

Introducción a las Metodologías de Desarrollo de Software

Proceso de Desarrollo de Software

Cursada 2020

Clases:

Miercoles: 16 a 18hs

Viernes: 16 a 20hs

Formato de las clases: Teorico-Práctica

Cursada promocionable:

7 o mas en el parcial (solo el parcial)

7 o mas en el TPE

Examenes

Parcial: 28 de octubre

Recuperatorio: 11 de noviembre

Prefinal: 25 de noviembre

Organización del Software

Ingeniería de Software

- **Ingeniería de Software (IS):** La aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software
- El objetivo de la IS es convertir el desarrollo de software en un proceso formal:
 - con resultados predecibles,
 - que permitan obtener un producto final de alta calidad
 - que satisfaga las necesidades y expectativas del cliente

Ingeniería de Software

El objetivo de la Ingeniería de Software es lograr productos de software de calidad (tanto en su forma final como durante su elaboración), mediante un proceso apoyado por métodos y herramientas



Desarrollo de Software

- El desarrollo de software se trata de desarrollar y entregar gran software. Un software es un gran software cuando complace a quienes lo necesitan, a quienes lo pidieron, a quienes lo usarán



Disciplinas en la Ingeniería de Software



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Construcción del producto

Disciplinas de Gestión



- Planificación de Proyecto
- Monitoreo y Control de Proyectos

Guía para la construcción del software



Disciplinas de Soporte

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

Brindan herramientas para la calidad

Disciplinas Técnicas

Construcción del Software



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Requerimientos: característica que el producto debe satisfacer se expresa en lenguaje natural que el cliente entiende



Disciplinas Técnicas

Construcción del Software



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Análisis: traduce los requerimientos funcionales del cliente en lenguaje específico de los desarrolladores



Disciplinas Técnicas

Construcción del Software



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Diseño: diseñar es crear, tomar decisiones respecto de cuál es la mejor forma de satisfacer los requerimientos definidos para el producto.
No sólo se diseña la solución de los requerimientos funcionales sino se tienen en cuenta los requerimientos no funcionales



Disciplinas Técnicas

Construcción del Software



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Implementación: escribir código, que permitirá controlar lo que una computadora hará



Disciplinas Técnicas

Construcción del Software



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Prueba/testing: Tiene como objetivo encontrar defectos.

Se deberá validar que el producto que se está probando es el que el usuario quería y verificar que el producto funciona correctamente.



Captura de Requerimientos

La parte más difícil de construir un sistema de software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual, es tan difícil como establecer los requerimientos técnicos detallados. Ninguna otra parte del trabajo afecta tanto al sistema resultante si se hace incorrectamente. Ninguna otra parte es tan difícil de rectificar más adelante.

Fred Brooks



Captura de Requerimientos

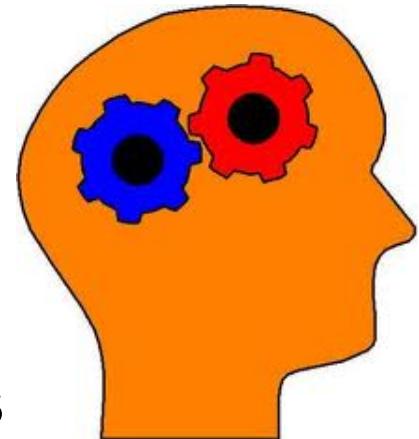
- La especificación de los requerimientos debe ser clara, sin ambigüedades, en forma consistente y compacta
- Un requerimiento es una condición o capacidad que debe cumplir el sistema que se desarrollará y que proviene directamente de los usuarios finales
- Un requerimiento es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo

Captura de Requerimientos

- De la definición, selección e implementación de los requerimientos depende el éxito del proyecto
- Los requerimientos son importantes porque permiten:
 - Llegar a un acuerdo entre el cliente y los usuarios de lo que el sistema debería hacer [ero no cómo lo hará
 - Ayudar al equipo de desarrollo a comprender lo que el sistema debería hacer
 - Delimitar el producto de software a construir
 - Servir de base para la planificación de las iteraciones de un proyecto
 - Definir la interfaz de usuario del sistema

Requerimientos Funcionales

- Relacionados con la descripción del comportamiento funcional del producto deseado
- La funcionalidad es especificada en términos de entradas, procesos y salidas
- Una vista dinámica podría considerar aspectos como el control, el tiempo de las funciones (de comienzo a fin) y su comportamiento en situaciones excepcionales



