

**Universidad de las Ciencias Informáticas**

**Facultades 5 y 7**



**TRABAJO DE DIPLOMA PARA OPTAR POR EL  
TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS**

**Título:** Gestión de requisitos en el desarrollo del software.

**Autoras:**

Yilitbet Cuellar Font  
Maylen Capote Vázquez

**Tutora:**

Ing. Rosalía Cué Delgado

**Ciudad de La Habana, junio de 2007**

## AGRADECIMIENTOS

*Quiero agradecer a todos los que de una forma u otra han contribuido a mi formación profesional a lo largo de estos cinco años:*

*A mi familia, por servirme de apoyo en los momentos más difíciles.*

*A mi tutora, por toda la ayuda que me ha brindado, por sus orientaciones y por la confianza depositada.*

*A mis profesores, por todo lo que me han enseñado.*

*A mis amigos, por estar siempre a mi lado cuando los he necesitado.*

Yilitbet

*A Dios, por escucharme y guiarme siempre por un mejor camino.*

*A la Revolución, por crear un sueño tan hermoso como lo es nuestra universidad.*

*A mi tutora, por su ayuda y dedicación en nuestro trabajo, enseñándonos siempre más.*

*A mis padres por lograr formar en mí: sensibilidad, fuerza y convicción de luchar para lograr más de mí cada día.*

*A los amigos, que tocaron mi corazón con amor, compasión, ánimo y consuelo para avanzar hasta aquí, Yanelys, Maria Isabel, Yara, Anaima, Kenia, Yeni, Mara, Dulce, Karelys, Lily, Yaritza, Yani y Yosbel.*

*A todos, muchas gracias; sin ustedes no hubiese sido posible la terminación de este trabajo.*

Maylen

**DEDICATORIA**

*A mi querida madre y a mi abuela, por sus noches de desvelo y su amor incondicional.*

*A mi padre, por enseñarme cada día el difícil arte de transitar por la vida.*

*A mi hermana del alma, tú sabes cuanto te quiero.*

*Yilitbet*

*A mi mamá, por ser inspiradora de mi vida, y siempre impulsar a mi superación profesional, ofreciéndome aliento constante para lograrlo.*

*A mi hermana, a quien quiero mucho.*

*A mi familia, por apoyarme durante todo este tiempo.*

*A mi novio, por ser amigo, compañero, alma gemela, paciente y fiel hasta el final.*

*Maylen*

**DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de este trabajo en el cual he utilizado información y documentación que es propiedad de la empresa SOFTEL lo cual está sujeto a un acuerdo de confidencialidad. Pongo a disposición de la Universidad (UCI) todo aquello que no comprometa dicho acuerdo.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_

\_\_\_\_\_

Maylen Capote Vázquez

\_\_\_\_\_

Yilitbet Cuellar Font

\_\_\_\_\_

Ing Rosalía Cué Delgado

## **RESUMEN**

En la actualidad, diversas compañías de desarrollo de software tienen una gestión de requisitos ineficiente debido a que utilizan algún enfoque informal de gestión de requisitos o ignoran los posibles riesgos en el avance del proyecto si no es bien definida. El objetivo de este trabajo es definir un proceso para la gestión de requisitos en los proyectos productivos de la universidad. Este proceso se podrá aplicar a proyectos de cualquier tamaño, independiente de la metodología utilizada.

En la investigación se desarrollaron entrevistas a los líderes de algunos proyectos productivos seleccionados con el objetivo de determinar el dominio de la gestión de requisitos, determinándose las actividades que se realizaban o no, y tomando como base el concepto de ingeniería y gestión de requisitos definidos por Rogers S. Pressman en su libro "Ingeniería de Software. Un enfoque práctico". Como resultado de la investigación se propone un proceso para la gestión de requisitos, que ayuda a la verificación de las deficiencias encontradas en las entrevistas en este tema.

De esta forma proyectos de desarrollo con características similares a los de la Universidad de las Ciencias Informáticas (UCI) podrán tomar como guía este trabajo, aprendiendo de las herramientas y las actividades de gestión de requisitos a utilizar. Finalmente, otra de las contribuciones de este trabajo es proveer evidencia empírica de la gestión de requisitos, ante la falta de casos reales en la literatura.

## **Palabras claves**

Ingeniería de requisitos, gestión de requisitos, trazabilidad

**INDICE**

INTRODUCCIÓN ..... 1

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA. .... 4

    1.1 Origen de la ingeniería de requisitos de software..... 4

    1.2 Principales factores del éxito y fracaso de proyectos de software..... 5

    1.3 Impacto de la ingeniería de requisitos de software. .... 6

    1.4 Ingeniería de requisitos. .... 6

    1.5 El concepto de requisito. .... 7

    1.6 El concepto de gestión de requisitos..... 8

    1.7 El concepto de trazabilidad. .... 8

    1.8 Gestión de requisitos en eXtreme Programming (XP)..... 8

    1.9 Gestión de requisitos en SCRUM. .... 11

    1.10 Gestión de requisitos en Microsoft Solution Framework (MSF). .... 12

    1.11 Gestión de requisitos en Rational Unified Process (RUP). .... 14

CAPÍTULO 2: CARACTERIZACIÓN Y DIAGNÓSTICO DE LOS PROYECTOS PRODUCTIVOS. .... 16

    2.1 Métodos empleados. La entrevista ..... 17

        2.1.2 Ventajas de la entrevista ..... 17

        2.1.3 Selección de entrevistados..... 17

        2.1.4 Herramientas utilizadas. .... 17

    2.2 Gestión de requisitos en proyectos productivos. .... 18

    2.3 Resultados de la encuesta. .... 18

        2.3.1 Tamaño del proyecto ..... 18

        2.3.2 Metodologías utilizadas ..... 19

        2.3.3 Comportamiento de la gestión de requisitos..... 20

        2.3.4 Tipos de requisitos..... 21

        2.3.5 Asignarle atributos a los requisitos..... 23

        2.3.6 Trazabilidad de los requisitos ..... 27

        2.3.7 Herramientas en la gestión de requisitos ..... 28

        2.3.8 Proceso de gestión de requisitos. .... 29

CAPÍTULO 3: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	31
3.1 Procesos. Elementos fundamentales.....	31
3.2 Concepto de gestión.....	33
3.3 La utilización de procesos en los proyectos productivos.....	33
3.4 El proceso de gestión de requisitos.....	35
3.5 Principales actividades del proceso de gestión de requisitos.....	35
3.6 Mapa del proceso.....	36
3.7 Descripción del proceso de gestión de requisitos.....	36
3.8 Descripción de las actividades propuestas.....	42
3.9 La gestión de requisitos en la práctica.....	50
3.10 Herramientas para la gestión de requisitos.....	50
3.11 Funcionalidades de las herramientas de gestión de requisitos.....	51
3.12 Herramientas de gestión de requisitos en el mercado.....	52
CONCLUSIONES.....	57
RECOMENDACIONES.....	58
BIBLIOGRAFÍA CITADA.....	59
ANEXOS.....	64
GLOSARIO DE TÉRMINOS.....	68

