Capítulo 4. ACTORES EN LA CREACIÓN DEL SOFTWARE

4.1.- Concepto

El desarrollo de software es una actividad que, dada su complejidad, debe desarrollarse en grupo. Además, esta actividad requiere de distintas capacidades, las que no se encuentran todas en una sola persona. Por ello, se hace necesario formar el grupo de desarrollo con las personas que cubran todas las capacidades requeridas. Cada una de esas personas aportará al grupo parte del total de las capacidades necesarias para llevar a cabo con éxito el desarrollo.

Hay que señalar que es posible que no se requieran todos los roles en un desarrollo. Eso dependerá del tamaño y del tipo del desarrollo. Por ejemplo, el desarrollo de un sistema de información de gran tamaño requerirá más roles que uno de menor tamaño. Es posible también que una persona realice las labores de más de un rol al mismo tiempo.

Esto, sobre todo en proyectos de desarrollo de software más pequeños. La metodología de desarrollo también incide en los roles o actores del equipo (se verá más adelante).

4.2.- Cliente / Usuario

Típicamente el proceso de desarrollo de un software comienza con un planteamiento de un problema en una organización o empresa, que solicita a un equipo de desarrollo la creación de una solución para dicho problema. Por esto, del lado de la empresa solicitante existen dos actores: el cliente y el usuario.

- El cliente: Un cliente es aquella persona responsable de llevar a cabo el buen desempeño del proyecto, por parte de la empresa que contrata el desarrollo. El cliente debe representar los derechos y asumir los deberes de dicha empresa ante el equipo de desarrollo. Por lo tanto, el cliente debe estar presente en todas las fases del desarrollo del producto, y realizar todas las actividades que se esperan de él, tales como la aceptación provisional y final del producto. Quien solicita la creación del programa, quien plantea el problema a resolver y quien paga.

En el desarrollo de software nos interesa tener clientes comprometidos. Es vital para el éxito del proyecto. Un cliente comprometido es aquel que participa en todas las etapas del proyecto, compartiendo deberes y responsabilidades. Entre sus tareas están:

- Liderar el proyecto de software cuando la organización así lo requiere.
- Debe conocer las distintas etapas y roles en la construcción de software.
- Definir los objetivos del proyecto negociando con sus clientes las características que le afecten.
- Definir y priorizar requisitos.
- Revisar y aprobar documentos en forma responsable.
- Difundir el estado del proyecto al resto de su ámbito de trabajo.
- Entregar los recursos necesarios para la realización del proyecto.

- El usuario: los usuarios corresponden a las personas que están operando día a día un sistema de software. Es la persona que conoce el problema, y utiliza la herramienta computacional para apoyar su trabajo. Un cliente y un usuario no siempre son lo mismo, ya que es posible que el cliente no opere el sistema de información.

4.3.- Administrador Del Proyecto

Es la persona que administra y controla los recursos asignados a un proyecto, con el propósito de que se cumplan correctamente los planes definidos. Los recursos asignados pueden ser recursos humanos, económicos, tecnológicos, espacio físico, etc. En un proyecto, siempre debe existir un administrador. No obstante, un administrador puede dirigir más de un proyecto.

Generalmente, el administrador del proyecto es quien mantiene contacto directo con el cliente, ya sea para la negociación, revisión del avance del proyecto y aprobación de módulos terminados.

Algunos de los objetivos de un administrador de proyecto son los siguientes:

1. Tener el producto “a tiempo”, “bajo presupuesto” y con los requisitos de calidad definidos.
2. Terminar el proyecto con los recursos asignados.
3. Coordinar los esfuerzos generales del proyecto, ayudando a cada uno de sus integrantes a cumplir sus objetivos particulares. Al final, se cumplirá el objetivo general.
4. Cumplir con éxito las diferentes fases de un proyecto, utilizando herramientas de administración.
5. Cumplir con las expectativas del cliente.

4.4.- Analista / Programador

El administrador del proyecto organiza un equipo de analistas y programadores que puedan convertir el planteamiento del problema del cliente en un software utilizable por los usuarios. Cada uno de estos roles tiene gran importancia en el proceso. A continuación se describe cada uno de estos:

- analista: en un proyecto de construcción de software se refiere a la especificación de un problema como la suma de subproblemas de menor complejidad. Como el experto en el problema es el cliente, se hace necesario trabajar junto a él para realizar la especificación correctamente. Los miembros del grupo que trabajan con el cliente para realizar el análisis y especificación del sistema a construir son precisamente los analistas. Una de las razones más frecuentes del fracaso de un desarrollo de software es la de realizar un análisis pobre. Debido al insuficiente esfuerzo dedicado a conocer y especificar el sistema que desea el cliente, los desarrolladores construyen sistemas que no cuentan con las características que el cliente desea.

- programador: quien desarrolla o plasma la solución planteada. Los programadores deben convertir la especificación del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de software de apoyo a la programación. El éxito del desarrollo de software depende grandemente de conocimiento. Este conocimiento no sólo corresponde a habilidades de programación y de administración de proyectos, sino que a una percepción y entendimiento de los últimos desarrollos de la industria del software.

4.5.- Otros Actores

En proyectos de desarrollo medianos o grandes existen más actores. Entre ellos están:

- Arquitecto de software.
- Administrador de base de datos (DBA).
- Probador (Tester).
- Documentador.

