

Ingeniería de software 1

INTRODUCCIÓN

SOFTWARE

son instrucciones, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación. Se caracteriza por ser un elemento lógico, que se desarrolla y no se desgasta

los productos de software pueden ser. Cada vez son más construidos como genéricos

- genéricos: sistemas aislados producidos por organizaciones desarrolladoras de software que lo venden en un mercado abierto
- personalizados: requeridos por un cliente en particular

se clasifica como

- de sistemas
- de aplicación
- científico y de ingeniería
- empujado
- de línea de productos
- aplicaciones web
- de IA

la ingeniería de software nació para hacerle frente a los retos ocurrientes a la hora del desarrollo de un sistema

INGENIERÍA DE SOFTWARE

es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema incluyendo la evolución de éste, luego que se comienza a ejecutar

disciplina de la ingeniería: hace que las cosas funcionen, usando teorías, métodos y herramientas

todos los aspectos de la producción: no sólo comprende los procesos técnicos del desarrollo de software si no también otras actividades como gestión de proyectos y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software

según IEE la IS es el uso de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software

- la cuantificación rigurosa de recursos, procesos y productos es una precondition para optimizar la productividades y calidad
- un ingeniero de sw debe cumplir contratos en tiempos y costos. Además de conocer tecnologías, productos y técnicas de administración de proyectos. La IS se desarrolla en un marco económico, social y legal, por lo tanto un ingeniero no debe usar sus capacidades y habilidades de forma deshonestas
- la IS se ocupa de todo el ciclo de vida de un producto

en el desarrollo de software participa el usuario, el desarrollador y el cliente

TÉCNICAS DE COMUNICACIÓN

es necesario comunicarse para saber lo que un usuario quiere, cómo, cuándo y por qué. Al hablar de necesidades, hablamos de *requerimientos*

requerimiento: característica del sistema que el sistema es capaz de hacer con el objetivo de satisfacer el propósito del sistema las fuentes de requerimientos más conocidas son: *documentación*, *stakeholders* y *especificación de sistemas similares*

con stakeholders se refiere a cualquier persona o grupo que se verá afectado por el sistema, directa o indirectamente. Pueden ser:

- interactuadores: personas o sistemas que interactúan directamente con el sistema
- indirecto: no usan el sistema pero que influyen en los requerimientos de algún modo
- dominio: características y restricciones del dominio que influyen en los requerimientos del sistema

ELICITACIÓN DE REQUERIMIENTOS

proceso de adquirir todo el conocimiento importante para producir un modelo de los requerimientos de un dominio de problema. Su objetivo es conocer el dominio del problema, conocer el sistema actual e identificar las necesidades de clientes y usuarios. Es una actividad social más que tecnológica

en el proceso de elicitación surgen los problemas de comunicación como dificultad de expresar las necesidades, cultura o vocabulario diferente, medios de comunicación inadecuados, intereses distintos en el sistema a desarrollar, limitaciones cognitivas del desarrollador, conflictos entre los participantes, fuentes de información poco claras, complejidad de los requisitos o del dominio del sistema

dentro de las técnicas de elicitación tengo

MÉTODOS DISCRETOS

son métodos menos perturbadores pero suelen recopilar insuficiente cantidad de información por lo tanto se deben usar junto con uno o más métodos

Muestreo de la documentación, formularios y datos existentes: recolección de hechos a partir de la documentación existente. Los documentos pueden ser organigramas, notas, registros contables, etc

Investigación y visitas al lugar: investigar el dominio, patrones de soluciones, revistas especializadas, buscar problemas similares en internet, consultar otras organizaciones

Observación del ambiente de trabajo: el analista se convierte en observador de las personas con el objetivo de aprender acerca del sistema. Tiene ciertas reglas como determinar quién y cuándo será observado, mantener perfil bajo, no interrumpir a la persona en su trabajo. La ventaja es que tendré datos confiables, se puede ver lo que se hace exactamente, hay un análisis de disposiciones físicas y es económica en comparación con otras técnicas pero la gente se siente incómoda siendo observada, las tareas están sujetas a interrupciones y quizá la persona observada está haciendo la tarea de forma correcta y no como lo hace habitualmente

MÉTODOS INTERACTIVOS

métodos que tienen como base hablar y escuchar a las personas de la organización para comprender las necesidades

Cuestionarios: documento que permite al analista recabar información de los encuestados, sirve para recolectar hechos de un gran número de personas, detectar un sentimiento generalizado, detectar problemas entre usuarios y cuantificar respuestas.