# **INTRODUCCIÓN**



## **INTRODUCCIÓN**

*“La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan dificultosa como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan dificultosa de rectificar posteriormente”. (FREDERICK P. BROOKS 1995)*

En la actualidad, el software se ha considerado el “alma máter” de muchos sistemas productivos. Las compañías dependen cada vez más de su realización, necesitan construir mejores software en menos tiempo y a menor costo. Este ha revolucionado la manera de operar de los negocios, las industrias y hasta de los individuos, al determinar los estándares de eficiencia personal y efectividad organizacional en todo el mundo.

Debido a la creciente participación del software en el mundo actual y de los avances producidos, se puede afirmar que hoy en día el desarrollo del software aún no responde a las exigencias de estos tiempos. Las necesidades y expectativas de los clientes y usuarios no son captadas satisfactoriamente.

Uno de los problemas que hay en la actualidad es que los proyectos no llegan a cumplir sus objetivos, una de las causas es por el uso de una inadecuada ingeniería de requisitos.

Dentro del proceso de la ingeniería de requisitos, la gestión de requisitos es fundamental para el éxito de los proyectos que se realizan habitualmente en cualquier sector de actividad, ya que saber lo que se va a hacer, cómo se va a hacer y dejar constancia de ello revierte en realizaciones de proyectos con mejor calidad, plazo y costo, garantizando que no se produzcan problemas ni desviaciones en su realización.

En los últimos años Cuba se ha estado insertando en el campo de la informática, la Revolución Cubana puso en marcha la gran idea de informatizar la sociedad cubana dando surgimiento a la creación de La Universidad de las Ciencias Informáticas (UCI), en el año 2002.

La UCI, no está excluida a las deficiencias que existen durante el proceso de desarrollo de software. Debido a esta dificultad se hace necesario definir un proceso de desarrollo para la realización de los proyectos productivos de la universidad, formando parte de él, el proceso de ingeniería de requisitos y dentro de este último, el conjunto de actividades que son necesarias para poder gestionarlos. Lo que queda concretado en el problema científico de la investigación que es:

¿Cómo realizar un proceso de gestión de requisitos que asegure la calidad del producto de software?

El objeto de estudio lo constituye la ingeniería de requisitos en el desarrollo del software, y el campo de acción en el cual se enmarca esta investigación es en el proceso de gestión de requisitos en los proyectos de desarrollo de software de la UCI.

Teniendo en cuenta el problema anterior, el objetivo general de la investigación está encaminado a definir un proceso para la gestión de requisitos que garantice la calidad del producto de software en los proyectos productivos de la UCI.

Para guiar la investigación, se plantea como idea a defender que: la definición de un proceso para la gestión de requisitos que permita crear las dependencias entre los requisitos y asignarle atributos para contribuir con la gestión del proyecto y así asegurar la calidad del producto de software en los proyectos productivos de la UCI.

Para el logro de este objetivo se trazaron las siguientes tareas:

- ◆ Definir los conceptos necesarios relacionados con la ingeniería de requisitos.
- ◆ Analizar el proceso de ingeniería de requisitos en las metodologías más usadas en la actualidad.
- ◆ Realizar entrevistas a los diferentes proyectos productivos con el propósito de conocer cómo realizan la gestión de requisitos.
- ◆ Definir un proceso para la gestión de requisitos.
- ◆ Proponer plantillas de trabajo a los artefactos que se generan al realizar las actividades.

- ◆ Estudiar las herramientas fundamentales que cumplan con el proceso descrito para la gestión de requisitos.

El trabajo fue estructurado en tres capítulos que a continuación se presentan:

### **Capítulo I: Fundamentación Teórica.**

El propósito de este capítulo es la formalización de todos los conceptos asociados al tema y que son necesarios para la comprensión de lo que se describe en el resto del trabajo, como los conceptos relacionados con los requisitos, su ingeniería, gestión, entre otros. Además se tratarán cuestiones como el origen e impacto de la ingeniería de requisitos. También se realizará un estudio del tema en las diferentes metodologías (ágiles y tradicionales), más utilizadas, con el objetivo de obtener sus mejores prácticas en la gestión de requisitos.

### **Capítulo II: Caracterización y diagnóstico de los proyectos productivos.**

El propósito de este capítulo es analizar los posibles niveles de aplicación en la UCI de la gestión de requisitos a partir de entrevistas realizadas en los proyectos productivos, determinando los principales problemas que existen, dándoles un diagnóstico previo.

### **Capítulo III: Descripción de la solución propuesta.**

En este capítulo se detallará la solución que radica en la definición de un proceso que permita gestionar los requisitos de software. Además de describir minuciosamente cómo se llevarán a cabo cada una de las actividades. Así como la importancia del uso de las herramientas para su ejecución.

Finalmente se mostrarán las conclusiones y recomendaciones, así como los anexos que permiten una mejor comprensión del contenido.

# CAPÍTULO 1

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.**

Los modelos de los procesos de la Ingeniería de Requisitos (IR), a pesar de su evolución, aún presentan carencias. Por tanto, para obtener un producto de calidad, se requiere una mejora en los procesos de IR. El objetivo de este capítulo es la caracterización del estado actual del problema de investigación, mediante la formalización de los conceptos asociados al tema que son necesarios para la comprensión de lo que se describe en el resto del trabajo, como son los conceptos de: ingeniería de requisitos, requisito, gestión de requisitos y trazabilidad. Además se abordan cuestiones como el origen e impacto de la ingeniería de requisitos y se realiza un estudio de dicho proceso en las metodologías de desarrollo más usadas en la actualidad.

