

Técnicas de elicitación de Requerimientos

Proceso que se realiza para adquirir el conocimiento necesario para producir un modelo de requerimientos de un dominio de un problema, conocer el sistema actual e identificar las necesidades explícitas e implícitas del mismo.

Técnicas Discretas

Insuficientes para recopilar información por sí solos, deben combinarse con otros métodos.

- Muestreo de documentación: Se hace en base a documentos existentes (base de datos, sistemas actuales, diagramas, etc.)
- Investigaciones y visitas al sitio: Investigar el dominio, consultar otras organizaciones, etc.
- Observación del ambiente de trabajo: El analista observa a los trabajadores y actividades manteniendo bajo el perfil.

Técnicas Interactivas

Se basan en hablar con las personas en la organización y escuchar para comprender.

- Planeación conjunta de requerimientos (JRP): Reuniones estructuradas que se realizan para analizar problemas y definir requerimientos. Ahorran tiempo e involucran a los usuarios y a la gerencia activamente pero es complicado de organizar. Además requieren entrenamiento y equipo específico → ej. proyectores.
- Brainstorming: Se ofrecen ideas sin análisis hasta que no haya más ideas, se buscan ideas creativas y en gran cantidad. Ayudan a entender el dominio y resolver la falta de consenso.
- Cuestionarios: Se usan para obtener actitudes, creencias, comportamientos y características de un grupo grande o disperso de personas. Son rápidos, económicos, anónimos y fáciles de analizar. Son rígidos, difíciles de hacer y tienen pocas preguntas.

Tipos de preguntas:

Abiertas: Abiertas a todas las opciones de respuesta.

Cerradas: Limitan o cierran todas las opciones de respuesta.

Entrevistas

Recaudan información de las personas mediante una conversación. Tienen un propósito específico y generalmente se basan en preguntas y respuestas. Generan inclusión con el entrevistado y adaptabilidad en las preguntas para una mejor retroalimentación. Son costosas y no aplicables a distancia.

Tipos de entrevista

Abiertas: Preguntas a responder como quiera el entrevistado. Espontáneas, además, revelan más preguntas. Dan datos irrelevantes y pueden hacer que se pierda el control de la entrevista.

Cerradas: Preguntas directas o cortas. Ahorra tiempo, son fáciles de controlar y solo dan datos relevantes. No obtienen detalles.

Sondeo: Preguntas enfocadas en un tema puntual.

Organizaciones

Piramidal: Preguntas Cerradas → Abiertas

Embudo: Preguntas Abiertas → Cerradas

Diamante: Preguntas Cerradas → Abiertas → Cerradas

Técnicas de Especificación de Requerimientos

Las especificaciones de requerimientos tienen como objetivo que los devs le expliquen al cliente como ven al sistema y sus funcionalidades y qué demostraciones llevar a cabo para que el cliente vea que el sistema resultante es lo que pidió.

Técnicas Estáticas

Describen el sistema a través de las entidades, sus atributos y sus relaciones con otros enfocados en el estado actual del sistema y no los cambios que pueden afrontar.

Técnicas Dinámicas

Consideran al sistema en función de los cambios que puede afrontar a futuro.

Historias de Usuario

Descripción corta y simple de un requerimiento de un sistema. Se escriben en lenguaje común del usuario y desde su perspectiva. Son independientes unas de otras, negociables, valoradas por los usuarios, son estimables en tiempo, pequeñas y verificables.

Beneficios: Cortas, fáciles de mantener, ideales para proyectos volátiles o para dividir proyectos en pequeñas partes.

Limitaciones: Difíciles de escalar, requieren contacto constante con los usuarios.

Casos de Uso

Modelado de las funcionalidades del sistema en eventos entre los usuarios y el sistema.

Beneficios: Captura requerimientos funcionales, permiten la comunicación con el usuario, descompone el alcance del mismo en piezas manejables.

Componentes

Diagrama: Interacciones entre el sistema y los actores.

Escenarios: Interacciones entre el actor y el sistema.

Caso de uso: Funcionalidad individual del sistema.

Actor: Papel desempeñado por los usuarios.

Relaciones: Entre ellas: Asociaciones, Extensiones, Uso o inclusión, Dependencia y Herencia.

