



Introducción a requerimientos y modelos de negocios

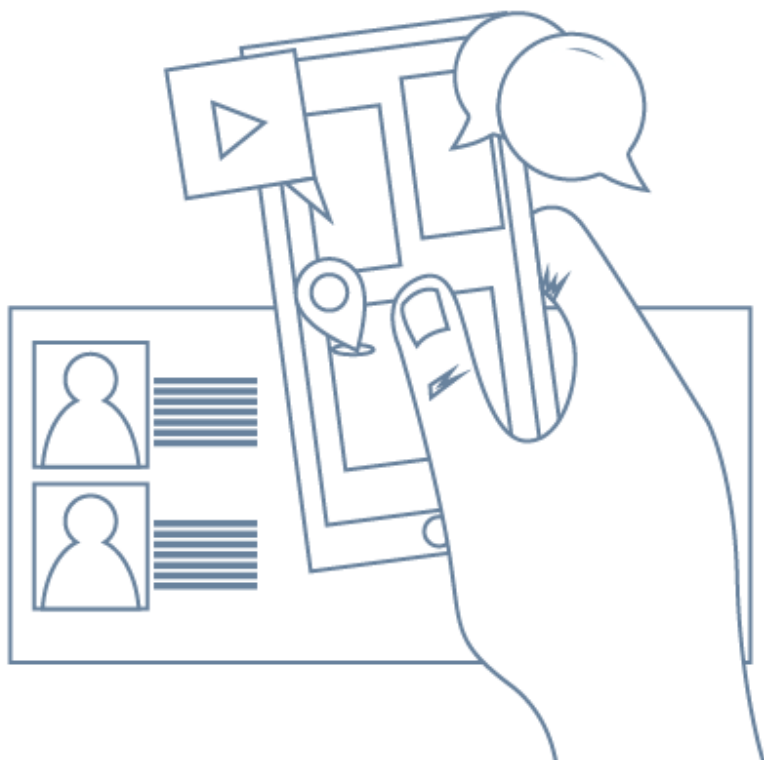
Requerimientos de software





Introducción a requerimientos y modelos de negocios

Requerimientos de software







ESCUELA DE CONSTRUCCIÓN E INGENIERÍA

Director de Escuela / Marcelo Lucero Yáñez

ELABORACIÓN

Experto disciplinar / Pablo Celedón Rodríguez

Diseñador instruccional / Camila Palacios Dussuel

VALIDACIÓN PEDAGÓGICA

Jefa de diseño instruccional y multimedia / Alejandra San Juan Reyes

Experto disciplinar / Rodrigo Pedrero

DISEÑO DOCUMENTO

Equipo de Diseño Instruccional AIEP



Contenido

Aprendizaje esperado de la semana.....	6
Ideas clave.....	6
1. ¿Qué es un requerimiento?	7
2. Tipos de requerimientos	8
2.1 Requerimientos de usuario.....	8
2.2 Requerimientos de sistema	8
2.3 Requerimientos funcionales	8
2.4 Requerimientos no funcionales	8
3. Ciclo de vida del software	11
4. Estructura de la gestión de requerimientos.....	13
4.1 La pirámide	13
4.2 Etapas de solicitud	14
5. Tipos de problemática según el tipo de organización	15
6. Técnicas de obtención de requerimientos funcionales y no funcionales.....	15
6.1 ¿Cuáles son las actividades de ingeniería de requerimientos?	15
6.2 Técnicas de obtención de requerimientos	16
7. Procesos incrementales de mejora.....	18
Conclusiones	19
Bibliografía.....	19
Enlaces y material multimedia	19