### **1.1 Origen de la ingeniería de requisitos de software.**

El término ingeniería de requisitos de software se viene tratando desde hace tiempo atrás cuando por primera vez en la conferencia sobre desarrollo de software patrocinada por el Comité de Ciencia de la OTAN celebrada en Garmisch, Alemania, en octubre de 1968 Fritz Bauer utilizó por primera vez el término ingeniería del software. Cuando, gracias a los avances en la tecnología, aumenta la capacidad de memoria de los ordenadores, se reducen los costos de desarrollo y mantenimiento del hardware, y comienzan a comercializarse los primeros ordenadores creciendo la demanda de software y generándose la crisis del software, término utilizado para designar la gran cantidad de problemas que presentaba, y aún presenta, el desarrollo de software, así como el alto índice de fracasos en los proyectos de desarrollo.

Fue entonces cuando por primera vez surgió la necesidad de aplicar los principios de otras ingenierías al desarrollo de software. En estudios realizados se ha demostrado que casi un tercio de los proyectos de desarrollo de software se cancelan durante su desarrollo y que la gran mayoría presenta graves desviaciones respecto a plazos y presupuestos iniciales. Atribuyéndoles entre sus principales fuentes de frustración la incapacidad para resolver las necesidades de los usuarios, como consecuencia fundamental de una pobre ingeniería de requisitos.

Ante esta situación de fracaso en la que se encontraban los proyectos software la solución era: construir software de forma similar a como se construía hardware, aviones, barcos, puentes o edificios, es decir, aplicar los métodos de la ingeniería a la construcción de software.

Debido a que las herramientas para construir software han evolucionado enormemente, se sigue produciendo software que no es satisfactorio para los clientes y usuarios, lo cual indica que los principales problemas que han dado origen a la crisis del software residen en las primeras etapas del desarrollo, cuando hay que decidir las características del producto software a desarrollar.

Todas estas circunstancias han convencido a la gran parte de la comunidad de la ingeniería del software de la necesidad, cada vez mayor, de una ingeniería de requerimientos.

### **1.2 Principales factores del éxito y fracaso de proyectos de software.**

Investigaciones realizadas en Estados Unidos, por la Oficina de Cuentas del Gobierno Norteamericano (*Government Account Office*, GAO) (DIV 1979), por el Grupo Standish en 1995 (GROUP 1995) y en 1996 en Europa, llevada a cabo por proyecto ESPITI (European Software Process Improvement Training Initiative) (INSTITUTE 1996) demostraron la existencia de problemas en el desarrollo de software a nivel general. Las dificultades se centraban fundamentalmente en los temas relacionados con la ingeniería de requisitos, en las temáticas de especificación, gestión y documentación de los requerimientos de software. También se detectaron los principales factores de éxito y fracaso de este estudio los que se revelarán a continuación.

Principales factores de éxito:

- ◆ Implicación de los usuarios.
- ◆ Apoyo de los directivos.
- ◆ Enunciado claro de los requerimientos.

Principales factores de fracaso:

- ◆ Falta de información por parte de los usuarios.

- ◆ Especificaciones y requerimientos incompletos.
- ◆ Especificaciones y requerimientos cambiantes.

### **1.3 Impacto de la ingeniería de requisitos de software.**

Aunque la ingeniería de requisitos de software no es el concepto que resolverá todos los problemas, promete tener un impacto altamente positivo en la reducción del tiempo del proyecto y la disminución de los recursos implicados, lo que facilitaría la reutilización de requisitos cuando fuese necesario. A continuación, se detallan algunos de los posibles beneficios que puede aportar una eficiente ingeniería de requisitos de software:

- ◆ Fomenta la calidad a través de una serie de características importantes, una de las cuales es la trazabilidad, la conexión de la información relacionada.
- ◆ Simplifica la evaluación de impacto de los cambios en los objetivos y requisitos.
- ◆ Valida la implementación de especificaciones técnicas de bajo nivel al demostrar cómo se cumplen requisitos de mayor nivel.
- ◆ Permite realizar un seguimiento del estado del proyecto al controlar el progreso del trabajo que se está realizando con el fin de cumplir los requisitos.
- ◆ Garantiza que el servicio, el software o el sistema que desarrolle y entregue cumpla con los requisitos del cliente en el plazo establecido y dentro del presupuesto fijado.
- ◆ Aumenta la comunicación y la colaboración, garantizando que todos los participantes tengan acceso a los objetivos del negocio comunes que guían la empresa o a los requisitos del cliente que definen el proyecto de desarrollo.

Con una adecuada ingeniería de requisitos se garantiza que el servicio, el software o el sistema que se desarrolle y se entregue cumplan con los requisitos del cliente en el plazo establecido y dentro del presupuesto fijado.

### **1.4 Ingeniería de requisitos.**

En esta investigación se utilizó como referencia el concepto de ingeniería de requisitos que define Pressman en (PRESSMAN 2005), mostrado a continuación.

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando sus necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional (THA 97). El proceso de ingeniería de requisitos puede ser descrito en 6 pasos distintos (SOM 97):

- ◆ Identificación de requisitos
- ◆ Análisis y negociación de requisitos
- ◆ Especificación de requisitos
- ◆ Modelado del sistema
- ◆ Validación de requisitos
- ◆ Gestión de requisitos

La ingeniería de requisitos, no es más que es una disciplina de la ingeniería de software, cuyo objetivo principal es determinar el propósito del sistema, al cual se desee informatizar, mediante la definición del conjunto de actividades y transformaciones que pretenden comprender las necesidades de un sistema software, las cuales son descritas y documentadas, siguiendo un determinado estándar.

### **1.5 El concepto de requisito.**

Una de las características de la ingeniería de requisitos es la falta de uniformidad en la terminología empleada, tanto para los conceptos básicos como para los procesos y los productos [Davis 1993, Brackett 1990, Pohl 1997]. Uno de los conceptos afectados por dicha falta de uniformidad es el de requisito. Para guiar este trabajo se ha tomado el concepto de requisito al cual se hace alusión en [IEEE 1990].

En el que se define que un requisito es:

- (a) una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo.
- (b) una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, norma, especificación u otro documento formal.



(c) Una representación en forma de documento de una condición o capacidad como las expresadas en (a) o en (b).

Según lo analizado se concluye, que un requisito no es más que diferentes funcionalidades o características que debe poseer un sistema para resolver un problema determinado.

Sin embargo, a pesar de la aparente simplicidad del concepto, es frecuente encontrar el término requisito calificado con adjetivos que pueden resultar confusos en un primer momento: de sistema, hardware, software, de usuario, de cliente, funcional, no funcional, etc.

