Introducción a la ingeniería de software – Clase 2

En la clase anterior ...

Ingeniería de software
Aplicación de la ingeniería al software: desarrollo, mantenimiento

Participantes y roles
Cliente + desarrollador + usuario
Múltiples relaciones.

Miembros del equipo
Diversos miembros participan de más de una actividad

Sistema
Sistema de software. Es producto y es herramienta.

Proceso
Conjunto de actividades y tareas a ejecutar para crear algún producto

Temas de hoy

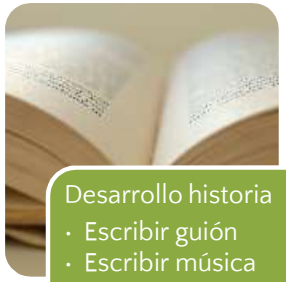
Flujos de proceso
Tipos y características.

Modelos de proceso
Ciclo de vida del software.
Diferencias de acuerdo el flujo de proceso

1 Proceso

Una posible organización del proceso

Investigación
- Entrevistar personas involucradas
- Escribir libro



Desarrollo historia
- Escribir guión
- Escribir música

Contratación
- Director
- Casting actores

Desarrollo película
- Filmar escenas
- Grabar banda Sonora

Producción
- Editar película
- Presentar internamente

Otra forma ...

Investigación
- Entrevistar personas involucradas
- Escribir libro

Desarrollo historia
- Escribir guión
- Escribir música

Desarrollo película
- Filmar escenas
- Grabar banda Sonora

Producción
- Editar película
- Presentar internamente

Contratación
- Director
- Casting actores

Flujo del proceso

Organización de las actividades y sus tareas con respecto a secuencia y tiempo:

- Lineal
- Iterativo
- Evolutivo
- Paralelo

Proceso del software

Actividades estructurales (Pressman):

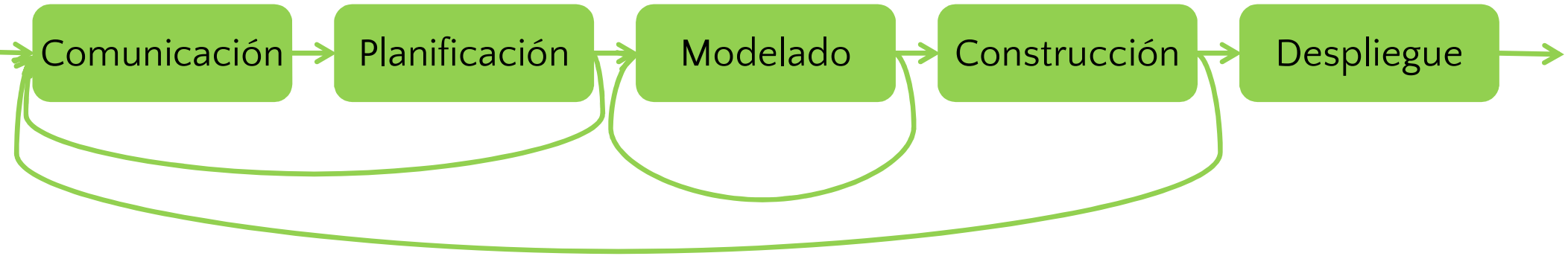
1. Comunicación
2. Planificación
3. Modelado
4. Construcción
5. Despliegue