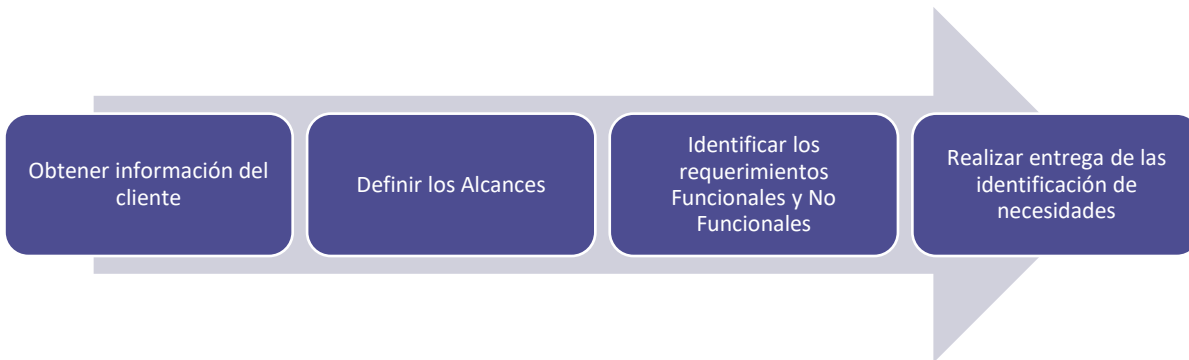


Aprendizaje esperado de la semana

Organizan requerimientos funcionales y no funcionales a partir de técnicas de obtención de requisitos del ciclo de vida de un software, considerando problemática y tipo de organización

Ideas clave

El proceso de ingeniería de requerimientos consiste en recopilar, analizar y verificar todas las necesidades que tiene un cliente para desarrollar un sistema de software. Como meta es entregar un análisis de los requerimientos de manera correcta y completa, donde aplicaremos técnicas y metodologías que iremos aprendiendo en el desarrollo del curso. Para que esta metodología sea efectiva se definen las siguientes actividades que se deben aplicar para la correcta identificación de necesidades de nuestros clientes





1. ¿Qué es un requerimiento?

En primera instancia, "se indica que requerimientos es la descripción para el desarrollo de un software donde se describen por medio de un documento formal o estipulado en contrato los atributos, capacidades, características y/o cualidades que necesita cumplir el software y/o sistema que el usuario detecta como una necesidad del negocio" (Zielczynski, Peter. "Requirements Management Using IBM® Rational® RequisitePro®", IBM Press., 2008. PMBOK® Guide).

Todo requerimiento debe cumplir con ciertas características para una mejor comunicación y que el jefe de proyecto se de a entender con claridad. Un requerimiento debe ser:

- a) Claro: el requerimiento debe ser entendible y no dar lugar a error o mala interpretación.
- b) Medible: debe ser capaz de lograr obtener resultados
- c) Necesario: debe existir un usuario con una necesidad que tenga un efecto.
- d) Trazabilidad: que se conozca de forma pública los avances del proyecto desde un inicio al final.
- e) Verificable: el proyecto debe poder comprobarse, donde se obtenga resultados de aprobación o rechazo del resultado de test.
- f) Cuantificable: se puede proyectar de forma monetaria el desarrollo del proyecto.
- g) Completo: un requerimiento debe ser especificado teniendo en cuenta todas las condiciones que puedan ocurrir.

Estas características son esenciales porque durante la petición del requerimiento ocurren las siguientes situaciones:

- a) A pesar de que el usuario tiene claro la necesidad que desea satisfacer, no siempre sabe expresarse con claridad, o con términos entendibles para quienes deben abordar la necesidad.
- b) Lenguaje distinto entre usuario y quien desarrolla la necesidad.
- c) No incluir al usuario a ser parte del desarrollo de la solución, provoca que no se llegue realmente a lo solicitado.

Otra definición considera que el análisis de requerimientos establece el proceso de definición de requerimientos como una serie de tareas o actividades, mediante las cuales se busca ganar conocimientos relevantes del problema y que se utilizarán para producir una especificación formal del software necesario para resolverlo. En este proceso, se deben conciliar diferentes puntos de vista y utilizar una combinación de métodos, personas y herramientas. El resultado final constituye la documentación de los requerimientos.



2. Tipos de requerimientos

Los requerimientos se clasifican en dos categorías: los que solicita el usuario y los requerimientos del sistema.

