

Ingeniería de Software

Clase práctica: 18/ 8/ 2021

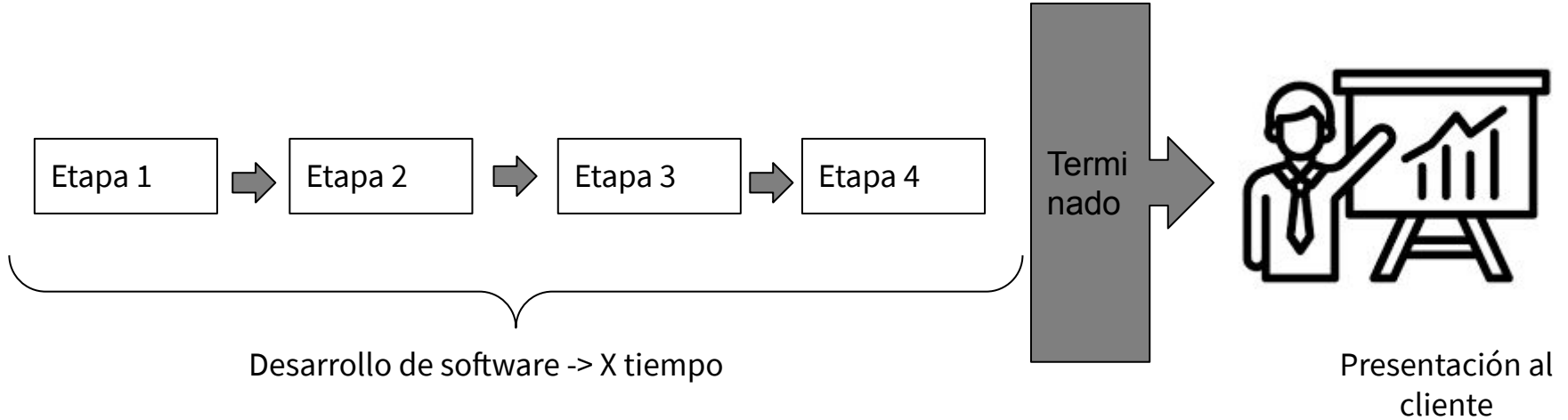


Desarrollo rápido de software

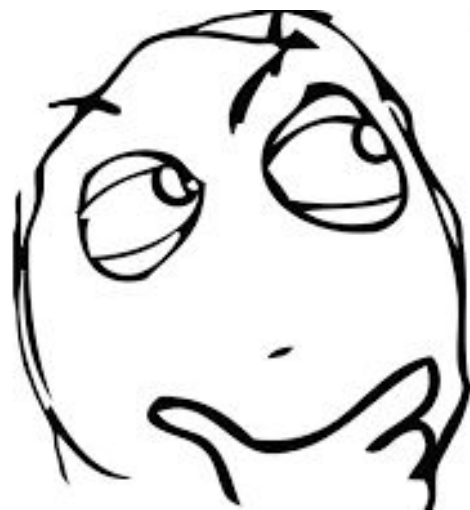
- El rápido desarrollo y la entrega son ahora los requisitos más importantes para los sistemas de software
 - a. Los requisitos cambian rápidamente
 - b. El Software tiene que evolucionar rápido.
- Desarrollo rápido de software
 - a. Especificación, diseño e implementación están intercalados
 - b. El sistema está desarrollado como una serie de versiones con las partes interesadas involucradas en la evaluación de versiones
 - c. Las interfaces del usuario se desarrollan a menudo utilizando un IDE y herramientas gráficas.



Entre 1980-1990....



el cliente....



el cliente pide un CAMBIO !!!



CAMBIO = + DINERO = + TIMEPO

Métodos Ágiles

El objetivo de los métodos ágiles es reducir los gastos generales del proceso de software (por ejemplo, limitando la documentación) y ser capaces de responder rápidamente a las necesidades cambiantes sin rehacer trabajo de manera excesiva.



