

Ingeniería de software 1

1) Conceptos de software e ingeniería de software

Software

Conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Tipos de productos de software

- Genéricos: sistemas aislados producidos por organizaciones desarrolladoras de software y que se venden en un mercado.
- Personalizados: sistemas requeridos por un cliente en particular y lo puede desarrollar la propia organización o un contratista.

Clasificación de software

- De sistema
- De gestión
- Científico
- Empotrado
- De tiempo real
- Basados en la web
- De inteligencia artificial

Características del software

- Elemento lógico
- Se desarrolla, no fabrica. Tiene más costo en la ingeniería que en la producción y hay una tendencia importante de construcción por componentes.

Evolución del software

El software no se desgasta y tampoco sigue una curva clásica de envejecimiento. El problema no está en el tiempo de operación, sino en los cambios.

Ingeniería de software

Disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de especificación seguido por la evolución.

Como disciplina de la ingeniería hará que las cosas funcionen aplicando teorías, métodos y herramientas.

Además no solo comprende procesos técnicos del desarrollo sino también incluye actividades de gestión de proyectos y desarrollo de herramientas, métodos y teorías.

- Según IEEE, la ingeniería de software es el uso de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software.

Con métodos sistemáticos cuantificables se refiere a que hay una cuantificación de los recursos, procesos y productos para optimizar la productividad y calidad.

En conclusión, al ingeniería de software trata de dar principios y métodos que permitan producir software confiable y eficiente al menor costo. Para ello, se establecerán métodos, herramientas y procedimientos.

Un **ingeniero de software** debe

- Tener conocimientos científicos, metodológicos, tecnológicos y administrativos.
- Estar familiarizado con lógica, estadística, simulación, notaciones, diseño y programación.
- Saber aplicar metodologías de documentación.
- Conocer tecnologías y productos.

Además deberá contar con **responsabilidad profesional y ética** ya que la ingeniería de software se desarrolla en un marco económico, social y legal.

- No debe usar capacidad y habilidades deshonestamente.
 - Respetar confidencialidad de empleados y clientes.
 - No falsificar nivel de competencia.
 - Conocer leyes vigentes sobre patentes y copyright.
-

2) Proceso de software

El proceso de software es el conjunto de actividades y resultados asociados que producen un producto de software. Las tareas principales de este proceso es: especificación, desarrollo, validación y evolución del software.

Modelo de proceso de software

Representación estructurada de cómo se lleva a cabo el desarrollo de software. Describe las actividades, tareas, roles e interacciones necesarias para producir un producto de SW.

Hay diferentes paradigmas

- Modelo en cascada: representa las actividades anteriores y las representa como fases de proceso separadas.
- Especificación de requerimientos, diseño, implementación, etc.
- Desarrollo iterativo: un sistema inicial se desarrolla rápidamente a partir de una especificación abstracta. Éste se refina basándose en las peticiones del cliente.
- Ingeniería basada en componentes: supone que las partes ya existen. El proceso se enfoca en la integración de las partes.

Los productos de software tienen un cierto número de atributos asociados que reflejan su calidad. Serán

- Mantenibilidad, es la posibilidad de hacer modificaciones ante cambios del negocio.

- Confiabilidad, no debe causar daños ante fallas.
- Eficiencia, uso apropiado de los recursos.
- Usabilidad, fácil uso sin esfuerzo adicional.

Modelo de procesos tradicionales

- Proceso: es el conjunto de actividades, restricciones y recursos que producen una determinada salida.
- Ciclo de vida del software: describe la vida del producto de software desde su concepción hasta la entrega, y mantenimiento.
- Modelo de procesos de software: representación abstracta de un proceso del software

Modelo	Definición
Modelo en cascada	Las etapas se representan cayendo en cascada donde cada etapa se debe completar antes que comience la siguiente. Es útil para diagramar lo que se necesita hacer. La desventaja es que no habrá resultados hasta que esté todo terminado, será difícil eliminar fallas en las últimas etapas.
Modelo en V	Demuestra cómo se relacionan las actividades de prueba con las de análisis y diseño.
Modelo prototipos	Un prototipo es un producto parcialmente desarrollado que permite que clientes y desarrolladores examinen aspectos del sistema propuesto y decidan si éste es adecuado o correcto. Podrá ser evolutivo o descartable. En los evolutivos tiene como objeto obtener el sistema a entregar, en cambio en los descartables no tiene funcionalidad y servirá para modelar.
Desarrollo por fases	Se desarrolla el sistema de manera que puede ser entregado en piezas. Implica que existen dos sistemas funcionando en paralelo, el operacional y el en desarrollo.