La ventaja es que es rápida, económica, anónima y estructurada pero obtengo un bajo número de respuestas, no responde a todas las preguntas, no da un análisis corporal y no se pueden dar aclaraciones de las respuestas incompletas

En los cuestionarios las preguntas pueden ser *abiertas* o *cerradas*. Las abiertas son las que dejan abiertas todas las posibles opciones de respuesta, mientras que en las cerradas se limitan las opciones de respuestas disponibles. la información obtenida será acerca de actitud, creencias, comportamiento y características. Los cuestionarios deben usarse cuando las personas están dispersas geográficamente, hay muchos involucrados, queremos opiniones generales o identificar problemas generales

Entrevistas: técnica mediante la cual el analista recolecta información de las personas mediante la interacción cara a cara. Resulta ser una conversación con un propósito específico que usa un formato de preguntas y respuestas en general la cual permitirá conocer opiniones y sentimientos.

Su ventaja es que el entrevistado se siente incluido en el proyecto, puedo obtener una retroalimentación, adaptar las preguntas y obtener información no verbal a partir de acciones y expresiones. La desventaja es que es costosa, requiere tiempo, recursos humanos y la entrevista depende en gran parte de las habilidades del entrevistador, no es aplicable a la distancia

La entrevista puede ser estructurada (cerrada) o no estructurada (abierta). En la estructurada el entrevistador tiene un conjunto específico de preguntas para hacérselas al entrevistado pero en la no estructurada el encuestador lleva a un tema en general sin preparación de preguntas específicas

Las preguntas pueden ser abiertas, cerradas o de sondeo. En las abiertas permite al encuestado responder de cualquier manera, en las cerradas las respuestas son directas o cortas y el sondeo permite obtener más detalle de un tema puntual

La ventaja de las preguntas abiertas es que revelan una nueva línea de preguntas y hará más interesante la entrevista pero puede dar muchos detalles irrelevantes y podría perderse el control de la entrevista

las preguntas cerradas ahorran tiempo y se puede mantener el control pero pueden ser aburridas y no obtendría detalles

Las entrevistas pueden organizarse de forma piramidal, embudo o diamante

Según Kendall, las entrevistas deben tener una preparación previa donde hay que leer los antecedentes, establecer los objetivos de la entrevista, seleccionar los entrevistados, planificar la entrevista y seleccionar el tipo de preguntas y su estructura

Whitten narra cosas que se deberían hacer y que no en una entrevista. Hay que vestirse adecuadamente, ser cortés, escuchar cuidadosamente, observar gestos y terminar a tiempo. Se deberá evitar revelar pistas, usar jerga, hablar en vez de escuchar o usar grabadores. Luego de la entrevista deberá hacerse un seguimiento en el cual se envía al entrevistado un resumen, permitiendo aclarar cualquier cosa que no se haya entendido

Planeación conjunta de requerimientos (JRP): proceso mediante el cual se conducen reuniones para analizar problemas y definir requerimientos. Esta técnica requiere extenso entrenamiento, reduce el tiempo, amplía la participación de los integrantes y se trabaja sobre lo que se va generando.

