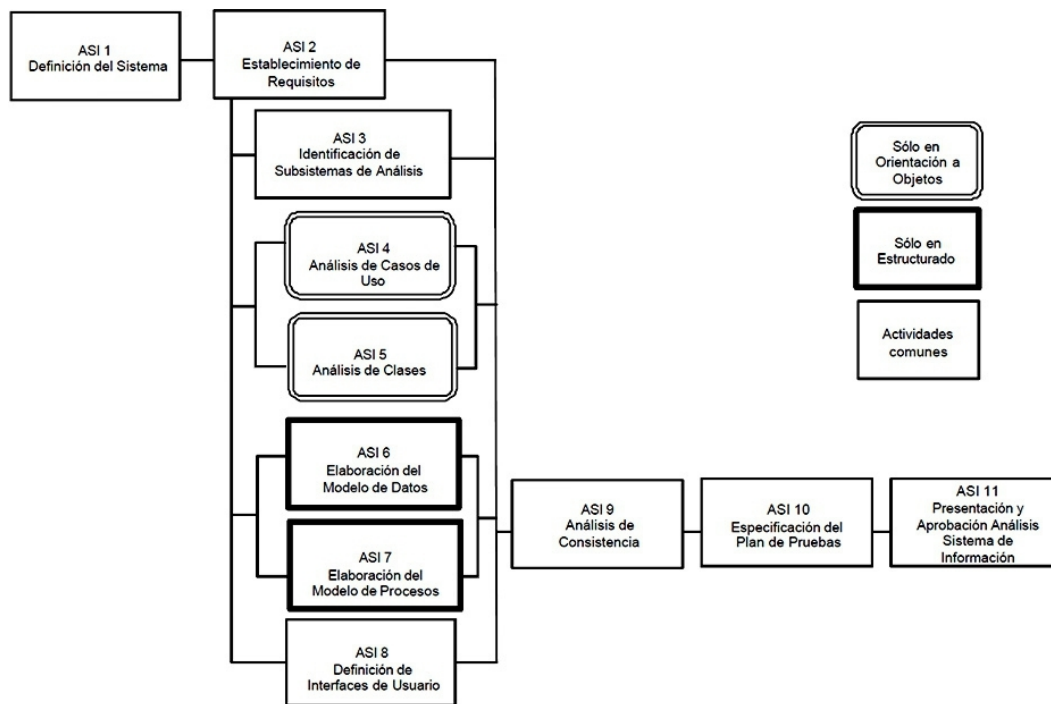
En esta actividad se aprueba la solución o se determina su inviabilidad.

2. ANÁLISIS DE SISTEMAS: ESPECIFICACIÓN FUNCIONAL DEL SISTEMA.

2.1. Análisis del sistema de información en métrica 3.

El objetivo del proceso de análisis del sistema de información (ASI) es la obtención de una especificación detallada del sistema de información que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema.

En la primera actividad, **Definición del Sistema (ASI1)**, se lleva a cabo la descripción inicial del sistema de información a partir de los productos generados en el proceso de Estudio de la Viabilidad del Sistema.



3. BÚSQUEDA Y DESCRIPCIÓN DE REQUISITOS FUNCIONALES.

3.1. Búsqueda de requisitos funcionales.

Las técnicas de recogida de información (requirements elicitation) surgen como un medio para mejorar la comunicación entre usuarios/clientes y los desarrolladores de software.

Para facilitar la colaboración entre usuarios y desarrolladores en el proceso de análisis se recurre a las técnicas de comunicación y recopilación de información. En la práctica es habitual utilizar combinaciones de varias de estas técnicas para recoger información de los usuarios (por ejemplo, brainstorming en sesiones de JAD). Las principales son:

- 3.1.1. Entrevistas.
- 3.1.2. Reuniones.
- 3.1.3. Desarrollo conjunto de aplicaciones (JAD).
- 3.1.4. Planificación conjunta de requisitos (JRP).
- 3.1.5. Prototipado.
- 3.1.6. Observación.
- 3.1.7. Estudio de documentación.
- 3.1.8. Cuestionarios.
- 3.1.9. Tormenta de ideas (brainstorming).

