|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **2** |
|  | TEMA |

El proceso del *software*

[2.1] Un modelo general de proceso

[2.2] Modelos de proceso prescriptivos

[2.3] Modelos de proceso especializado

[2.4] El proceso unificado

[2.5] Modelos del proceso personal y del equipo

[2.6] ¿Qué es la agilidad?

[2.7] ¿Qué es un proceso ágil?

[2.8] Scrum

[2.9] Conjunto de herramientas para el proceso ágil

[2.10] Referencias

Esquema



Ideas clave

2.1. Un modelo general de proceso

Como ya comentamos en el tema anterior, el proceso del *software* se compone de cinco actividades estructurales (comunicación, planeación, modelado, construcción y despliegue), cada una de las cuales está compuesta por un grupo de acciones y, estas a su vez, por un grupo de tareas.

Las actividades, acciones y tareas están organizadas por un flujo de proceso:



Los **patrones de proceso** proporcionan soluciones basadas en la experiencia a problemas del proceso *software*. Estos patrones facilitan al equipo de *software* la resolución del problema cuando se presenta.

El proceso *software* debe **evaluarse** para garantizar el cumplimiento de plazos, calidad, requisitos, etc. Existen diversas metodologías de evaluación y mejora del proceso *software*: SCAMPI (CMMI), CBA IPI (CMM), SPICE (ISO/IEC 15504) o ISO 9001:2000.

2.2. Modelos de proceso prescriptivos

Los modelos tradicionales de procesos son:

* **Modelo en cascada** (o ciclo de vida clásico): modelo secuencial en el que lleva a cabo una actividad de *software*, que se realiza por completo, y cuando finaliza comienza la siguiente actividad.
* **Modelo incremental**: aplica repetidas secuencias lineales de modo escalonado. Al final de cada secuencia se obtiene un incremento parcial del producto a entregar.
* **Modelo evolutivo**: modelo iterativo en el que se desarrolla en cada ciclo versiones del *software* cada vez más completas. Dos modelos evolutivos son el modelo de prototipos y el modelo en espiral.
* **Modelo concurrente**: modelo que permite representar elementos concurrentes e iterativos de los modelos anteriores. Así, las diferentes actividades, acciones o tareas existen de modo concurrente, pudiendo estar cada una de ellas en un estado diferente, y estableciéndose una serie de eventos que originan los cambios de estado.

2.3. Modelos de proceso especializado

Los modelos de proceso especializado son:

* **Modelo basado en componentes**: en el que se estructura el *software* en componentes (fragmentos de *software* prefabricados) y, posteriormente, se diseñan, integran y prueban.
* **Modelo de métodos formales**: que hacen uso de una notación matemática estricta para especificar, desarrollar y verificar el sistema.
* **Modelo orientado a aspectos**: donde se definen, especifican, diseñan y construyen aspectos, que caracterizan propiedades globales funcionales y no funcionales de componentes.

2.4. El proceso unificado



El proceso unificado consta de cinco fases:



2.5. Modelos del proceso personal y del equipo

El **proceso personal del *software*** se basa en la medición personal del producto generado y su calidad. Se organiza en cinco actividades estructurales: planeación, diseño de alto nivel, revisión del diseño, desarrollo y mediciones para la mejora del proceso.

El **proceso del equipo de *software*** tiene como objetivo construir un equipo capaz de autodirigirse para el desarrollo de un proyecto. Las actividades estructurales están orientadas a que el equipo realice la planificación, diseño y construcción del *software*, a la vez que realiza la medición cuantitativa del proceso y el producto.

2.6. ¿Qué es la agilidad?

La **agilidad** es un cambio de filosofía en el proceso *software* que tiene como origen el manifiesto por el desarrollo ágil firmado en el año 2001.

La agilidad establece **estructuras de equipo y actitudes** que faciliten la comunicación, se prioriza la entrega rápida de *software* funcional, incorpora al cliente como parte del equipo de desarrollo y defiende la planificación flexible de los proyectos.

La incorporación de un cambio en el proceso *software* supone un coste en tiempo y dinero que es mayor cuanto más avanzado esté el proyecto, ya que implicará rediseño, nueva codificación, pruebas de regresión, etc. Los procesos ágiles bien diseñados deberían reducir este coste del cambio.

2.7. ¿Qué es un proceso ágil?

Un **proceso de *software* ágil** es un proceso de adaptación incremental rápida a los cambios en las necesidades y condiciones técnicas de un proyecto, apoyándose en la retroalimentación constante del cliente.

El manifiesto por el desarrollo ágil establece **doce principios de agilidad** que deben aplicarse en base a las necesidades que se tenga.

El proceso ágil se debe adaptar a las necesidades de las personas y del equipo, que deben ser competentes, tener un enfoque común, colaborar entre sí y con todos los participantes, organizarse ellos mismos, tener habilidad para la toma de decisiones y la resolución de problemas, además de confianza y respeto en el equipo.

2.8. Scrum

Scrum Es un método de desarrollo ágil ideado por Jeff Sutherland en los años 90 y desarrollado posteriormente por Schawaber y Beedle.



Todos los días el equipo tiene una breve reunión de 15 minutos para comentar lo que se hizo el día anterior, los obstáculos encontrados y lo que se va a hacer ese día. Al final del *sprint*, se muestra al cliente la nueva funcionalidad para su evaluación.

