

Modelos de Procesos.

Procesos de software que son fundamentales.

1- Especificación del software. = Definir su funcionalidad y las restricciones de sus operaciones.

2- Diseño e implementación del software. = Debe desarrollarse para cumplir con las especificaciones.

3- Validación del software. Hay que validar el software para asegurarse de que cumple lo que el cliente quiere.

4- Evolución del software: el software tiene q' evolucionar para satisfacer las necesidades cambiantes del cliente.

Prototipos = Son los requerimientos que el usuario ha establecido.

Modelos de procesos que examinaremos:

Modelos tradicionales

- Formados por un conjunto de fases o actividades en el q' no tienen en cuenta la nat. evolutiva del software

- * Clásico, lineal o en cascada.

- * Estructurado.

- * Basado en prototipos.

- * Desarrollo rápido de aplicaciones (RAD).

Modelos evolutivos.

- Son modelos que se adaptan a la evolución que sufren los requisitos del sistema en función del tiempo.

- * En espiral.

- * Evolutivo.

- * Incremental.

Modelos utilizados a reutilización.

- Basado en componentes.

Procesos unificados.

Modelos para sistemas orientados a objetos.

- Modelos con un alto grado de iteratividad y solapamiento entre fases.

- * De agrupamiento, * Fuente.

- * Basado en componentes

- * Proceso Unificado.

Proceso: Ágiles.

- Programación Extrema XP.
- Desarrollo de software adaptativo.
- Scrum, Crystal.

Modelos para
Sistemas web

- UML -based webengineering.

Def. de cada modelo de procesos.

Modelo Tradicional o modelo cascada: Proceso dirigido por un plan; se debe planear y programar todas las actividades del proceso.

Def. de requerimiento

Diseño del sistema
y del software

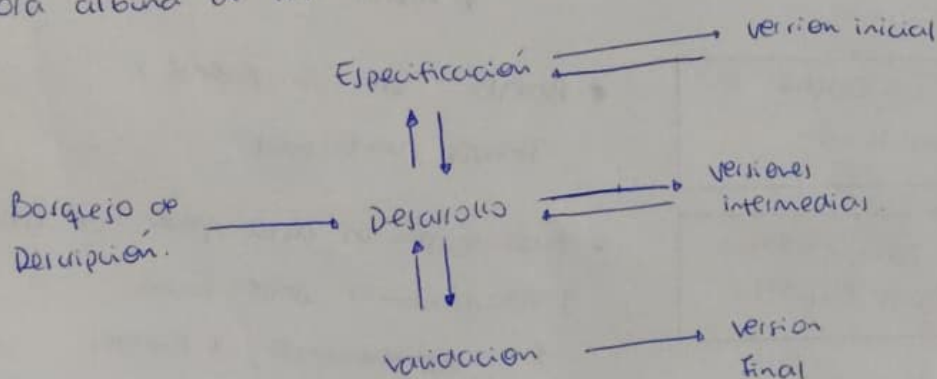
Implementación
y prueba de
unidad

Integración
y prueba
del sistema

Operación y
mantenimiento

Evolutivo (o incremental)

Se va realizando conforme a la aceptación del cliente. Refleja la forma en que se resuelven los problemas. Rara vez se trabaja por adelantado, se avanza en una serie de pasos hacia la solución y se retrocede cuando se detecta que se cometió un error. Es más fácil realizar cambios y es más barato. Cada incremento incorpora alguna de las funciones que necesita el cliente.



en comparación con el modelo cargado el mejor ya que se reduce el costo pq se va realizando de a poco y con los requerimientos cambiantes, la documentación el mejor q' el modelo cargado (ya el en era).

Interface: Medio por el cual el usuario interactúa con un dispositivo tecnológico. Esto abarca todos los puntos de contacto entre la persona y el equipo. (interfaz ya es en si todas las plantillas juntas o, cuando vamos realizando por ejemplo en Figma + q' es una herramienta de diseño de interfaz, y vos vos colocando las fechas, horas y vos modificando a tu antojo, vos creando plantillas q' te llevan a otra parte al pulsar un botón y esas cosas.

