



Meza Peña Augusto

No. Cuenta. 414107090

Metodologías de desarrollo de software (Resumen)

Metodologías Ágiles y las Tradicionales

Ingeniería de Software

Prof.: Ing. Orlando Zaldívar Zamorategui

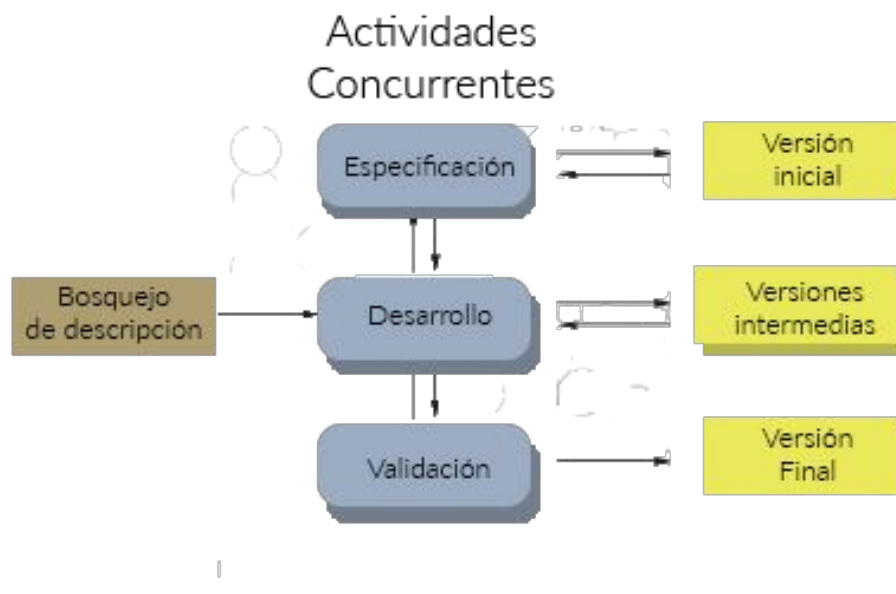
Métodos Ágiles	Métodos Tradicionales
<p>Basado del Manifiesto Ágil</p> <p>Tienen mayor valor para:</p> <ul style="list-style-type: none"> ● Individuos y pequeñas empresas ● Código funcional ● Colaboración entre clientes ● Responder al cambio 	<p>Tienen mayor valor para:</p> <ul style="list-style-type: none"> ● El proceso de desarrollo ● Productos de Software ● Acuerdo de clientes ● Control y planeación de proyectos
Centrada en los recursos humanos	Centrada en el proceso
Énfasis en el código y la cooperación	Énfasis en el producto y la organización
Pocos procesos con minima documentacion	Procesos bien definidos y documentados
Formalidad en el proceso reducida al mínimo	Altamente dependiente del proceso de desarrollo
Un cambio en los requerimientos no atrasa el proyecto	Un cambio en los requerimientos atrasa severamente el proyecto
Resultados dependientes del equipo de desarrollo, malo para sistemas críticos	Resultados predecibles, bueno para sistemas críticos
De medianos a pequeños proyectos	Grandes proyectos
Equipo reducido de desarrollo	Gran equipo de desarrollo

Método en cascada:



- Definición de requerimientos: Son las funcionalidades que tendrá el sistema, sus limitaciones y sus respectivos detalles.
- Diseño del sistema y del software: Identificar y planificar las abstracciones del sistema y la comunicación entre sus componentes.
- Implementación y prueba de unidad: El diseño se realiza subdividiendo en programas, llamados unidades. La prueba consiste en verificar que la unidad cumpla con los requerimientos.
- Integración y prueba del sistema: Ahora las unidades se prueban como un sistema completo para verificar que se cumplen todos los requerimientos. Una vez terminada se entrega el producto al cliente.
- Operación y mantenimiento: El sistema se instala y se echa a andar, se corrigen los errores que no se detectaron, y se optimiza el sistema.
- Uno de los inconvenientes en este método es que es esencial terminar la fase para poder continuar con el método.
- Una de los atributos del método en cascada es que es transparente en su proceso
- Este método se recomienda cuando se conocen a fondo los requerimientos del cliente o cuando se trate de sistemas muy grandes o críticos.