Metodologías ágiles

La ingeniería de software ágil combina una filosofía y un conjunto de directrices de desarrollo.

Busca satisfacer al cliente y hacer una entrega temprana de software incremental, habiendo equipos pequeños y con alta motivación.

Hay varias metodologías pero todas minimizan riesgos, el software desarrollado en una unidad de tiempo es llamado una iteración que deben durar de una a cuatro semanas.

Cada iteración incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación.

Extreme programming

Metodología basada en la sencillez, comunicación, retroalimentación, valentía y respeto.

Se hacen pruebas unitarias continuamente, hay programación en pareja, se corrigen todos los errores antes de añadir nueva funcionalidad, se refactoriza código y está simplificado.

SCRUM

Proceso donde se aplican mejores prácticas para trabajar en equipo.

Se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptos del proyecto.

No se generan artefactos, se inyecta calidad en el código desde el inicio, las entregas son rápidas, hay un ambiente de motivación y respeto.

Elementos

- Product owner: conoce y marca prioridades del producto.
- SCRUM master: persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo.
- SCRUM team: personas responsables de implementar funcionalidades.
- Usuarios o clientes: beneficiarios del producto.
- Product backlog: lista que contiene toda la funcionalidad deseada en el producto.
- Sprint backlog: lista que contiene toda la funcionalidad que el equipo se comprometió a desarrollar durante un sprint.

Será iterativo e incremental.

Estudio de viabilidad

Importante para sistemas nuevos. Es un informe para recomendar la conveniencia o no de un desarrollo.

Responde a si el sistema contribuye a los objetivos de la organización, si se puede implementar con la tecnología actual, el costo y tiempo y si es integrable con otros sistemas.

3) Ingeniería de requerimientos

Disciplina para desarrollar una especificación completa, consistente y no ambigua que servirá como base para acuerdos comunes entre todas las partes involucradas y en donde se escriben las funciones que realizará el sistema. También se intercambian puntos de vista para recopilar y modelar lo que el sistema realizará.

Permite gestionar las necesidades del proyecto estructuradamente, hacer cronogramas, disminuir costos y retrasos, mejorar la calidad y comunicación y evitar rechazos de usuarios finales.

Comunicación

Base principal para obtener necesidades del cliente. Es el punto más importante porque es la fuente principal de errores como falta de guías formales, falta de participación del usuario, mala interpretación de las necesidades o falta de comunicación.

Requerimiento

Característica del sistema o descripción de algo que el sistema es capaz de hacer.

Tipos de requerimientos

- **Requerimientos funcionales:** describen una interacción entre el sistema y su ambiente, cómo debe comportarse el sistema ante determinado estímulo.
- **Requerimientos no funcionales:** restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución del problema.
 - **Del producto,** especifican el comportamiento del producto (usabilidad, eficiencia, rendimiento, espacio, fiabilidad y portabilidad).
 - **Organizacionales:** políticas y procedimientos existentes en la organización del cliente y en la del desarrollador.
 - **Externos:** legalidad, privacidad, seguridad, éticos.
 - **Del dominio:** reflejan las características y restricciones del dominio de la aplicación del sistema.
 - **Por prioridad:** que deben ser absolutamente satisfechos, que son deseables pero no indispensables, que son posibles pero que no podrían eliminarse.
 - **Del usuario:** declaraciones en lenguaje natural y en diagramas de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales debe operar.