Permite ahorrar tiempo, involucra a los usuarios y surge un desarrollo creativo pero es difícil organizar los horarios y es complicado encontrar un grupo integrado y organizado

Brainstorming: técnica para generar ideas al alentar a los participantes a que ofrezcan ideas en un corto tiempo. El principio es que cuantas más ideas se sugieren, mejores resultados se conseguirán y las ideas de una persona pueden hacer que aparezcan otras por contagio

MODELO DE PROCESO DE SOFTWARE

es una representación simplificada de un proceso de software que representa una visión. Algunos de los modelos son

MODELO EN CASCADA: las etapas se representan cayendo en cascada. Cada etapa se debe completar para que comience la siguiente. Algunas de sus dificultades es que no hay resultados concretos hasta que esté todo terminado, las fallas más triviales se encuentran al comienzo del período de prueba, la eliminación de fallas es difícil

DESARROLLO ITERATIVO: un sistema inicial se desarrolla rápidamente a partir de una especificación abstracta y se va refinando según peticiones del cliente

DESARROLLO BASADO EN COMPONENTES: supone que las partes ya existen y el proceso se enfoca en la integración de las partes

MODELO EN V: demuestra cómo se relacionan las actividades de prueba, análisis y diseño

MODELO DE PROTOTIPOS: un prototipo es un producto parcialmente desarrollado que permite que clientes y desarrolladores examinen algunos aspectos del sistema propuesto, y decidan si éste es adecuado o correcto para el producto terminado. Si es evolutivo el objetivo será obtener el sistema a entregar, en cambio si es un prototipo descartable no tendrá funcionalidad y se usará como herramienta de modelado

MODELO DE DESARROLLO POR FASES: se desarrolla el sistema de tal manera que puede ser entregado en piezas. Esto implica que existen dos sistemas funcionando en paralelo: el sistema operacional y el sistema en desarrollo. Puede ser incremental (el sistema es particionado en subsistemas de acuerdo con su funcionalidad) o iterativo (entrega un sistema completo desde el principio y luego aumenta la funcionalidad de cada subsistema con las nuevas versiones)

MODELO EN ESPIRAL: combina actividades de desarrollo con la gestión del riesgo, mejora los ciclos de vida y elimina errores y alternativas no atractivas al comienzo

Esta es una alternativa de especificación para tratar mejor la incertidumbre, la ambigüedad y la volubilidad de los proyectos reales.

el proceso de software es un conjunto de actividades y resultados asociados que producen un producto de software

los modelos de proceso se caracterizan por

- establecer todas las actividades
- usar recursos, sujeto a restricciones
- se puede componer por subprocesos

- cada actividades tiene entradas y salidas
- las tareas se organizan en una secuencia

el proceso implica la construcción de producto el cual tendrá un ciclo de vida el cual comienza en su concepción y sigue hasta la etapa de mantenimiento

los modelos pueden ser

PRESCRIPTIVOS: prescriben un conjunto de elementos del proceso, como actividades del marco de trabajo, acciones de la IS, tareas, calidad y mecanismos de control

DESCRIPTIVOS: descripción en la forma en que se realizan en la realidad

ambos modelos deben ser igual

los modelos tradicionales se conforman por fases o actividades. Estos modelos son los cascada, modelo en V o basado en prototipos. Los modelos evolutivos son modelos que se adaptan a la evolución que sufren los requisitos en función del tiempo, algunos son en espiral, evolutivo o incremental

REQUERIMIENTOS

dentro de los tipos de requerimientos tenemos

FUNCIONALES: describen la interacción entre el sistema y su ambiente, describe lo que el sistema debe hacer mostrando detalladamente la funcionalidad del mismo, son independientes y se puede expresar en distintas formas

NO FUNCIONALES: describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema

la ingeniería de requerimientos es el proceso en el cual se transforman los requerimientos declarados por los clientes a especificaciones precisas, no ambiguas, consistentes y completas. También es el proceso en el cual se intercambian puntos de vista para modelar el sistema.