3.2. Descripción de requisitos funcionales.

3.2.1. La especificación de requisitos del software (ERS).

Se trata de especificar lo que desea el usuario sin considerar cómo se va a solucionar, aunque la ERS sí puede limitar la variedad de soluciones de diseño aplicables.

3.2.2. Características de una buena ERS (IEEE STD. 830).

1. **Correcta:** Una ERS es correcta si y sólo si cada requisito establecido es alguno de los que debe satisfacer el software.
2. **No ambigua.** Si un término puede tener varios significados, debe incluirse en un glosario en el que se detalle específicamente que se quiere decir con él.
3. **Completa.**
4. **Consistente.**
5. **Graduada en importancia.**
6. **Verificable**
7. **Fácil de modificar.**
8. **Trazable**

3.2.3. Estructura de la ERS según el IEEE (STD. 830).

1. Introducción.
 - 1.1. Propósito.
 - 1.2. Ámbito.
 - 1.3. Definiciones, acrónimos y abreviaturas.
 - 1.4. Referencias.
 - 1.5. Visión general.
2. Descripción general.
 - 2.1. Perspectiva del producto.
 - 2.2. Funciones del producto.
 - 2.3. Características del usuario.
 - 2.4. Restricciones.
 - 2.5. Asunciones y dependencias.
 - 2.6. Retraso de requisitos.
3. Requisitos específicos.
- Apéndices.
- Índice.

Una ERS no ha de seguir obligatoriamente este esquema pero cualquier ERS que esté bien escrita debería incluir toda la información que aquí se expone.

3.2.4. Evolución de la ERS (IEEE STD. 830).

La ERS normalmente necesitará modificarse a medida que progresa el producto software. Es casi imposible especificar algunos detalles en el momento en el que se inicia el proyecto.

4. ESPECIFICACIÓN DE SOLUCIONES TÉCNICAS.

Las especificaciones técnicas se pueden representar de diversas formas:

- **Gráficas:** cada técnica gráfica utiliza una serie de iconos en el que cada uno representa un componente de un aspecto particular del modelo.
- **Textuales:** se usan para especificar, con más detalle, los componentes definidos en los gráficos mediante una gramática definida.
- **Marcos o plantillas ("templates"):** especifican la información relativa a un componente de un modelo que ha sido declarado en un diagrama.
- **Matriciales:** permiten estudiar las referencias cruzadas entre componentes.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Sirven de soporte para las metodologías de ingeniería del software, en todos los aspectos del ciclo de vida de desarrollo del software.

La siguiente clasificación es la más habitual basada en las fases del ciclo de desarrollo que cubren:

- **Upper CASE,** herramientas que ayudan en las fases de planificación, análisis de requisitos y estrategia del desarrollo, usando, entre otros diagramas UML.
- **Middle CASE,** herramientas para automatizar tareas en el análisis y diseño de la aplicación.
- **Lower CASE,** herramientas que semiautomatizan la generación de código, crean programas de detección de errores, soportan la depuración de programas y pruebas.

Permiten representar, gestionar y documentar los resultados de cada una de las fases del ciclo de desarrollo software. Permitiendo la comprobación y consistencia de los mismos y automatizar de este modo gran parte del proceso complejo.

CONCLUSIÓN.

Las especificaciones funcionales son documentos que amplían en detalle un requerimiento de desarrollo, incluyendo una descripción completa de la lógica esperada, los datos y los criterios de prueba, con el fin de que el desarrollador disponga de la información suficiente para que pueda diseñar una especificación técnica, construir el desarrollo y realizar las pruebas técnicas necesarias para asegurar la calidad del desarrollo entregado.

Si bien es necesario un correcto análisis de los requerimientos del negocio, para garantizar la calidad de la especificación funcional son necesarias además otras consideraciones de diseño que se deben tener en cuenta dependiendo del desarrollo requerido.

Además de una explicación clara de la forma en que se espera que funcione el desarrollo, es necesario identificar con claridad el tipo de desarrollo que se requiere y algunos elementos de diseño específicos para cada tipo de modo que el desarrollador pueda plasmar en su desarrollo estas especificaciones de forma precisa y eficaz.