Método incremental basado en un plan:



- Este desarrollo parte de un desarrollo parcial (un bosquejo) y de ahí ir agregando las funciones, por supuesto recibiendo siempre la retroalimentación del cliente.
- Las primeras funciones en agregarse son las primordiales del sistema.
- Hay una relación directa entre las versiones y el desarrollo, especificación y validación.
- Entre sus ventajas se reduce la dificultad de modificar los requerimientos ya avanzado el proyecto.
- Tiene la desventaja de que para el coordinador del proyecto es difícil medir los tiempos. Cada que se agregan funciones el sistema se vuelve más y más pesado pues se desarrolla encima de otra capa en lugar de linealmente.

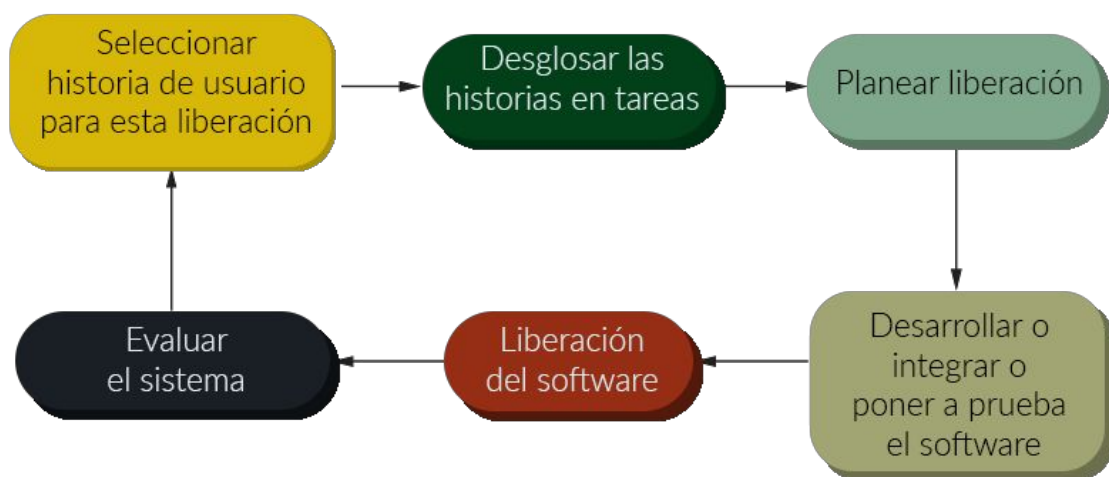
Método basado en la reutilización:



- Es común que nos encontremos con la reutilización de software en los sistemas computacionales actuales.

- **Análisis de componentes:** En cuanto se termina la fase de especificación se buscan los componentes que más se parezcan a lo que se necesita para el sistema
- **Modificación de los requerimientos:** Una vez que ya se tienen los componentes se actualizan los requerimientos para mostrar que se logró y si se necesita se regresa a la etapa anterior para evaluar las posibilidades.
- **Diseño de sistema con reutilización:** En esta etapa se diseña el sistema utilizando los componentes adquiridos y se diseñan los que se requieran para cumplir con los requerimientos.
- **Desarrollo e Integración:** Ya que se tienen todos los módulos diseñados se implementan y se unen para formar el sistema completo, ya utilizando los componentes reutilizados y los diseñados
- Las ventajas de este método son que reduce considerablemente el costo del sistema así como los riesgos y todo esto deriva en una entrega mucho más rápida.
- Sin embargo se pierde un poco el control del sistema y además puede ser que no se cumplan todos los requerimientos del cliente.

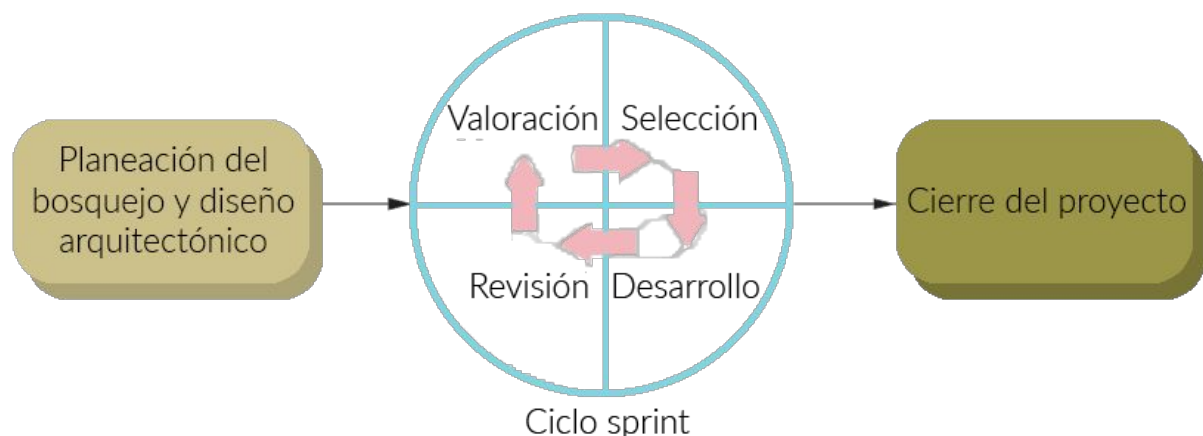
Programación extrema:



- En este método los requerimientos se les llama historias de usuario
- Si llegan a cambiar los requerimientos las historias no implementadas son desechadas y se crean nuevas.

- Hay ocasiones en que se requiere diseñar la arquitectura del sistema o la documentación, en ese caso se generan versiones que no producen código y que solo se enfocan en lo anterior, se les conoce con el nombre de picos.
- Al principio se desarrolla una versión mínima, ya sea con lo más urgente o lo más útil para el cliente.
- El equipo de desarrollo debe asegurarse que el código está optimizado
- Los desarrolladores trabajan en parejas (no siempre las mismas), así uno puede comprobar el trabajo del otro, y cuando se llegue a requerir la creación de una prueba esta sea imparcial.
- En cuanto los desarrolladores han terminado una tarea esta se integra a la nueva versión, siempre y cuando se cumplan todas las pruebas, estas versiones no deben consumir mucho tiempo, normalmente dos semanas.
- Para validar que la versión es satisfactoria debe estar in situ un cliente para corroborarlo y si no es satisfactoria se desecha y se pasa para la siguiente versión.
- Un contrapeso de la programación extrema es que poco a poco corrompe el sistema y lo vuelve más difícil de cambiar.
- En la programación extrema el código se prueba conforme a las pruebas diseñadas al principio después con cada historia se van agregando pruebas y el cliente a su vez agrega las pruebas que considere necesarias.

Método Scrum:



- Es por esto que los sistemas deben administrarse usando de la mejor manera el tiempo y los recursos humanos (el talento) disponibles.

- Planeación del bosquejo: Aquí se establecen los objetivos generales del proyecto y su diseño general.
- Ciclo Sprint: En cada ciclo se desarrolla una versión, estos son de longitud de tiempo fija. En la fase de valoración se establece el trabajo a realizar del proyecto, las prioridades y los riesgos. En la selección se incluye a todo el equipo para trabajar con el cliente, esto para seleccionar las mejores funcionalidades para el próximo sprint. Una vez que se tiene el trabajo el equipo se organiza para sacarlo adelante, se asignan prioridades y se tienen pequeñas reuniones diarias para mostrar los avances y estar al tanto del proyecto en general. En esta etapa solo el “maestro del Scrum” habla con el cliente y mantiene un canal de comunicación entre todas las partes involucradas en el proyecto. Al final el trabajo se revisa y se procede a liberar la versión.
- Por último tenemos el cierre del trabajo, se incluye la documentación requerida y los manuales de usuario.
- La idea central del Scrum es que no hay un administrador, el “maestro del Scrum” actúa más bien como un medio de comunicación, mide el avance del proyecto, y facilita la conversación entre los miembros del proyecto durante las reuniones diarias. Estas reuniones facilitan lo que se le conoce como responsabilidad colectiva. Es por esta colectividad que todos participan en la toma de decisiones, ya sea por una funcionalidad o porque haya que rediseñar el sistema.
- Ya que se desarrolla por tareas con un énfasis en código simple, el producto final es comprensible y manejable.
- Un cambio en los requerimientos no atrasa el proyecto
- El equipo de desarrollo conoce todo el sistema y puede ayudar en cualquier parte
- Los clientes obtienen retroalimentación inmediata y pueden supervisar los avances.
- La colectividad motiva a un sentimiento general de terminar el proyecto, lo que también genera confianza con el cliente.

- Su implementación en proyectos grandes y en sistemas críticos es complicado porque uno de los requerimientos es que el equipo conozca todo el sistema, algo complicado o incluso imposible en muchos de los sistemas grandes.