### **1.6 El concepto de gestión de requisitos.**

Muchas son las definiciones en el tema de la ingeniería de requisitos que se le atribuyen al término gestión de requisitos, algunos muy abarcadores y otros muy sintetizados. Para guiar esta investigación se hace énfasis en el siguiente:

La gestión de requisitos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y sus cambios en cualquier momento (PRESSMAN 2005).

### **1.7 El concepto de trazabilidad.**

Al abordar el tema de la gestión de requisitos no se debe dejar de mencionar la trazabilidad de requisitos que no es más que la habilidad para describir y seguir la vida de un requisito en ambos sentidos, hacia sus orígenes o hacia su implementación, a través de todas las especificaciones generadas durante el proceso de desarrollo de software. (LETELIER y ANAYA)

### **1.8 Gestión de requisitos en eXtreme Programming (XP).**

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa fundamentalmente en la realimentación continua entre el cliente y el equipo de trabajo, la comunicación entre todos los participantes, y el enfrentamiento de los cambios. Es usado en proyectos pequeños con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.

Desde el punto de vista que describe Pressman, el proceso de ingeniería de requisitos que refiere la metodología XP, se basa fundamentalmente en los tres primeros pasos, dejando atrás lo fundamental de la investigación, que es la gestión de requisitos.

Uno de los procesos que define es la identificación de los requisitos con los que contará el sistema, especificando claramente los objetivos del sistema o producto, e investigando cómo se ajustan a las necesidades del negocio y serán utilizados diariamente. Los requisitos son recopilados en reuniones en las que participan clientes e ingenieros del sistema. Hay que añadir que ese es el momento de solucionar cualquier problema de alcance, o comprensión.

El análisis y la negociación de requisitos, en XP se tratan a través de las historias de usuarios. El cliente describe brevemente las características que el sistema debe poseer como parte de la identificación de requisitos, en muchos casos sólo se propone utilizar un nombre y una descripción, mas quizás una estimación de esfuerzo en días. Beck en su libro (BECK 2000) presenta un ejemplo de ficha en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado cosas por terminar y comentarios.

Se opina que la sola descripción de una historia de usuario no es suficiente, pero siguiendo la sencillez de XP, cualquier otra información asociada podría estar justificada con respecto al beneficio que aporte. Es decir, la información de una historia de usuario podría variar y ajustarse a las características específicas del proyecto.

En XP no se define la especificación de requisitos, pues se sugiere realizar en grandes sistemas, y es aplicada generalmente en proyectos pequeños. Por el contrario no significa que otra metodología que se aplique en pequeños proyectos incluya este proceso. Tampoco crea un modelado del sistema, pues según la literatura estudiada no se refleja que evalúe los componentes del sistema y sus relaciones entre sí, ni determina cómo están reflejados los requisitos, ni cómo se ha concebido la estética en el sistema.

Todo proyecto sea grande o pequeño debe abordar la validación de requisitos, para asegurar que todos han sido establecidos sin ambigüedad, inconsistencias y se ajusten a los estándares establecidos en un comienzo para el proceso, proyecto y producto.

Según Pressman el primer paso para la validación de requisitos es una revisión técnica formal donde intervienen los desarrolladores y el cliente. XP realiza semanalmente reuniones entre los clientes y desarrolladores con la finalidad de analizar, el avance en el proyecto, así como las técnicas y estrategias que se deben seguir.

Respecto a la gestión de requisitos expuesta, no concuerda con lo planteado. Una vez identificados los requisitos, en esta metodología no se les asigna identificadores, ni se crean matrices para su seguimiento. Este proceso se determina mediante una guía, que se encuentra en el artículo (SANCHEZ y ÁVILA). Su principal objetivo es orientar al usuario a la hora de concebir o seguir un cambio en una historia, y los pasos a seguir se describen posteriormente:

- ◆ Si la modificación que se desea realizar no implica cambios en el contenido de la historia entonces se crea una nueva tarea de programación para cubrir esa funcionalidad. De esta forma se registran las mejoras y modificaciones detectadas por el equipo de desarrollo pero que no afectan a los requisitos iniciales registrados en la historia de usuario
- ◆ Si la modificación implica cambios en la información de la historia se toma una decisión de acuerdo a la granularidad del cambio.
- ◆ Si la modificación es de una granularidad similar a la propia historia de usuario implica que se debe crear una nueva historia de usuario. En este caso es necesario mantener referencias entre las dos historias para disponer de trazabilidad.
- ◆ Por otra parte si el cambio no alcanza el nivel de una nueva historia se modifica la historia de usuario afectada y se registra el cambio en el control de versiones.

En fin XP no realiza las actividades propuestas por la gestión de requisitos que plantea la identificación de cada requisito y la creación de las dependencias entre los mismos mediante el uso de las matrices y la trazabilidad. No refiere estas actividades antes de realizar un cambio en los requisitos, o sea que las realiza sin su previa definición.

### 1.9 Gestión de requisitos en SCRUM.

SCRUM no es propiamente un método o metodología de desarrollo, se inicia como un modelo para el desarrollo de productos tecnológicos. Una de las características más importantes es su facilidad de entender y explicar, lo que ayuda mucho a su implantación. Es empleada en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; donde se presenten situaciones frecuentes en el desarrollo de determinados sistemas de software.

Por otra parte puede ser aplicado a distintos modelos de calidad (como podría ser CMMI) puesto que estos dicen qué debes hacer, pero no dicen cómo. Ahí es donde entra SCRUM jugando su papel como modelo de gestión del proyecto.

SCRUM es una metodología ágil basada en el desarrollo incremental e iterativo. Se basa fundamentalmente en la gestión de proyectos de software. No es una metodología de análisis, ni de diseño, como podría ser RUP, Sólo abarca prácticas de gestión sin entrar en las prácticas de desarrollo como lo hace XP.

Para controlar los requisitos define métodos de gestión y control para complementar la aplicación de otros métodos ágiles como XP que, centrados en prácticas de tipo técnico, carecen de ellas. Entre sus elementos básicos se encuentran Product Backlog y Sprint Backlog, los que se explicarán más adelante.

SCRUM al igual que XP realiza la identificación de los requisitos, organizándolos en una lista denominada Product Backlog, quedando determinados los objetivos y el alcance del sistema. Los requisitos son recopilados mediante reuniones con el cliente, y no es necesario que desde un principio queden todos definidos, sino que se pueden ir incorporando a la lista a medida que el proyecto se desarrolle.

