Unidad II

Modelos de la ingeniería del software

La ingeniería de software dispone de varios modelos, paradigmas y filosofías de desarrollo, en los cuales se apoya para la construcción del software, entre ellos se puede citar:

- Modelo en cascada o Clásico (modelo tradicional)
- Modelo de prototipos
- Modelo en espiral
- Desarrollo por etapas
- Desarrollo iterativo y creciente o Iterativo e Incremental
- RAD (Rapid Application Development)
- Desarrollo concurrente
- Proceso Unificado
- RUP (Proceso Unificado de Rational)

Naturaleza de la IS

La ingeniería de software es una disciplina que está orientada a aplicar conceptos y métodos de ingeniería a la fabricación de software de calidad.

Matemáticas

Los programas tienen muchas propiedades matemáticas. Por ejemplo la corrección y la complejidad de muchos algoritmos son conceptos matemáticos que pueden ser rigurosamente probados. El uso de matemáticas en la IS es llamado *métodos formales*.

Creación

Los programas son construidos en una secuencia de pasos. El hecho de definir propiamente y llevar a cabo estos pasos, como en una línea de ensamblaje, es necesario para mejorar la productividad de los desarrolladores y la calidad final de los programas. Este punto de vista inspira los diferentes procesos y metodologías que se encuentran en la IS.

Gestión de Proyecto

El desarrollo de software de gran porte requiere una adecuada gestión del proyecto. Hay presupuestos, establecimiento de tiempos de entrega, un equipo de profesionales que liderar. Recursos (espacio de oficina, insumos, equipamiento) por adquirir. Para su administración se debe tener una clara visión y capacitación en Gestión de Proyectos.

Arte[editar · editar código]

Los programas contienen muchos elementos artísticos. Las interfaces de usuario, la codificación, etc. Incluso la decisión para un nombre de una variable o una clase. Donald Knuth es famoso por argumentar a la programación como un arte.

2.1. Modelo de capacidad de madurez.

El Modelo de Madurez de Capacidades o CMM (*Capability Maturity Model*), es un modelo de evaluación de los procesos de una organización. Fue desarrollado inicialmente para los procesos relativos al desarrollo e implementación de software por la Universidad Carnegie-Mellon para el SEI (Software Engineering Institute).

El SEI es un centro de investigación y desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la Universidad Carnegie-Mellon. "CMM" es una marca registrada del SEI.

Este modelo establece un conjunto de prácticas o procesos clave agrupados en Áreas Clave de Proceso (KPA - *Key Process Area*). Para cada área de proceso define un conjunto de buenas prácticas que habrán de ser:

- Definidas en un procedimiento documentado
- Provistas (la organización) de los medios y formación necesarios
- Ejecutadas de un modo sistemático, universal y uniforme (institucionalizadas)
- Medidas
- Verificadas

A su vez estas Áreas de Proceso se agrupan en cinco "niveles de madurez", de modo que una organización que tenga institucionalizadas todas las prácticas incluidas en un nivel y sus inferiores, se considera que ha alcanzado ese nivel de madurez.

Los niveles son:

0 - Inexistente. Las Organizaciones carencen completamente de cualquier proceso reconocible e incluso se desconoce la existencia de un problema a resolver.

1 - Inicial. Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El éxito de los proyectos se basa la mayoría de las veces en el esfuerzo personal, aunque a menudo se producen fracasos y casi siempre retrasos y sobrecostes. El resultado de los proyectos es impredecible.

2 - Repetible. En este nivel las organizaciones disponen de unas prácticas institucionalizadas de gestión de proyectos, existen unas métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente.

3 - Definido. Además de una buena gestión de proyectos, a este nivel las organizaciones disponen de correctos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detalladas y un nivel más avanzado de métricas en los procesos. Se implementan técnicas de revisión por pares (*peer reviews*).

4 - Gestionado. Se caracteriza porque las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.