(el pa paper)

Prototipo: * es la representación de alto detalle sobre el proyecto, muestra las interacciones y flujos del sistema.

* Permite representar cómo va a funcionar y se va a comportar el sistema.

* es una representación o simulación de las interacciones q' va a tener nuestra solución del sistema.

Se va mostrando el prototipo del sistema q' vamos a estar desarrollando y así vamos a poder ir realizando la interacción con las interfaces y con el diseño q' se va realizando.

✓ interactivo como el sistema final.

✓ imita con la experiencia exacta del usuario con la aplicación real.

✓ permite el alcance de los interesados p/ realizar pruebas exhaustivas.

Borrador Estructura
Sketch. alambrica.

El prototipo va ir evolucionando, vamos a tener el borrador ¹, después el wireframe ², que es de una forma más detallada, maquetado ³, mockup ³ más interactiva del contenido, color, gráficos y el contenido real, y por último el prototipo q' es con el q' el usuario va interactuar como si fuera el sistema real.

- 1) Identificar los objetivos del proyecto y las necesidades del usuario
- 2) Especificar las funcionalidades y requerimientos del sistema
- 3) Diseñar el sistema. Organización, etiquetado, y búsqueda.
- 4) Realizar el prototipo del sistema.

Tipos de prototipos.

- 1) Sketching + Primer dibujo. es un boceto para tener la idea de como va a ser
- 2) Wireframes + Prototipo de baja fidelidad. (Diseño del contenido, generalmente se hacen en escala de color gris o blanco y negro)

- 3) Prototipo de media fidelidad (Mockup).

Se trata de la parte visual del proyecto. y debe contener amplitud de detalles como, colores, tipografías, contenido. Es el q' mejor representa al producto final. pero aun falta incluir las interacciones q' se realizarán en la interacción.

- 4) Maqueta + Prototipo de alta fidelidad. (Aquí validamos si la idea funciona o no, en este momento el usuario se perciben los problemas de usabilidad. mediante los tests de interactividad y los comentarios de los usuarios. vemos todos los detalles que necesitan una revisión. Por ej: campos de textos, botones mal ubicados, colores mal elegidos. Es la fase más determinante para validar la usabilidad de todo el producto.

Prácticas Ágiles. — Metodología Ágil ¿Cuál es la mejor?

Scrum = Se agrupan en tareas q' duran 30 días generalmente y que proporcionan un incremento completo del producto. (Es un enfoque ágil popular q' se utiliza para formular hipótesis, probarlas, reflexionar sobre los resultados y hacer mejoras). Trabaja en un desarrollo incremental paso a paso, se utiliza porque por multifuncionales q' están trabajando en el desarrollo. Se hacen con el cliente reuniones demostrativas para que valoren como han ido desarrollada los requisitos que proporciono. Además se pueden añadir cambios y nuevos tiempos de entrega según su objetivo de negocio. (Se divide en una iteración de dos a 4 semanas).

Metodo Programacion extrema XP

Trabaja con 10 a 15 iteraciones. Cada iteracion o ciclo se divide en etapas:

- 1 - Analisis y exploración. → Dura 2 semanas, en ella se definen el alcance del proyecto. El cliente redacta q' necesitan los usuarios
- 2 - Planificación. → El orden en q' se va a implementar las necesidades de los usuarios y coordinar tiempo de entrega
- 3 - Diseño.
- 4 - Implementación.
- 5 - Pruebas.

- Se destaca por el diseño sencillo, un lenguaje facil de entender y sin mucho tecnicismo en la comunicacion con el cliente.
- Se atacan los problemas para su analisis y se permite redefinir los programas buscando claridad y eficiencia.
- es una metodologia que permite a los equipos producir software de alta calidad, permite a los ingenieros de software emplear los enfoques de ingenieria mas actualizados, permite pruebas unitarias y funciones automatizadas

Metodo Kanban (justo a tiempo).

Se caracteriza por ser una metodologia visual. Se utiliza tarjetas sobre un muro de desarrollo incremental. Se subdivide el trabajo en partes, con tarjetas que evolucionan de una fase a otra en lineas horizontales imaginarias.