Fuentes de requerimientos

- **Documentación.**
- **Stakeholders:** cualquier persona que se verá afectado por el sistema, directa o indirectamente. Pueden ser usuarios finales, ingenieros, gerentes.
Puedo clasificarlos en distintos puntos de vista
 - **De los interactuadores:** personas o sistemas que interactúan directamente con el sistema.
 - **Indirecto:** stakeholders que no usan el sistema ellos mismos pero influyen en los requerimientos de alguna forma.
 - **Del dominio:** características y restricciones del dominio que influyen en los requerimientos.
- **Especificación de sistemas similares.**

Tipos de documentos

- **Documento de definición de requerimientos:** listado completo de todas las cosas que el cliente espera del sistema.
- **Documento de especificación de requerimientos:** definición en términos técnicos.
- **Documento de especificación de requerimientos de software IEEE 380 (SRS):** colección de buenas prácticas para escribir especificaciones de requerimientos.

Validación de requerimientos

Proceso de certificar la corrección del modelo de requerimientos contra las intenciones del usuario. Intenta mostrar que los requerimientos definidos es lo que estipula el sistema. Es importante porque los errores en requerimientos pueden conducir gastos a futuro.

Cuando antes se valida, menos costoso será y se evitará una bola de nieve de defectos.

Para realizarlo puede hacerse manual o automatizadamente.

- **Informal:** los desarrolladores muestran los requerimientos con tantos stakeholders como sea posible.

- Formal: el equipo de desarrollo debe conducir al cliente, explicándole las implicaciones de cada requerimiento.

Elicitación de requerimientos

Proceso de adquirir todo conocimiento relevante para producir un modelo de los requerimientos de un dominio del problema. Tiene como objetivo conocer el problema para comunicarse con clientes y usuarios y así entender sus necesidades, conocer el sistema actual e identificar las necesidades.

Este proceso es de carácter social entonces surgirán problemas de comunicación

Del cliente	Del desarrollador
Dificultad para expresar las necesidades	Cultura o vocabulario diferente
No ser consciente de sus necesidades	Intereses distintos en el sistema a desarrollar
No entender cómo la tecnología puede ayudar	Medios de comunicación inadecuados
	Conflictos personales o políticos

Técnicas de elicitación

- Métodos discretos: son menos perturbadores que otras formas de averiguar los requerimientos. Se consideran insuficientes para recopilar información cuando se utilizan por sí solos, por lo que deben utilizarse junto con uno o varios de los métodos.
- Métodos interactivos: pueden usarse para obtener los requerimientos de los miembros de la organización. Aunque son distintos en su implementación, estos métodos tienen muchas cosas en común. La base es hablar con las personas en la organización y escuchar para comprender.

Métodos discretos	Métodos interactivos
Muestreo de la documentación, los formularios y datos existentes	Cuestionarios
Investigación y visitas al lugar	Entrevistas
Observación del ambiente de trabajo	Planeación conjunta de requerimientos
	Brainstorming

Muestreo de la documentación, formularios y datos existentes

Se observarán organigramas, notas internas, registros contables o solicitudes de proyectos de sistemas de información anteriores.

Permiten conocer el historial que origina el proyecto.

Investigación y visitas al sitio

Permite investigar el dominio, obtener patrones de soluciones a partir del mismo problema en otra organización, ver revistas especializadas, buscar problemas similares en internet o consultar otras organizaciones.

Observación del ambiente de trabajo

El analista se convierte en observador de las personas y actividades para aprender acerca del sistema.

Para eso, habrá ciertos lineamientos: quién y cuándo será observado, tener el permiso de la persona y explicar por qué se observará, tener perfil bajo, tomar nota de lo observado, no interrumpir a la persona en su ambiente de trabajo.

Ventajas	Desventajas
Datos confiables	La gente se siente incómoda siendo observada
El analista puede ver exactamente lo que se hace	Algunas actividades del sistema puede ser realizadas en horarios incómodos
Análisis de disposiciones físicas, tránsito, iluminación, etc.	Las tareas están sujetas a interrupciones
Económica en comparación con otras técnicas	Tener en cuenta que la persona observada puede estar realizando las tareas de la forma correcta y no como lo hace habitualmente

Cuestionarios

Es un documento que permite al analista recabar información y opiniones de los encuestados.