Requerimientos No Funcionales

- Juegan un papel **crucial** en el **diseño** y **desarrollo** del sistema de información
- Pueden definirse como **consideraciones** o **restricciones** asociadas a un servicio del sistema
- Suelen llamarse también **requerimientos de calidad** o **no comportamentales** en contraste con los comportamentales
- Pueden ser tan **críticos** con los funcionales



Captura de Requerimientos

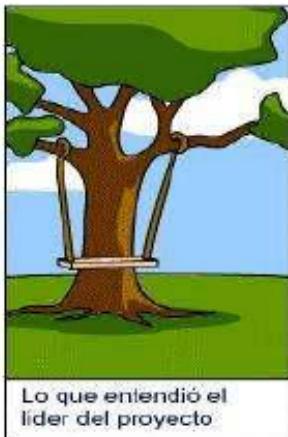


Captura de Requerimientos

Problemas



La solicitud del usuario



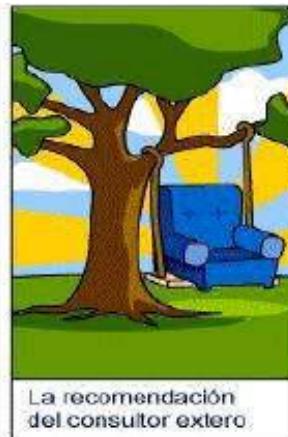
Lo que entendió el líder del proyecto



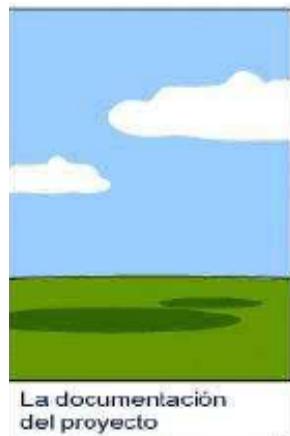
El diseño del analista de sistemas



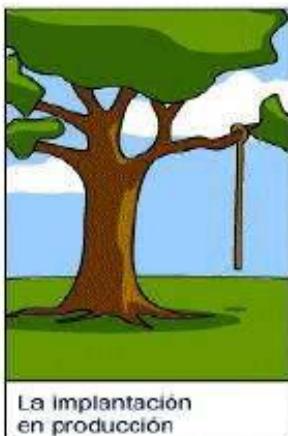
El enfoque del programador



La recomendación del consultor externo



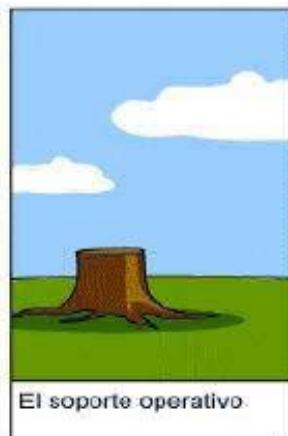
La documentación del proyecto



La implantación en producción



El presupuesto del proyecto



El soporte operativo



Lo que el usuario realmente necesitaba

Captura de Requerimientos

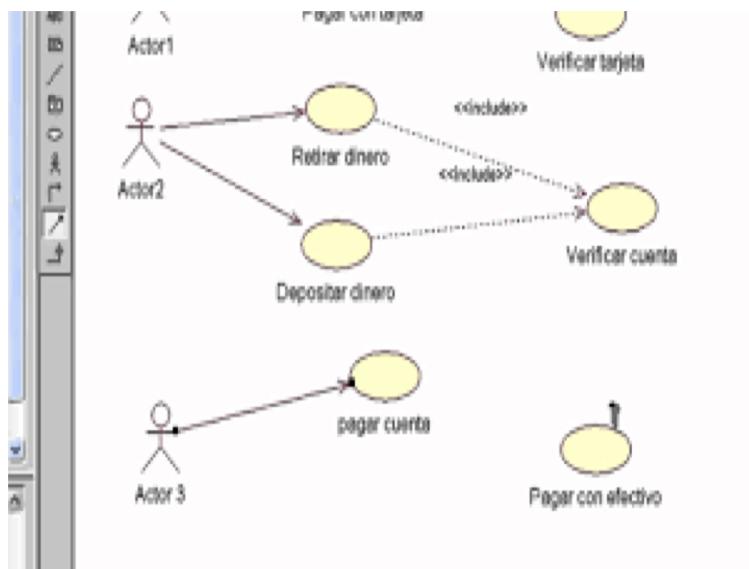
Problemas

- Dificultad en el seguimiento de los cambios en los requerimientos
- Dificultad en la especificación de los requerimientos
- Errores en la detección de características esperadas para el software
- Mala organización
- Los requerimientos no son siempre obvios y provienen de fuentes diferentes
- Los requerimientos no son siempre fáciles de expresar de manera clara mediante palabras
- Existen diferentes tipos de requerimientos a diferentes niveles de detalle
- El número de requerimientos puede volverse inmanejable si no se controla
- Existen varias partes interesadas y responsables, lo que significa que los requerimientos necesitan ser gestionados por grupos de personas de funciones cruzadas
- Los requerimientos cambian ya sea durante el desarrollo del proyecto como una vez que el sistema está funcionando