STORIES Peticion Tareas.	TO DO Hacer	In progress Haciendo	Testing Comprobas.	Done. hecho.

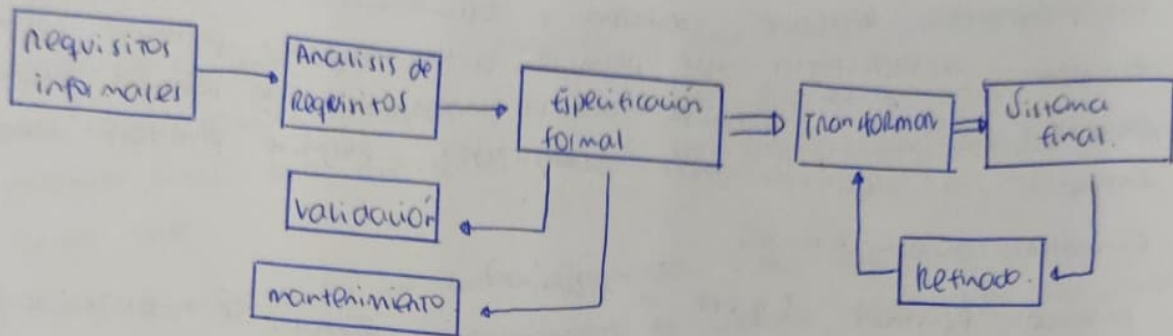
su rasgo distintivo es la flexibilidad con la que permite identificar el avance de los proyectos. (no se debe sacrificar calidad y eficiencia).

Crystal.

Coleccion de enfoques de desarrollo de software agil. se centra en las personas y su interaccion entre ellas mientras trabajan en un proyecto de desarrollo de software. Al principio del proyecto, los procesos y las herramientas no son fijas, sino q' deciden teniendo en cuenta los requisitos comerciales y las necesidades tecnicas del proyecto.

Modelos Genéricos - Tradicionales

- ✓ Modelo cascada = lejana en distintas fases de especificación y desarrollo.
- ✓ Desarrollo evolutivo = la especificación y el desarrollo están separados.
- ✓ Prototipado = un modelo sirve de prototipo p/ la construcción del sistema final.
- ✓ Transformación formal = un modelo matemático del sistema se transforma formalmente en la implementación.
 - Análisis de requisitos.
 - Especificación formal.
 - Transformación.
 - Integración y pruebas del sistema final.



- ✓ Desarrollo basado en reutilización = El sistema es ensamblado a partir de componentes existentes.

Modelo cascada - Fases.

- * Análisis de requerimiento y definición.
- * Diseño del sistema y del software.
- * Implementación y prueba de módulos.
- * Integración y prueba del sistema.
- * Operación y mantenimiento.

La dificultad en este modelo reside, en la dificultad de hacer cambios entre etapas.

Documentos del modelo cascada. (pdf modelos-tradicionales pag 6 de 24).

Desarrollo Evolutivo. (Problemas y Aplicaciones).

Problemas.

- ✓ Poca visibilidad en el proceso.
- ✓ Los sistemas están pobremente especificados.
- ✓ Se requieren habilidades especiales.

Aplicabilidad

- ✓ Para sistemas interactivos pequeños o medianos.
- ✓ Para módulos de sistemas grandes. (p.ej. la interfaz de usuario)
- ✓ Para sistemas de corta vida.

Prototipado.

* Prototipado evolutivo: El objetivo es trabajar con clientes hasta evaluar a un sistema final, a partir de una especificación inicial, se debe comenzar con una especificaciones bien entendidos.

* Prototipado desechable: el objetivo es entender los requerimientos del sistema. Se puede comenzar con especificaciones poco entendidas.

Problemas y Riesgos con los modelos.

□ Cascada.

✓ Alto riesgo en sistemas nuevos debido a problemas en las especificaciones y en el diseño.

✓ Bajo riesgo para desarrollos bien comprendidos utilizando tecnología conocida.

□ Prototipado.

✓ Bajo riesgo para nuevos (~~especificaciones~~) aplicación debido a que las especificaciones se llevan a cabo paso a paso.

✓ Alto riesgo debido a falta de visibilidad.

□ Evolutivo.

