

Metodologías de desarrollo

Prof. Ernesto Rodríguez

Universidad del Itsmo

erodriguez@unis.edu.gt

- Conjunto de practicas y lineamientos para estructurar el proceso de desarrollo.
- Existen varias opciones, cada una con ventajas y desventajas según la naturaleza del proyecto.
- Muchas metodologías tienen herramientas asociadas a ellas.
- Generalmente, las metodologías no se siguen al pie de la letra, queda a criterio del equipo de desarrollo decidir como utilizar la metodología.

Grados de libertad en el Ciclo de vida de un Sistema

La metodología se debe seleccionar y adaptar según la naturaleza del sistema a desarrollar. Algunas de las variables son:

Participación del cliente	Ninguna participación ⇔ Parte del equipo
Requisitos	Rígidos ⇔ Flexibles
Planeación	Predictivo ⇔ Adaptivo
Confiabilidad del sistema	con errores ⇔ sin errores
Validación	automatica ⇔ manual ⇔ formal
Equipo	Pequeño y centrado ⇔ Grande y distribuido
Tamaño del sistema	usuarios contados ⇔ millones de usuarios
Tiempo de desarrollo	dias ⇔ años
Presupuesto del cliente	start-up ⇔ multi-nacional

- **Principio fundamental:** No se avanza a la siguiente etapa hasta terminar la etapa anterior.
- No es un modelo realista para la mayoría de sistemas.
- Sin embargo, puede utilizarse en algunos casos:
 - El costo de comunicación con el cliente es alto.
 - Alto grado de certeza en los requisitos. Ejemplo:
 - Migrar un sistema a tecnología nueva.
 - Delegar un componente de un sistema a otro equipo.
 - Un sistema con propósitos demostrativos.
 - El sistema requiere un grado muy alto de confiabilidad: Modificar requisitos \Rightarrow nuevos puntos de fallo.

Waterfall

The classic waterfall development model

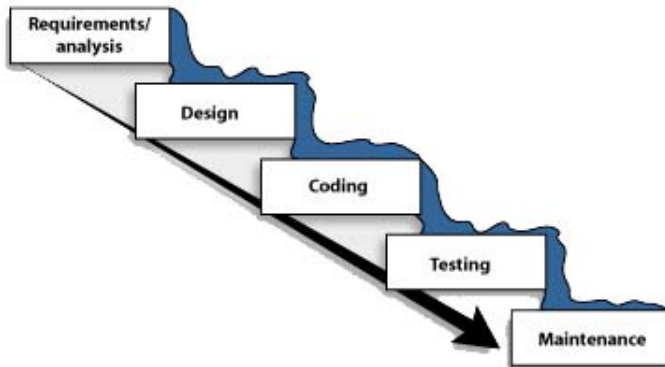


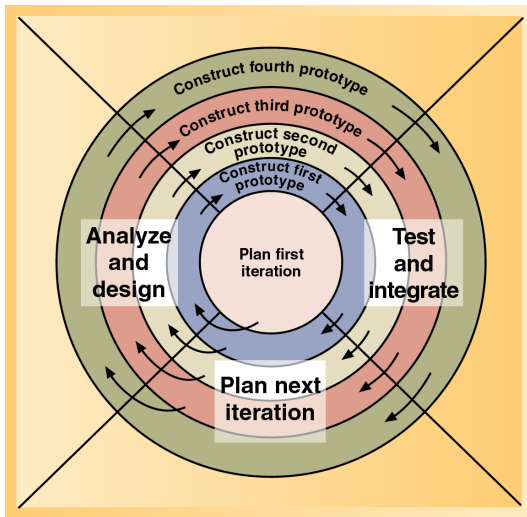
Figure: *Obtenido de [2]*

Problemas con Waterfall

- Los requisitos cambian constantemente.
- Difícil utilizar los recursos humanos eficientemente.
- Los errores cometidos en una etapa se amplifican en las siguientes etapas.
- Muy difícil corregir los errores, en especial cuando el sistema ya se terminó. Ej. Xorg

- El proceso de desarrollo se estructura mediante iteraciones.
- En cada iteración se repiten los mismos pasos.
- Cada iteración se retro-alimenta de las iteraciones anteriores.
- Las iteraciones se pueden ordenar según diferentes criterios:
 - Enfocarse en un sub-sistema a la vez.
 - En orden de riesgo.
 - Implementar funciones claves primero, luego complementos.
- Con buenas prácticas y un buen pipeline de desarrollo, los errores causados por cambios de requisitos se pueden mitigar.
- Existen varios modelos[1]:
 - Extreme programming.
 - Rational unified process (RUP)
 - Scrum

Metodologías espirales



Ventajas de las metodologías espirales

- Inclúyen el cambio como parte del proceso:
 - Aprender de iteraciones pasadas.
 - Minimizan el riesgo y costo de introducir cambios.
 - Minimizan la propagación de errores.
- Entregan valor temprano:
 - Satisfacer al cliente desde temprano.
 - Reduce el riesgo de fallar.
 - Se enfoca en la funcionalidad más importante en cada etapa.
- En algunos casos, el cliente es parte del equipo:
 - Le permite al cliente dar retro-alimentación desde temprano.
 - Mantiene al cliente informado.
 - Facilita el diseño del sistema.
- Mayor utilización de recursos humanos:
 - Cada miembro del equipo puede trabajar en su campo durante todas las iteraciones.
 - No hay necesidad de esperar a que completen fases anteriores para trabajar.
 - Todos los miembros tienen la misma importancia en todas las iteraciones.