Captura de Requerimientos

Técnicas de Especificación

Casos de Uso (UML)



Historias de Usuario (métodos ágiles)

Historia: Registrarse
Como: Lector del Blog
Quiero: suscribirme al Blog

Para: poder realizar comentarios a las entradas de mi interés

Historia: Ingresar al Blog
Como: Lector del Blog
Quiero: Ingresar al Blog con mi usuario y contraseña
Para: realizar comentarios y mantenerme en contacto con otros lectores que comparten mis intereses

2

2

Probar el Software

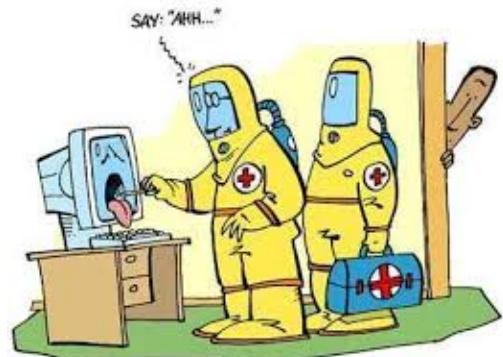
Testing

- Pertenece a una actividad denominada Verificación y Validación V&V
- V&V son las actividades que aseguran que el software respeta su especificación y satisface las necesidades de los clientes
- **Validación:** Estamos construyendo el producto correcto?
- **Verificación:** Estamos construyendo el producto correctamente?
- La verificación consiste en corroborar que el programa respeta su especificación; y validación significa corroborar que el programa satisface las expectativas del usuario
- Según Dijkstra: “*Si uno de los casos de prueba detecta un error el programa es incorrecto, pero si ninguno de los casos de prueba encuentra un error no podemos decir que el programa es correcto*”

Probar el Software

Testing

- Proceso DESTRUCTIVO de tratar de encontrar defectos en el código.
- Se debe ir con una actitud negativa para demostrar que algo es incorrecto
- El desarrollo exitoso del testing es cuando se encuentran errores NO cuando no se encuentran