2.1 Requerimientos de usuario

El usuario declara de forma que sea comprensible la relación del usuario con el sistema desde su punto de vista, explicando con un lenguaje natural (casos de uso) y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.

2.2 Requerimientos de sistema

Se establece una descripción en detalle de las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema deberá ser preciso, y definir exactamente lo que se va a solicitar.

Otra forma de clasificar los requerimientos en base de lo solicitado es dividirlos en funcionales y no funcionales.

2.3 Requerimientos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema en detalle. Estos requerimientos se utilizan para determinar que hará el Software, definiendo las relaciones de su operación y su implementación, sin olvidar que deben ser explícitos también en lo que el sistema no debe hacer y que validaciones se deben realizar, teniendo en cuenta cual será el comportamiento del sistema.

Ejemplo de requerimiento Funcional:

- ✓ El usuario tiene la posibilidad de buscar en el conjunto inicial de la base de datos o buscar un subconjunto de ella.
- ✓ El sistema deberá proporcionar visores adecuados para que el usuario lea documentos en el almacén de datos.
- ✓ A cada pedido se le deberá asignar un identificador único, que el usuario podrá copiar al área de almacenamiento permanente de la cuenta.

2.4 Requerimientos no funcionales

Define propiedades o restricciones de los servicios o funciones ofrecidos por el sistema. Incluye restricciones de tiempo, sobre el proceso de desarrollo y estándares. Dentro de estos requerimientos encontramos todo lo referente a fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. El proceso de requerimiento puede especificarse en un sistema en particular de lenguaje de programación o método de desarrollo. Además, estos surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las herramientas utilizadas, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad, etcétera.



Los requerimientos no funcionales, a su vez, se clasifican en:

Requerimientos del Producto: se especifica el comportamiento del producto. Por ejemplo: el rendimiento del sistema (memoria, rapidez, etc.), fiabilidad (tasa de fallos aceptables) y usabilidad.

Requerimientos organizacionales: se ejecutan en base a políticas y procedimientos existentes en la organización del cliente y desarrollados. Por ejemplo: El desarrollo del sistema y documentación a implementar debe ser en base a los procesos, otro ejemplo es ajustarse a los tiempos de trabajo de la empresa.

Requerimientos externos: estos requerimientos se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador. En esta clasificación de requerimientos encontramos los que tienen que ver con Requerimientos Legislativos, Requerimientos Éticos, etc.



REQUERIMIENTOS NO FUNCIONALES. (FUENTE: INGENIERÍA DE SOFTWARE, SÉPTIMA EDICIÓN, IAN SOMMERVILLE, 2005, EDITORIAL PEARSON EDISON WESLEY



Estos requerimientos son importantes de definir. En caso contrario, el sistema es inútil en su uso.

Algunas recomendaciones para determinarlos son:

Propiedad	Medida
Rapidez	Transacciones procesadas por segundo Tiempo de respuesta al usuario y a eventos Tiempo de actualización de la pantalla
Tamaño	K Bytes Número de chips de RAM
Facilidad de Uso	Número en promedio en fallos Probabilidad de no disponibilidad Tasa de ocurrencia de fallos Disponibilidad
Fiabilidad	Tiempo de reinicio de fallos Porcentaje de eventos que provocan fallos Probabilidad de corrupción de los datos después de fallos
Robustez	Porcentaje de declaraciones dependientes de objetivos Número de sistemas objetivo

En síntesis, se debe recordar que todo requerimiento depende del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos.

Los requerimientos funcionales y no funcionales son de gran importancia para el desarrollo de una aplicación en software, por lo tanto, siempre deben ser escritos con claridad, contener la mayor especificación de las necesidades expuestas por el cliente, esto con el fin de tener un soporte base desde el cual se trabajarán y no presentar ambigüedades en la definición y el resultado del producto.