✓ Alto riesgo debido a la necesidad de tecnología avanzada y habilidades del grupo desarrollador.

Modelo de procesos Híbridos.

* Los sistemas grandes están compuestos usualmente de varios subsistemas.

* No es necesario utilizar el mismo modelo de procesos p/ todos los subsistemas.

* El prototipado es recomendable cuando existen especi. de alto riesgo.

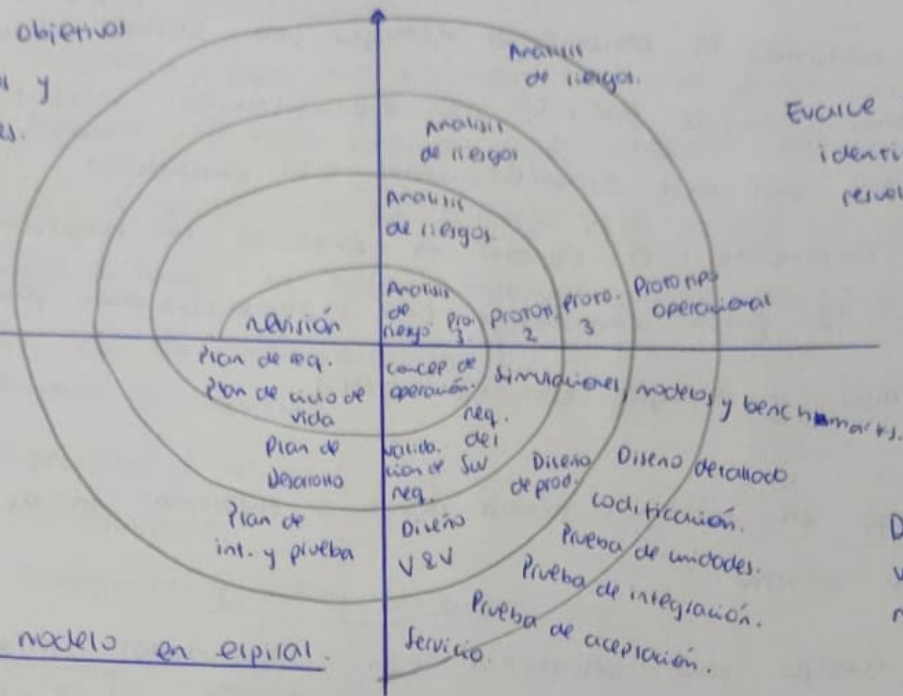
* El modelo cascada es utilizado en desarrollos bien comprendidos.

Modelo de Procesos en espiral.

Definir objetivos
alternativos y
restricciones.

Planear la
1ra fase.

Fases del modelo en espiral.



Evalúe alternativa,
identifique y
resuelva riesgos

Desarrolla y
verifica el 1er
nivel del producto.

* Planteamiento de objetivos.

✓ Se identifican los objetivos específicos por cada fase del proyecto.

* Identificación y reducción de riesgos.

✓ Los riesgos clave se identifican y analizan, y la información sirve para minimizar los riesgos.

* Desarrollo y validación.

✓ Se elige un modelo apropiado para la 1ra fase del desarrollo.

* Planificación.

✓ Se revisa el proyecto y se hacen planes para la siguiente ronda del espiral.

Ventajas del modelo espiral.

* Centra su atención en la reutilización de componentes, y eliminación de errores en fase inicial.

* Los objetivos de calidad son el primer objetivo.

* Integra desarrollo con mantenimiento.

* Provee un marco de desarrollo de hardware/software.

Problemas del modelo espiral.

* El desarrollo contractual especifica el modelo del proceso y los resultados a entregar por adelantado.

* Requiere de experiencia en la identificación de riesgos.

* Requiere refinamiento para uso generalizado.

Que modelo utilizar?