Desarrollo de Software

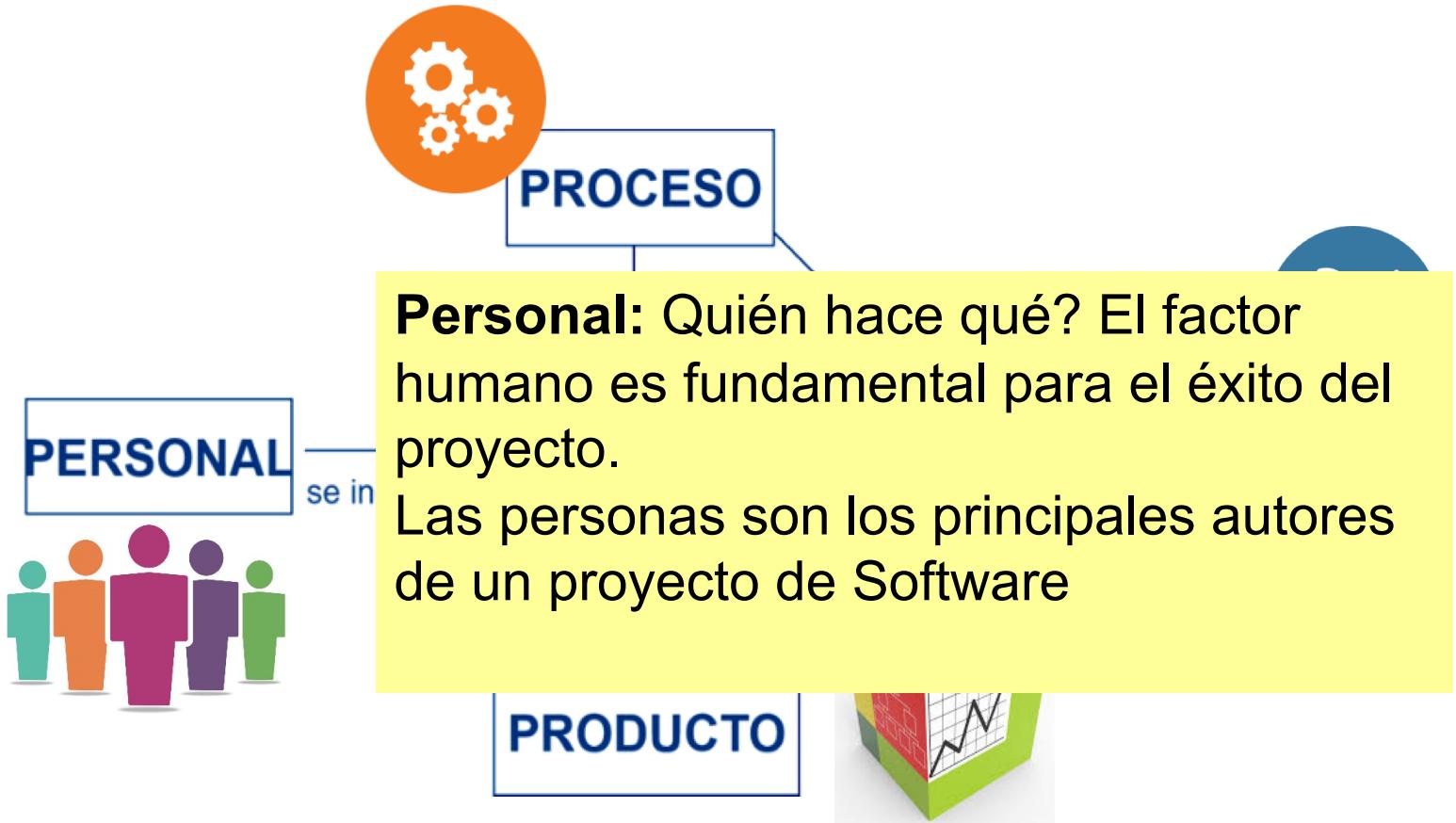
Las 4 P

El software como producto se desarrolla en forma gradual, cada versión es única y se obtiene como resultado de la ejecución de un Proyecto



Desarrollo de Software

Las 4 P



Desarrollo de Software

Las 4 P



Proceso: Cómo se hacen las cosas?.

Un proceso de software define un conjunto completo de actividades necesarias para transformar los requerimientos de un usuario en un producto