3. Ciclo de vida del software

El ciclo de vida de un proyecto consta de una serie de fases por las que se atraviesan desde su inicio hasta su cierre. A pesar de que cada proyecto cuenta con un ciclo de vida diferente, existen ciertas características genéricas que se cumplen:

- Las fases del proyecto por las que pasa son generalmente secuenciales, y en ocasiones se superponen y se ejecutan en paralelo.
- Usualmente, los costos y esfuerzos son menores al inicio y al final, y en la fase de ejecución suele incrementar. Esto es, dependiendo la necesidad del proyecto como también que se requiere grandes inversiones de tiempo y esfuerzo en el arranque y luego se estabiliza con el tiempo. Cada gestor de proyectos debe saber identificar cual es el caso.
- El nivel de riesgo con la incertidumbre es mayor al comienzo debido a que el proyecto aún es inmaduro y contiene aspectos no definidos y faltan decisiones por tomar.
- Como gestor de proyectos se debe saber prever los riesgos que se presenten en etapas avanzadas para solucionarlos o evitarlos.

Pero, antes de conocer el ciclo de vida de un software, se debe entender que todo proyecto varía en su tamaño y complejidad, existiendo una estructura genérica de ciclo de vida, que es:

- Inicio del proyecto
- Organización y Preparación
- Ejecución del trabajo
- Cierre del Proyecto
- Control y seguimiento

Este ciclo de vida, si lo llevamos a un proyecto de software, se clasifica en 5 etapas de alto nivel:

Inicio: etapa inicial donde se define los objetivos del proyecto y los recursos necesarios para su ejecución, se define hacia donde se quiere ir y las características implícitas y explícitas del proyecto. Principal mente se define el requerimiento en un documento formal del que se requiere y se formaliza.

Planificación: planeamiento detallado que guía la gestión del proyecto, aquí se establece Carta Gantt del proyecto, estimando costos, tiempo y recursos. Y análisis y diseño coordinado este acorde a la arquitectura de la empresa.



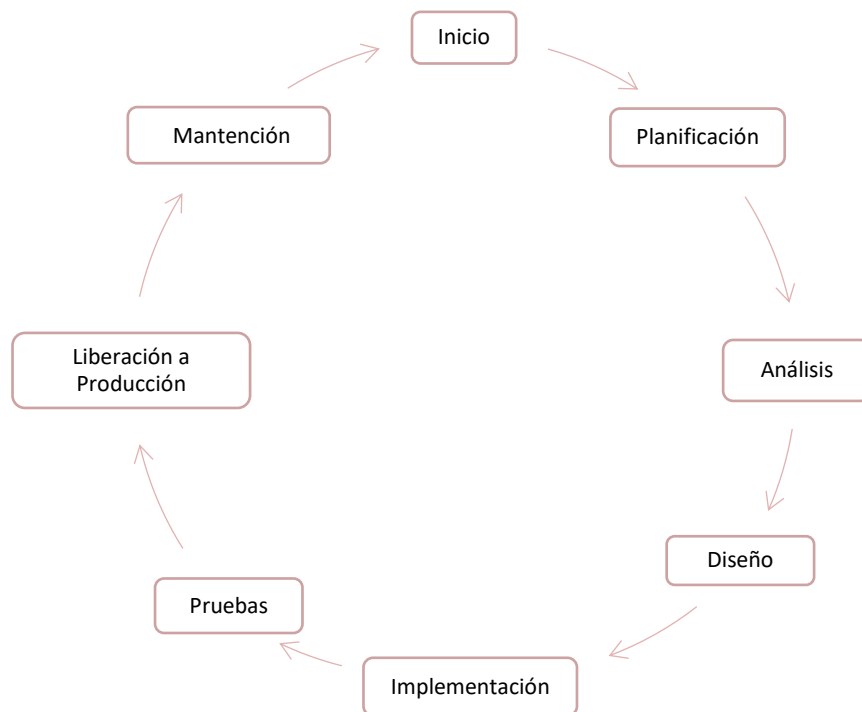
Implementación: conjunto detallado de actividades que coordina la realización del producto. Se inicia la decodificación de algoritmos y estructuras de datos según corresponda. Una vez desarrollado el producto se debe probar en test antes del paso a producción.