Una vez coleccionados los requisitos se procede a su análisis y negociación. Son ordenados de menor a mayor en otra lista llamada Product Backlog en dependencia de la prioridad que se les confiera. Cualquier persona tiene privilegios para añadir funcionalidades al Product Backlog, pero sólo una puede ordenarlo. A esta persona se le denomina Product Owner y es el responsable del producto final.

Posteriormente del Product Backlog, se toman las primeras funcionalidades, y son descompuestas en pequeñas tareas las cuales son anotadas en otra lista denominada Sprint Backlog, y constituyen las tareas próximas a implementar.

Al final del período establecido para implementar los requisitos que se seleccionaron en la lista Sprint Backlog, se presenta el producto y se toma del Product Backlog ordenado las funcionalidades que serán implementadas próximamente.

En la metodología SCRUM solo se necesitan dos listas, el Product Backlog y el Sprint Backlog para la gestión de requisitos que ellos plantean.

Según lo establecido al inicio como gestión de requisitos, SCRUM no maneja este término completamente, se limita solo a la identificación, análisis y validación de los requisitos sin determinar su seguimiento y control, es decir solo pone en práctica algunas de las actividades que describe la propia gestión.

### **1.10 Gestión de requisitos en Microsoft Solution Framework (MSF).**

Una de las metodologías tradicionales de desarrollo es MSF, compleja, adaptable, flexible y con tecnología agnóstica porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología. Entre sus ventajas principales se encuentra la posibilidad de adaptarse a proyectos de cualquier tamaño y complejidad. La base de dicho modelo comprende prácticas y técnicas de otras metodologías. Permite optimizar recursos, manejar riesgos y administrar cambios para un óptimo beneficio a los usuarios. Proporciona soluciones contrastadas para planear, diseñar, desarrollar e implementar soluciones de la tecnología de la información (TI) con éxito. Es la metodología empleada por Microsoft para el desarrollo de software.

En una de sus fases de desarrollo, denominada visión y alcance identifica las metas y objetivos a alcanzar, término que le atribuye a requisito de software. Deben ser determinadas por los líderes y responsables del proyecto y respetarse durante la ejecución del proyecto en su totalidad. El equipo debe tener una visión clara de lo que se quisiera lograr para el cliente y ser capaz de indicarlo en términos que

motivarán a todo el equipo. Además utiliza la lista de fuentes de requisitos para hacer su recopilación. Las metas son descritas en el documento de visión/ámbito.

Posteriormente en la fase de planificación se procede a realizar otro de los procesos definidos: el análisis y negociación de los requisitos, que en MSF se le atribuye el nombre de descomposición del trabajo.

El equipo de trabajo prepara las especificaciones funcionales, o metas dividiéndolas en partes más pequeñas y manejables, y asignándole una prioridad. Aquellas funcionalidades de mayor prioridad serán implementadas en la en la fase más temprana del proyecto, ampliándose al máximo el tiempo disponible para reaccionar ante los problemas.

También se realiza el proceso de diseño de la solución, y prepara los planes de trabajo, estimaciones de costos y cronogramas de los diferentes entregables del proyecto. Esto se documenta en una estructura de división de trabajo (WBS), que sugiere que los objetivos importantes provisionales estén vinculados a la finalización o el establecimiento de la línea base de los elementos a entregar. Los objetivos importantes provisionales también forman un segundo nivel natural. Por debajo de este nivel, se identifican todos los elementos para entregar y se desglosan las tareas necesarias para crear cada uno de ellos.

Cuando se han documentado y realizado las estimaciones de las tareas y las actividades del proyecto en la WBS, se identifican las dependencias entre ellas, las cuales deben identificarse a los niveles de tareas más pequeños.

En el documento visión/ámbito, cuando se establece su línea base, ya el equipo considera que todas las funcionalidades están bajo control. Por lo que los cambios en el ámbito deben ser revisados y aprobados tanto por el equipo como por el cliente.

Parte de un buen control del cambio implica tomar buenas decisiones sobre las compensaciones. Los cambios son controlados mediante la matriz de compensaciones, o el triángulo de tres variables interdependientes. MSF utiliza herramientas para dicho control.

Lo expuesto anteriormente muestra una medida de cómo se gestionan los requisitos en MSF, mantenido de una forma u otra la trazabilidad o relación entre las especificaciones funcionales o metas como se les llaman en MSF.

### **1.11 Gestión de requisitos en Rational Unified Process (RUP).**

RUP es un proceso de Ingeniería del Software que proporciona una visión disciplinada para la asignación de tareas y responsabilidades en las organizaciones de desarrollo de software. Es un “modelo-producto” desarrollado y mantenido por Rational Software, integrado en su conjunto de herramientas de desarrollo, y distribuido por IBM.

Integra un conjunto de “buenas prácticas” para el desarrollo de software en un marco de procesos válido para un rango amplio de tipos de proyectos y organizaciones, dentro de estas se encuentra la gestión de requisitos, aspecto tratado en este trabajo.

La ingeniería de requisitos en RUP comienza con su identificación. Desde el comienzo del proyecto en la fase de inicio, los requisitos son recogidos mediante entrevistas con el cliente y son guardados en el documento Visión. En este documento también se recogen los cambios que se puedan introducir. Posteriormente se procede al análisis y negociación de los mismos. Los requisitos pueden ser definidos en casos de uso, casos de prueba, características funcionales, no funcionales, etc. Los casos de uso se pueden examinar en un diagrama de casos de uso del negocio o del sistema en dependencia de lo que se desee modelar.

Esta metodología también define la especificación de requisitos, actividad encargada de asignarle atributos a los requisitos. Entre los atributos principales que define RUP se encuentran: estado, estabilidad, riesgo, esfuerzo, prioridad, entre otros.

La validación de los requisitos en esta metodología se realiza a través de los casos de prueba que son definidos para cada uno de los requisitos funcionales del sistema. En ellos se determina si han cumplido o no con su funcionalidad. Este proceso se lleva a cabo en el flujo de trabajo de prueba.

No importa el cuidado tenido a la hora de definir los requisitos, en ocasiones estos cambiarán. Reemplazar un requisito no solo se reduce a quitar un requisito y sustituirlo por uno nuevo, sino también hay que analizar el impacto que este cambio puede ocasionar en los ya existentes. Es necesario entonces cerciorarse de dar a los requisitos una estructura que sea resistente a los cambios. Como solución a esta nueva interrogante RUP utiliza la trazabilidad para representar las dependencias entre los requisitos y otros artefactos del ciclo de vida del desarrollo y de esa manera gestionarlos.

