

Unidad IV

Análisis del proyecto de software

4.1. Modelado: análisis, diseño, documentación.

Analisis:

Pressman establece que la tarea del analisis de requisitos es un proceso de descubrimiento,refinamiento,modelo y especificacion.Se refina en detalle el ambito del software y se crean modelos de los requisitos datos,flujo de informacion y control y del comportamiento operativo.Se realizan soluciones alternativas y se asignan a diferentes elementos del software.

El analisis de requisitos permite al desarrollador o desarrolladores especificar la funcion y el rendimiento del software,indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software.Puede dividirse en cinco areas de esfuerzo que son:

- Reconocimiento del problema
- Evaluacion y sintesis
- Modelado
- Especificacion
- Revision

Diseño:

Es cuando se traducen los requerimientos funcionales y no funcionales en una representacion de software.El diseño es el primer paso en la fase de desarrollo o cualquier producto o sistema de ingenieria.

De acuerdo con Pressman el objetivo del diseño es producir un modelo o representacion de una entidad que va a construir posteriormente.

De acuerdo con McGlaughlin hay tres caracteristicas que sirven como parametros generales para la evaluacion de un buen diseño:

El diseño debe implementar todos los requerimientos explicados obtenidos en la etapa de analisis.

El objetivo debe ser una guia que puedan leer y entender los que construyen el codigo y los prueban y mantienen el software.

El diseño debe proporcionar una idea completa de lo que es el software.



Documentacion:

Se presenta en base a los metodos orientados a objetos propuestos por Martin y Odell. Se utilizan las siguientes tecnicas para documentar los componentes mas relevantes de la herramienta de software:

Diagramas de eventos: Para ilustrar la manera en que el usuario del software interactua con los eventos virtuales.

Diagramas de contexto: Para ubicar el campo de accion que abarca el software.

Tarjetas CRC: Utilizadas para representar todas las clases dentro de un diseño.



4.2. Construcción: codificación, pruebas y evaluación, manual del usuario, manual técnico.

Construccion:

Es la creacion detallada de software operativo.

Lenguajes de construccion pueden ser varias clases:

De configuracion.

De Toolkits.

De programacion.

Los principios fundamentales de la construccion de software son:

- Minimizar la complejidad
- Anticipar los cambios
- Pensar en la verificacion posterior
- Aplicar estandares



Coodificacion:

La escritura del codigo fuente es el principal esfuerzo de costrucion de software:

- Aplicar tecnicas para crear codigo fuente comprensible
- Manejar condiciones de error
- Prevenir brechas de seguridad a nivel de codigo
- Uso eficiente de recursos escasos
- Organizar el codigo fuente
- Documentar el codigo

Pruebas:

Frecuentemente realizadas por los mismos que escriben el codigo. El proposito de estas pruebas es reducir el tiempo entre el momento en el que los fallos se insertan en el codigo y el momento en el que son detectados

Implica la ejecucion del programa permite:

Evaluar la calidad de un producto.
Mejorarlo identificando defectos y problemas.

El ambito o destino de las pruebas de software pueden variar en tres niveles:

Modulo unico:Pruebas unitarias
Grupo de modulos:Pruebas de integracion
Sistema completo:Pruebas del sistema



Técnicas de prueba:

Principio básico: Intentar ser lo más sistemático posible en identificar un conjunto representativo de comportamiento de programa.

Objetivo: romper el programa, encontrar el mayor de fallos posibles. Existen 2 enfoques diferentes:

- Caja negra: Funcional: Se basan en el comportamiento de entrada y salida.
- Caja blanca: Estructural: Información del software que a sido diseñada o codificada.

Evaluación:

Evaluación de Proyectos es un proceso que permite emitir un juicio sobre la conveniencia del proyecto. Este criterio está presente en cada etapa del “Ciclo de proyecto”:

1 ^a Etapa:	Pre-inversión
2 ^a Etapa:	Inversión

Al igual que en las etapas, en cada fase se realiza una evaluación de acuerdo a sus características.

La etapa de Pre-inversión consta de las siguientes fases

1 ^a Etapa	1 ^a Fase	Concepción de la idea
1 ^a Etapa	2 ^a Fase	Perfil
1 ^a Etapa	3 ^a Fase	Pre-factibilidad (Estudio de Alternativas)
1 ^a Etapa	4 ^a Fase	Factibilidad (Ante- proyecto)

	definitivo)
--	-------------

En la etapa de Inversión también se realiza el proceso de evaluar en cada una de sus fases.