Puesta en Producción: se instala en producción y se presenta al usuario y/o cliente teniendo la aceptación (proceso de validación) si cumple con los requerimientos solicitados y sin errores. Esta etapa es la más compleja debido a que se presentan dificultades en la práctica de errores lo que incurre en gastos y tiempo.

Control: es la etapa transversal del ciclo, donde se controla cada fase del proyecto en curso revisando que no difiera del requerimiento y costos asociados, aplicándose acciones correctivas si fuese necesario al existir alguna desviación. Esta etapa es desde inicio a fin y debe incluir liderazgo, capacitación y documentación.


Se debe recordar que durante el desarrollo del proyecto existen entregables por fase a definir con usuario.

Se muestra a continuación el ciclo de vida en detalle:



CICLO DE VIDA EN DETALLE DE PROYECTO DE SOFTWARE. (FUENTE: INGRID YÁÑEZ, 2018)

Es relevante preguntarnos; **¿Por qué es importante conocer el proceso de ciclo de vida de un software?** Porque, al conocerlo, se podrá:

-
- 
- ✓ Predecir riesgos futuros
 - ✓ Reducir riesgos e incertidumbre
 - ✓ Conocer en que partes del proyecto se debe poner mayor énfasis.
 - ✓ Diseñar los más adecuados indicadores de seguimiento.
 - ✓ Controlar las fases del proyecto.

4. Estructura de la gestión de requerimientos

4.1 La pirámide

Como bien se sabe, los requisitos pueden ser derivados de diversas indoles, origen o características de acuerdo a lo que el usuario defina. Por ello, existe un orden de solicitud desde que el usuario indica su problema hasta llegar en detalle al requerimiento final, este proceso de solicitud se explica en forma de pirámide debido a que va en escala el nivel de detalle del requisito, es decir, entre menor sea el nivel de la etapa mayor detalle del requisito existe, pero esto siempre lo define el usuario solicitante del negocio el nivel de detalle a dar de acuerdo a su solicitud.

La pirámide cuenta con la siguiente estructura como se muestra en la figura:



PIRÁMIDE DE ESTRUCTURA DE GESTIÓN DE REQUERIMIENTOS. (FUENTE:
[HTTP://WWW.LIDERDEPROYECTO.COM/MANUAL/LOS_REQUERIMIENTOS.HTML](http://www.liderdeproyecto.com/manual/los_requerimientos.html)**)**



4.2 Etapas de solicitud

En las etapas de solicitud se definen:

- Necesidad: un interesado demanda un requerimiento.
- Característica: un servicio proporcionado por el sistema, por lo general formulado por un analista de negocios.
- Caso de uso: una descripción del comportamiento del sistema descrito como una secuencia de acciones.
- Requisito complementario: otro requisito (generalmente no funcional) que no puede ser contemplado en los casos de uso.
- Escenario: una secuencia específica de acciones o una ruta de acceso específica a través de un caso de uso. Ayudan a derivar en casos de uso a partir de los casos de prueba y facilitan el diseño e implementación a través de los casos de uso.
- Caso de prueba: una especificación de las entradas necesarias para una prueba, las condiciones de ejecución y resultados esperados. Tiene el papel de comprobar si los casos de uso derivados de los casos de prueba y los requisitos complementarios se aplican correctamente.



5. Tipos de problemática según el tipo de organización

Son variados los problemas que se pueden presentar dependiendo de la organización. La complejidad de los procesos al momento de realizar el levantamiento del requerimiento dependerá del área que analice y el proceso que desarrolle dentro de la organización. Es por esto que se debe tener algunas consideraciones para evitar todo tipo de problemáticas:

- Entender el Problema del negocio
- Entender el ambiente de la operación
- Trabajar con el personal especializado, conocedor de los procesos.
- Relación fluida con el cliente, comunicación constante.