- Organiza el desarrollo en sprints de 2-4 semanas de duración.
- Los requisitos son capturados como historias en un backlog.
- Cada sprint debe producir un producto entregable.
- Antes de cada sprint debe haber una session de planeación.
- Los errores se van agregando al backlog según van siendo reportados.
- Requiere mucha transparencia y comunicación del equipo, para poder corregir y agendar errores. \Rightarrow Encontrar errores es bueno.

Composición de Scrum

- Ocupaciones:
 - Lider de proyecto
 - Equipo
 - Scrum master
- Ceremonias:
 - Planeación del sprint
 - Revisión del sprint
 - Retrospectiva del sprint
 - Reunion diaria
- Artefactos:
 - Backlog del sistema
 - Backlog del sprint
 - Burndown chart

- A menudo referido como Product Manager (PM)
- Responsable de crear y priorizar la lista el backlog del sistema.
- Responsable de crear el sprint backlog en cada sprint.
- Interactua con los clientes y personas asociadas al sistema ⇒ Refinar prioridades y requerimientos
- Asiste al equipo proporcionandole los recursos que necesiten para el desarrollo.
- Encargado de procesos administrativos.

- Supervisa el proceso:
 - Se asegura que los artefactos se utilicen adecuadamente.
 - Se asegura que las ceremonias se lleven a cabo.
 - Se asegura que todos los miembros participen debidamente.
- Asesora al equipo:
 - Ayuda al equipo a resolver problemas difíciles
 - Ayuda a que exista comunicación efectiva en el equipo.
- El scrum master **no** debe ser el líder del proyecto.

- Generalmente pequeño (2-7 personas) y centralizado.
- Multifuncional:
 - Personas con diferentes destrezas.
 - Tiene todas las destrezas necesarias para desarrollar el producto.
- Auto-organizado:
 - Cada miembro decide cuando esfuerzo poner en cada sprint.
 - Las tareas se deben asignar según las habilidades, no solo por la ocupación oficial.
 - El equipo colabora para cumplir con sus tareas.

Planeación del sprint

- Se realiza antes de comenzar cada sprint.
- Se refina el backlog del sistema:
 - Agregar o quitar requerimientos.
 - Cambiar prioridad de requerimientos.
- Definir la meta del sprint.
- Seleccionar los requerimientos para avanzar a esa meta:
 - Refinar requerimientos \Rightarrow crear tareas a partir de los requerimientos.
 - Estimar los recursos necesarios para dicho requerimiento.
 - Crear el backlog para el sprint.
- Asignar tareas a los miembros del equipo.

Ejecución del sprint

- Cada miembro lleva a cabo su tarea asignada:
 - Diseño
 - Programación:
 - Pair programming
 - Revisión de código
 - Pruebas unitarias
 - Integración continua
 - Tareas administrativas:
 - Reuniones con los clientes.
 - Reportes.
 - Focus groups.
 - Varía según la empresa y el cliente
- Una *reunion diaria*
- El sprint debe concluir con un sistema **funcional**
- El equipo puede dejar tareas pendientes para el siguiente sprint.
- Los requisitos y ámbito no cambian durante el sprint.

- Objetivos:
 - Mantener al equipo informado sobre los avances en el sprint.
 - Exponer los problemas y obstaculos.
 - Coordinar al equipo.
- Caracteristicas:
 - Corta, menos de 15 minutos \Rightarrow parados
 - Diaria, a la misma hora
 - No es para resolver problemas.
 - Pueden agendarse reuniones posteriores para discutir problemas.
 - Ordenado: una persona presenta a la vez.
- Posible formato por miembro:
 - 1 ¿Que hize ayer?
 - 2 ¿Que hare hoy?
 - 3 ¿Que obstaculos tengo?



Aniruddha Ray.

Scrum and agile.

[https:](https://es.slideshare.net/ekmusibat/scrum-and-agile-sdlc-101)

[//es.slideshare.net/ekmusibat/scrum-and-agile-sdlc-101.](https://es.slideshare.net/ekmusibat/scrum-and-agile-sdlc-101)



Embeded Systems.

Waterfall model.

[https://embeddedsystemsvvce.wordpress.com/tag/waterfall-model/.](https://embeddedsystemsvvce.wordpress.com/tag/waterfall-model/)