Permite gestionar necesidades del proyecto estructuradamente, mejora la capacidad de predecir cronogramas de proyectos, disminuye costos, mejora la calidad del software y evita rechazos de usuarios finales

ESTUDIO DE VIABILIDAD

a partir de una descripción resumida del sistema se elabora un informe que recomienda la conveniencia o no de realizar el proceso de desarrollo

ESPECIFICACIÓN DE REQUERIMIENTOS

PROPIEDADES DE LOS REQUERIMIENTOS

necesario: su omisión provoca una deficiencia

conciso: fácil de leer y entender

completo: no necesita ampliarse

consistente: no contradictorio con otro

no ambiguo: tiene una sola implementación

verificable: puede testearse

el objetivo es permitir que los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema, indica a los diseñadores qué funcionalidad y características va a tener el sistema e indica al equipo de prueba qué demostraciones habrá que hacer

VALIDACIÓN DE REQUERIMIENTOS

proceso de certificar la corrección del modelo de requerimientos contra las intenciones del usuario.

Trata de mostrar que los requerimientos definidos son los que estipula el sistema. Se describe el ambiente en el que debe operar el sistema. Es importante porque los errores en los requerimientos pueden conducir a grandes costos si se descubren más tarde las técnicas de validación pueden ser

informales: los desarrolladores deben tratar los requerimientos con tantos stakeholders como sea posible

formal: el equipo de desarrollo debe conducir al cliente, explicándole las implicaciones de cada requerimiento

construir prototipos

generar casos de pruebas

TÉCNICAS DE ESPECIFICACIÓN DE REQUERIMIENTOS

ESTÁTICAS: describe el sistema a través de las entidades u objetos, sus atributos y sus relaciones con otros. No describe como las relaciones cambian con el tiempo

DINÁMICAS: considera un sistema en función de los cambios que ocurren a lo largo del tiempo.

Se considera que el sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado. Como TD, DTE, RP, etc

HISTORIAS DE USUARIO

es una representación de un requerimiento de SW escrito con una o dos frases usando el lenguaje común del usuario una HU debe poder escribirse sobre una nota pequeña y resulta ser una forma rápida de administrar los requisitos de los usuarios, permiten responder rápidamente a los requisitos cambiantes y discutirlos con clientes.

Generalmente se espera que la estimación de tiempo sea de entre 10 horas y un par de semanas, las estimaciones mayores a dos semanas indican que la historia es muy compleja y debe dividirse en varias historias

Las HU se caracterizan por ser estimables, pequeñas y verificables.

Un criterio de aceptación es el criterio por el cual se define si una historia fue desarrollada de forma correcta y se si puede realizarse y expresan el resultado de intercambios con el cliente.

Su beneficio es que al ser corta, puede implementarse rápidamente, necesitan poco mantenimiento, permite dividir los proyectos en pequeñas entregas y se puede estimar fácilmente el esfuerzo de desarrollo pero sin criterios de aceptación pueden quedar abiertas, requiere un contacto permanente con el cliente, no se puede implementar en proyectos grandes y requiere de desarrolladores competentes

CASOS DE USO

proceso de modelado de las “funcionalidades” del sistema en término de los eventos que interactúan entre los usuarios y el sistema.

Captura requerimientos funcionales, usa lenguaje común y fácil de entender, permite estimar el alcance del proyecto y el esfuerzo a realizar

Está compuesto por

DIAGRAMA DE CASOS DE USO: ilustra las interacciones entre el sistema y los actores. El diagrama se compone por el CU (representa un objetivo y describe la secuencia de actividades), actores (representa un papel desempeñado por un usuario que interactúa, puede ser una persona o sistema externo)

ESCENARIOS: descripción de la interacción entre el actor y el sistema

A la hora de modelarlo hay que identificar a los actores, identificar los CU para los requerimientos, construir el diagrama y hacer los escenarios

REDES DE PETRI

usadas para especificar sistemas de tiempo real en los que son necesarios representar aspectos de concurrencia. Las tareas deben estar sincronizadas para permitir la comunicación entre ellas y así hacer tareas paralelamente

Los arcos indican a través de una flecha la relación entre sitios y transiciones y viceversa.