5 - Optimizado. La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

> Así es como el modelo CMM establece una medida del progreso, conforme al avance en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El alcanzar estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables. Con la excepción del primer nivel, cada uno de los restantes Niveles de Madurez está compuesto por un cierto número de Áreas Claves de Proceso, conocidas a través de la documentación del CMM por su sigla inglesa: KPA.
>
> Cada KPA identifica un conjunto de actividades y prácticas interrelacionadas, las cuales cuando son realizadas en forma colectiva permiten alcanzar las metas fundamentales del proceso. Las KPAs pueden clasificarse en 3 tipos de proceso: Gestión, Organizacional e Ingeniería.
>
> Las prácticas que deben ser realizadas por cada Área Clave de Proceso están organizadas en 5 Características Comunes, las cuales constituyen propiedades que indican si la implementación y la institucionalización de un proceso clave es efectivo, repetible y duradero.
>
> Estas 5 características son: i)Compromiso de la realización, ii) La capacidad de realización, iii) Las actividades realizadas, iv) Las mediciones y el análisis, v) La verificación de la implementación.

Las organizaciones que utilizan CMM para mejorar sus procesos disponen de una guía útil para orientar sus esfuerzos. Además, el SEI proporciona formación a evaluadores certificados (*Lead Assesors*) capacitados para evaluar y certificar el nivel CMM en el que se encuentra una organización. Esta certificación es requerida por el Departamento de Defensa de los Estados Unidos, pero también es utilizada por multitud de organizaciones de todo el mundo para valorar a sus subcontratistas de software.

Se considera típico que una organización dedique unos 18 meses para progresar un nivel, aunque algunas consiguen mejorarlo. En cualquier caso requiere un amplio esfuerzo y un compromiso intenso de la dirección.

Como consecuencia, muchas organizaciones que realizan funciones de factoría de software o, en general, outsourcing de procesos de software, adoptan el modelo CMM y se certifican en alguno de sus niveles. Esto explica que uno de los países en el que más organizaciones certificadas exista sea India, donde han florecido las factorías de software que trabajan para clientes estadounidenses y europeos.

2.2. Marco de trabajo para el proceso.

Debido a que el software, como cualquier capital, es conocimientomaterializado y dado que el conocimiento en un inicio es dispersotácito, latente y en gran medida incompleto, el desarrollo delsoftware es un proceso de aprendizaje social.

- El proceso es un dialogo en el cual el conocimiento que elsoftware debe convertir se conjunta y se materializa en esteultimo.
- El proceso proporciona interacción entre los usuarios y lasherramientas en evolución, y entre los diseñadores y sus herramientas.

Es un proceso iterativo en el que las herramientas en evoluciónsirven como un medio para la comunicación, en el cual cadanueva etapa del dialogo logra obtener mas conocimiento útil delas personas implicadas.

2.3. Modelos de la ingeniería del software: modelo de cascada, modelo de prototipos, modelo de espiral, modelo de Proceso Unificado Racional (RUP).

En Ingeniería de software el desarrollo en cascada, también llamado modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.
Un ejemplo de una metodología de desarrollo en cascada es:

1.- Análisis de requisitos
2.- Diseño del Sistema
3.- Diseño del Programa
4.- Codificación
5.- Pruebas
6.- Implantación

7.- Mantenimiento

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo. La palabra cascada sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.
Si bien ha sido ampliamente criticado desde el ámbito académico y la industria, sigue siendo el paradigma más seguido al día de hoy.

El Modelo de prototipos, en Ingeniería de software, pertenece a los modelos de desarrollo evolutivo. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos.

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción de un prototipo, el cual es evaluado por el cliente para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

Boehm, ideó y promulgó un modelo desde un enfoque distinto al tradicional en Cascada: El Modelo Evolutivo Espiral. Su Modelo de Ciclo de Vida en Espiral tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por

la de riesgo más asumible y se hace un ciclo de la espiral. Si el cliente quiere seguir haciendo mejoras en el software, se vuelve a evaluar las distintas nuevas alternativas y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto software desarrollado sea aceptado y no necesite seguir mejorándose con otro nuevo ciclo.

