

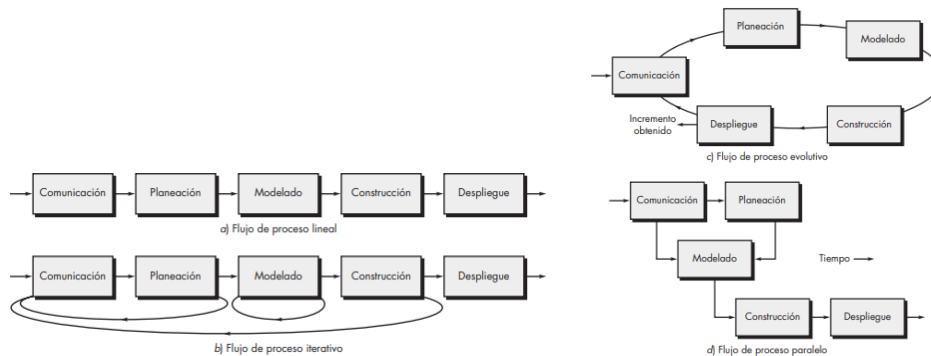
Resumen: Capitulo 2 de Pressman

Nombre: Marco Rodriguez Vargas

Carnet: 2022149445

Un **proceso** es la colección de actividades de trabajo, acciones y tareas que se realizan cuando va a crearse algún producto terminado. Cada una de las actividades, acciones y tareas se encuentra dentro de una estructura o modelo que define su relación tanto con el proceso como entre sí. Una estructura general para la ingeniería de software se define en cinco actividades estructurales (comunicación, planeación, modelado, construcción y despliegue).

Además, a lo largo de todo el proceso se aplica un conjunto de **actividades sombrilla**: seguimiento y control del proyecto, administración de riesgos, aseguramiento de la calidad, administración de la configuración, revisiones técnicas, entre otras.



Definición de actividad estructural: Para un proyecto de software pequeño solicitado por una persona (en una ubicación remota) con requerimientos sencillos y directos, la actividad de comunicación tal vez no incluya algo más que una llamada telefónica con el participante apropiado.

Tareas del trabajo: (1. Hacer contacto con el participante por vía telefónica. 2. Analizar los requerimientos y tomar notas. 3. Organizar las notas por escrito en una formulación breve de los requerimientos. 4. Enviar correo electrónico al participante para que revise y apruebe).

Si el proyecto fuera considerablemente más complejo, con muchos participantes y cada uno con un distinto conjunto de requerimientos (a veces en conflicto), la actividad de comunicación puede tener seis acciones distintas: concepción, indagación, elaboración, negociación, especificación y validación.

Identificación de un conjunto de tareas: Debe escogerse el conjunto de tareas que se adapte mejor a las necesidades del proyecto y a las características del equipo.

Patrones del proceso: describe un problema durante el trabajo de ing. de software, identifica el ambiente en el que surge el problema y sugiere soluciones para el mismo. Da en formato [Amb98]: un método consistente para describir soluciones del problema en el contexto del proceso del software. Al combinar patrones, un equipo de software resuelve problemas y construye el proceso que mejor satisfaga las necesidades de un proyecto.

Ambler sugiere tres tipos:

- Patrón de etapa: define un problema para el proceso. Incluye múltiples acciones y tareas del trabajo, un patrón de la etapa incorpora múltiples patrones de la tarea. Un ejemplo de patrón de etapa sería EstablecerComunicación. (Incorporaría el patrón de tarea RecabarRequerimientos y otros más).
- Patrón de tarea: define un problema con una acción o tarea de trabajo de la ing. de software y que es relevante para el éxito de la práctica de ingeniería de software (por ejemplo, RecabarRequerimientos es un patrón de tarea).
- Patrón de fase: define la secuencia de las actividades estructurales que ocurren, aun cuando el flujo general de las actividades sea de naturaleza iterativa. (Un ejemplo de patrón de fase es ModeloEspiral o Prototipos).

Para esto se ocupa un **contexto inicial** (Condiciones en las que se aplica el patrón).

Problema: El problema específico que debe resolver el patrón. **Solución:** Describe cómo implementar con éxito el patrón. **Contexto resultante:** Describe las condiciones que resultarán una vez que se haya implementado con éxito el patrón. **Patrones relacionados:** Proporciona una lista de todos los patrones de proceso directamente relacionados con éste. **Usos y ejemplos conocidos:** Indica las instancias específicas en las que es aplicable el patrón.

Evaluación y mejora del proceso: Un proceso del software no es garantía de que el software se entregue a tiempo, que satisfaga las necesidades de los consumidores o que tenga las características técnicas que conducirán a características de calidad de largo plazo. Enfoques para la evaluación y mejora de un proceso del software:

1. **Método de evaluación del estándar CMMI para el proceso de mejora (SCAMPI):** modelo de cinco fases para evaluar el proceso: inicio, diagnóstico, establecimiento, actuación y aprendizaje. El método SCAMPI emplea el SEI CMMI como la base de la evaluación [SEI00].
2. **Evaluación basada en CMM para la mejora del proceso interno (CBA IPI):** proporciona una técnica de diagnóstico para evaluar la madurez relativa de una organización de software; usa el SEI CMM como la base de la evaluación [Dun01].
3. **SPICE (ISO/IEC 15504):** estándar que define requerimientos para la evaluación del proceso. El objetivo del estándar es ayudar a las organizaciones a desarrollar una evaluación objetiva de cualquier proceso del software definido [ISO08].
4. **ISO9001:2000 para software:** estándar genérico que se aplica a cualquier organización que desee mejorar la calidad general de los productos, sistemas o servicios que proporciona. Por tanto, el estándar es directamente aplicable a las organizaciones y compañías de software [Ant06].