A los lugares se les asignan tokens (fichas) que se representan mediante un número o puntos dentro del sitio. Esta asignación de tokens a lugares constituye la marcación. Luego de una marcación inicial se puede simular la ejecución de la red. El número de tokens asignados a un sitio es ilimitado. El conjunto de tokens asociado a cada estado sirve para manejar la coordinación de eventos y estados. Una vez que ocurre un evento, un token puede “viajar” de uno de los estados a otro.

Las reglas de disparo provocan que los tokens “viajen” de un lugar a otro cuando se cumplen las condiciones adecuadas.

La ejecución es controlada por el número y distribución de los tokens. La ejecución de una Red de Petri se realiza disparando transiciones habilitadas.

Una transición está habilitada cuando cada lugar de entrada tiene al menos tantos tokens como arcos hacia la transición.

Disparar una transición habilitada implica remover tokens de los lugares de entrada y distribuir tokens en los lugares de salida (teniendo en cuenta la cantidad de arcos que llegan y la cantidad de arcos que salen de la transición).

Para que varios procesos colaboren en la solución de un problema es necesario que compartan información y recursos pero debe ser controlado para asegurar la integridad y correcta operación del sistema

TABLAS DE DECISIÓN

herramienta que permite presentar de forma concisa las reglas lógicas que hay que utilizar para decidir acciones a ejecutar en función de las condiciones y la lógica de decisión de un problema específico. Describe el sistema como un conjunto de posibles

CONDICIONES satisfechas por el sistema en un momento dado REGLAS para reaccionar ante los estímulos que ocurren cuando se reúnen determinados conjuntos de condiciones y ACCIONES a ser tomadas como un resultado.

Para construir tablas de decisión, el analista necesita determinar el tamaño máximo de la tabla; eliminar cualquier situación imposible, inconsistencia o redundancia, y simplificar la tabla lo más que pueda.

Es esencial que verifique la integridad y precisión de sus tablas de decisión. Pueden ocurrir cuatro problemas principales al desarrollar tablas de decisión: que estén incompletas, que existan situaciones imposibles, contradicciones y redundancia.

ANÁLISIS ESTRUCTURADO

Para entender los requerimientos, se debe poder reconocer además como se mueven los datos, los procesos o transformaciones que sufren dichos datos y sus resultados. La técnica de análisis estructurado permite lograr una representación gráfica que permite lograr una comprensión mas profunda del sistema a construir y comunicar a los usuarios lo comprendido.

DIAGRAMA DE FLUJO DE DATOS: herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y almacenamientos de datos. Representa la transformación de entradas a salidas y es también llamado diagrama de burbujas. Es una herramienta comúnmente utilizada por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja

METODOLOGÍAS ÁGILES

una metodología ágil es aquella en la que "se da prioridad a las tareas que dan resultados directos y que reduce la burocracia tanto como sea posible"

VALORES

- individuos e interacciones más que procesos y herramientas
- software operante más que documentaciones completas
- colaboración con el cliente más que negociaciones contractuales
- respuestas al cambio más que apegarse a una rigurosa planificación

PRINCIPIOS

- nuestra mayor prioridad es satisfacer al cliente a través de fáciles y continuas entregas de software valuable
- los cambios de requerimientos son bienvenidos, aún tardíos, en el desarrollo
- entregas frecuentes de software, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre una entrega y la siguiente
- usuarios y desarrolladores deben trabajar juntos durante todo el proyecto
- construir proyectos alrededor de motivaciones individuales
- darles el ambiente y soporte que ellos necesitan y confiar en el trabajo dado
- el software que funciona es la medida clave de progreso
- los procesos ágiles promueven un desarrollo sostenible
- atención continua a la excelencia técnica y buen diseño incrementa la agilidad
- simplicidad