Sirve para recolectar hechos de un gran número de personas, detectar un seguimiento generalizado, detectar problemas entre usuarios y cuantificar respuestas.

Ventajas	Desventajas
Respuesta rápida	Número bajo de respuestas
Económico	No responde a todas las preguntas
Anónimo	Preguntas rígidas
Estructurado de fácil análisis	No se puede hacer análisis corporal
	No se pueden aclarar respuestas incompletas
	Difícil de preparar

Tipos de cuestionarios

- Formato libre: para ofrecer más flexibilidad en la respuesta y son preguntas WH-.
- Formato fijo: requieren la selección de una respuesta entre varias predefinidas. Suelen ser respuestas de SÍ o NO.

Con los cuestionarios obtendré información sobre

- Actitud: lo que las personas dicen que quieren.
- Creencias, o sea lo que creen que es verdad.
- Comportamiento.
- Características de personas o cosas.

Usaremos cuestionarios cuando las personas están dispersas geográficamente, hay muchos involucrados (ya sea clientes o usuarios), queremos opiniones o identificar problemas generales.

Diseñando un cuestionario

Suelen ser difíciles de desarrollar pero se recomienda

- Dejar espacios en blanco para que estén bien distribuidos y para responder.
- Facilitar el marcado de las respuestas.

- Mantener el estilo.
- Ordenar las preguntas agrupándolas por contenido similar.

Entonces el procedimiento para desarrollarlo será

1. Determinar qué hechos, opiniones y de quién, quieren recolectarse.
2. Determinar qué tipo de preguntas dan mejores resultados a los hechos buscados.
3. Escribir las preguntas y examinarlas en cuanto a errores o malas interpretaciones. No proveer un sesgo personal en la pregunta.
4. Ensayar en una pequeña muestra de encuestados.
5. Duplicar y distribuir el cuestionario.

Para la redacción, se debe tener en cuenta

- Redacción auto contenida y precisa.
- Hilo conductor entre las preguntas.
- Usar el lenguaje de los encuestados.
- Evitar preguntas muy específicas.
- Hacer preguntas breves y que no sean ofensivas.

Entrevistas

Técnica en la cual el analista recolecta información a través de la interacción cara a cara. Es una conversación con un propósito específico, se basa en un formado de preguntas y respuestas en general. Permitirá conocer opiniones y sentimientos del entrevistado

Ventajas	Desventajas
El entrevistado se siente incluido en el proyecto	Costoso
Hay una retroalimentación del encuestado	Las entrevistas dependen de las habilidades del entrevistador
Puedo adaptar las preguntas según el entrevistado	No aplicable a la distancia
Información no verbal	

Tipos de entrevistas

- Estructuradas (cerradas): el encuestador tiene un conjunto específico de preguntas para hacérselas al entrevistado, dirigiéndose al usuario sobre un requerimiento puntal y no permite adquirir un amplio conocimiento del dominio.
- No estructuradas (abiertas): el encuestador lleva a un tema en general, no hay preparación de preguntas específicas y se inicia con preguntas que no dependen del contexto para conocer el problema, la gente involucrada, etc.

Tipos de preguntas

- Abiertas: permite al encuestado responder de cualquier forma. Esto permite revelar nueva línea de preguntas, hacer más interesante la entrevista y dar más espontaneidad pero pueden darse muchos detalles irrelevantes, se pierde el control de la entrevista y hace parecer que el entrevistador no tiene los objetivos claros.

- Cerradas: respuestas directas o cortas. Ahorrará tiempo, se mantendrá fácilmente el control de la entrevista y dará datos relevantes pero puede aburrir al encuestado y no habrá muchos detalles.

Preparación previa

1. Leer antecedentes, para tener un vocabulario en común.
2. Establecer objetivos de la entrevista
3. Seleccionar los entrevistados: minimizando el número de entrevistas, deberán conocer con antelación el objetivo y preguntas a realizar.
4. Planificar la entrevista: establecer fecha, hora, lugar y duración.
5. Selección del tipo de preguntas a usar y su estructura.