Incluso, hace énfasis en que si un requerimiento no ha sido cambiado, pueden modificarse los atributos y la trazabilidad asociados a ellos. Esto puede o no ocurrir, el encargado de mantener esta información sobre una base en curso es el analista del sistema que cuenta con una solución ante el impacto que puede ocasionar un cambio en los requisitos. Se plantea que ello podría acarrear modificaciones en los requerimientos relacionados, por lo que es necesario revisar el impacto en el documento visión y luego en cada uno de los elementos asociados por las dependencias, para lograr una mejor comprensión del mismo. El manejo del cambio incluye actividades como establecer una línea base, determinándose las dependencias que son importantes resaltar.

Por lo anteriormente analizado se puede llegar a la conclusión, de que cada metodología implementa la gestión de requisitos teniendo en cuenta el alcance que la misma se propuso. Es la propia metodología la encargada de definir cómo realizar la ingeniería de requisitos. Pues a pesar de su importancia, en los proyectos (sobre todo proyectos pequeños) no se define claramente cómo esta se debe realizar.

La metodología RUP es la que más se acerca al concepto de ingeniería de requisitos y dentro de este al de gestión que se define en "Ingeniería del software, Un enfoque practico". En él la gestión, como se ha analizado, se realiza a través de la trazabilidad entre los diferentes elementos, lo que permite el seguimiento de los cambios en los mismos a lo largo del ciclo de vida del proyecto.



## **CAPÍTULO 2**

## **CAPÍTULO 2: CARACTERIZACIÓN Y DIAGNÓSTICO DE LOS PROYECTOS PRODUCTIVOS.**

En este capítulo se presentan algunos datos sólidos sobre la situación interna de los proyectos productivos de la UCI en el tema de la gestión de requisitos, además, se dan a conocer importantes resultados obtenidos al aplicarle una entrevista a los líderes de los proyectos que fueron definidos por la dirección de producción de cada facultad.

La UCI ha puesto en marcha la gran idea de la dirección de la Revolución Cubana de informatizar la sociedad cubana. En ella junto a la docencia, otra de las actividades más importantes lo constituye la producción, las cuales están estrechamente relacionadas desarrollándose simultáneamente. Desde los primeros años los estudiantes son vinculados a proyectos productivos, en los que se encuentran reunidos los recursos humanos y la tecnología de punta necesaria para alcanzar las metas propuestas.

El proceso productivo se basa fundamentalmente en la colaboración con las diferentes entidades externas a esta institución. La vinculación universidad-entidad externa es una de las estrategias, de las que ambas obtienen un beneficio. Esta estrategia propicia la colaboración y el intercambio de información entre las instituciones, del cual siempre se adquiere un nuevo conocimiento, y aporta mayor experiencia para enfrentarse a trabajos futuros.

El desarrollo de cualquier producto software trae consigo la definición de la metodología, herramienta, y el proceso a utilizar. A pesar de la diversidad de criterios de los desarrolladores en el tema de la gestión de requisitos, y de la resistencia de algunos a utilizarla o no, es un proceso de suma importancia para el desarrollo y seguimiento de los requisitos a lo largo del ciclo de vida del producto. Aparece entonces la interrogante ¿Cómo se comporta la gestión de requisitos en los proyectos productivos de la UCI?

Para dar respuesta a esta interrogante, se realizaron entrevistas a los líderes de los diferentes proyectos productivos, los cuales fueron seleccionados después de consultar al vicedecano de producción de cada una de las facultades.

## **2.1 Métodos empleados. La entrevista**

El método empleado fue la entrevista que no es más que una técnica de recopilación de datos empíricos, un proceso de comunicación entre dos o más personas, generalmente de forma oral donde las preguntas al entrevistado se hacen por vía directa, se debe desarrollar preguntas que permitan respuestas precisas. Es una conversación planificada cuyo objetivo es obtener información mediante una buena comunicación. El tipo de entrevista seleccionada fue la cerrada la que le ofrece a los entrevistados un conjunto de respuesta que se pueda seleccionar para que se basen en un mismo conjunto de posibles respuestas, siendo más fácil de administrar y evaluar.

### **2.1.2 Ventajas de la entrevista**

La entrevista es una técnica eficaz para obtener datos relevantes y significativos desde el punto de vista de las ciencias sociales, para averiguar. La información que se obtiene es muy superior que cuando se limita a la lectura de respuesta escrita, a través de ella se pueden captar los gestos, los tonos de voz, los énfasis, etc., que aportan una importante información sobre el tema y las personas entrevistadas.

La ventaja esencial de la entrevista reside en que son los mismos actores sociales quienes proporcionan los datos relativos a sus conductas, opiniones, deseos, actitudes, expectativas, etc. Cosas que por su misma naturaleza es casi imposible observar desde fuera.

### **2.1.3 Selección de entrevistados**

Realizar entrevistas toma tiempo; por lo tanto no es posible utilizar este método para recopilar toda la información que se necesite en la investigación. La entrevista fue aplicada a los líderes de los proyectos productivos que cada facultad consideró más importante. Dicha entrevista fue confeccionada a partir de los datos brindados por los vicedecanos de producción de cada una de las facultades que facilitaron los proyectos que se podían entrevistar.

### **2.1.4 Herramientas utilizadas.**

Para procesar las entrevistas se utilizó del paquete de Microsoft Office 2003, Microsoft Office InfoPath 2003. Esta herramienta mejora el proceso de recolección de información y facilita la reutilización de esa información en toda la organización mediante la creación de formularios los cuales se pueden exportar al

Microsoft Excel y generar gráficos que muestren claramente su análisis. InfoPath 2003 da soporte a tres actividades claves: la creación de formularios dinámicos, el llenado de estos formularios y el envío de estos formularios a los sistemas y procesos que necesitan la información recopilada en los mismos.

## **2.2 Gestión de requisitos en proyectos productivos.**

En los proyectos productivos se agrupan estudiantes con conocimientos básicos, que guiados por un líder de proyecto trabajan en conjunto en la realización de un determinado producto software para resolver las necesidades del cliente y contribuir a un mejor desempeño.

Para llevar a cabo una buena gestión de requisitos ante cualquier tipo de software, independientemente de la metodología que este utilice es necesario seguir un proceso por la importancia que este tiene, como se explica en el epígrafe 3.4. Sin embargo ¿Cómo se lleva a cabo la gestión de requisitos en los proyectos productivos de la UCI?