El **Proceso Unificado de Rational** (*Rational Unified Process* en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software, también desarrollado por Rational, que incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el **Rational Method Composer** (RMC), que permite la personalización de acuerdo con las necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el ***Rational Unified Process***, que se vendiera como producto independiente...

2.4. Tendencias modernas de modelos de la ingeniería del software.

XP:Programacion extrema:

De todas las metodologias habiles esta recibe mas atencion.Valores :
Retroalimentación.
Comunicación.
Simplicidad .
Coraje.

Contruye un proceso de diseño evolutivo refactora un sistema simple en cada iteraccion que se centra en la iteraccion actual no se hace nada anticipadamente.Combina la disiplina con la adaptabilidad.Kent Beck escribio Extreme Programming Explained qu es clave de la XP.

La familia de cristal de Cockburn:

Tipos diferentes de proyectos requieren tipos diferentes de metodologia.

Costituye:el numero de personas de un proyecto y las consecuencias de los errores.Alistar requiere que las personas sigan un proceso diciplinado,explora a

metodologia menos disiplinada que aun pueda tener éxito intercambiando productividad.El cristal es menos productivo que la XP maas personas seran capaces de seguirlo.

Codigo abierto:Es un estilo de software,en particular su proceso se engrana a equipos fisicamente distribuidos,la mayoria de los procesos adaptables exigen procesos locales.La mayoria tiene uno mas mantenedores.Un mantenedor es la unica persona a la que se permite hacer cambioen el almacen de codigo fuente,otras personas tambien pueden hacer cambios pero necesitan madarlas a mantenedor para que las revise aplique.

El desarrollo de software adaptable de Highsmith:

Trabajando con metodologías predictivas. él las desarrolló, instaló, enseñó, y concluyó que son defectuosas: particularmente para los negocios modernos. En ASD hay tres fases, no lineales: especulación, colaboración, y aprendizaje. En un ambiente adaptable, aprender desafía a todos - desarrolladores y sus clientes - a examinar sus asunciones y usar los resultados de cada ciclo de desarrollo para adaptar el siguiente.

Scrum:

Scrum divide un proyecto en iteraciones (que ellos llaman carreras cortas) de 30 días. Antes de que comience una carrera se define la funcionalidad requerida para esa carrera y entonces se deja al equipo para que la entregue. el punto es estabilizar los requisitos durante la carrera.

Todos los días el equipo sostiene una junta corta (quince minutos), llamada scrum, dónde el equipo discurre lo que hará al día siguiente. la literatura de scrum se enfoca principalmente en la planeación iterativa y el seguimiento del proceso.

Desarrollo manejado por rasgo:

Fue desarrollado por Jeff de Luca y Peter Coad. Las iteraciones duran dos semanas. Tiene cinco procesos. los primeros tres se hacen al principio del proyecto.

•desarrollar un modelo global
•construir una lista de los rasgos
•planear por rasgo
•diseñar por rasgo
•construir por rasgo
Los últimos dos se hacen en cada iteración. Cada proceso se divide en tareas y se da un criterio de comprobación.
Los desarrolladores entran en dos tipos: dueños de clases y programadores jefe.

DSDM (MÉTODO DE DESARROLLO DE SISTEMA DINÁMICO)

Empieza con un estudio de viabilidad y negocio. Viabilidad considera si DSDM es apropiado para el proyecto. Negocio es una serie corta de talleres para entender el área de negocio dónde tiene lugar el desarrollo. También propone arquitecturas de esbozos del sistema y un plan del proyecto.

El resto del proceso forma tres ciclos entretejidos: el ciclo del modelo funcional produce documentación de análisis y prototipos, el ciclo de diseño del modelo diseña el sistema para uso operacional, y el ciclo de implantación se ocupa del despliegue al uso operaciona