Flujo del proceso - Lineal

- Lineal: secuencia de actividades

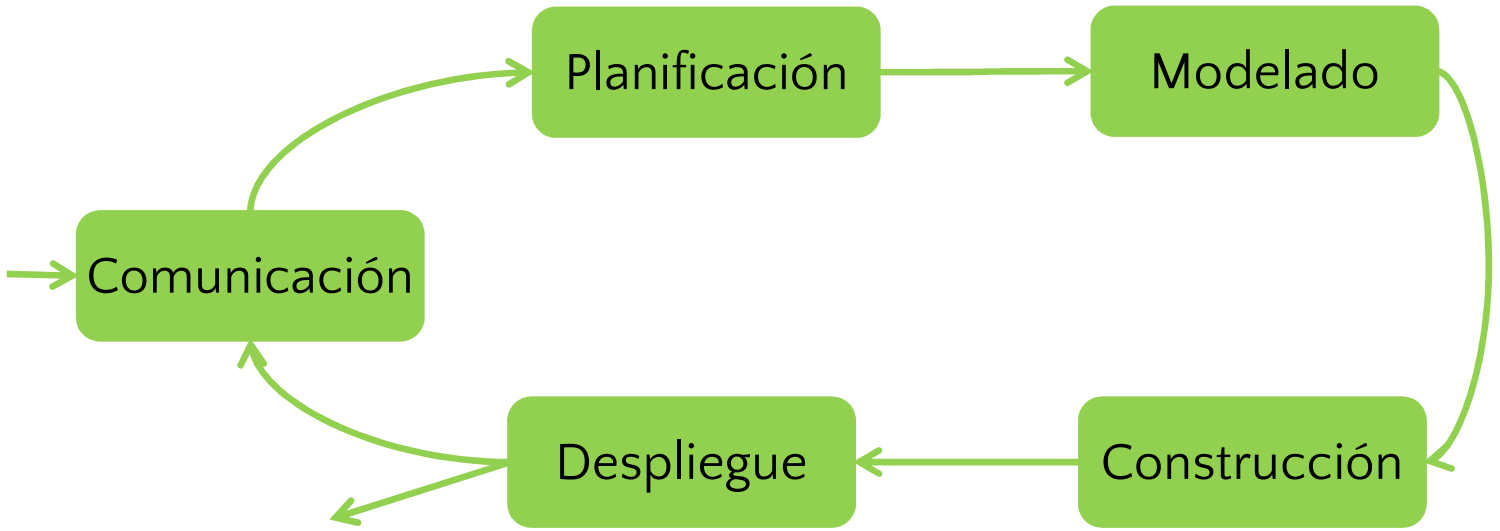
Flujo del proceso - Iterativo

- Iterativo: repite una o más de las actividades antes de pasar a la siguiente



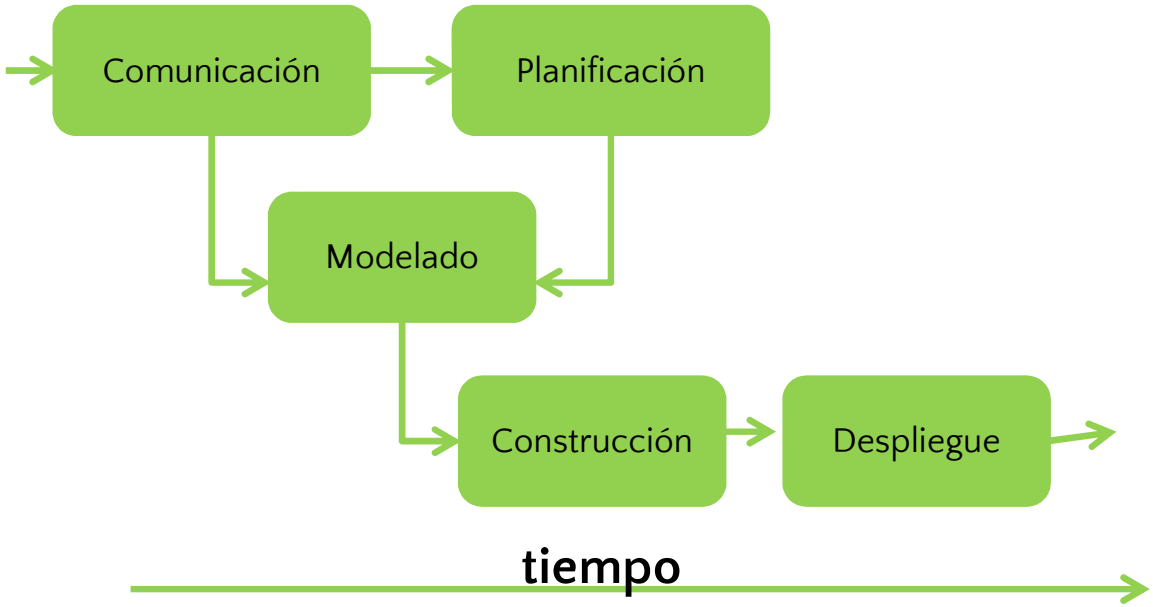
Flujo del proceso - Evolutivo

- Evolutivo: cada circuito lleva a una versión más completa del producto



Flujo del proceso - Paralelo

- Paralelo: una o más actividades en paralelo con otras



2

Modelos de proceso

Ciclo de vida del software

Modelos de proceso

- Cascada
  - Modelo en V
- Incremental
- Iteraciones: modelo evolutivo
  - Prototipos
  - Espiral
- Concurrente (*)
- Especializados

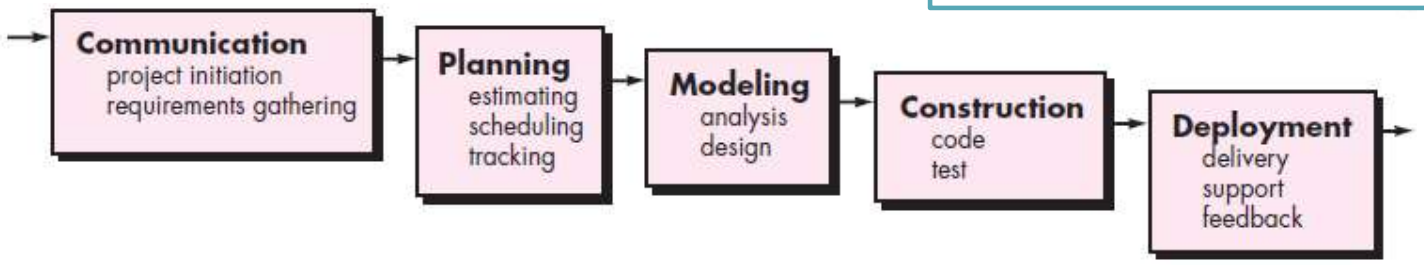
Prescriptivos

Modelo de cascada

- El trabajo fluye de manera lineal desde la primera actividad hasta la última

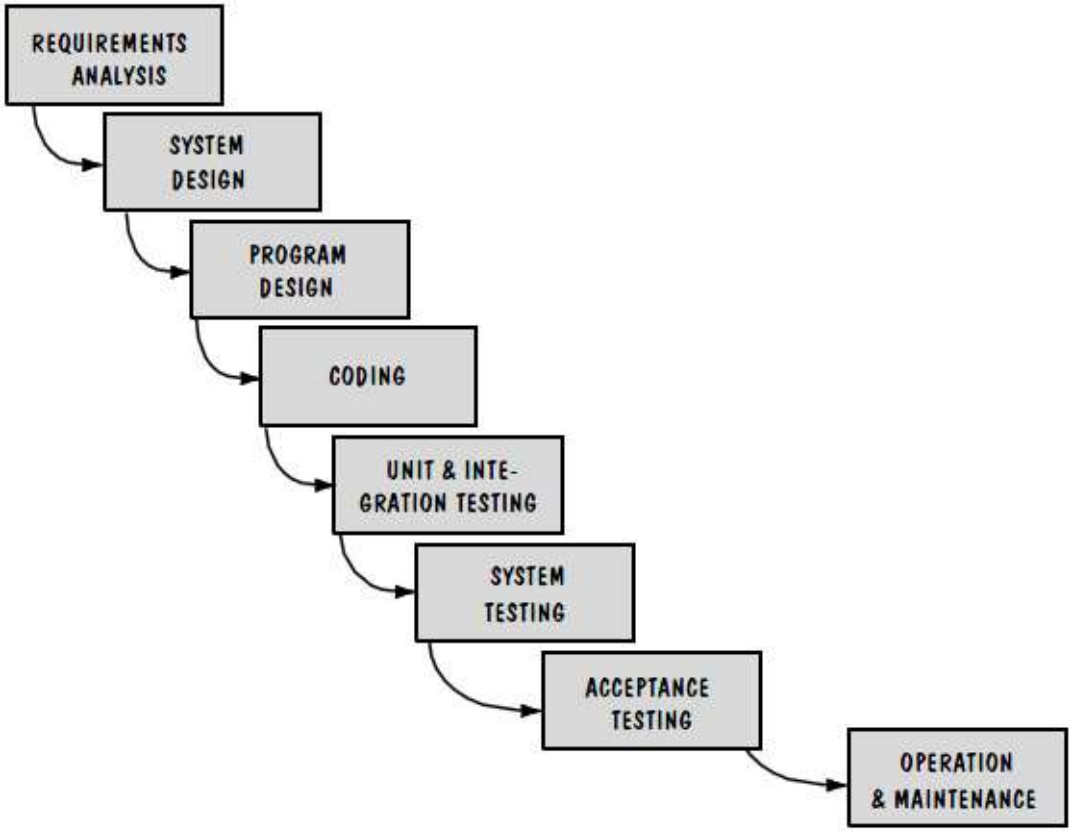
- Requerimientos bien definidos o para mejora de sistema existente.