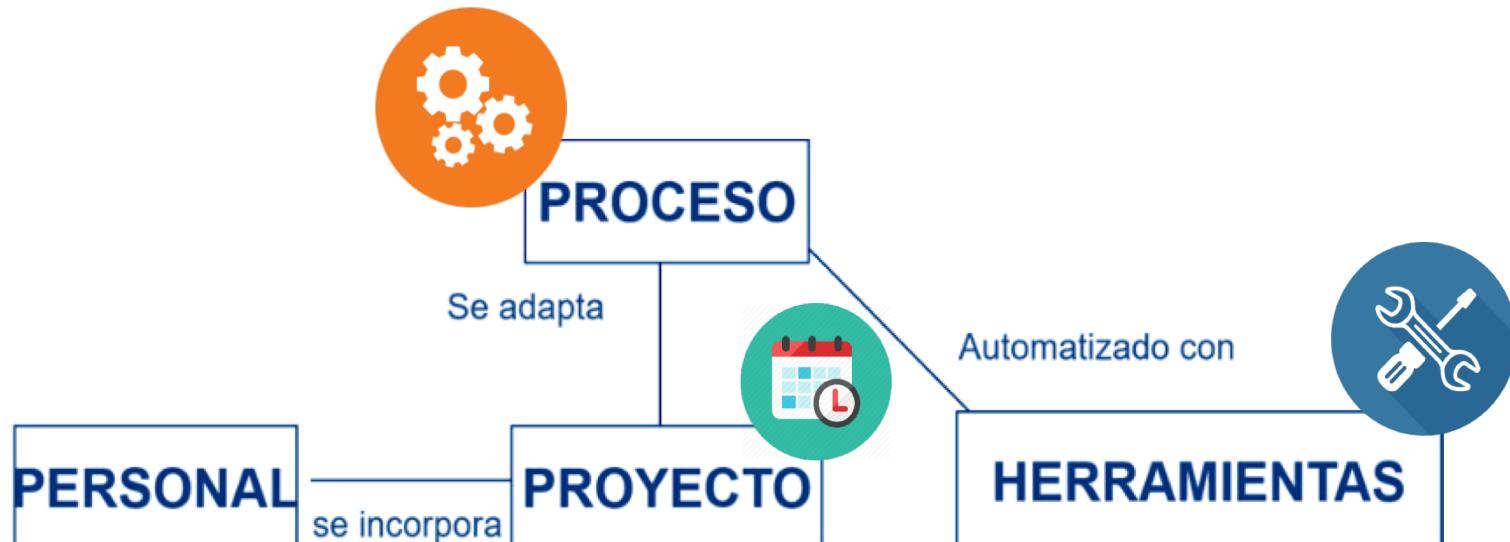
Obtiene como resultado

PRODUCTO



Desarrollo de Software

Las 4 P



Proyecto: Una ejecución única.
Es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único, en este caso un producto de software

Desarrollo de Software

Las 4 P



Producto: El resultado.

Es el conjunto de artefactos o componentes que se obtienen como salida de cada una de las actividades definidas en un proceso, que se ejecutan en un proyecto

PE



se incorpora

Obtiene como resultado

PRODUCTO



Las 4 P

Personas (*roles*)



- Las personas asumen diferentes roles, conformando el **Equipo de Desarrollo**:
 - Analista de sistemas
 - Diseñadores
 - Programadores
 - Arquitectos
 - Analistas de pruebas
 - Líderes de proyectos
 - Revisores técnicos
- Factores para conformar el Equipo de Desarrollo:
 - Experiencia en el dominio de aplicación
 - Experiencia en la plataforma y el lenguaje de programación
 - Habilidad para resolver problemas
 - Habilidad de comunicación
 - Adaptabilidad
 - Actitud
 - Personalidad



Las 4 P

Proyecto

- Integra personas, utiliza proceso y herramientas para obtener como resultado un producto de software
- Al principio se debe adaptar el proceso y el ciclo de vida al proyecto
- Un proyecto tiene las siguientes características:
 - **Temporal:** Posee fecha de comienzo y finalización
 - **Productos, servicios o resultados únicos:** Los resultados que se obtienen por similares que sean dos proyectos, tienen características que los hacen únicos
 - **Orientado a objetivos:** Los objetivos deben ser claros, cuando todos comprenden lo que hay que lograr; y alcanzables cuando es factible de hacerse
 - **Elaboración gradual:** Desarrollo en pasos que aumenta mediante incrementos

Las 4 P

Producto

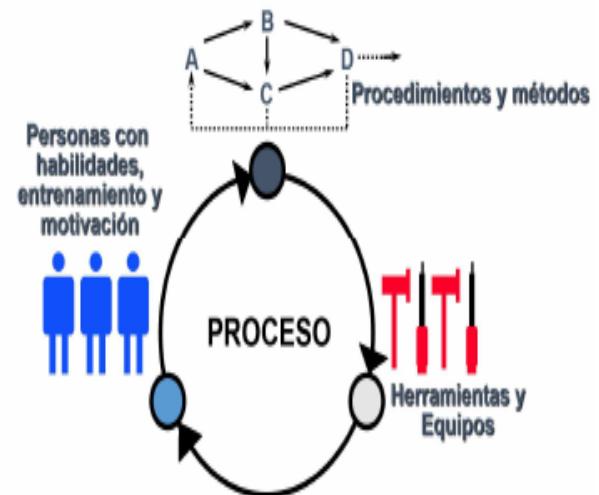
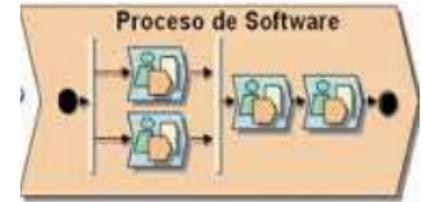


- Se obtiene como consecuencia de una evolución y refinamiento continuo de los modelos
- El producto es más que código, el producto que se obtiene es un sistema de software
- Un **Artefacto** se refiere a cualquier documentación creada, producida, cambiada o utilizada por las personas en el desarrollo del sistema
- Un **sistema de software** es la sumatoria de todos los artefactos que se necesitan para representarlo en una forma comprensible para las máquinas, los trabajadores y los interesados