Manifiesto Ágil

Estamos descubriendo mejores formas para desarrollar software, al hacerlo y al ayudar a otros a hacerlo. Gracias a este trabajo llegamos a valorar:

- ***A los individuos y las interacciones sobre los procesos y las herramientas***
- ***Al software operativo sobre la documentación exhaustiva***
- ***La colaboración con el cliente sobre la negociación del contrato***
- ***La respuesta al cambio sobre el seguimiento de un plan***

Principio	Descripción
Participación del cliente	Los clientes deben intervenir estrechamente durante el proceso de desarrollo. Su función consiste en ofrecer y priorizar nuevos requerimientos del sistema y evaluar las iteraciones del mismo.
Entrega incremental	El software se desarrolla en incrementos y el cliente especifica los requerimientos que se van a incluir en cada incremento.
Personas, no procesos	Tienen que reconocerse y aprovecharse las habilidades del equipo de desarrollo. Debe permitirse a los miembros del equipo desarrollar sus propias formas de trabajar sin procesos establecidos.
Adoptar el cambio	Esperar a que cambien los requerimientos del sistema y, de este modo, diseñar el sistema para adaptar dichos cambios.
Mantener simplicidad	Enfocarse en la simplicidad tanto en el software a desarrollar como en el proceso de desarrollo. Siempre que sea posible, trabajar de manera activa para eliminar la complejidad del sistema.

Problemas con métodos ágiles

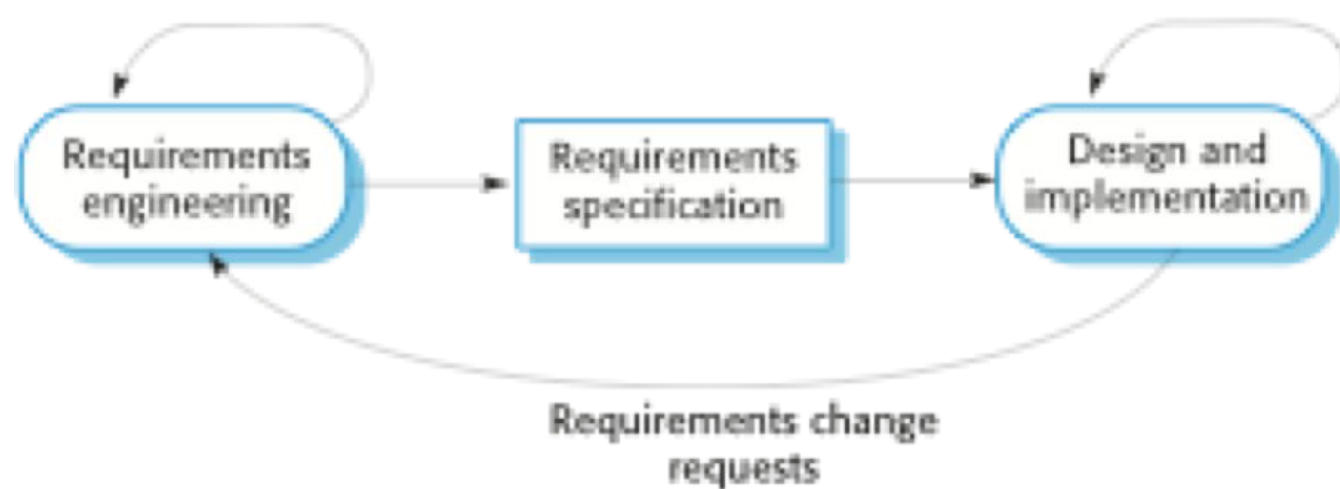
- Puede ser difícil mantener el interés de los clientes que están involucrados en el proceso.
- Los miembros del equipo pueden ser inadecuados para la intensa participación que caracteriza a los métodos ágiles.
- Priorizar cambios puede ser difícil donde hay múltiples partes interesadas.
- Mantener simplicidad requiere un trabajo extra
- Los contratos pueden ser un problema

Plan vs Ágil

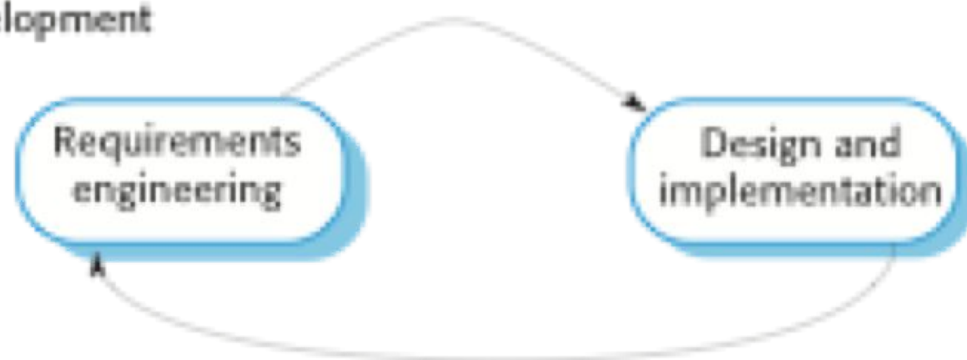
- Desarrollo guiado por plan:
 - Identifica etapas separadas en el proceso de software con salidas asociadas a cada etapa.
 - Las salidas de una etapa se usan como base para planear la siguiente actividad del proceso.
 - Las iteraciones se producen dentro de las actividades.
- Desarrollo Ágil
 - Especificación, diseño, implementación y pruebas están intercalados.