¿Qué desventajas presenta este modelo?



R.Pressman

Modelo de cascada



S. Pfleeger

Modelo de cascada

Ventajas

- Es simple de explicar y entender para personas no familiarizadas con el desarrollo de software
- Explicita productos intermedios necesarios para avanzar de etapa

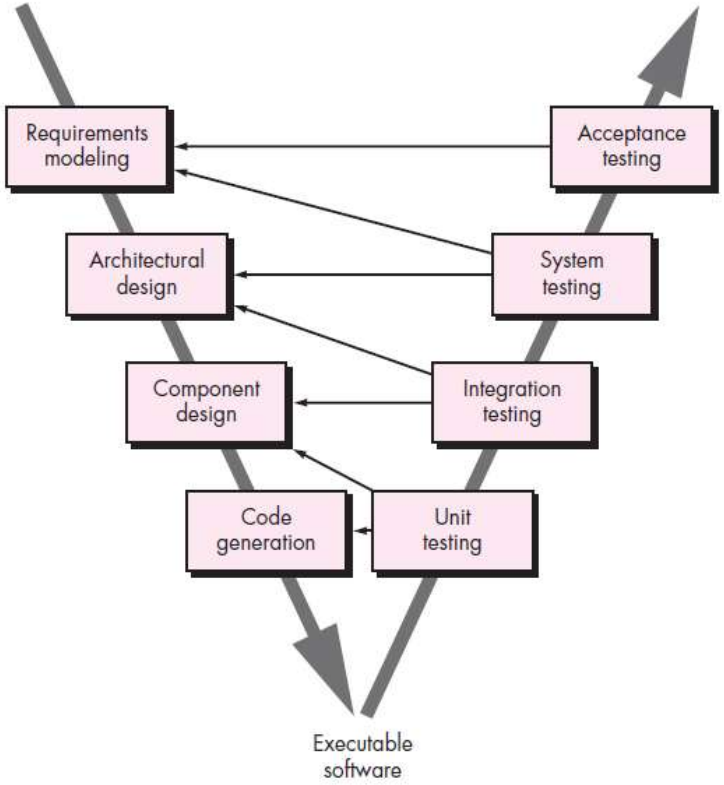
Desventajas

- No refleja la realidad más frecuente. No es flexible, no se puede aplicar cuando hay requerimientos dinámicos.
- El cliente debe esperar por una versión funcional

Modelo V (cascada en V)

Variante del modelo de cascada

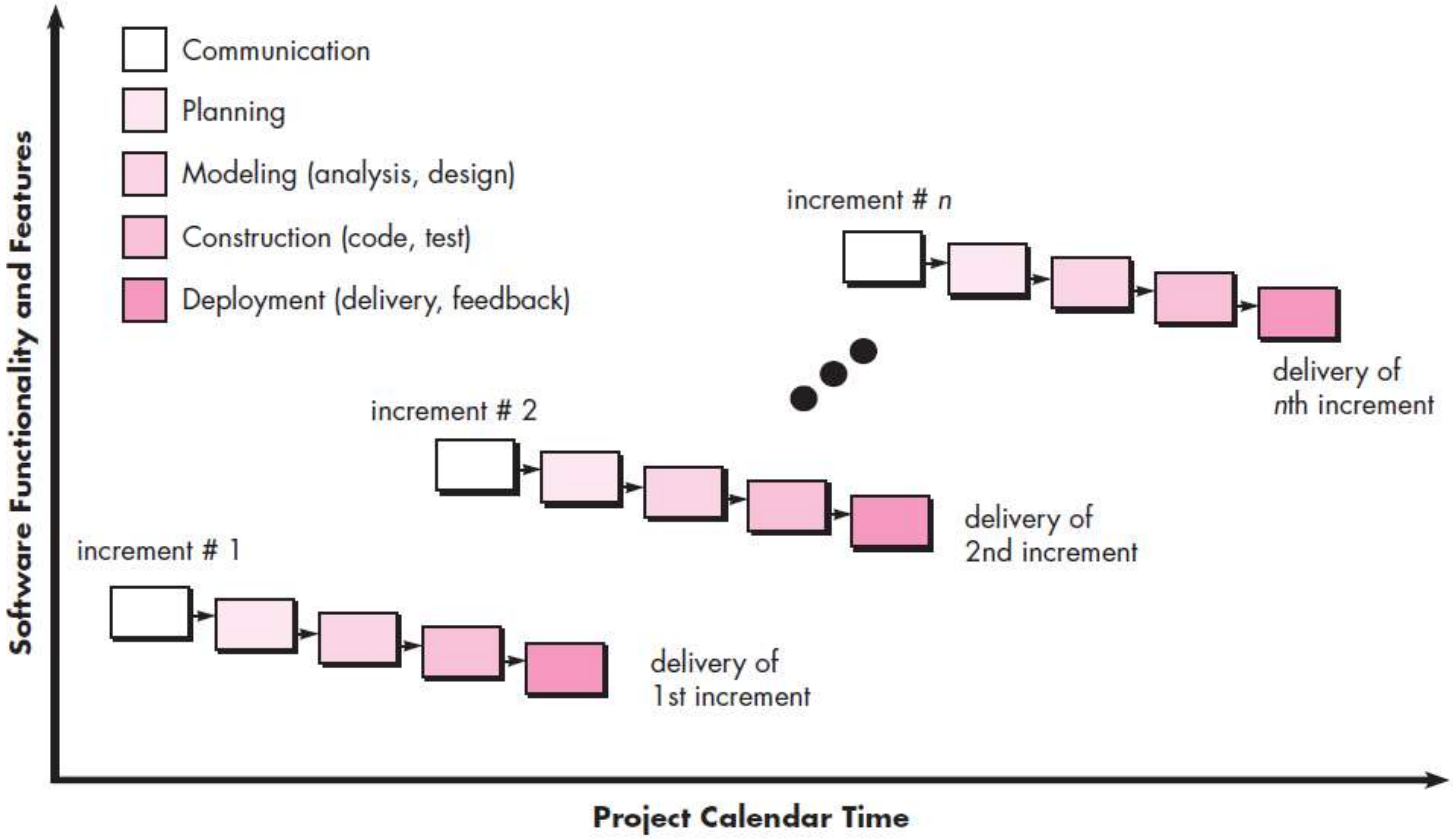
- Permite visualizar las acciones de verificación y validación
- Hace más explícita la necesidad de iteración para rehacer tareas si hay cambios o errores