Las 4 P

Proceso

- Proporciona un marco de trabajo, una estructura que puede tomarse como referencia para definir el plan que guiará el proyecto
- Contiene además de la definición de las actividades, procedimientos y métodos, la identificación de las herramientas que permitan la automatización de las actividades y facilite el trabajo
- El proceso contiene actividades relacionadas con las disciplinas técnicas, de gestión y de soporte o protectoras
- **El Ciclo de Vida del Proceso de Desarrollo de Software** indica las fases/etapas del proceso y el orden en el que se llevarán a cabo



Ciclo de Vida de un Sistema

- Para el diseño y desarrollo de un producto de software se aplican metodologías, modelos y técnicas que permiten resolver problemas
- Se han definido varias metodologías de desarrollo que ayudan a desarrollar software de calidad
- De acuerdo a las características del producto a desarrollar será el ciclo de vida que se debe utilizar

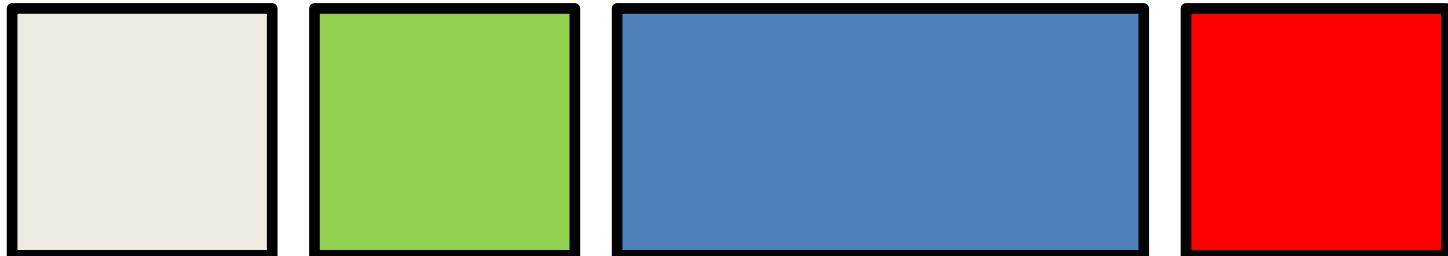
Modelos de Ciclo de Vida

Un *modelo de proceso o ciclo de vida*, es una representación abstracta de un proceso

- Cada modelo representa un proceso desde una perspectiva particular y así proporciona información parcial sobre el proceso
- Cada modelo de ciclo de vida ayuda al desarrollador a ordenar el trabajo a ser realizado durante la construcción del producto
- Cada modelo de ciclo de vida indica en qué orden se desarrollarán las etapas del ciclo de vida
- Existen varios Ciclos de Vida
 - Secuencial
 - Iterativo/Incremental
 - etc.