- * Para sistemas bien comprendidos se utiliza el modelo de cascada. la fase de analisis de riesgos es relativamente facil.
- * Con requerimientos estables y sistemas de seguridad criticos, se utiliza modelos formales.
- * Con especificaciones incompletas, se utiliza el modelo de prototipado

Visibilidad de procesos

Modelos de procesos	Visibilidad de procesos.
Modelos de cascada	Buena visibilidad, cada actividad produce un documento o resultado.
Desarrollo evolutivo	Visibilidad pobre, muy caro al producir documentos en cada iteración.
Modelos formales.	Buena visibilidad, en cada fase deben producirse documentos.
Desarrollo orientado a la reutilización.	Visibilidad moderada, importante contar con documentación de componentes reutilizables.
Modelos de espiral.	Buena visibilidad, cada segmento y cada anillo de espiral debe producir un documento.

Modulo 3 - Metodologías ágiles.

Hace referencia a metodos de la Ing. de software q' se basa en el desarrollo

iterativo e incremental, donde las condiciones y soluciones van evolucionando mediante la colaboración de grupos auto organizados y multidisciplinarios. (Son técnicas q' permiten incorporar cambios con rapidez en el desarrollo de software).

En conclusión las metodologías ágiles son una serie de técnicas empleadas para la gestión de proyectos y estas responden al cumplimiento del MANIFIESTO ÁGIL.

- 1) Individuos e iteraciones por encima de procesos y herramientas:
Las personas son el principal factor de éxito, el más importante construir un buen equipo a un buen entorno.
- 2) Software funcionando contra documentación extensiva.
Solo producir documentos cuando sea necesario, estos deben ser cortos y centrados en lo fundamental.
- 3) Colaboración con el cliente por encima del contrato de regulación:
Interactuar constantemente con el cliente (cuando sea necesario) y el equipo de desarrollo.
- 4) Respuesta ante el cambio sobre seguir un plan determinado. La planificación debe ser flexible y abierta, esto facilita la habilidad de responder ante posibles cambios.

Principios del manifiesto ágil

- 1) tiene como objetivo principal satisfacción del cliente al garantizar la entrega oportuna y temprana del software de valor.
- 2) están creados para aceptar positivamente los req. cambiantes (buenos y malos) se encuentran en una etapa tardía del desarrollo del proyecto.
De esta manera los procesos ágiles le dan una ventaja competitiva a la empresa q' los utiliza.
- 3) Se hacen entregas frecuentes de software perfectamente funcionales, en periodos que van desde un par de semanas hasta un par de meses, dep. de la duración del proyecto.
- 4) Las personas encargadas de la empresa y los desarrolladores de la metodología ágil, deben trabajar en conjunto y mantener una comunicación cotidiana en todo el tiempo del desarrollo del proyecto.
- 5) Se construyen los proyectos destinados a los individuos motivados.
- 6) Toda la información que ingresa y sale debe ser comunicada entre los desarrolladores de las metodologías ágiles y las personas de la empresa cara a cara, esto para garantizar la eficiencia y eficacia en el proceso.

7) le va determinar q' todo el proceso este funcionando cuando el software, producto o servicio empiece a funcionar.

8) los procesos q' intervienen en la metodologías ágiles ayudan a promover el desarrollo sostenido. De esta manera, tanto los desarrolladores, patrocinadores como los usuarios deberán de mantener un ritmo constante de manera indefinida.

9) Para garantizar una excelente aplicación, los desarrolladores deberán atenderse mutuamente para responder inquietudes y aclarar preguntas.

10) Menos es más, siempre mantener la simplicidad para maximizar la cant. de trabajo q' se elabora.

11) Las mejores arquitecturas, requisitos y diseños se originan de equipos q' se auto organizan apropiadamente.

¿Por qué elegir metodología Ágil?

comentarios.

Los equipos q' eligen la metodología ágil pueden responder ante los cambios en el mercado o ante el feedback de los clientes rápidamente sin arrastrar la planificación de un año entero. Con la planificación justa y necesaria y el lanzamiento de incrementos pequeños de forma frecuente permite q' el equipo recopile ~~información~~ comentarios sobre cada cambio y los integre en los planes futuros sin unos costos mínimos.

Aquí los protagonistas no son los ~~no~~, sino las personas, ya q' las verdaderas interacciones humanas son más importantes q' los procesos rígidos. Colaborar con clientes y compañeros es mucho más importante q' las disposiciones predefinidas, así como entregar una solución q' resuelva problemas del cliente es más importante q' una documentación minuciosamente detallada.