La gestión de requisitos es un proceso que no se tiene en cuenta en la totalidad de los proyectos productivos de la UCI. Para evaluar su estado actual, en esta investigación se ha realizado un análisis de los mismos, con el objetivo de medir si es empleada o no la gestión, y hasta que punto se efectúa.

Como se explicó anteriormente para la selección de la muestra se visitó a cada uno de los vicedecanos de producción de las 10 facultades, donde cada uno de ellos suministró la información de los proyectos en los cuales se aplicaría la entrevista. El total de los proyectos dados por los vicedecanos fue de 28, pudiéndose aplicar la entrevista en el 78.57% de los proyectos asignados, constituyendo un por ciento significativo para la investigación.

## **2.3 Resultados de la encuesta.**

En las entrevistas realizadas se confeccionaron una serie de preguntas alrededor del tema que se está investigando. Como se muestra en el Anexo 1.

### **2.3.1 Tamaño del proyecto**

Un atributo importante para la gestión de proyectos informáticos es conocer su tamaño, pues mientras mayor sea, más complicada será su gestión, teniendo mayor cantidad de requisitos, personas., procesos,

por mencionar algunos de los factores que provocan el alto grado de complejidad del software a la hora de ser desarrollado. Como resultado de la entrevista se tiene que el 68% del total son proyectos grandes (Ver Fig1).

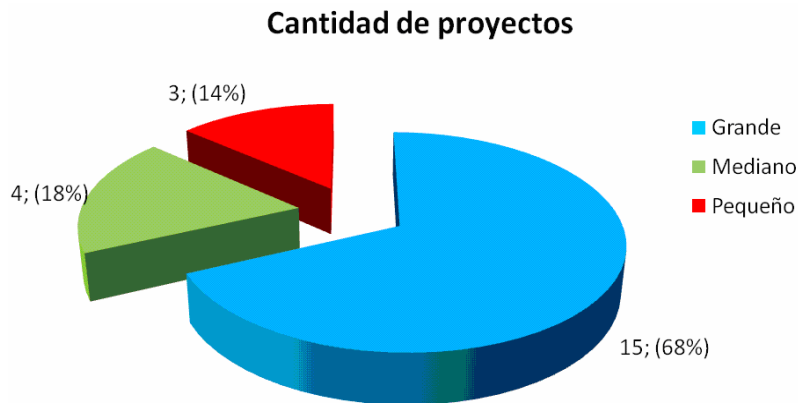


Fig. 1 Porcentaje de proyectos según su tamaño

### 2.3.2 Metodologías utilizadas

Todo proceso de desarrollo del software debe ser guiado por una metodología, que es la encargada de definir los procesos que deben seguirse para producir un producto de software con calidad.

Existen diferentes tipos de metodologías, cada proyecto atendiendo a sus características, selecciona cual va a utilizar y define cómo la ejecutará, con el propósito de que cada miembro del equipo de proyecto tenga una guía que garantice alcanzar la meta final: el producto de software.

Como resultado de la entrevista las dos metodologías más utilizadas son XP y RUP. De los 22 proyectos entrevistados 15 utilizan RUP, de ellos 12 son proyectos de gran tamaño y 3 proyectos medianos por lo que se justifica su utilización. Los restantes proyectos cuyo plazo de entrega son cortos, y cuentan con un equipo de desarrollo pequeño, utilizan metodologías ágiles como XP, siendo la que más se ajusta a proyectos con dichas características (Ver Fig. 2).

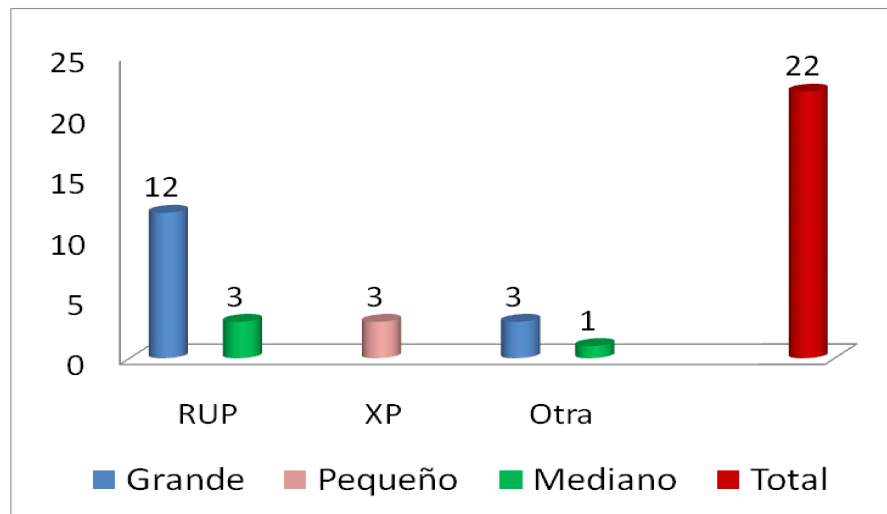


Fig. 2. Tamaño de proyectos vs. Metodologías de desarrollo.

### 2.3.3 Comportamiento de la gestión de requisitos.

Algunos consideran que la gestión de requisitos, es una tarea importante dentro de la ingeniería de requisitos, que solo se resume a la identificación y clasificación de los mismos. Otros por otra parte plantean que solo es necesario en proyectos de mayor envergadura y que en aquellos de menor complejidad se puede prescindir de ella.

La gestión de requisitos no se resume solo a eso sino que es una actividad que comprende desde el inicio del proyecto en la etapa de análisis y especificación de requisitos y se lleva a cabo durante todo su ciclo de desarrollo. Además no solo abarca las actividades de identificación, sino también el control, y seguimiento de los cambios en los requisitos en cualquier momento. Es aplicable a cualquier tipo de proyecto, aunque de no hacerse en proyectos pequeños las consecuencias serían menores. Mostrándose a continuación ¿Cómo se manifiesta esta temática en la universidad?

Muchos de los proyectos encuestados muestran que de alguna u otra manera realizan la gestión de requisitos aunque, de ellos solo un pequeño por ciento lo hace correctamente, los demás centran sus esfuerzos solo en las tareas de identificación y clasificación, sin llegar a gestionarlos en su plenitud. Por otro lado el 22% no la realiza (Ver Fig. 3), argumentando como principales causas las siguientes.

- ◆ Falta de conociendo relacionada con el tema.
- ◆ No es factible en proyectos pequeños.
- ◆ Están en la fase de captura de requisitos aún y no han concebido si la van a realizar o no.