6. Técnicas de obtención de requerimientos funcionales y no funcionales

Ahora, veremos cuál es la mejor de forma de poder llegar a definir y realizar el levantamiento de requerimientos respecto a las necesidades del usuario por medio de actividades y técnicas de obtención y posterior documentación y aceptación de requerimientos.

6.1 ¿Cuáles son las actividades de ingeniería de requerimientos?

Cuando un usuario y/o cliente realiza una solicitud de desarrollo de software, es difícil lograr el éxito, debido a que se debe lograr satisfacción del cliente, finalizando el proyecto en tiempo, forma (alcance definido) y dentro del presupuesto inicial.

Por este escenario, la ingeniería de requerimientos es parte importante del trabajo para el responsable del proyecto debido a que lo ayudan a comprender el requerimiento definiendo las actividades involucradas, siendo su meta entregar una especificación del requerimiento de software correcta y completa, para esto se debe analizar la necesidad, confirmar la viabilidad, negociar una solicitud razonable, especificar la solución de forma clara y sin supuestos, validar la especificación y gestionar los requerimientos y estos se transformen en un sistema operacional disminuyendo riesgos y sobrecostos.

Las actividades que debe ejecutar el responsable del proyecto son:

Recolección: se analiza y se descubre en conjunto al cliente el problema que el sistema debe resolver, los servicios que debe prestar y las restricciones que debe tener el sistema.

Análisis: en esta fase se estudia sobre los requerimientos del cliente los problemas existentes y como solucionarlos. Generalmente se ejecuta este paso después de realizar el bosquejo del documento de requerimientos. Se debe discutir temas con el equipo, se ven alternativas de solución, se investiga y se reúnen con cliente para discutir requerimientos.



Especificación: se documentar requerimientos del cliente, detallando cada proceso del requerimiento, la necesidad, mejora a ejecutar y todos los detalles. Esta fase se realiza en conjunto con el análisis aplicando notaciones de modelado como el UML (lenguaje modelado unificado) para levantamiento de caos de uso para la obtención de requerimientos.

Validación: en esta etapa se ratifica el requerimiento, es decir que este sea consistente, aceptable y completo a lo que se debe implementar por medio de una descripción.

Por medio de este proceso, se obtiene un documento de especificaciones estructurado de los requerimientos, siendo el documento final y de carácter formal.

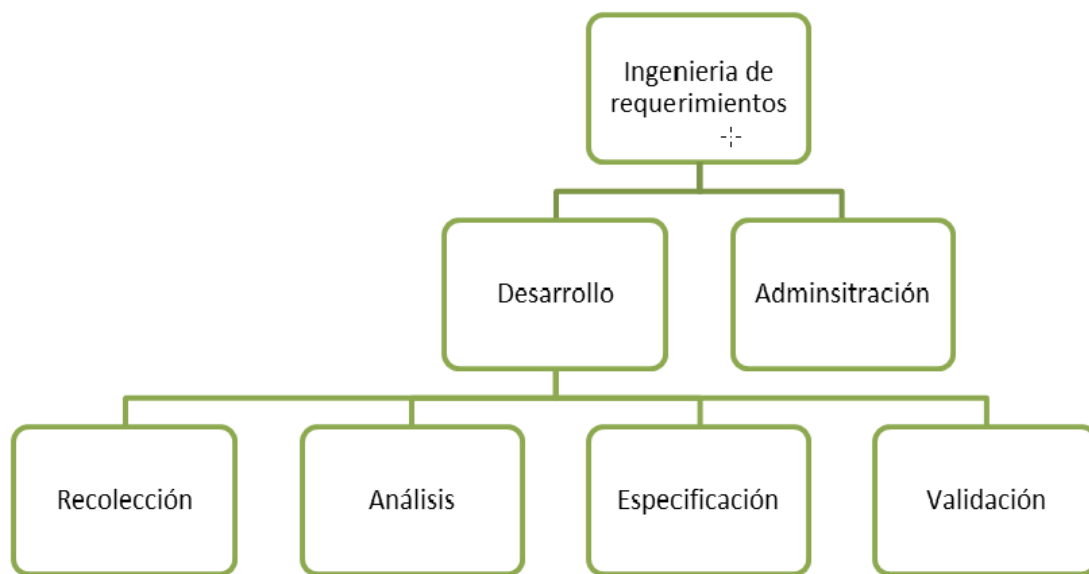


DIAGRAMA DE INGENIERÍA DE REQUERIMIENTOS. (FUENTE: INGRID YÁÑEZ, 2018)