Modelo incremental

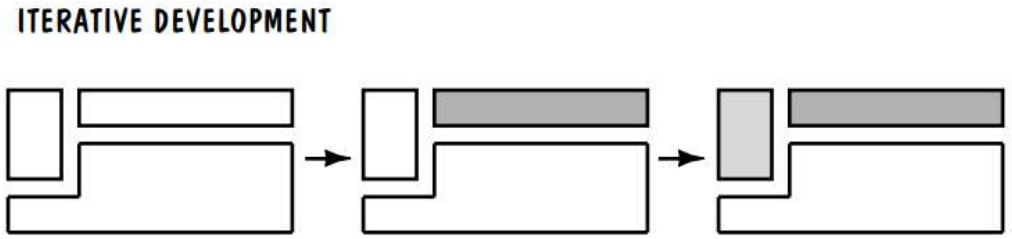
- Secuencias lineales aplicadas de manera escalonada
- Puede haber un grado de paralelismo entre las distintas secuencias
- Cada secuencia produce “incrementos” en el SW
- Útil cuando no se dispone de personal suficiente para la implementación completa
- Ayuda a la administración de riesgos técnicos

Modelo incremental



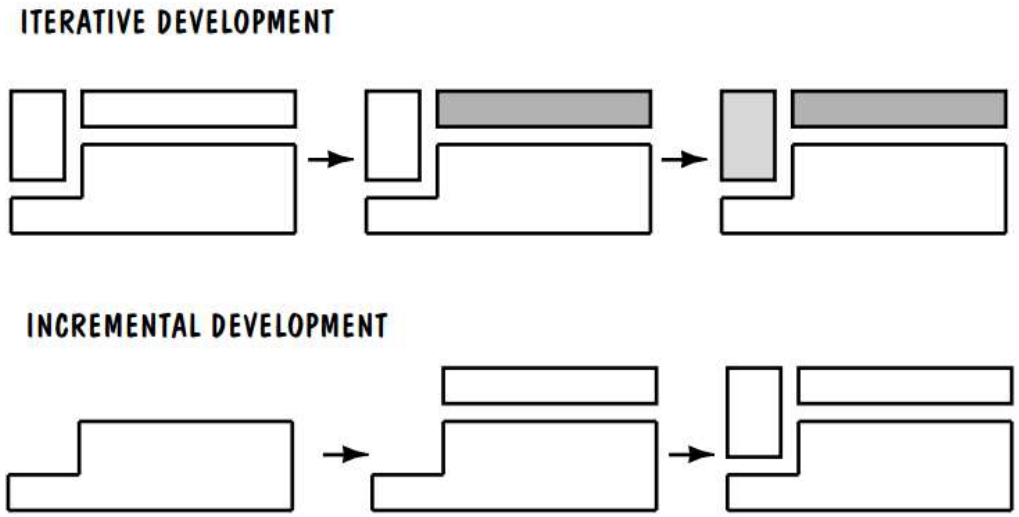
Modelo por iteraciones

- El sistema se entrega completo al comienzo
- Las iteraciones modifican (mejoran, corrigen, extienden) la funcionalidad



Modelo por iteraciones

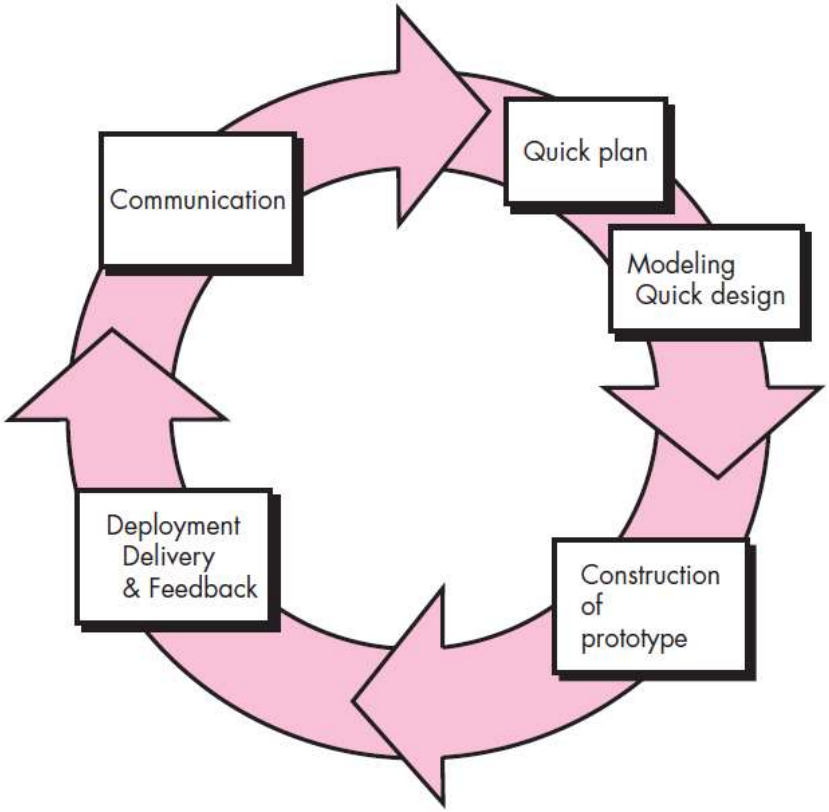
- Iterativo vs. Incremental



Modelo evolutivo - prototipos

- Inicialmente se planea rápidamente una iteración para producir un primer prototipo
- Cada iteración refina el conocimiento, el plan y el modelo, y genera un nuevo prototipo
- Útil cuando inicialmente el cliente solo define objetivos generales y no identifica requerimientos detallados
- Algunos prototipos son desechables, otros evolucionan