Ventajas de la metodología Ágil

- * Respuesta rápida a cambios de req. durante el proyecto.
- * minimiza costos.
- * Mayor velocidad y eficiencia.
- * Se identifican errores rápidamente ya q' se van haciendo pruebas a medida q' se avanza.
- * el equipo de desarrollo conoce el estado del proyecto.
- * mejora la calidad del producto.
- * simplifica la sobrecarga de pruebas.

Características de la M. Ágil.

- * Interacción al cliente mediante entrega de productos tempranos, funcionales y continuos.
- * los cambios en los req son permitidos.
- * Equipos auto organizados.
- * entregas frecuentes en el menor tiempo posible.
- * el equipo de desarrollo y el cliente deben trabajar juntos durante todo el proyecto
- * el método más eficiente y efectivo es el dialogo cara a cara dentro del equipo.
- * simplicidad
- * Aumenta la productividad.
- * Difusión y transferencia del conocimiento.

Desarrollo Ágil del software.

Enfoca su enfoque en la toma de decisiones. Se refiere a met. de I. sot. basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

Cada iteración del ciclo de vida incluye planificación, análisis de requisitos, diseño, implementación, pruebas y documentación. Adquiere una gran importancia el concepto de "finalizado" (Done) ya q' el obje. de cada iteración no es agotar toda la funcionalidad p' justificar el lanzamiento del producto. Sino incrementar el valor por medio de "software q' funciona".

los métodos ágiles enfatizan la comunicación cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta. llamados "plataformas de lanzamiento".

La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y directores de proyecto. Ja q' es preferible las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.

Modulo 3. Metodologías Ágiles más utilizadas.

* Scrum, XP, Kanban.

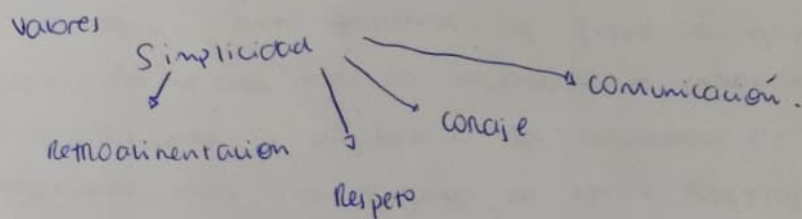
el manifiesto ágil compuesto por 12 principios agrupados en 4 valores fundamentales

- 1- Individuos e interacciones sobre procesos y herramientas.
- 2- Software funcionando sobre documentación extensiva.
- 3- Colaboración con el cliente sobre negociación contractual.
- 4- Respuesta ante el cambio sobre seguir un plan.

1) (XP).