6.2 Técnicas de obtención de requerimientos

La obtención de requerimientos involucra a diferentes personas que participan en el proceso, lo que conlleva a dificultades para definir de forma correcta el requerimiento, por lo que para poder comunicar de forma clara los requerimientos se requieren técnicas como se describe a continuación:

Entrevistas: por medio de esta técnica se logra obtener información cualitativa, es decir opiniones y descripciones. Es necesario que el entrevistador por medio de una conversación estructurada realice preguntas abiertas en forma de plantear una conversación en un tiempo no mayor a 2 horas y el entrevistador previamente deberá prepararse por medio de investigación y documentación la situación de la organización y de los aspectos relevantes.



Desarrollo de Prototipos: consiste en realizar un demo de la aplicación pedida, esta técnica generalmente se usa cuando la aplicación no está definida o por dudas del usuario en no saber que se quiere. La construcción de prototipos incrementa los costos en las etapas iniciales de un proyecto, pero esto se recupera en etapas posteriores gracias al mejor entendimiento de los requerimientos por parte de los desarrolladores. En algunos casos también se utiliza como un medio para formalizar la aceptación previa del cliente de los requisitos del proyecto.

Observación: este método consiste en observar la forma en que llevan a cabo los procesos y verificar si se siguen todos los pasos especificados.

Estudio de comunicación: por medio de documentación ya sean manuales, procedimientos, reportes se analizan principalmente para obtener un dominio de la operación y el vocabulario que se utiliza. Ya que por medio de esto no se puede saber cómo realmente se desarrollan las actividades y donde se encuentra el poder de tomar decisiones.

Cuestionario: por medio de cuestionarios se reúne información de forma estandarizada asegurándose el encuestador de que los datos proporcionados sean los requeridos. El encuestado debe asegurar que las personas seleccionadas para contestar el cuestionario sean los indicados y parte del proceso.

Tormenta de Ideas (Brainstorming): reunión de cuatro a diez personas donde se realiza lista de ideas sin juzgarlas y después tomar todas estas ideas y analizar en detalle cada propuesta.

Escenarios: estos se utilizan para documentar el comportamiento del sistema cuando se le presentan eventos específicos. Cada evento de interacción distinto, o la selección de un servicio del sistema, sedo documentos como un escenario de eventos distinto. Los escenarios de eventos incluyen una descripción del flujo de datos y las acciones del sistema, y documenta las excepciones que pueda surgir. Las convenciones para los diagramas utilizados en los escenarios de eventos.

Etnografía: es una técnica de observación que se utiliza para entender los requerimientos sociales y organizacionales.

7. Procesos incrementales de mejora

Tradicionalmente, la mejora de procesos de software se realiza en un solo ciclo. Esto es más sencillo para los que la administran, pero se complica para el equipo de mejora. No obstante, la mejora incremental permite al equipo de mejora y a la gerencia identificar resultados a corto plazo, con el inconveniente de que genera mayor complejidad en la administración. Sin embargo, los beneficios bien valen la pena.

Según la página especializada en desarrollo de SW, SG Software Guru, el proceso incremental de mejora tiene como objetivo asegurar la implantación de la capacidad de los modelos de software a través de ciclos de vida incrementales orientados a organizaciones y objetivos de negocio como se presenta en la figura 1. Esta figura ilustra la arquitectura del proceso, el cuál provee un acercamiento disciplinado para la asignación de tareas y responsabilidades en la organización. Como podrán apreciar, estamos tomando la misma idea del ciclo de vida del Proceso Unificado, y aplicándola a la mejora de procesos.