Modelo evolutivo - prototipos



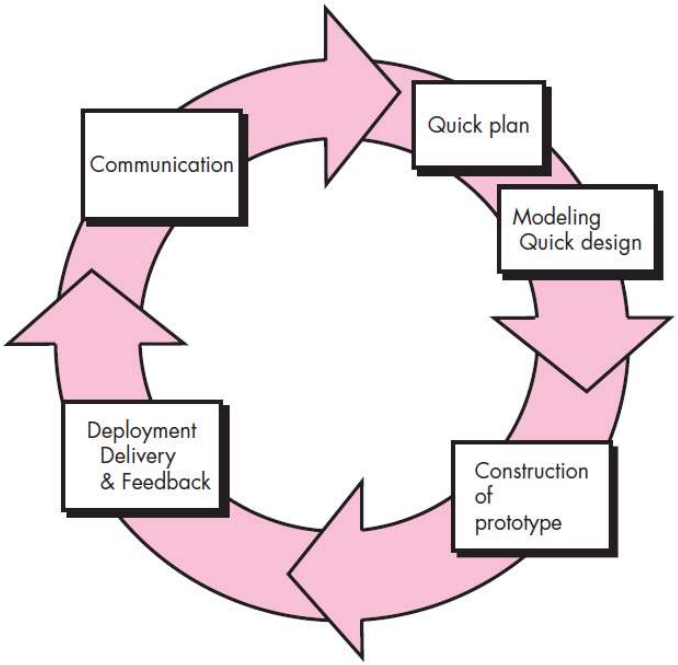
Modelo evolutivo - prototipos

Ventajas

- Dar una idea real más temprana al usuario
- Obtener mejor definición de los requerimientos

Desventajas

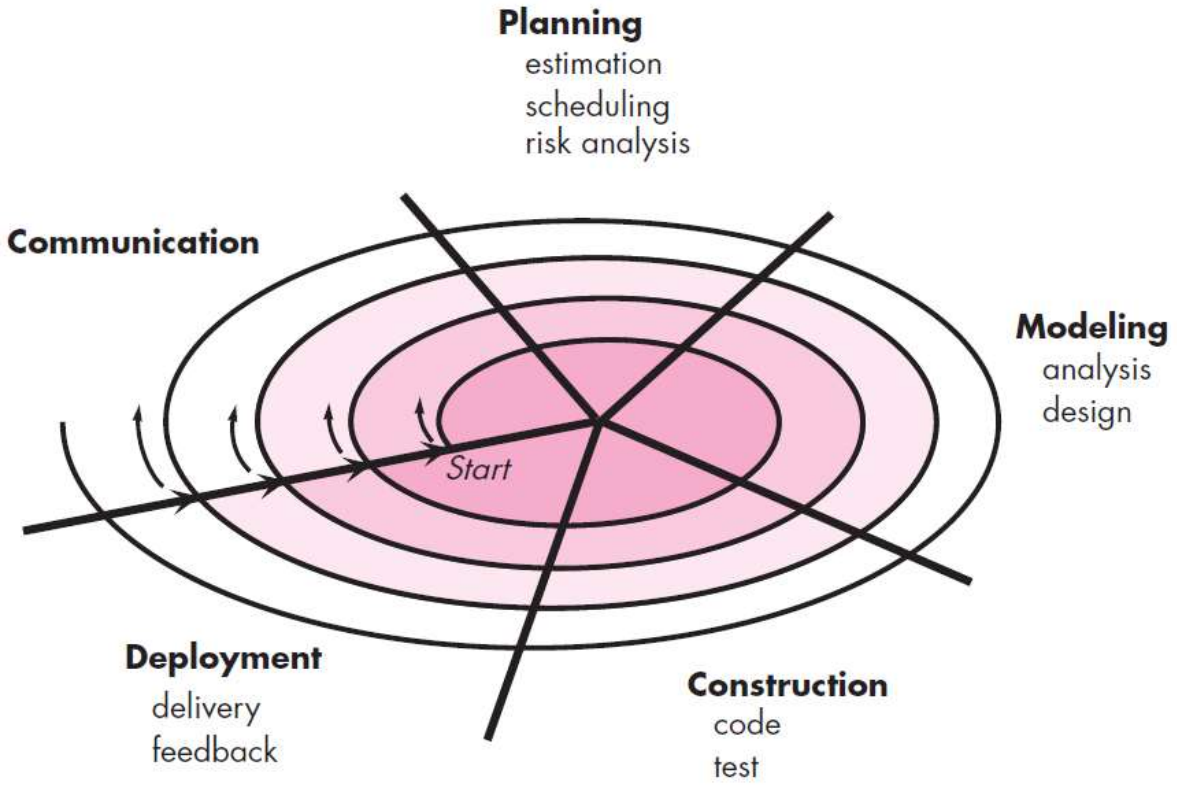
- Riesgo de perder de vista que lo que se obtiene es un prototipo y no el producto definitivo



Modelo evolutivo - espiral

- Se itera en forma de espiral
- Las primeras iteraciones pueden producir modelos o prototipos
- En cada iteración se ajustan todos los documentos: plan, definición, modelos, costos, incluso el número de iteraciones necesarias
- Es adaptable para usarse durante todo el ciclo de vida del sistema

Modelo evolutivo - espiral



Modelos de proceso especializados

Tienen las características de los procesos vistos.
Se aplican para enfoques específicos.

