

1.- Ingeniería del sistema

1.1.- Análisis del sistema

El análisis de sistemas se centra en todos los elementos del sistema y no solo en el software y se realiza con los siguientes objetivos:

1. Identificar las necesidades del cliente.
2. Evaluar la viabilidad del sistema.
3. Realizar un análisis económico.
4. Realizar un análisis técnico.
5. Asignar funciones al software, al hardware, a la gente, a la base de datos y a otros elementos del sistema.
6. Establecer restricciones de costo y tiempo.
7. Crear una definición del sistema.

1. Identificación de las necesidades:

El analista se entrevista con el cliente y su representante. Para empezar, el analista da asistencia al cliente definiendo los objetivos del sistema: la información que se va a obtener, la información que se va a suministrar, las funciones y el rendimiento requerido.

El analista debe distinguir entre lo que "necesita" el cliente (elementos críticos para la realización) y lo que "quiere" (elementos deseables pero no esenciales).

- a- Funcionamiento y rendimiento requerido.
- b- Aspectos de fiabilidad y calidad.
- c- Fines generales del sistema.
- d- Limitaciones de costo y agenda
- e- Requisitos de fabricación.
- f- Mercado y competencia.
- g- Tecnología disponible.
- h- Ampliaciones futuras.

La información recogida durante la etapa de identificación de necesidades se especifica en un "*documento de conceptos del sistema*".

2. Estudio de viabilidad:

El desarrollo de un sistema basado en computadora se caracteriza por la escasez de recursos y dificultad de cumplir los plazos de entrega. Es necesario y prudente evaluar la viabilidad de un proyecto lo antes posible.

Durante la ingeniería de sistemas centramos nuestra atención en cuatro áreas de interés básico:

- a- *Viabilidad económica*. Una evaluación del costo de desarrollo frente al beneficio final producido por el sistema.

b- *Viabilidad técnica*. Un estudio de la funcionalidad, el rendimiento y las restricciones que pueden afectar a la posibilidad de realización de un sistema aceptable.

c- *Viabilidad legal*. Una determinación de cualquier infracción, violación o ilegalidad que pudiera resultar del desarrollo del sistema.

d- *Alternativas*. Una evaluación de los enfoques alternativos para el desarrollo del sistema.

No será necesario llevar a cabo un *estudio de viabilidad* para sistemas en los que la justificación económica es obvia, el riesgo técnico es bajo, se esperan pocos problemas legales y no existe una alternativa razonable.

El estudio de viabilidad puede documentarse en un informe separado de los otros documentos importantes de gestión e incluirse como apéndice en la *especificación del sistema*.

3. Análisis económico

El análisis de costo-beneficio señala los costos del desarrollo del proyecto y los contrasta con los beneficios tangibles e intangibles del sistema.

4. Análisis técnico

Durante el análisis técnico, el analista evalúa los méritos técnicos del concepto de sistema, mientras que al mismo tiempo recoge información adicional sobre el rendimiento, fiabilidad, facilidad de mantenimiento y posibilidad de producción.

El análisis técnico empieza con una definición de la viabilidad técnica del sistema propuesto. ¿Que tecnologías se requieren para conseguir la funcionalidad y el rendimiento del sistema? ¿Que nuevos materiales, métodos, algoritmos o procesos se requieren y cuál es el riesgo de su desarrollo?

5. Asignación y compromisos

Cada función del sistema, con su rendimiento requerido y sus características de interfaz, es asignada a uno o más elementos del sistema.

6. Establecer restricciones de costo y tiempo.

La *restricción de tiempo* se refiere a la cantidad de tiempo disponible para completar un proyecto, sea en días cronológicos o en horas-persona.

La *restricción de coste* se refiere a la cantidad presupuestada para el proyecto, sea como costo a incurrir o costo de oportunidad a considerar

7. Crear una definición del sistema.

Cree una definición del sistema que forme el fundamento de todo el trabajo de Ingeniería.

La especificación del sistema

La *especificación del sistema* es un documento que sirve como base para la ingeniería del hardware, la ingeniería de software, la ingeniería de bases de datos y la ingeniería humana. Describe la función y el rendimiento de un subsistema basado en computadora y las restricciones que gobernarán su desarrollo. La especificación limita cada uno de los elementos del sistema asignado. La *especificación del sistema* también describe la información (control y datos) que sirve de entrada y de salida al sistema.

Revisión de la especificación del sistema

La revisión de la especificación del sistema evalúa la corrección de la definición contenida en la especificación del sistema. La revisión ha de ser realizada por el técnico y por el cliente, para asegurar que:

1. Se ha perfilado adecuadamente el ámbito del proyecto.
2. Se ha definido correctamente la funcionalidad, el rendimiento y las interfaces.
3. El análisis del entorno y del riesgo del desarrollo justifican el sistema.
4. El desarrollador y el cliente tienen los mismos objetivos.

La revisión de la especificación del sistema se realiza en dos partes. Inicialmente se aplica un punto de vista de gestión. Después se realiza una evaluación técnica de los elementos y funciones del sistema.

2.- Análisis de requerimientos

Análisis de requerimientos

Comienza con un refinamiento detallado del ámbito del programa establecido durante la ingeniería del sistema y refinado durante la planificación del proyecto de software. Se crean los modelos del flujo de la información y del control, del comportamiento en operación y del contenido de los datos. Se analizan las soluciones alternativas.

Tanto el desarrollador como el cliente juegan un papel activo en la especificación y el análisis de requerimientos. El cliente intenta volver a plantear detalladamente el concepto de la función y del comportamiento del software. El desarrollador actúa como interrogador, como consultor y como persona que resuelve problemas.