2.9. Conjunto de herramientas para el proceso ágil

Las herramientas para el proceso ágil suelen incorporar el apoyo automatizado para la planificación del proyecto, el desarrollo de casos y la obtención de requisitos, el diseño rápido, la generación de código y la realización de pruebas.

Algunas herramientas de este tipo son OnTime (Axosoft), Ideogramic UML (Ideogramic) o Together Tool Set (Borland).

2.10. Referencias

Fowler, M. & Highsmith, J. (2001). *The Agile Manifesto*. Recuperado de: <http://agilemanifesto.org/>

Jacobson, I., Booch, G. & Rumbaugh, J. (1999). The Unified *Software* Development Process. Massachusetts: Addison-Wesley.

Pressman, R. S. (2010). *Ingeniería del software. Un enfoque práctico* (7ª ed.). México: McGraw Hill.

Schwaber, K. & Beedle, M. (2001). *Agile Software Development with SCRUM*. Prentice Hall.

Lo + recomendado

No dejes de leer...

**Manifiesto Ágil**

Este artículo contiene la publicación original realizada del manifiesto ágil en la revista *Software Development* en agosto de 2001 por Martin Fowler y Jim Highsmith.

Accede al artículo desde el aula virtual o a través de la siguiente dirección web:

<http://agilemanifesto.org/iso/es/manifesto.html>

No dejes de ver…

**Scrum**



El modelo de desarrollo ágil Scrum define un conjunto de prácticas para definir el proceso de desarrollo *software*. Los participantes en un proyecto Scrum tienen distintos roles (ScrumMaster, ProductOwner, Stakeholders…). El desarrollo del proyecto se divide en pequeños ciclos de desarrollo completo (*sprint*) con entrega al cliente para la retroalimentación. En este vídeo podrás descubrir todo esto contado de una manera clara.

Accede al vídeo desde el aula virtual o a través de la siguiente dirección web:

<https://www.youtube.com/watch?v=PlLHc60egiQ>

+ Información

Enlaces relacionados

Estándares de UML

La página web de OMG recoge todas las especificaciones de UML. Se trata de un buen sitio de consulta para resolver dudas.



Accede a la página desde el aula virtual o a través de la siguiente dirección web:

<http://www.omg.org/spec/UML/2.4.1/>

Recursos externos

Herramientas de Modelado UML

Para realizar diagramas UML se pueden utilizar herramientas muy completas y gratuitas como StarUML, ArgoUML, Frame UML o Tiny UML.

Accede a las herramientas desde el aula virtual o a través de la siguiente dirección web:

<http://staruml.io/>

<http://argouml.tigris.org/>

<http://sourceforge.net/projects/frameuml/>

<http://sourceforge.net/projects/tinyuml/>

Actividades

Trabajo: Cambio de modelo de proceso

Supón que eres directivo de una empresa de desarrollo de *software* en el que se trabaja con un modelo de proceso clásico en cascada y se decide realizar un plan de cambio para trabajar con un modelo de proceso ágil. Elabora el plan de cambio en el que describas las tareas a realizar.

El documento deberá tener una **extensión máxima de 3 páginas**. Se valorará la claridad de explicación.

Test

**1.** En el flujo de proceso iterativo…

A. Se ejecutan las actividades en secuencia.

B. Se realizan las actividades en forma circular.

C. Se repite una o más de las actividades antes de pasar a la siguiente.

D. Se ejecuta una o más actividades en paralelo con otras.

**2.** ¿Qué modelo de proceso se le conoce también como ciclo de vida clásico?

A. Modelo en cascada.

B. Modelo incremental.

C. Modelo evolutivo.

D. Modelo concurrente.

**3.** Indica cuales son modelos comunes del proceso evolutivo:

A. Patrones y pruebas instantáneas.

B. Prototipos y modelo en espiral.

C. Análisis de riesgo y retroalimentación.

D. Modelo unificado y modelo ágil.

**4.** El desarrollo basado en componentes:

A. Incorpora muchas características del modelo en cascada.

B. No requiere de un modelo de proceso específico.

C. Fomenta la reutilización del *software*.

D. Evita la realización de pruebas de integración.

**5.** El proceso unificado propone las fases:

A. Análisis, diseño, implementación y mantenimiento.

B. Concepción, elaboración, construcción, transición y producción.

C. Comunicación, planificación, modelado, construcción y despliegue.

D. Diagramado de clases, de secuencia, de implementación y de estados.

**6.** En los modelos ágiles comparados con los modelos de proceso tradicionales:

A. El coste económico del cambio es mayor.

B. El tiempo de proyecto cuando se presenta un cambio de alcance es mayor.

C. La planificación del proyecto es más flexible.

D. Las entregas al cliente se hacen cada más tiempo.

**7.** Según los principios de agilidad:

A. Es más eficiente y eficaz las reuniones de equipo por videoconferencia.

B. Los desarrolladores y las personas de negocio deben tener espacios independientes.

C. Los requisitos deben congelarse al principio del proyecto y posteriormente no deben cambiarse.

D. Hay que realizar entregas frecuentes de *software* al cliente.

**8.** En Scrum al patrón del proceso se le conoce como:

A. Actividad.

B. Melé.

C. *Sprint*.

D. Retraso.

**9.** ¿Qué modelo de proceso combina varios flujos de proceso?

A. El modelo en cascada combina el flujo lineal y evolutivo.

B. El modelo incremental combina el flujo lineal y paralelo.

C. El modelo concurrente combina el flujo evolutivo y paralelo.

D. Cada modelo de proceso sigue un tipo de flujo y no pueden combinarse.