Capítulo 9. REQUISITOS DE SOFTWARE

9.1.- Concepto

Conocer qué tiene que hacer el software es el punto de partida y la parte más importante del proceso de desarrollo. Si los desarrolladores no conocen de forma precisa el problema a resolver, es poco probable que se obtenga una solución correcta y útil. Un requerimiento de Software expresa las necesidades o restricciones de un producto de software que contribuyen a la solución de un problema del mundo real.

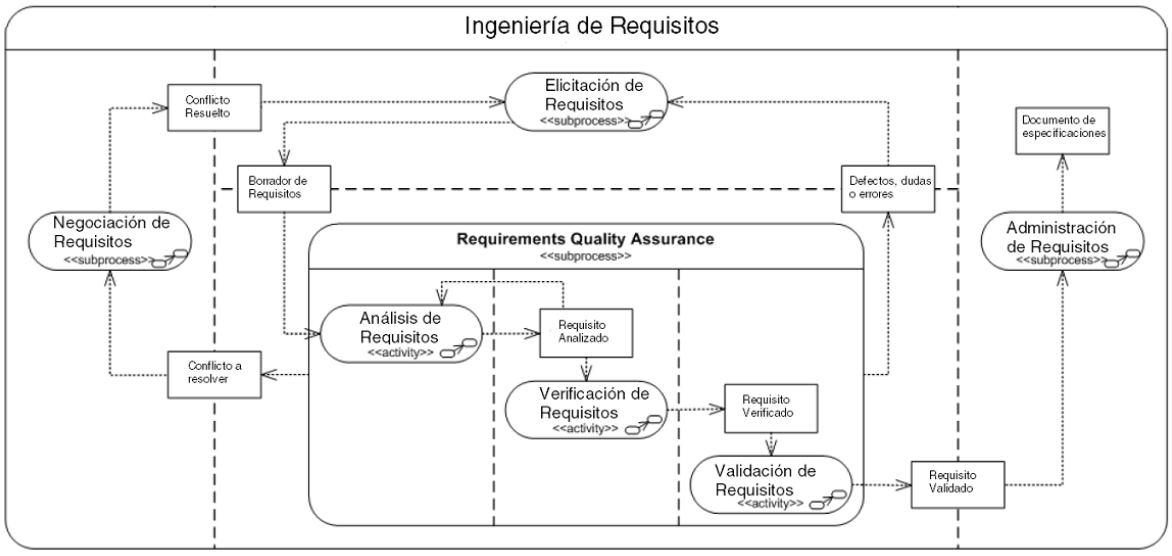
Así pues, la correcta obtención de los requisitos es uno de los aspectos más críticos de un proyecto software, independientemente del tipo de proyecto que se trate, dado que una mala captura de los mismos es la causa de la mayor parte de los problemas que surgen a lo largo del ciclo de vida

El área de conocimiento de requerimientos de software abarca lo concerniente a la elicitación, análisis, especificación y validación de requerimientos de software así como su administración durante el ciclo de vida de un producto de software.

La forma en la cual los usuarios, los procesos y los dispositivos funcionan en las organizaciones generalmente es compleja. Por extensión, los requerimientos de software particulares son generalmente una combinación compleja de varias personas

en diferentes niveles de la organización, quienes de una forma u otra están involucrados o conectados con el entorno en el cual el software operará.

El proceso de requerimientos involucra varias actividades. En el siguiente esquema se muestran las actividades y la relación que hay entre estas:



9.2.- Requisitos Funcionales

Los requerimientos funcionales describen las funciones que el software debe poseer o ejecutar. Por ejemplo: el sistema debe validar el correo electrónico de los clientes, el sistema debe enviar un correo electrónico al cliente una vez hecha la compra, etc. Son conocidos también como características o funciones. Un requerimiento funcional también puede ser descrito como un conjunto de pasos de prueba que pueden ser escritos para validar el comportamiento del software.

Ejemplos de requerimientos funcionales son:

- el sistema no debe permitir inscribir en una sección de un curso una cantidad de participantes superior a la capacidad del salón donde verán clase.
- el sistema no debe permitir el acceso a usuarios que no se hayan autenticado.
- los clientes de la empresa pueden registrarse remotamente y hacer solicitudes de préstamos.

- las solvencias pueden ser impresas en papel o guardadas como documentos pdf.

9.3.- Requisitos no Funcionales

Los requerimientos no funcionales son aquellos que actúan como restricciones de la solución. Los requerimientos no funcionales son conocidos también como restricciones o requerimientos de calidad. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, 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, entre otros.

Los requerimientos no funcionales suelen referir a:

- Usabilidad: factores humanos, ayuda, documentación.
- Portabilidad: donde debe ejecutarse, podrá ser instalado por los usuarios, etc.
- Confiabilidad: frecuencia de fallas, tiempo de recuperación.
- Performance: tiempo de respuesta, tasa de procesamiento, precisión, capacidad de carga.
- Soportabilidad: adaptabilidad, mantenibilidad, configurabilidad, internacionalización.
- Seguridad: limitaciones de acceso, donde puede y donde no puede ejecutarse.
- Interfaces: restricciones en la comunicación con sistemas externos
- Restricciones: en el uso de sistemas o paquetes existentes: plataformas, Lenguajes de programación, Ambientes de desarrollo, Herramientas, manejadores de bases de datos, etc.