Diagrama de transición de estado

Máquinas de estado finito: Conjunto de estados donde el sistema reacciona a ciertos eventos posibles (externos o internos)

Redes de Petri

Utilizadas para especificar sistemas de tiempo real en los que son necesarios representar aspectos de concurrencia. Es multigrafo, dirigido.

Son asincrónicas y el orden en que ocurren los eventos es uno de los permitidos. La ejecución es no determinística. Pueden ser aplicadas para la modelación de sistemas de eventos discretos.

Componentes

Sitios: Son los estados del modelo.

Transiciones: Son los eventos de nuestro sistemas.

Arcos: Indican la relación entre sitios y transiciones.

Tokens: Sirven para habilitar o deshabilitar transiciones.

Tablas de decisión

Herramienta que representa las reglas lógicas para decidir acciones a ejecutar en función de las condiciones y la lógica de decisión de un problema específico.

Componentes: Condiciones (V o F), Acciones simples, 2^n donde n es la cantidad de condiciones.

Análisis Estructurado

Representación gráfica para lograr una comprensión más profunda del sistema a construir y comunicar a los usuarios lo comprendido enfocándose en el procesamiento de los datos en los procesos. Busca reconocer cómo se mueven los datos, sus procesos o transformaciones y sus resultados.

Stakeholders

Cualquier persona o grupo que se verá afectado por el sistema, directa o indirectamente, entre ellos, usuarios finales, ingenieros, gerentes, expertos del dominio, etc.

Puntos de vista

Interactuadores: Personas u otros sistemas que interactúen directamente con el sistema, Pueden influir en los requerimientos.

Indirectos: Stakeholders que no usan el sistema pero inferen en los requerimientos.

Dominio: Características o restricciones del dominio que influyen en los requerimientos.

Requerimientos

Característica del sistema o descripción de algo que él mismo es capaz de hacer en pos de satisfacer el propósito del sistema.

Requerimiento funcional: Describen lo que el sistema debe hacer o incluso cómo no debe comportarse. Descripción detallada de una función o capacidad que un sistema de software debe cumplir o ser capaz de resolver.

Requerimiento no funcional: Restricciones que limitan las elecciones de construcción de soluciones al problema. Relacionadas con la calidad, el rendimiento, la seguridad, la privacidad, aspectos legales, la eficiencia, la fiabilidad, la usabilidad, etc.

Modelos de Proceso de Software

Representación simplificada de un proceso de software que presenta una visión del mismo. Establece todas las actividades y pueden estar compuestos por subprocesos.

Modelos Prescriptivos: Prescriben un conjunto de elementos del proceso, actividades del marco de trabajo, acciones de la ingeniería de software, tareas, aseguramiento de la calidad, mecanismos de control y están interrelacionados.

Modelos Descriptivos: Describen cómo se realizan en la realidad.

Modelos Tradicionales: Conjunto de fases o actividades que no toma en cuenta la evolución del software.

Modelo en Cascada: Es simple y secuencial, se debe terminar la parte anterior para pasar a la siguiente. Desventajas → Necesita terminar para ver resultados, difíciles de manejar, necesita más pruebas al final ya que es donde ocurren más errores/fallos. No se adapta bien a los cambios.

Modelos en V: Demuestra cómo se relacionan las actividades de prueba con las de análisis y diseño. Sugiere que la prueba unitaria y de integración también sea utilizada para verificar el diseño del programa → Al vincularse el lado derecho con el izquierdo implica que, si se encuentran problemas en las verificaciones y validaciones, entonces el lado izquierdo de la V es el que se ejecuta de vuelta para solucionar el problema.

Modelos de Prototipos:

Prototipo: Producto parcialmente desarrollado que permite que se examinen algunos aspectos del sistema.

Características recomendadas para su uso: El sistema ha de ser experimentable. El costo ha de ser barato (<10%). Debe ser de desarrollo rápido. Enfocado en la interfaz de usuario. Equipo de devs reducido. Herramientas y lenguajes adecuados.

Tipos:

Evolutivo: Busca obtener el sistema final y permite construir rápidamente partes del sistema para analizar y tener una comprensión de lo que se necesita y lo propuesto como solución.

Descartable: Sin funcionalidad.

Modelos Evolutivos: Se adaptan a la evolución de los requisitos del sistema en el tiempo.

-Modelo de desarrollo por fases: El sistema se desarrolla por fases permitiendo que se entregue por partes.