La gráfica muestra el esfuerzo variado a través del tiempo. Por ejemplo, en iteraciones tempranas, el esfuerzo se encuentra concentrado en el desarrollo de la iniciativa y en las iteraciones tardías en la entrega. Esta contiene dos dimensiones:

- El eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida del proceso claramente. La primera dimensión ilustra el aspecto dinámico del desempeño del proceso el cual está expresado en términos de fases, el ciclo de vida incremental y los entregables.
- El eje vertical representa a las disciplinas que de manera lógica agrupa las actividades por naturaleza. Esta segunda dimensión representa el aspecto estático de la descripción del proceso en términos de componentes, disciplinas, actividades, flujos de trabajo, artefactos y roles. (Ver Figura 1)

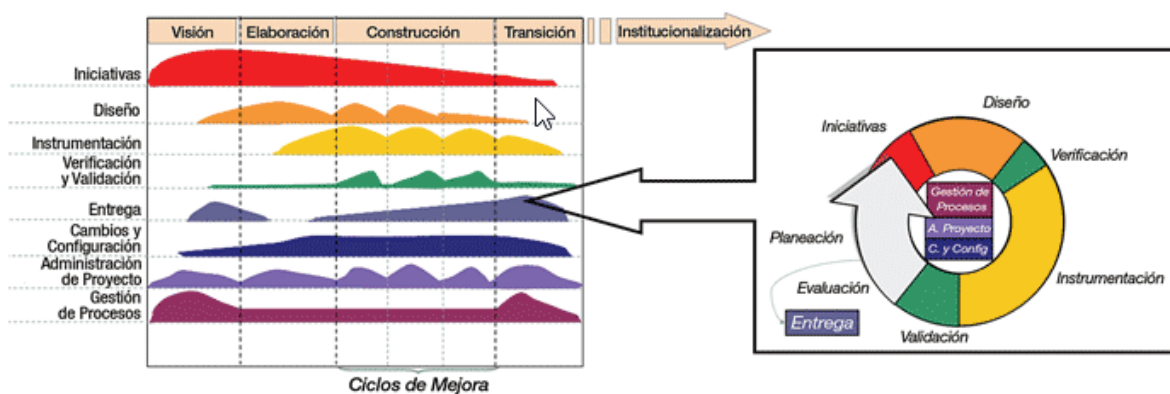


Figura 1. Modelo de ciclo de vida para la mejora de procesos incremental



Conclusiones

Durante esta semana, pudimos revisar el concepto de requerimiento, considerando su utilidad para describir, por medio de un documento formal o estipulado en contrato, los atributos, capacidades, características y/o cualidades que necesita cumplir el software que buscamos desarrollar, de modo de lograr lo que el cliente espera en el plazo determinado y dentro del presupuesto.


Revisamos los tipos de requerimiento que existen, además de las etapas del proyecto que desarrollamos y por qué es tan importante determinar los requerimientos al comienzo de este, además de los tipos de problemática que puede aparecer y las formas de levantar requerimientos.

Para finalizar, nos adentramos en los procesos incrementales de mejora, entendiéndolo como un ciclo constante para el mejoramiento de nuestros productos.

Bibliografía

- Young, Ralph R. (Abril 2002). Recommended Requirements Gathering Practices. STSC
- Gottesdiener, Ellen. (Marzo 2008). Good Practices for Developing User Requirements. STSC
- Zielczynski, Peter. (2008). Requirements Management Using IBM® Rational® RequisitePro®, IBM Press. PMBOK® Guide
- Bittner, K. (2000) Why Use Cases Are Not Functions. USA.
- SG Software Guru <https://sg.com.mx/>

Enlaces y material multimedia

MÓDULO: Introducción a requerimientos y modelos de negocios		Unidad: Requerimientos de software
Recurso	Descripción	
 Video	En el siguiente enlace, encontrarás un video para profundizar el análisis de requerimientos: https://www.youtube.com/watch?v=50aSAUFrALk	