Modelo de software prescriptivo: pone orden en el caos del desarrollo.

Modelo de la cascada: Secuencial, se cuestiona su eficacia porque es raro que un proyecto siga un flujo lineal. Es difícil para el cliente enunciar todos los requerimientos desde el inicio. No se dispondrá de una versión del proyecto hasta que haya avanzado mucho.

Modelo incremental el modelo incremental aplica secuencias lineales en forma escalonada a medida que avanza el calendario de actividades. Cada secuencia lineal produce “incrementos” de software susceptibles de entregarse [McD93] de manera parecida a los incrementos producidos en un flujo de proceso evolutivo. Conforme avance el proyecto el cliente tendrá funcionalidades mas avanzadas.

Modelos de proceso evolutivo: modelo de proceso diseñado explícitamente para adaptarse a un producto que evoluciona con el tiempo. Los modelos evolutivos son iterativos. Se caracterizan por la manera en la que permiten desarrollar versiones cada vez más completas del software.

El modelo espiral: es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Tiene el potencial para hacer un desarrollo rápido de versiones cada vez más completas.

Modelos concurrentes: permite que un equipo de software represente elementos iterativos y concurrentes de cualquiera de los modelos de proceso descritos en este capítulo.

Modelos de proceso especializado

Desarrollo basado en componentes: construye aplicaciones a partir de fragmentos de software prefabricados.

El método de métodos formales: permiten especificar, desarrollar y verificar un sistema basado en computadora por medio del empleo de una notación matemática rigurosa.

Desarrollo de software orientado a aspectos: los constructores de software complejo implementan de manera invariable un conjunto de características, funciones y contenido de información localizados. Estas características localizadas del software se modelan como componentes (clases orientadas a objetos) y luego se construyen dentro del contexto de una arquitectura de sistemas.

El proceso unificado: analizan la necesidad de un proceso del software “impulsado por el caso de uso, centrado en la arquitectura, iterativo e incremental”. Es un intento por obtener los mejores rasgos y características de los modelos tradicionales del proceso del software, pero en forma que implemente muchos de los mejores principios del desarrollo ágil de software.

Modelos del proceso personal y del equipo: El mejor proceso es el que está cerca de las personas que harán el trabajo. Si un modelo del proceso se ha desarrollado en un nivel corporativo u organizacional, será eficaz si acepta una adaptación para que cubra las necesidades del equipo de proyecto que hace el trabajo. En la situación ideal se crearía un proceso que se ajustara del mejor modo a los requerimientos, y al mismo tiempo cubriera las más amplias necesidades del equipo y de la organización.

Proceso personal de software (PPS): pone el énfasis en la medición personal tanto del producto del trabajo que se genera como de su calidad

- **Planeación:** Esta actividad aísla los requerimientos y desarrolla las estimaciones tanto del tamaño como de los recursos.

- Diseño de alto nivel. Se desarrollan las especificaciones externas para cada componente que se va a construir y se crea el diseño de componentes.
- Revisión del diseño de alto nivel. Se aplican métodos de verificación formal para descubrir errores en el diseño.
- Desarrollo. Se mejora y revisa el diseño del componente. El código se genera, revisa, compila y prueba.
- Post mórtem. Se determina la eficacia del proceso por medio de medidas y mediciones obtenidas (ésta es una cantidad sustancial de datos que deben analizarse con métodos estadísticos).

Proceso del equipo de software (PES): El objetivo de éste es construir un equipo “autodirigido” para el proyecto, que se organice para producir software de alta calidad. Un equipo autodirigido tiene la comprensión consistente de sus objetivos; define la responsabilidad de cada miembro del equipo; da seguimiento cuantitativo a los datos del proyecto (sobre la productividad y calidad); identifica un proceso de equipo para el proyecto y una estrategia para implementarlo; define estándares locales aplicables al trabajo del equipo; evalúa en forma continua el riesgo y reacciona en consecuencia; y da seguimiento, administra y reporta el estado del proyecto. El PES define las siguientes actividades estructurales: inicio del proyecto, diseño de alto nivel, implementación, integración y pruebas, y post mórtem.

Tecnología del proceso: permiten que una organización de software construya un modelo automatizado de la estructura del proceso, conjuntos de tareas y actividades sombrilla. Se analiza para determinar el flujo de trabajo normal y se examinan estructuras alternativas del proceso que podrían llevar a disminuir el tiempo o costo del desarrollo.

Si el proceso es deficiente, no cabe duda de que el producto final sufrirá. Pero también es peligrosa la dependencia excesiva del proceso.

1. ¿Cuál es el problema que plantea el capítulo?

Definición de proceso. Los diferentes tipos de modelo de proceso y cómo manejarlos

2. ¿Por qué el problema es interesante o importante?

Es interesante porque se mencionan diversas formas de poder solucionarlo y mejorarlo, además de ser adaptables a las diferentes situaciones y escenarios.

3. ¿Cuál es la solución propuesta por el autor?

El autor brinda diferentes soluciones dependiendo del contexto en el que se tenga que trabajar y bajo las condiciones existentes.

4. ¿Qué tan exitosa es esta solución?

El éxito dependerá de que tan bien se pueda aplicar y adaptar el modelo de proceso seleccionado para la solución además del equipo de trabajo y el manejo de este. El proceso debe ser eficiente o sino el producto final sufrirá las consecuencias.