- Desarrollo basado en componentes
- Métodos formales
- Orientado a aspectos (*)

Desarrollo basado en componentes

- Componentes de software como productos
- Se integran en el software a construir
- Naturalmente evolutivo, tiene características del modelo en espiral
  1. Investigar y evaluar productos disponibles
  2. Considerar aspectos de integración
  3. Diseñar la arquitectura
  4. Integrar componentes en la arquitectura
  5. Realizar pruebas exhaustivas

Desarrollo basado en componentes

Ventajas

- Aumenta la reutilización
- Se hace más medible el proceso y el producto

Desventajas

- Perdida de control
- Costo de integración

Métodos formales

- Especificación matemática del software y sus características
- Especificar, desarrollar y verificar los productos de manera rigurosa

Ventajas

- Posible verificar y validar de manera rigurosa
- Permite descubrir ambigüedades e inconsistencias
- Baja costos de mantenimiento

Desventajas

- Pocos desarrolladores con la capacitación adecuada
- Puede encarecer el desarrollo
- Dificil de utilizar para comunicarse con los clientes

3 Metodologías ágiles

Manifiesto ágil

Grupo conocido como “Alianza Ágil” firma:

Estamos descubriendo formas mejores de desarrollar software , desarrollándolo y ayudando a otros a desarrollarlo. Ese trabajo nos ha hecho valorar:

- Los **individuos y sus interacciones** más que a los procesos y las herramientas
- El **software que funciona** más que la documentación exhaustiva

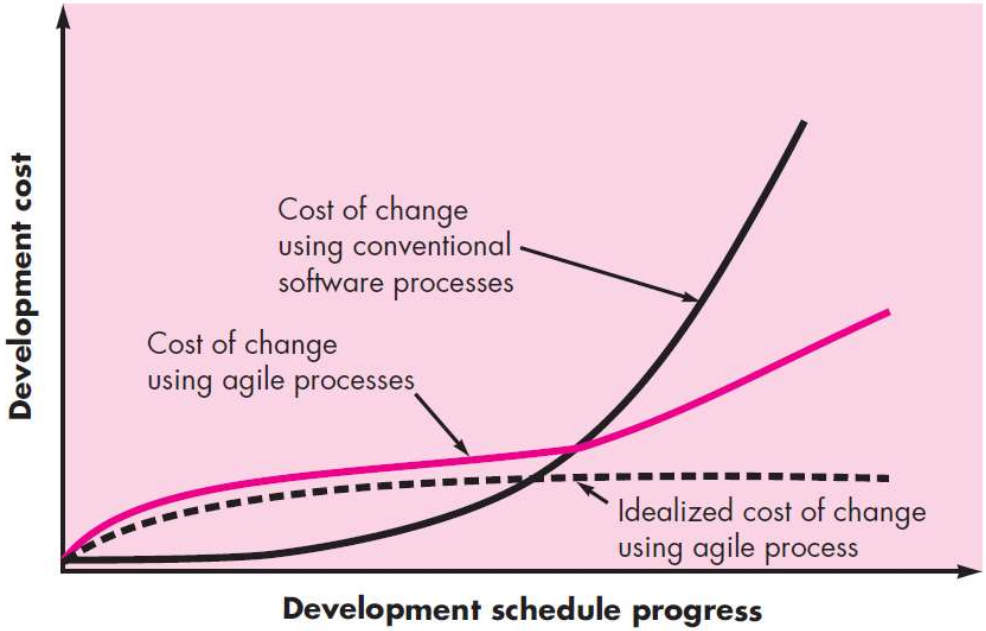
Manifiesto ágil (continuación)

Ese trabajo nos ha hecho valorar:

- La **colaboración con el cliente** más que la negociación de contratos
- **Responder al cambio** más que seguir un plan

Costo del cambio

- El costo se incrementa de manera no lineal con el avance del proyecto



Proceso ágil

- Suposiciones
  - Es difícil predecir: persistencia de los requerimientos, cambios de prioridades del cliente
  - En muchos tipos de software el diseño y la construcción se superponen: deben ser simultáneos
  - Análisis, diseño, construcción y pruebas no son tan predecibles como quisieramos: es difícil planificar
- El proceso debe ser **adaptable**, debe hacerlo **incrementalmente** y debe permitir la **retroalimentación** con el cliente

Principios de agilidad

1. La prioridad más alta es satisfacer al cliente mediante la entrega pronta y continua de software valioso.
2. Los requerimientos cambiantes son bienvenidos, aún en etapas avanzadas del desarrollo. Los procesos ágiles aprovechan el cambio para la ventaja competitiva del cliente.
3. Entregar software funcional con frecuencia, entre dos semanas y dos meses, de preferencia lo más pronto posible.

Principios de agilidad

4. Las personas de negocio y los desarrolladores deben trabajar juntos, a diario, durante todo el proyecto.
5. Desarrollar proyectos con gente motivada. Darles el ambiente y el apoyo que necesiten, y confiar en que harán bien su trabajo.
6. El método más eficiente y eficaz para transmitir información a los integrantes de un equipo, y entre éstos, es la conversación cara a cara

Principios de agilidad

7. La medida principal del avance es el software que funciona
8. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deben poder mantener un ritmo constante en forma indefinida.
9. La atención continua a la excelencia técnica y el buen diseño mejora la agilidad

Principios de agilidad

10. La simplicidad es esencial – el arte de maximizar la cantidad de trabajo no realizado.
11. Las mejores arquitecturas, requerimientos y diseños surgen de los equipos con organización propia.
12. El equipo reflexiona a intervalos regulares sobre cómo ser más eficaz, para después afinar y ajustar su comportamiento en consecuencia.

Procesos ágiles

- Diferentes modelos enfatizan distintos principios. Todos siguen un “espíritu ágil”.

- Enfatizan la importancia de factores humanos:
  - Competencia, Enfoque común, Colaboración, Habilidad para tomar decisiones, Capacidad para resolver problemas difusos, Confianza y respeto mutuos, Organización propia.