Tipos:

Incremental: Se subdivide en subsistemas por funcionalidad. Se entrega por subsistemas.

Iterativo: Entrega un sistema completo y va aumentando las funcionalidades de los subsistemas con nuevas versiones.

Modelo en Espiral:

Características: Combina actividades de desarrollo con la gestión del riesgo. Incorpora objetivos de calidad. Trata de mejorar los ciclos de vida clásicos y prototipos. Elimina errores y alternativas poco atractivas para el inicio. Permite iteraciones. En cada ciclo se identifican primero los objetivos de su porción y sus alternativas.

Restricciones: Cada ciclo se completa con una revisión que incluye todo el ciclo anterior y el plan para el siguiente ciclo a desarrollar.

Metodologías Ágiles

Enfoque iterativo e incremental de desarrollo de software que emerge como una posible solución a las metodologías con énfasis en el control de proceso, definiendo roles, actividades, herramientas y documentación.

Objetivos: Producir software de alta calidad con un costo efectivo y tiempo apropiado. Esbozar los valores y principios que permiten desarrollar software rápidamente y responder a situaciones cambiantes.

Ser alternativa a procesos de desarrollo de software rígidos.

Dan prioridad a las tareas con resultados **directos** y **reducen la burocracia** lo más posible.

Desventajas: Los clientes no intervienen por completo en el desarrollo del software. Priorizar los cambios es difícil en sistemas grandes. Posiblemente no haya tiempo para las simplificaciones que hay que realizarle al sistema. El cambio a otro modelo de trabajo es duro. Complejidad a la hora de reglamentar los documentos de requerimientos.

Extreme Programming (XP)

Consiste en llevar a todo el equipo reunido en la presencia de prácticas simples, con información suficiente para ver dónde están y ajustar las prácticas a su situación particular.

Prácticas/Características:

- Testing.
- Refactoring del código.
- Programación de a pares.
- Integración continua de piezas de código
- Semana de 40 horas laborales.

- Cliente en lugar de desarrollo.
- Estándares de codificación.
- Simplicidad de código.

Los elementos claves son: Historias de Usuario, Roles, Proceso y Prácticas.

Roles: Conformados por programador, Jefe de Proyecto, Cliente, Entrenador, Encargado de pruebas y Rastreador.

Scrum

Marco de trabajo para el desarrollo colaborativo en productos complejos. Proporciona un paradigma de trabajo que soporta la innovación, permite la auto-organización del equipo y garantiza entregas parciales y regulares, priorizadas por su beneficio al receptor del proyecto. Es usada en proyectos con requerimientos dinámicos y tecnología de punta.

Principios:

- Eliminar el desperdicio.
- Construir la calidad con el producto.
- Crear conocimiento.
- Diferir en las decisiones.
- Entregar rápido.
- Respetar a las personas.
- Optimizar el todo.

Roles:

Propietario (conoce y marca las prioridades).

Jefe (garantiza el seguimiento de la metodología).

Equipo (implementan las funcionalidades).

Usuario/Cliente (beneficiarios finales).

Artefactos:

Product Backlog (lista de todas las funcionalidades).

Sprint Backlog (lista de funcionalidades a desarrollar en un Sprint)

Burndown Chart (acumulativo del trabajo hecho día a día)

Proceso: Iterativo e incremental con fases de desarrollo solapadas en ejecución.

Software

Instrucciones, procedimientos y datos asociados que forman parte de las operaciones de un sistema de computación.

Características:

- Es un elemento lógico.
- Se desarrolla.
- No se desgasta.
- No sigue una curva de envejecimiento normal porque el problema está en los nuevos cambios y no en el tiempo de operación..

Calidad

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

Calidad de Software

De gran importancia en la industria por la gestión de calidad y las técnicas de gestión de la misma. Compuesta por 2 calidades dependientes entre ellas.

Calidad del Producto: La estandarización del producto define las propiedades que debe satisfacer el producto resultante.

Calidad del Proceso: La estandarización del proceso define la manera de desarrollar el producto.

Calidad de producto de software	Se evalúa la calidad mediante	ISO/IEC 25000	Está 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 qué 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 (reemplazada por ISO 33000)	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 en general	se evalúa mediante	ISO 9001	La Norma ISO 9001 determina los requisitos para establecer un Sistema de Gestión de la Calidad. 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.