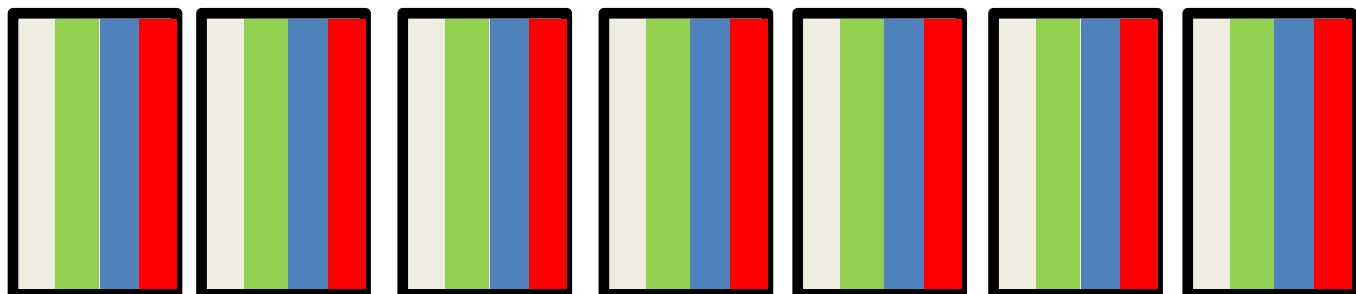
Modelos de Ciclos de Vida

100% Secuencial



Requerimientos Arquitectura Desarrollo Test

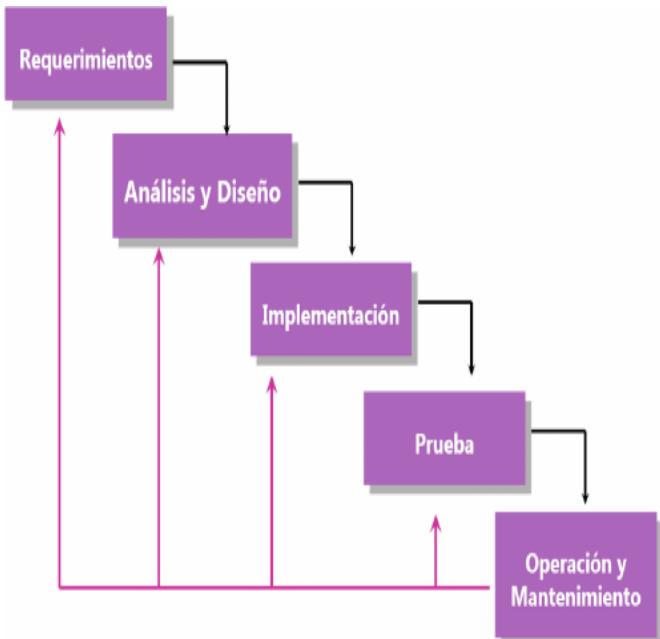
100% Iterativo



Modelos de Ciclos de Vida

Ciclo de Vida en Cascada

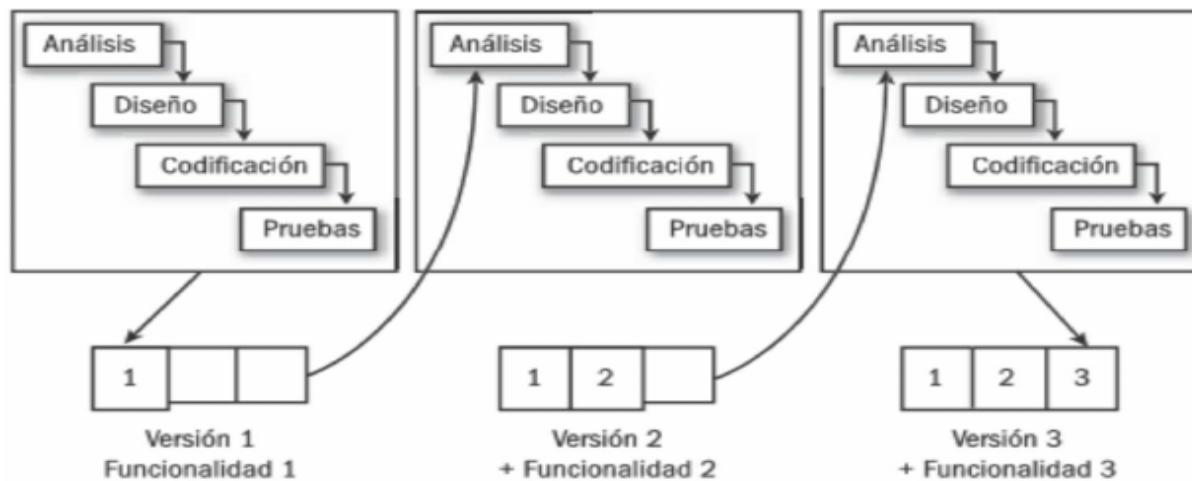
- Considera que las actividades (o etapas) del proceso de desarrollo: Requerimientos, análisis, diseño, implementación y prueba, son etapas separadas que para continuar con la siguiente se debe completar totalmente la anterior
- La retroalimentación con el cliente se realiza una vez que el producto ha sido terminado
- Es muy costoso volver a las etapas anteriores para realizar modificaciones por un malentendimiento de los requerimientos



Modelos de Ciclos de Vida

Ciclo de Vida Incremental

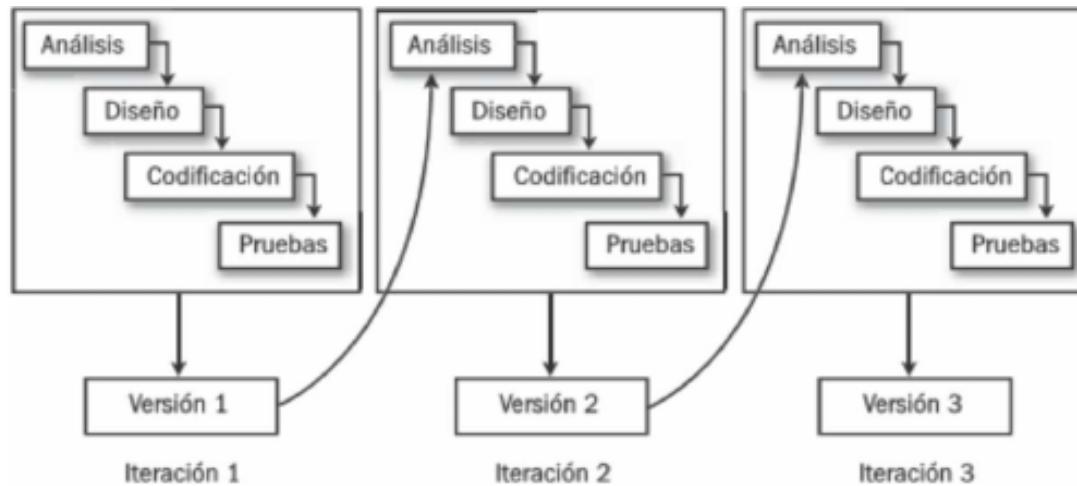
- El software se desarrolla gradualmente, por funcionalidades que incrementan el producto
- Se aplican pequeños ciclos de vida en cascada
- Luego de cada ciclo es posible realizar una entrega de software parcial al cliente