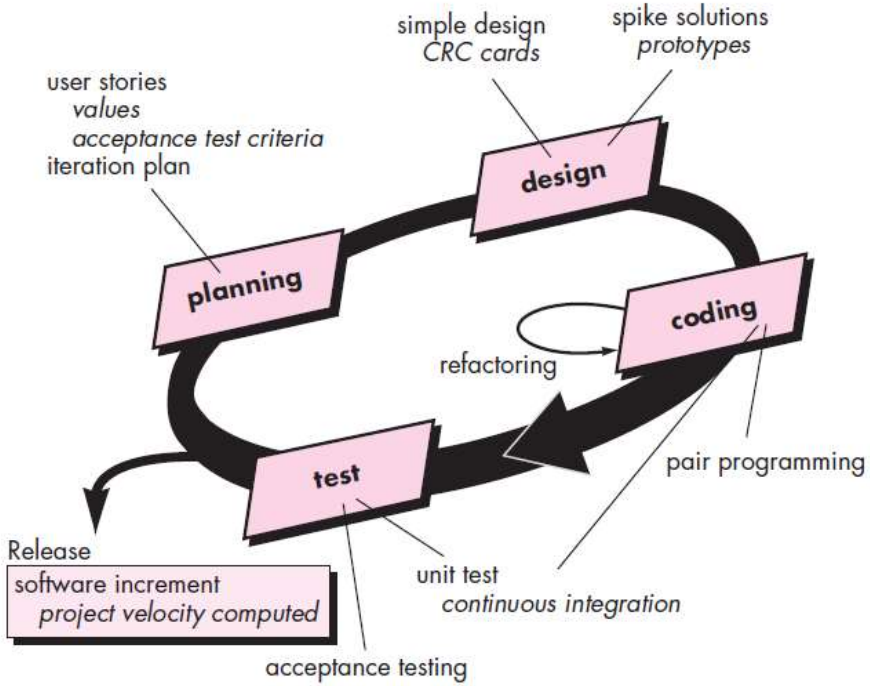
Programación Extrema (XP)

- Del inglés: eXtreme Programming
- Variante: IXP = Industrial eXtreme Programming

- Pone el énfasis en la colaboración estrecha pero informal entre clientes y desarrolladores para establecer **metáforas** que comuniquen conceptos importantes, en la retroalimentación continua y en evitar documentación voluminosa

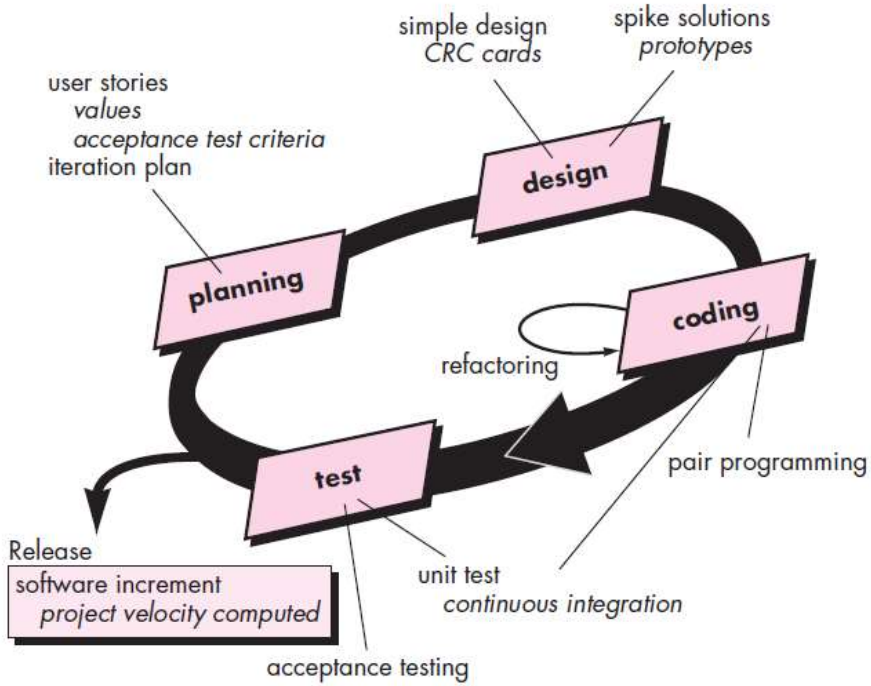
Programación Extrema (XP)

- Cuatro actividades estructurales: planificación, diseño, codificación y pruebas.



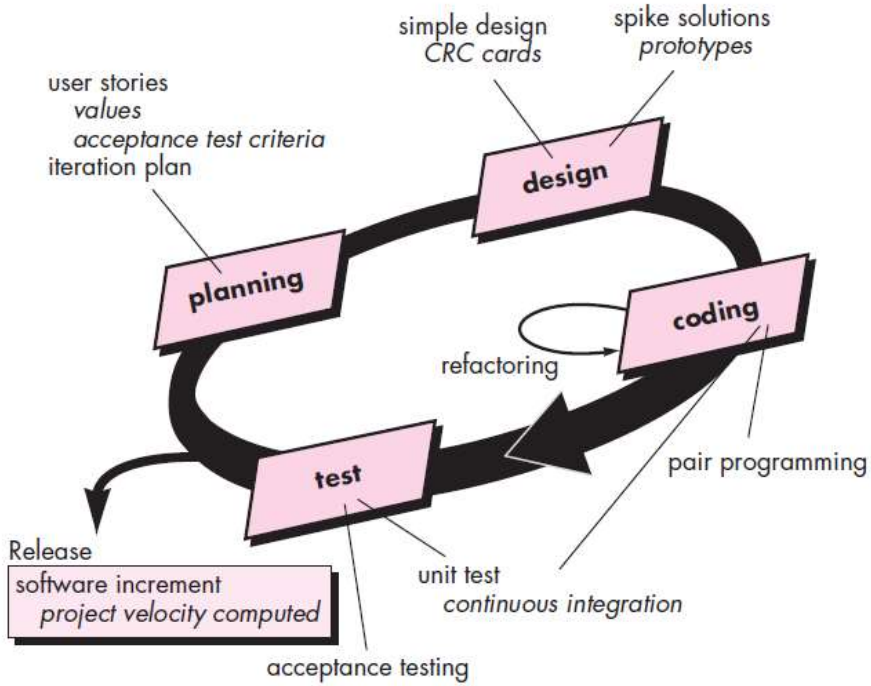
Programación Extrema (XP)

- Planificación:
  - recabar requerimientos,
  - crear historias y asignarles valores,
  - asignar costo medido en semanas de desarrollo.
  - Luego de la primera entrega se calcula “velocidad”
- Diseño:
  - Mantenerlo sencillo
  - Si es difícil: realizar prototipos



Programación Extrema (XP)

- Codificación:
  - ¡No codificar! Sino diseñar pruebas unitarias.
  - Luego desarrollar el código.
  - Utilizar **programación por parejas**.
- Pruebas:
  - Automatizarlas siempre que sea posible.
  - Efectuar pruebas de integración a diario.