METODOLOGÍA ÁGIL	METODOLOGÍA NO ÁGIL
pocos artefactos	más artefactos
pocos roles	más roles
no existe un contrato tradicional o al menos es bastante flexible	existe un contrato prefijado
el cliente es parte del equipo de desarrollo	el cliente interactúa con el equipo de desarrollo mediante reuniones
grupos pequeños y trabajando en el mismo sitio	grupos grandes
menos énfasis en la arquitectura	la arquitectura es esencial

DEVENTAJAS

- aunque es atractiva la idea de involucrar al cliente, estos están sujetos a otras presiones
- priorizar los cambios podría ser difícil, sobre todo en sistemas donde existen muchos participantes
- mantener la simplicidad requiere trabajo adicional
- muchas organizaciones, especialmente las grandes compañías, pasan años cambiando su cultura, de tal modo que los procesos se definan y continúen

PRINCIPALES METODOLOGÍAS ÁGILES

- **XP extreme programming:** disciplina de desarrollo de software basado en los valores de la sencillez, la comunicación, la retroalimentación, valentía y respeto. Consiste en llevar a todo el equipo reunido en la presencia de prácticas simples con suficiente información para ver dónde están. Se caracteriza por HU, roles, proceso y prácticas.
Los programadores diseñan, programan y realizan las pruebas y no hay distinción entre analistas, diseñadores o codificadores. El jefe de proyecto organiza y guía las reuniones y asegura condiciones adecuadas para el proyecto. Finalmente, el cliente es parte del equipo, determina qué construir, cuándo y establece las pruebas funcionales. El entrenador es responsable del proceso y tiende a estar en un segundo plano a medida que el equipo madura. El encargado de pruebas ayuda al cliente con las pruebas funcionales y se asegura de que las pruebas funcionales se superan. El rastreador observa sin molestar y conserva datos históricos
el ciclo de vida en XP consiste en
 - exploración: los clientes plantean las HU que son de interés en la primera entrega, el equipo se familiariza con las herramientas y tecnologías y se construye un prototipo
 - planificación: el cliente establece la prioridad de cada HU, se estima el esfuerzo y se toman acuerdos sobre el contenido de la primera entrega
 - iteraciones: el plan de entrega está compuesto por iteraciones de no más de tres semanas, esta fase incluye varias iteraciones sobre el sistema antes de ser entregado
 - producción: fase que requiere pruebas adicionales y revisiones de rendimiento además se toman decisiones sobre la inclusión de nuevas características
 - mantenimiento: mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que se desarrolla nuevas iteraciones
 - muerte: es cuando el cliente no tiene más HU para incluirlas en el sistema, se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo
- las prácticas llevadas a cabo son
 - testing: los programadores escriben pruebas unitarias las cuales deben correrse sin problema para que el desarrollo continúe
 - refactoring: actividad constante de reestructuración del código para que sea más legible
 - programación de a pares: todo el código de producción es escrito por dos programadores en una máquina
 - propiedad colectiva del código: cualquiera puede cambiar código en cualquier parte del sistema en cualquier momento
 - integración continua: cada pieza de código es integrada en el sistema una vez que esté lista
 - semana de 40-horas: se debe trabajar un máximo de 40 horas por semana
 - cliente en el lugar de desarrollo: el cliente tiene que estar presente y disponible todo el tiempo para el equipo
 - estándares de codificación: los programadores escriben todo el código de acuerdo con reglas que enfatizan la comunicación a través del mismo
- **Scrum:** proceso en el que se aplican de manera regular, un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Se realizan entregas parciales y regulares del resultado final del proyecto
principios:
 - eliminar el desperdicio: no generar artefactos ni perder tiempo haciendo cosas que no le suman valor al cliente
 - construir la calidad con el producto: la idea es inyectar la calidad directamente en el código desde el inicio
 - crear conocimiento: en la práctica no se puede tener el conocimiento antes de empezar el desarrollo
 - diferir las decisiones: tomar las decisiones en el momento adecuado, esperar hasta ese momento ya que uno tiene más información a medida que va pasando el tiempo
 - entregar rápido: debe ser una de las ventajas competitivas más importantes
 - respetar a las personas: la gente trabaja mejor cuando se encuentra en un ambiente que la motive
 - optimizar todo: optimizar todo el proceso, ya que el proceso es una unidad y para lograr tener éxito y avanzar, hay que tratarlo como tal
- roles:
 - product owner: conoce y marca las prioridades del proyecto o producto
 - scrum master: persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que puede aparecer
 - scrum team: personas responsables de implementar la funcionalidad elegidas
 - usuarios: beneficiarios finales del producto
- artefactos:
 - product backlog: lista que contiene toda la funcionalidad deseada del producto
 - sprint backlog: lista de funcionalidades que el equipo realizó durante un sprint
 - burndown chart: muestra un acumulativo del trabajo hecho