La etapa de Inversión consta de las siguientes fases:

2 ^a Etapa	Diseño definitivo
2 ^a Etapa	Montaje y Operación

Como se puede apreciar, el trabajo de evaluación se encuentra en todo momento, cada etapa tiene su forma de evaluar. En la etapa de Pre-inversión, la evaluación es realizada en las distintas fases, comenzando por la concepción de la idea y terminando en la factibilidad del proyecto. En esta etapa la Evaluación suele ser llamada "Evaluación Ex-Ante".

En la etapa de Inversión la Evaluación se da tanto en el Diseño definitivo como en el Montaje y Operación del proyecto. En esta etapa la Evaluación suele ser llamada "Evaluación Ex-Post".

El camino al éxito de nuestros proyectos, solo se conseguirá siendo imparciales a los datos o resultados que otorgue la Evaluación correspondiente, adicionalmente se necesita cambiar algunos paradigmas tradicionales y desarrollar un sistema de control adecuado, de este modo se conseguirá el éxito.

La Evaluación de Proyectos es "un instrumento o herramienta que genera información, permitiendo emitir un juicio sobre la conveniencia y confiabilidad de la estimación preliminar del beneficio que genera el Proyecto en estudio".



Manual de usuario:

Un manual de usuario se trata de una guía que ayuda a entender el funcionamiento de algo. Es un documento de comunicación técnica que busca brindar asistencia a los sujetos que usan un sistema o servicio.

Elaboración del Manual De Usuario.

Pasos del manual del usuario:

1. Portada: De que se trata el documento y ¿quién lo elaboró?
2. Introducción: Describe el uso del documento (¿para qué sirve?) y ¿de qué habla?
3. Análisis y requerimientos del sistema (¿que se ocupa para poder instalarlo y usarlo?)
3. Explicación del funcionamiento: Debes de poner paso a paso y con pantallas bien explicadas cómo funciona el programa
4. Glosario

Debe ser escrito de tal manera, que cualquier persona pueda entenderlo con la menor dificultad posible.

Es recomendable, detallar todos aquellos pasos que se llevan a cabo para usar el programa.

Especificar los alcances y las limitaciones que tiene el programa.

Un buen punto de partida para un manual de usuario, es hacer de cuenta que las personas que lo van a leer no tienen el más mínimo conocimiento sobre computadores.

Manual Técnico.

Este documento contiene toda la información sobre los recursos utilizados por el proyecto, llevan una descripción muy bien detallada sobre las características físicas y técnicas de cada elemento. Por ejemplo: características de procesadores, velocidad, dimensiones del equipo, garantías, soporte, proveedores y equipo adicional.

Su extensión depende de la cantidad de recursos y equipo utilizado y generalmente se presenta en forma de fichas técnicas en donde se describe en cada una las características de cada recurso.

Elaboración del Manual Técnico.

Un manual técnico es aquel que va dirigido a un público con conocimientos técnicos sobre algún área, mientras que, por ejemplo, un manual de usuario va dirigido a un público más general, el cual no necesariamente debe tener conocimientos específicos en el área de interés.

En este caso el manual técnico, debe incluir:

Paradigma de programación seleccionado y sus beneficios.

Lenguaje de programación seleccionado y sus beneficios frente a otros lenguajes.

Estandarización de código utilizada.

Diseño del sistema.



4.3. Medida, métrica e indicador.

Medida:

Proporciona una indicación cuantitativa de la cantidad, dimensiones o tamaño de algunos atributos de un producto.

Se debe medir el software para:

Indicar la calidad del producto.

Evaluar la productividad del agente que desarrolla el producto.

Evaluar los beneficios en términos de productividad y calidad mediante el uso de nuevos métodos y herramientas de ingeniería de software.

Establecer una línea de base para la estimación.

Ayudar a justificar el uso de nuevas herramientas o de formación adicional.



Las métricas

Son medidas cuantitativas que permiten obtener una visión de la eficacia del proceso de software y los proyectos que se llevan a cabo utilizando ese proceso como marco de trabajo.

Permiten valorar el estado de un proyecto en curso, así como también rastrear los riesgos potenciales y descubrir las aéreas problema antes que se vuelvan “críticas”, también permite ajustar el flujo de trabajo o las tareas y evaluar la habilidad del equipo del proyecto. Las métricas del proyecto se usan con fines tácticos

Indicador:

Son datos esencialmente cuantitativos, que nos permiten darnos cuenta de cómo se encuentran las cosas en relación con algún aspecto de la realidad que nos interesa conocer.

Los Indicadores pueden ser medidas, números, hechos, opiniones o percepciones que señalen condiciones o situaciones específicas.

4.4. Tipos de métricas: métricas de proceso, métricas de proyecto, métricas orientadas a punto de función, métricas orientadas al tamaño, métricas para la calidad del software.