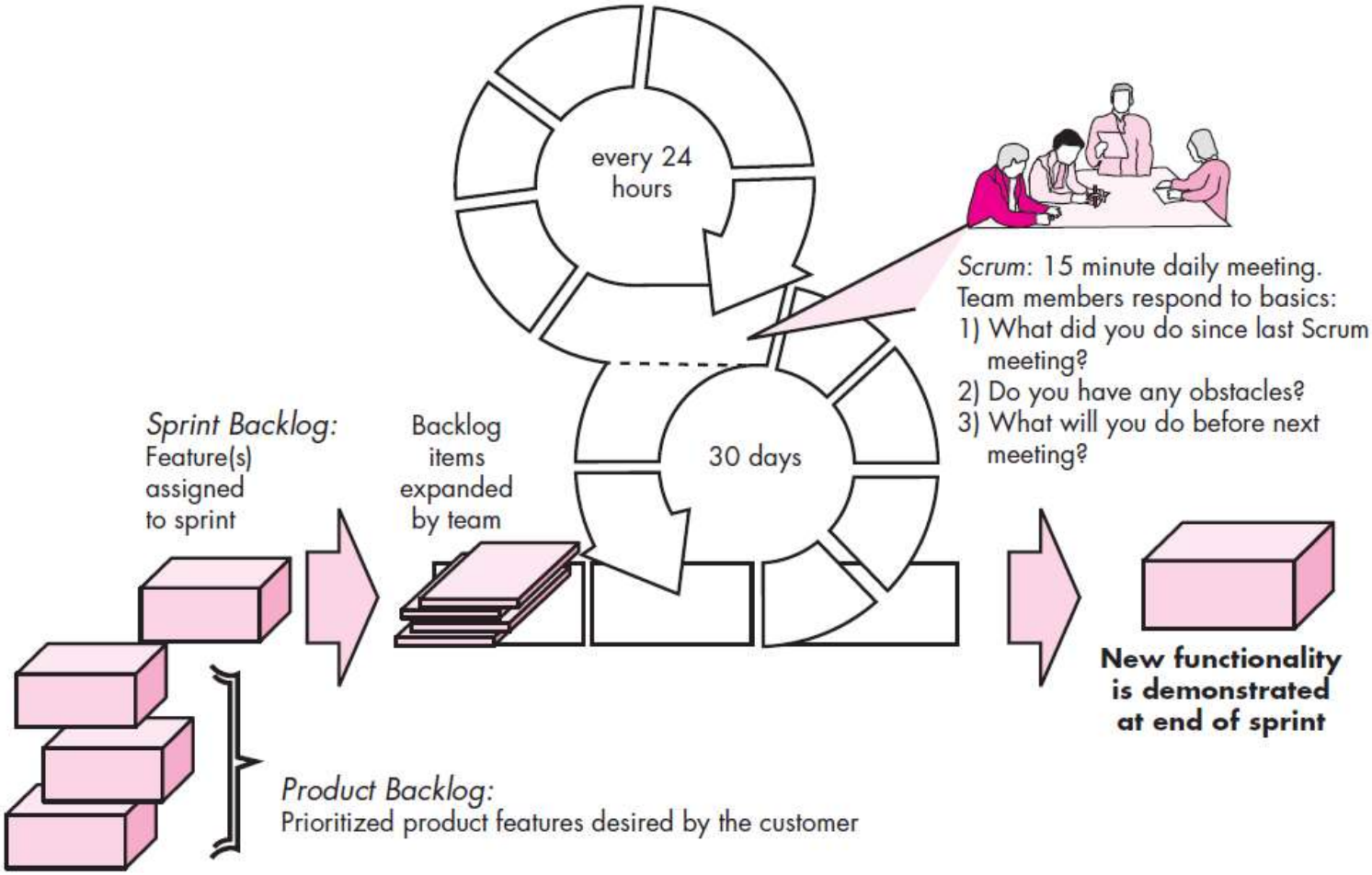


Scrum

Scrum

- Actividades estructurales:
  - Requerimientos
  - Análisis
  - Diseño
  - Evolución
  - Entrega
- Cada actividad realiza las tareas bajo un patrón de proceso llamado *sprint*

Scrum


every 24 hours
30 days
Scrum: 15 minute daily meeting.
Team members respond to basics:
1) What did you do since last Scrum meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog:
Feature(s) assigned to sprint
Backlog items expanded by team
New functionality is demonstrated at end of sprint
Product Backlog:
Prioritized product features desired by the customer

Scrum

- Backlog (pendientes o acumulados)
- Sprints: unidades de trabajo necesarias para alcanzar un requerimiento. No introducir cambios.
- Reuniones scrum: breves y diarias, para responder tres preguntas claves
  - ¿Qué hiciste desde la última reunión del equipo? (ayer)
  - ¿Qué obstáculos estás encontrando?
  - ¿Qué planeas hacer hasta la próxima reunión?
- Demostraciones preliminares: entrega de incremento

Modelos de proceso ágiles

- **XP**
- Desarrollo adaptativo de software (DAS)
- **Scrum**
- Método de desarrollo de sistemas dinámicos (MDSD)
- Cristal
- Desarrollo impulsado por las características (DIC)
- Desarrollo esbelto de software
- Modelado ágil
- Proceso unificado ágil

Resumen de hoy:

Flujo de proceso
Organización de las actividades y sus tareas con respecto a secuencia y tiempo

Ciclo de vida
Proceso de desarrollo de sw
Ciclo de vida del sw
Diferentes modelos de proceso

Metodologías ágiles
Modelo de proceso centrado en el cambio

Bibliografía

- *Ingeniería de software . Teoría y Práctica* - S. L. Pfleeger

  Capítulo 2 – Modelado del proceso y del ciclo de vida

- *Ingeniería del software. Un enfoque práctico* - R. Pressman

  Capítulo 2 – Modelos del proceso

  Capítulo 3 – Desarrollo Ágil