el scrum es iterativo e incremental y busca poder atacar todos los problemas que surgen durante el desarrollo del proyecto. El nombre scrum se debe a que durante los Sprints, lo que serían las fases de desarrollo, se solapan, de manera que no es un proceso de cascada por cada iteración, si no que tenemos todas éstas etapas juntas que se ejecutan una y otra vez, hasta que se crea suficiente DSDM (Dynamic Systems Development Method). Scrum está pensado para ser aplicado en proyectos en donde el “caos” es una constante, aquellos proyectos en los que tenemos requerimientos dinámicos, y que tenemos que implementar tecnología de punta.

Esos proyectos difíciles, que con los enfoques tradicionales se hace imposible llegar a buen puerto

- Crystal Methods
- ASD Adaptive Software Development
- FDD Feature-Driven Development

DESARROLLO DE SOFTWARE BASADO EN MODELOS

hacia fines de los 70' De Marco introdujo el concepto de desarrollo de software basado en modelos. Destacó que la construcción de un sistema de software debe ser precedida por la construcción de un modelo, tal como se realiza en otros sistemas ingenieriles. Un modelo del sistema consiste en una conceptualización del dominio del problema y actúa como una especificación precisa de los requerimientos que el sistema de software debe satisfacer

Model Driven Development (MDD) promueve enfatizar los siguientes puntos claves:

- Mayor nivel de abstracción en la especificación tanto del problema a resolver como de la solución correspondiente.
- Aumento de confianza en la automatización asistida por computadora para soportar el análisis, el diseño y la ejecución.
- Uso de estándares industriales como medio para facilitar las comunicaciones, la interacción entre diferentes aplicaciones y productos, y la especialización tecnológica.
- Los modelos son los conductores primarios en todos los aspectos del desarrollo de software.

Beneficios:

- Incremento en la productividad (modelos y transformaciones).
- Adaptación a los cambios tecnológicos.
- Adaptación a los cambios de requisitos.
- Consistencia (automatización).
- Re-uso (de modelos y transformaciones).
- Mejoras en la comunicación con los usuarios y la comunicación entre los desarrolladores (los modelos permanecen actualizados).
- Captura de la experiencia (cambio de experto).
- Los modelos son productos de larga duración (resisten cambios).
- Posibilidad de demorar decisiones tecnológicas

CALIDAD

Conjunto de propiedades o características de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas

norma: documento, establecido por consenso y aprobado por un organismo reconocido (nacional o internacional), que proporciona para un uso común y repetido, una serie de reglas, directrices o características

organización internacional de normalización, también llamada Organización Internacional de Estandarización (International Organization for Standardization, ISO) es una organización para la creación de estándares (o normas) internacionales compuesta por diversas organizaciones nacionales de normalización

sistema de información: conjunto de personas, datos, procesos y tecnología de información que interactúan para recopilar, procesar, guardar y proporcionar como salida la información necesaria para brindar soporte a una organización en la actualidad hace necesario que las empresas de tecnología hagan mucho hincapié en los estándares (o normas) de calidad

la calidad tiene determinados componentes

- calidad del software: de las aplicaciones de software construidas o mantenidas o con el apoyo de la IS
- calidad de la información: relacionada con la calidad de los datos
- calidad del dato: que ingresa en el sistema de información
- calidad de la infraestructura: incluye la calidad de las redes y sistemas de software
- calidad de la gestión: incluye el presupuesto, planificación y programación
- calidad del servicio: incluye los procesos de atención al cliente

la *calidad de software* se refiere al conjunto de propiedades o características de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas. Está dividida en