Metricas de proceso:

Permiten obtener un conjunto de indicadores de proceso que conduzcan a la mejora de los procesos de software a largo plazo, las cuales se usan con fines estratégicos

Metricas de proyecto:

Permiten valorar el estado de un proyecto en curso, así como también rastrear los riesgos potenciales y descubrir las aéreas problema antes que se vuelvan “críticas”, también permite ajustar el flujo de trabajo o las tareas y evaluar la habilidad del equipo del proyecto. Las métricas del proyecto se usan con fines tácticos

Metricas orientadas a punto de funcion:

Son medidas indirectas del software y del proceso. Se centran en la funcionalidad o utilidad del programa. “Emplean como un valor de normalización una medida de la funcionalidad que entrega la aplicación. La métrica orientada a la función utilizada con mayor amplitud es el punto de función (PF)

Metricas orientadas al tamaño:

Son medidas directas del software y del proceso por el cual se desarrolla. Se obtiene considerando las medidas de productividad y normalizándolas por el tamaño del código, es decir las líneas de código LDC.

Se lista cada proyecto de desarrollo de software de los últimos años y los correspondientes datos orientados al tamaño de cada uno, se basan en a utilización de registros sencillos para las medidas más relevantes al desarrollo de nuestro proyecto.

Metricas para calidad del software:

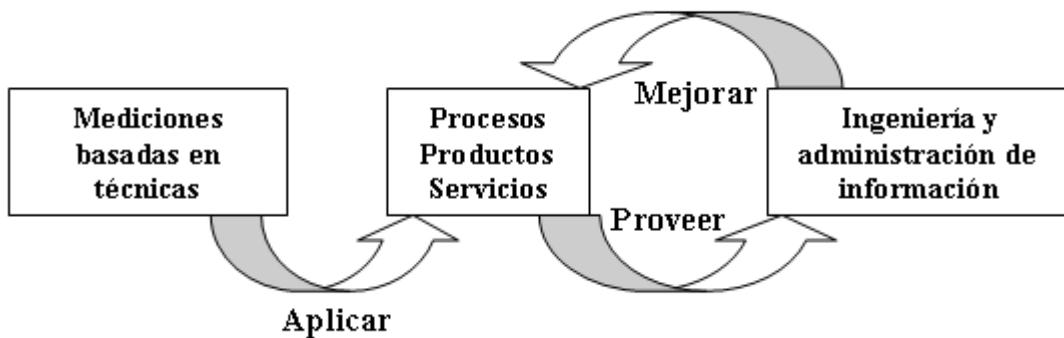
Existen medidas de la calidad del software como: la corrección, la facilidad de mantenimiento, la integridad y la facilidad de uso ofrecen indicadores utilices para el equipo del proyecto, Gilb[GIL88]sugiere definiciones y mediciones para cada una de ellas:

Corrección: es el grado en que el software desempeña la funcionara la que fue creado y se mide en defectos por KLOC.

Facilidad de Mantenimiento:es la sencillez con que un programa puede corregirse si se encuentra unerror; al adaptarse si su entorno cambia o mejorar si el cliente cambia los requisitos y se mide en forma indirecta en TMC (tiempo medio de cambio).

Integridad: es la habilidad de un sistema para resistir ataques que requiere la definición de amenaza y seguridad y se calcula:

$$\text{integridad} = 1 - (\text{amenaza} \times (1 - \text{seguridad}))$$



4.5. Implementación y mantenimiento: entrega, retroalimentación del cliente.

Implementacion:Es la actividad durante la cul los desarrolladores traducen el diseño a codigo.

Mantenimiento:Es el proceso de mejora y optimizacion del software asi como tambien correccion de los defectos.

Entrega: Proporciona el conjunto de actividades necesarias para llevar a cabo las entregas software y de documentación asociada a un proyecto, desde la formalización de la entrega hasta la revisión y validación de la misma, con la finalidad de homogeneizar todas las entregas software y facilitar su revisión y tratamiento.

Retroalimentacion del cliente:

El proceso de retroalimentación del cliente es una parte crítica del sistema de gestión de la calidad, y por lo tanto debe recibir una atención adecuada durante una auditoría de tercera parte. La retroalimentación del cliente es uno de los indicadores primarios de desempeño que puede ser utilizado para juzgar la eficacia general del SGC. Por lo tanto, es importante para el auditor verificar que:

- a) Las entradas a este proceso incluya datos relevantes, representativos y confiables.
- b) Estos datos se analizan eficazmente.
- c) La salida de este proceso proporciona información útil para la revisión por la dirección y otros procesos del SGC, para aumentar la satisfacción del cliente y llevar hacia la mejora continua