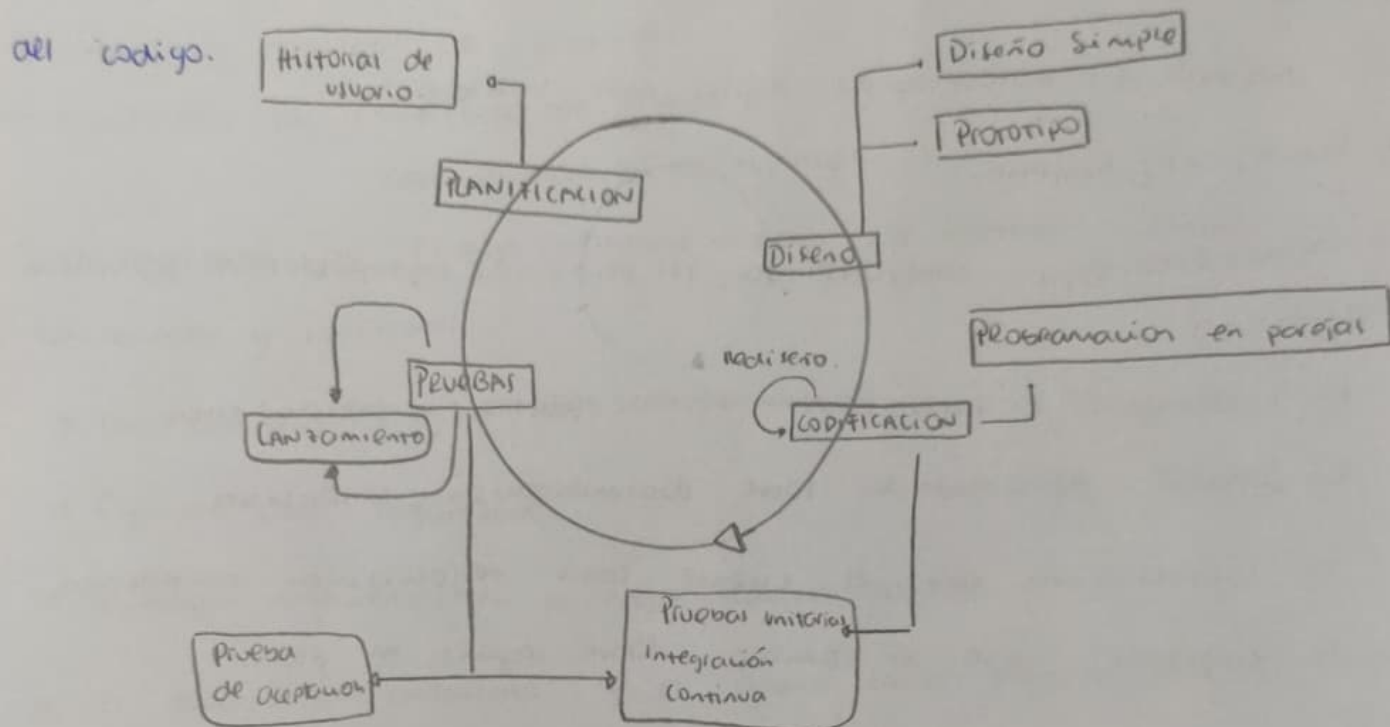
Conjunto de reglas y buenas prácticas p/ el desarrollo de software en ambientes muy cambiantes con requisitos imprevisibles, esta enfocada en la retroalimentación continua entre el equipo de desarrollo y el cliente.

Por ello, iniciando el proyecto se debe definir todos los requisitos - p/ luego invertir el esfuerzo en manejar los cambios que se presentan y así minimizar las posibilidades de error. XP tiene como base la simplicidad y como objetivo la satisfacción del cliente



Características de x.p.

- Desarrollo iterativo.
- Programación en parejas.
- Pruebas unitarias continuas.
- Corrección periódica de errores.
- Integración del equipo de programación con el cliente.
- Simplicidad, propiedad de código compartida y refactorización del código.



La programación extrema optimiza los tiempos y se adapta al desarrollo de sistemas grandes y pequeños sin mayor documentación, el código es claro y simple, así mismo completa los conocimientos entre los miembros del equipo, gracias a la programación en parejas.

Sin embargo una desventaja de esta metodología ágil, es que luego de entregar el sistema puede ir cambiando según sean las peticiones del cliente.

2) Scrum.

Esta metodología es un marco de trabajo de procesos ágiles q' trabaja con el ciclo de vida iterativo e incremental, donde se va liberando el producto por partes de forma periódica, aplicando las buenas prácticas de trabajo colaborativo en equipo, Así facilitando el hallazgo de soluciones óptimas a los problemas q' pueden ir surgiendo en el proceso de desarrollo del proyecto.

Con Scrum se realizan entregas regulares y parciales del producto final. Esta especialmente indicado para proyectos en entornos complejos donde se necesitan obtener resultados de manera inmediata y donde son fundamentales los tres aspectos: la innovación, la productividad, la flexibilidad y la competitividad.

Quié conforma el equipo scrum?: se cuenta con roles específicos y c/u de ellos es imprescindible p' q' se lleve a cabo el proceso de forma satisfactoria.

El interesado • Stakeholder: Es el cliente, su responsabilidad radica en definir los requerimientos, recibir el producto al final de cada iteración y proporcionar el feedback (comentario) correspondiente.

Dueño del producto • Product Owner: es el intermediario de la comunicación entre el cliente (Stakeholder) y el equipo de desarrollo. Este debe priorizar los requerimientos según sean las necesidades de la sociedad.