- calidad del producto obtenido: estandarización del producto define las propiedades que debe satisfacer el producto de software resultante
 - calidad del proceso de desarrollo: la estandarización del proceso define la manera de desarrollar el producto de software
- ambas son dependientes, sin un buen proceso de desarrollo es casi imposible obtener un buen producto

CMM

es un modelo de evaluación de los procesos de una organización. Proporciona un marco estructurado para evaluar los procesos actuales de la organización, establece prioridades de mejora e implementa esas mejoras

CMMI

posee dos vistas que permiten un enfoque diferente según las necesidades de quien vaya a implementarlo

- escalonado: centra su foco en la madurez de la organización
- continuo: enfoca las actividades de mejora y evaluación en la capacidad de los diferentes procesos. Presenta 6 niveles de capacidad. Los niveles de capacidad indican qué tan bien se desempeña la organización en un área de proceso individual

ISO- International organization for standardization

la familia ISO 9000 es un conjunto de normas de “gestión de la calidad” aplicables a cualquier tipo de organización con el objetivo de obtener mejoras en la organización y, eventualmente arribar a una certificación, punto importante a la hora de competir en los mercados globales

- ISO- 9001: 2015
- IRAM- ISO 9001: 2015
- ISO 90003:2004, proporciona una guía para identificar la evidencia dentro del proceso de software para satisfacer los requisitos de la ISO 9001

los beneficios de trabajar con un sistema de gestión de calidad es

- ISO 9001: asegura que su negocio cumpla con los requisitos legales y del cliente
- aumenta el rendimiento de la organización. El sistema de gestión de calidad ayuda a implementar procesos simplificados y mejorar la eficiencia operacional
- asegura la toma de decisiones y mejora la satisfacción del cliente
- optimiza las operaciones para así cumplir y superar requisitos de los clientes
- mejora el rendimiento financiero

Calidad de producto de software	Se evalúa la calidad mediante	ISO/IEC 25000	Esta compuesto por distintos modelos. Define características que pueden estar presentes o no en el producto. La norma nos permite evaluar si están presentes o no, y de que manera evaluarlas. EJ: Seguridad, Compatibilidad, Seguridad. Etc.
Calidad de proceso de desarrollo de software	Se evalúa la calidad mediante	ISO/IEC 12207	ISO/IEC 12207 establece un modelo de procesos para el ciclo de vida del software. Define cómo debería ser el modelo de proceso para ser completo y con calidad. Actividades, tareas etc.
		ISO/IEC 15504	Es una norma internacional para establecer y mejorar la capacidad y madurez de los procesos. Define que se debe tener en cuenta para evaluar el modelo de proceso y concluir si es completo y con calidad.
		ISO/IEC 90003	Proporciona una guía sobre cómo aplicar la ISO 9001 en procesos de software
		CMMI	Proporciona un marco estructurado para evaluar los procesos actuales de la organización, establecer prioridades de mejora, e implementar esas mejoras. Se utiliza para organizaciones desarrolladoras de software de medianas a grandes dimensiones
Calidad de Procesos/Servicios /Productos en general	se evalúa mediante	ISO 9001	La Norma ISO 9001 determina los requisitos para establecer un Sistema de Gestión de la Calidad, de producto y/o servicio. Forma parte de la familia ISO 9000, que es un conjunto de normas de “gestión de la calidad” aplicables a cualquier tipo de organización con el objetivo de obtener mejoras en la organización y, eventualmente arribar a una certificación, punto importante a la hora de competir en los mercados globales.