Conducción de la entrevista

- Respetando horario y tiempos definidos.
- Asegurar el lugar donde se hará la entrevista.
- Iniciar saludando y presentándose.
- Mencionar propósito y duración.
- Escuchar con atención observando al entrevistado, tomando nota.
- Hacer breve conclusión.

Debo

Vestirme adecuadamente, ser cortés, escuchar cuidadosamente, mantener el control, observar los gestos, ser paciente y terminar a tiempo.

Evitar

Suponer que una respuesta no lleva a ningún lado, usar jerga, hablar en vez de escuchar, suponer respuestas, usar grabadores.

Luego de la entrevista

Enviar al entrevistado un resumen, permite aclarar cualquier cosa que no se haya entendido durante la entrevista.

Planeación conjunta de requerimientos (JRP)

Proceso donde se hacen reuniones de grupo altamente estructuradas para analizar problemas y definir requerimientos.

Para planearlos habrá que seleccionar una ubicación, participantes y armar una agenda.

Ventajas	Desventajas
Reduce tiempo de exploración de requisitos	Difícil de organizar porque hay muchos involucrados
Participación de integrantes	Difícil encontrar participantes integrados y organizados
Se trabaja sobre lo que se va generando	
Ahorra tiempo	
Desarrollo creativo	

Participantes

- Patrocinador: tiene autoridad sobre los departamentos que participan y es el responsable del proyecto que toma las decisiones finales.
- Facilitador: dirige las sesiones.
- Usuarios y gerentes: usuarios comunican requerimientos y los gerentes los aprueban.
- Secretarios: registran las sesiones y publican resultados.
- Equipo de IT: escuchan y toman nota de los requerimientos.

Brainstorming

Se generan ideas al alentar a los participantes para que ofrezcan tantas ideas como sea posible. Se hacen reuniones con las personas involucradas en la resolución de un problema.

Principios de Brainstorming

- Cuantas más ideas se sugieren, mejores resultados se conseguirán.
- La producción de ideas en grupo es más efectivo que individualmente.
- Las ideas de una persona pueden aparecer por contagio de otras.

Hay varias fases: descubrir hechos, producir ideas y descubrir soluciones.

4) Modelos del sistema

Técnicas de especificación de requerimientos

- Estáticas: el sistema se describe mediante objetos, atributos y relaciones pero NO describe cómo cambian con el tiempo.
- Dinámicas: considera un sistema en función de los cambios que ocurren a lo largo del tiempo. Considera que el sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado. Son las TD, DTE, Redes de Petri, CU, HU, etc.

Tablas de decisión

Representa de forma concisa las reglas lógicas que hay que usar para decidir acciones a ejecutar según condiciones.

El sistema tiene posibles **condiciones** satisfechas por el sistema en un momento dado, **reglas** para reaccionar ante los estímulos que ocurren cuando se reúnen condiciones y **acciones** a ser tomadas como resultado.

Hay $2N$ reglas donde N es la cantidad de condiciones.

A partir de un enunciado se debe identificar las condiciones y acciones. Si hay condiciones opuestas debe colocarse una de ellas porque por la negativa se obtendrá la otra.

Historias de usuario

Representación de un requisito de software escrito en una o dos fases usando el lenguaje común del usuario y sirve para especificar requisitos.

Permite responder rápidamente a los requisitos cambiantes.

Se espera que cada HU tenga una duración de 10 horas a un par de semanas. Si dura más de dos semanas indica que la historia es compleja y debe dividirse en varias historias.

Hay que indicar a quién beneficia, qué se quiere y cuál es el beneficio → Como ... quiero ... para poder ...

Se caracterizan por

- Independientes unas de otras
- Negociables.
- Valoradas por los clientes o usuarios.
- Pequeñas, si es larga será difícil de estimar.
- Verificables, cubre requerimientos funcionales entonces son verificables.

Son beneficiosas porque son implementaciones rápidas, requieren poco mantenimiento, dan una relación cercana con el cliente, puedo dividir al proyecto en pequeñas entregas y puedo estimar el esfuerzo de desarrollo.