Maestro Scrum • Scrum Master: Actúa como facilitador ante todo el equipo de desarrollo, elimina todo aquellos impedimentos q' identifique durante el proceso, Así mismo se encarga de q' el equipo siga todos los valores y los principios ágiles, las reglas y los procesos de Scrum, incentivando al grupo de trabajo.

• Scrum Team (Equipo de desarrollo): se encarga de desarrollar los casos de uso definidos en el product Backlog (Pila de productos). Es un equipo auto-gestionado. lo quiere decir q' no existe un jefe de equipo. Por lo cual todos los miembros se deben encargar de realizar las estimaciones y en base a la velocidad obtenida en las iteraciones irán construyendo el Sprint Backlog.

las reuniones un pilar importante

un punto fundamental en el proceso que contiene scrum son las reuniones (reuniones) con ella se fomenta la comunicación y transparencia del proceso. las reuniones q' aplican son:

1- Reunion de planificación: Se debe realizar al inicio de cada sprint, esto con el objetivo de planificar la cantidad de trabajo a la q' el equipo se va a comprometer a construir durante el proximo sprint.

2- Reunion diaria: Son reuniones cuyo lapso dura un maximo 15 min, en ella se realiza una retroalimentación de q' se hizo el dia ayer, q' se hara hoy y cuales han sido los problemas q' han surgido hasta el momento. El objetivo es q' el equipo establezca un plan para las proximas 24 horas.

3- Reunion de revisión: Se lleva a cabo al final de cada Sprint, se exponen los puntos completados y los q' no.

4- Reunion de retrospectiva: Una vez culminado el sprint se efectua esta reunión. q' tiene como objetivo q' los integrantes del equipo reflexionen y saquen resultados de posibles acciones de mejora. A estas reuniones deben asistir todo el equipo scrum. (Dueño del producto, Equipo desarrollo y Scrum master) es una de las reuniones más importantes ya q' es un espacio de reflexión y mejora continua.

CARACTERÍSTICAS ESPECÍFICAS.

- Este desarrollo agil se basa en dos pilares fundamentales: el primero es el ciclo de vida incremental, con fiste en aprender del proceso anterior para mejorar o incrementar el valor de las versiones de los productos, y finalmente entregar uno de alta calidad.
- el segundo pilar. Son las revisiones del equipo de desarrolladores y miembros de la empresa.
- Reunirse a conversar y revisar los avances conseguidos es muy importante p/ scrum, pq consideran q' esto va a garantizar una comunicación transparente entre ambas partes.
- Se estima q' se debe hacer una reunión diaria con almenos 15 min

• Luego se procede a revisar cada sprint una vez que se ha finalizado con el desarrollo el aquí q' se discute qué se logró completar y que no. Estas reuniones son muy importantes ya que cada miembro puede dar su opinión de cada sprint y así se contribuya a la mejora inmediata de el proceso p/ la próxima vez.

3) Kanban.

- Comunicación en tiempo real.
- Utilizado para controlar el avance del trabajo en una línea de producción q' clarifican tareas en sub-etapas, esto con la intención de determinar los niveles de productividad en cada fase.

TODO	Doing	DONE.
Hacer	haciendo	Hemos.

Se asignan responsabilidades en un tablero q' representan los procesos del flujo de trabajo, por lo menos deben existir 3 columnas. (pendiente, en proceso, terminado).

Ventajas.

- Planificación de tareas
- Tiempos de ciclo reducidos.
- Rendimiento del equipo de trabajo.
- Métricas visuales
- Menos cuellos de botella.
- Entrega continua.

Características específicas:

- Regula el flujo de avance continuo, es por eso q' se utilizamos en proyectos TIC
- Presenta toda la información relativa a la producción en forma muy visual donde se pueden identificar fácilmente los componentes y los estados del proyecto.

- Se basa en el WIP (work in progress o trabajo en curso).
- Utiliza un cuadro de 3 columnas de tareas a realizar.
(Pendientes, en proceso, y terminadas).
- Toda los miembros del equipo tienen acceso al diagrama.
- Promueve la productividad y eficiencia del equipo de trabajo.