El análisis y la especificación de requerimientos puede parecer una tarea sencilla, pero el contenido de la comunicación es muy denso y se puede llegar a malas interpretaciones, falta de información o ambigüedades.

El análisis de requerimientos es la tarea que establece un puente entre la asignación del software a nivel de sistema y el diseño del software. El análisis de requerimientos facilita la especificación de la función y del rendimiento del software, la descripción de la interfaz con los otros elementos del sistema y el establecimiento de las restricciones de diseño que debe considerar el software. El análisis de requerimientos permite al ingeniero de software refinar la asignación del software y construir modelos de los ámbitos del proceso, de los datos, y del comportamiento que serán cubiertos por el software. El análisis de requerimientos proporciona al diseñador una representación de la información y de las funciones que se puede traducir en un diseño de datos, arquitectónico y procedimental. Por último, la especificación de los requerimientos suministra al técnico y al cliente un medio para valorar la calidad del software.

Tipos de requerimientos

- **Requerimientos funcionales:** describen una interacción entre el sistema y su ambiente, como debe comportarse el sistema ante determinado estímulo.
- **Requerimientos no funcionales:** describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema. Ejemplo el rendimiento, de calidad, etc.

Tareas de análisis

En el análisis de requisitos se pueden identificar las siguientes tareas:

1. Reconocimiento del problema: Se estudia la *especificación del sistema* y el *plan del proyecto de software*. Se debe establecer una comunicación adecuada de forma que se facilite el reconocimiento del problema.
2. Evaluación y síntesis: El analista debe evaluar el flujo y la estructura de la información, definir y elaborar todas las funciones del software, entender el comportamiento del programa en el contexto de los sucesos que afectan al sistema, establecer las características de la interfaz del sistema y descubrir las restricciones de diseño. Tras evaluar los problemas actuales y la información deseada, el analista comienza a sintetizar una o más soluciones. El proceso de evaluación y síntesis continúa hasta que el analista y el cliente estimen que se puede especificar el software de forma adecuada. A lo largo de la evaluación y síntesis de la solución, el analista se centra básicamente en el "qué", no en el "cómo".

3. Modelización: Durante la evaluación y la síntesis de la solución, el analista crea modelos del sistema para entender mejor el flujo de datos y de control, el procesamiento funcional, el comportamiento en operación y el contenido de la información. También se puede construir un prototipo.
4. Especificación: Las tareas asociadas con el análisis y la especificación intentan proporcionar una representación del software que pueda ser revisada y aprobada por el cliente. En el mundo real es difícil tener una buena especificación que el cliente no altere a lo largo del desarrollo, pero conviene aproximarse.
Una vez que se han descrito la información básica las funciones, el rendimiento, el comportamiento y la interfaz, se especifican los criterios de validación que han de servir para demostrar que se ha llegado a un buen entendimiento de la forma de implementar con éxito el software. Estos criterios sirven como base para las actividades de prueba que se llevarán a cabo más adelante dentro del proceso de la ingeniería del software. Para definir las características y los atributos del software, se escribe una especificación formal de requerimientos. Además, para los casos en los que no se desarrolle un prototipo, se puede esbozar un *manual de usuario preliminar*.
5. Revisión: Los documentos del análisis de requerimientos sirven como base para una revisión por parte del cliente y del desarrollador. La revisión de los requerimientos casi siempre produce modificaciones en la función, en el rendimiento, en la representación de la información, en las restricciones o en los criterios de validación. Además, se vuelve a evaluar el *plan del proyecto de software*.

Metodologías para el análisis:

Luego del relevamiento, seleccionamos los datos serán útiles para el sistema en construcción. En base a esto vamos a crear, descubrir, inventar qué hay que hacer para lograr el objetivo.

Para todo esto existen distintas metodologías. Las tres principales son:

- La tradicional
- La estructurada
- La orientada a objetos

Metodología tradicional:

El analista escribía lo que entendía de los requerimientos del en un enorme documento que consistía primariamente en una narrativa denominada a menudo como 'especificación funcional'.

Problemas:

1. Eran monolíticos: había que leer completamente la especificación para poder entenderla.
2. Eran redundantes: se repetía la misma información en diversas partes del documento, el problema estaba cuando había que cambiar cualquier aspecto de los requerimientos del usuario.
3. Eran ambiguos: el reporte detallado, de los requerimientos podía ser interpretado de diferentes maneras por el usuario, el analista, el diseñador y el programador.
4. Eran imposibles de mantener: Por las razones descritas anteriormente.

Metodología estructurada:

Surge de una nueva forma de diagramación. Aquí la documentación en su mayoría es gráfica, muy fácil de entender. Cuando se termina de programar la documentación ya está hecha, ya que fue necesario hacerla en primera instancia para poder empezar a trabajar.

En la metodología estructurada se baja de nivel, se va de lo general a lo particular. Es una tecnología TOP DOWN.

Se concentra en las funciones que tiene que ejecutar el sistema, en su búsqueda y desarrollo.

Herramientas:

- Diagramas de Flujos de Datos
- Diccionarios de Datos
- Diagramas Entidad-Relación
- Diagramas de Transición de Estado
- Especificaciones de procesos

Metodología Orientada a objetos

Va de lo particular a lo general. Es una tecnología BOTTOM UP.

En el análisis orientado a objetos cobra mucho más importancia el aspecto de "modelado" del sistema, examinando el dominio del problema como un conjunto de objetos que interactúan entre sí.

Herramientas:

- Casos de uso
- Modelado de clases-responsabilidades-colaboraciones CRC
- Modelo de objeto
- Diagrama de estado