Pero requieren contacto permanente con el cliente, difícil de escalar a proyectos grandes y requiere desarrolladores muy competentes.

Casos de uso

modela funcionalidades en término de los eventos que interactúan entre los usuarios y el sistema.

Facilita y alienta la participación de los usuarios.

Captura requerimientos funcionales, puede descomponer el sistema, usa el lenguaje común, puedo estimar el alcance del proyecto, el esfuerzo y da una línea base para planes de prueba y documentación

Elementos del modelo de CU

- Diagrama de casos de uso: ilustra interacciones entre el sistema y los actores.
 - Casos de uso: representa un objetivo individual del sistema y describe la secuencia de actividades para alcanzarlo.
 - Actores: interactúa con el sistema. Será una persona, sistema o dispositivo externo que emita eventos.
 - Relaciones: asociaciones (relación entre actor y CU), extensiones (un CU extiende funcionalidad de otro CU), uso (reduce la redundancia entre dos o más CU), dependencia (relación donde un CU no puede realizarse hasta que se haya realizado otro CU), herencia (relación entre actores donde un actor hereda las funcionalidades de uno o varios actores).
- Escenarios: descripción de la interacción entre el actor y el sistema.

Para modelarlo deberá identificar actores, los CU, construir el diagrama y realizar los escenarios.

Un CU debe representar una funcionalidad concreta donde la descripción de los escenarios debe tener más de un paso, caso contrario no se representará la interacción entre los componentes.

Diagrama de transición de estados

Describe al sistema como un conjunto de estados donde el sistema reacciona a ciertos eventos posibles.

Para construirlo debo identificar los estados, identificar los estados, analizar condiciones y acciones para pasar de un estado a otro y finalmente verificar la consistencia.

Redes de Petri

Especifica sistemas de tiempo real donde hay que representar aspectos de concurrencia ya que las tareas de concurrencia deben estar sincronizadas para permitir la comunicación entre ellas.

NO son tareas secuenciales entonces como ocurren paralelamente se necesita de alguna forma controlar los eventos para cambiar de estado.

5) Calidad

Capacidad de un producto o servicio para satisfacer las necesidades del usuario. La importancia de los sistemas de información (SI) en la actualidad hace necesario que las empresas de tecnología hagan mucho hincapié en los estándares de calidad.

Componentes

- Calidad de la infraestructura: incluye calidad de las redes, sistemas de software, etc.
- De software: de las aplicaciones de software construidas y mantenidas.
- De datos: que ingresan en el sistema de la información.
- De información: relacionada con la calidad de los datos.
- De gestión: incluye el presupuesto, planificación y programación.
- De servicio: procesos de atención al cliente.

La calidad del software mejoró en el último tiempo a partir de que se le dio importancia a la gestión de calidad y a las técnicas de gestión de calidad.

Puede dividirse en calidad del producto (bien tangible resultado de un proceso) obtenido y calidad del proceso (manera de desarrollar el producto) de desarrollo.

Calidad de los datos ISO/IEC 25012

Entiende por calidad de datos a la capacidad de los datos de satisfacer necesidades explícitas e implícitas bajo determinadas condiciones de uso.

- Inherente: capacidad de los datos de tener el potencial intrínseco para satisfacer las necesidades explícitas e implícitas.
- Dependiente del sistema: capacidad del sistema informático de alcanzar y preservar la calidad de los datos cuando los datos se usan en determinadas condiciones.

Proceso de evaluación ISO/IEC 25040

- Establecer los requisitos de la evaluación.
- Especificar la evaluación.

- Diseñar la evaluación.
- Ejecutar la evaluación.
- Finalizar la evaluación.

Modelo de calidad de proceso de software ISO 12207/ISO 15504

CMMI o “Modelo de capacidad y madurez- integración” incluye 4 disciplinas: software, ingeniería de sistemas, desarrollo integrado de procesos y productos y gestión de proveedores.

ISO 9000 y el desarrollo de software

- ISO 90003: 2018 establece directrices para la interpretación en el proceso de software y da una guía para identificar la evidencia dentro del proceso de software para satisfacer los requisitos de la ISO 9001.