La mayoría de los proyectos de software incluyen prácticas de los enfoques ágil y basado en un plan

Plan-based development



Agile development



Ágil

- Extreme Programming (XP)
- Scrum

Scrum

- Su atención se centra en la gestión del desarrollo iterativo en lugar de prácticas ágiles específicas.

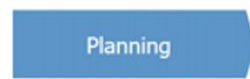
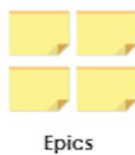
Hay tres fases en Scrum.

- La fase inicial es una fase de anteproyecto, donde se establecen los objetivos generales del proyecto y se diseña la arquitectura de software.
- Esto es seguido por una serie de ciclos, donde cada ciclo desarrolla un incremento del sistema.
- La fase de cierre del proyecto concluye el proyecto, completa documentación requerida, tales como marcos de ayuda del sistema y manuales de usuario y evalúa las lecciones aprendidas del proyecto.



Scrum

- Sprint dura entre 1-4 semanas
- Ceremonias
 - Planning
 - Daily
 - Sprint review
 - Retrospective
- Roles
 - Scrum master
 - Scrum Team
 - Product Owner
 - Stakeholders



SPRINT

Review

Retrospect

Implementation

User story

Forma en que representamos los requerimientos del sistema desde una visión de usuario. En las User story debemos tener en cuenta lo siguiente:

Como <quién> Quiero <qué> Para <objetivo>.

*Ejemplo: Como **Vendedor**, quiero registrar los productos y cantidades que me solicita un cliente para crear un pedido de venta.*

User story

¿Como valido la User story?

Para eso tenemos los **criterios de aceptación**:

Aquí se especifica qué salidas obtendremos cuando finalice el proceso de ejecución de la funcionalidad, y nos sirve para verificar que está terminada la funcionalidad. Está relacionada con las pruebas que se realizarán para verificar el cumplimiento de la expectativa de diseño, usabilidad, rendimiento, y la satisfacción del usuario.

Criterios de aceptación

Generalmente ayudan al equipo de desarrollo a responder a las preguntas:

- ¿He construido el producto correcto?
- ¿He construido el producto correctamente?

Los criterios de aceptación deben ***describir siempre un contexto, un evento y la respuesta o consecuencia esperada del sistema***. La forma más utilizada para describir los criterios de aceptación es conocida como **Given-When-Then**. Aquí un ejemplo:

Dado un usuario que aún no se ha identificado en el sistema

Cuando intenta acceder a alguna funcionalidad de la parte privada

Entonces se le redirige automáticamente a la página de login para que pueda identificarse

Aunque describen comportamiento de la aplicación, se utiliza siempre un lenguaje de negocio, no técnico.

Ejemplo:

User story

COMO usuario **QUIERO** acceder a la agenda de enfermería de forma segura **PARA** usar sus funcionalidades

Acceptance criteria

- Dado un usuario que quiere loguearse en nuestro sistema, cuando ingresa su nombre de usuario y contraseña y estos son correctos, entonces el usuario puede ingresar exitosamente a la Home
- Dado un usuario que quiere loguearse en nuestro sistema, cuando ingresa el nombre de usuario DEBE tener forma de email, en caso contrario se mostrará el mensaje de error pertinente.
- Dado un usuario que quiere loguearse en nuestro sistema, no ingresa nada en los campos usuario y contraseña, entonces el sistema muestra un mensaje de error indicando que debe ingresar esos campos obligatorios.

User Stories

Theme

Epic

Epic

User
Stories

User Stories

User Stories

Task

Task

Task

Task

Task

Task

Task

Ejercicio 1

Nos dividimos en grupos y creamos user stories en base a los enunciados (link en aula virtual)