Modelos de Ciclos de Vida

Ciclo de Vida Iterativo

- Busca reducir la brecha entre las necesidades del usuario y el producto final
- Después de cada iteración se le entrega al cliente una versión del sistema
- El cliente evalúa el proyecto, lo corrige o propone mejoras



Metodología de Desarrollo de Software

- Una **metodología** define una forma disciplinada para desarrollar software con el objetivo de hacerlo más predecible y eficiente
- Una **metodología** describe el ciclo de vida a utilizar, es decir cómo las etapas del ciclo de vida se desarrollarán, y los artefactos a generar durante el desarrollo del producto
- De acuerdo a las características del producto se definirá la metodología de desarrollo de software a utilizar. Existen varios tipos de metodologías:
 - Orientado a dato o función: Sistemas de información clásicos
 - Métodos ágiles: Requerimientos cambiantes
 - Métodos formales: Validación de requerimientos formalmente (sistemas donde la vida humana es la entidad principal del proyecto)

Disciplinas en la Ingeniería de Software

Disciplina de Gestión



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Construcción del producto



Disciplinas de Gestión

- Planificación de Proyecto
- Monitoreo y Control de Proyectos

Guía para la construcción del software



Disciplinas de Soporte

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

Brindan herramientas para la calidad

Disciplinas de Gestión

Guía para la Construcción de Software



- Planificación de Proyecto
- Monitoreo y Control de Proyectos

Estimar: Predecir el tiempo y el costo que llevará desarrollar un producto de software, basándose en el tamaño de lo que se desea construir



Disciplinas de Gestión

Guia para la Construcción de Software



Disciplinas de Gestión

- Planificación de Proyecto
- Monitoreo y Control de Proyectos

Planificación de Proyecto: Un plan es a un proyecto, lo que una hoja de ruta es a un viaje.

Planificar es definir qué es lo que haremos, cuándo lo haremos, cómo vamos a hacerlo y quién lo hará



Disciplinas de Gestión

Guía para la Construcción del Software



Disciplinas de Gestión

- Planificación de Proyecto
- Monitoreo y Control de Proyectos

Monitoreo y Control de Proyectos: El monitoreo y control compara los planes realizados con el avance real de un proyecto. Da visibilidad respecto de la situación del proyecto en un momento de tiempo, cuánto hicimos, cuánto nos falta para terminar, cuánto consumimos y cuánto tiempo consum



Disciplinas en la Ingeniería de Software



Disciplinas Técnicas

- Requerimientos
- Análisis y Diseño
- Implementación
- Prueba
- Despliegue

Construcción del producto

Disciplinas de Gestión



- Planificación de Proyecto
- Monitoreo y Control de Proyectos

Guía para la construcción del software



Disciplinas de Soporte

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

Brindan herramientas para la calidad



Disciplinas de Soporte

Herramientas de Calidad

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

Gestión de configuración de software: Una de las características del software es que es fácil de modificar, maleable y dado que vamos a necesitar cambiarlo muchas veces, es necesario crear mecanismos que nos ayuden a controlar el software a lo largo de su ciclo de vida





Disciplina de Soporte

Aseguramiento de Calidad

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

Aseguramiento de Calidad:

El aseguramiento de la calidad del software es una actividad de protección que se aplica a lo largo de todo el proceso de construcción del software



Herramientas de Soporte

Revisar Técnicamente el Software

- Gestión de Configuración de Software
- Aseguramiento de Calidad
- Métricas

Revisar técnicamente el software: Tiene por objetivo detectar tempranamente errores que se cometan al desarrollar

Dado que el software es invisible, es esperable que posea errores