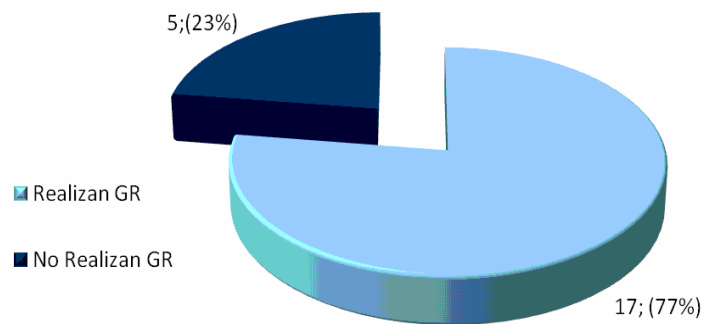


Fig. 3 Porcentaje de proyectos que gestionan los requisitos.

#### 2.3.4 Tipos de requisitos.

Cuando se enfrenta a un producto de software para determinar aquellas funcionalidades con las que el producto una vez concluido debe cumplir, se debe definir los tipos de requisitos que poseerá, que no son más que aquellas condiciones o capacidades que el usuario necesita para resolver un problema o lograr un objetivo.

Cada uno de los proyectos productivos de la UCI tiene características particulares y sus propios métodos para definir los requisitos. Aunque su gestión no está presente en todos, muchos se acercan en los tipos de requisitos que seleccionan; bajo este término se pueden distinguir varios tipos de requisitos.

#### ◆ Solicitudes de los clientes

Las solicitudes de los clientes, se encuentran entre los tipos de requisitos que más se especifican en los proyectos productivos de la universidad. Consiste en determinar las características relacionadas con las necesidades del cliente, entre los cuales figuran los directamente explicitados por él, con el objetivo de

definir por completo todos los aspectos relacionados con dicha solicitud. Además incluye los requisitos no indicados por el cliente pero necesarios para el uso especificado o previsto, cuando sea conocido. Ejemplo los requisitos impuestos por la organización o los legales y reglamentarios relacionados con el producto solicitado.

◆ **Casos de uso**

Los casos de uso se definen mayormente al igual que las solicitudes de los clientes en los proyectos entrevistados. Constituyen una secuencia de acciones realizadas por el sistema, que producen un resultado valioso para un actor en particular (LAGUNA). Se utilizan para capturar los requisitos funcionales del sistema y simplificar la construcción de los modelos de objetos. Otro elemento es que describen el sistema, su entorno y la relación entre ambos.

◆ **Características funcionales**

Los requisitos funcionales (RF) son los que más se definen en los proyectos entrevistados, siendo las características que describen de qué forma el usuario va a utilizar el sistema, declarando los servicios que debe proporcionar. Permiten especificar la manera en que se debe reaccionar ante determinadas entradas y cómo actuará el sistema en situaciones particulares. También pueden declarar explícitamente lo que el sistema no debe hacer.

◆ **Características no funcionales**

Los requisitos no funcionales (RNF), por su importancia, son identificados en la mayoría de los proyectos. Se conocen como el conjunto de características de calidad, que es necesario tener en cuenta al diseñar e implementar software. Están estrechamente relacionados con el producto: no siendo parte de su razón fundamental, pero precisos para hacerlo funcionar de la manera deseada.

◆ **Casos de Prueba**



Muy pocos proyectos ven la importancia de obtener los casos de prueba, obviando que son necesarios para validar que un producto de software cumpla con los requerimientos establecidos, sean funcionales o no funcionales.

Es importante precisar que con la presente investigación se citaron estos tipos de requisitos que aunque no son los únicos que existen, si los que más se identifican en los proyectos de la universidad.

La relación anterior expuesta se puede observar en la figura 4.

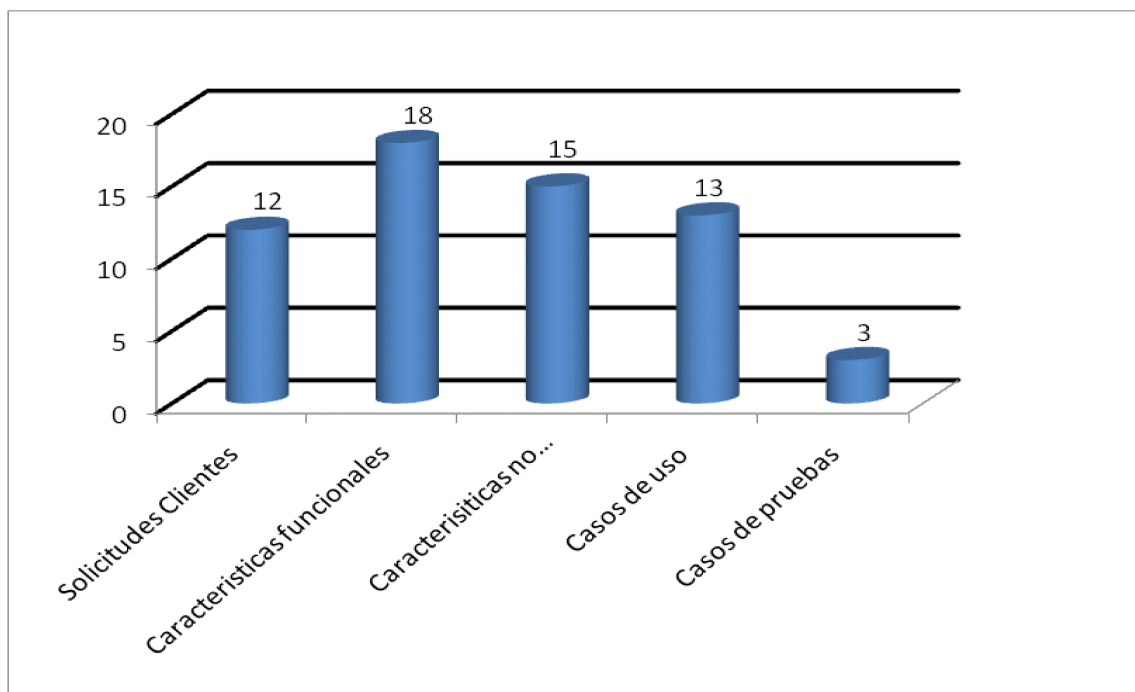


Fig. 4 Cantidad de proyectos vs Tipos de requisitos

### 2.3.5 Asignarle atributos a los requisitos

La definición de atributos a los requisitos es de suma importancia pues son las propiedades que caracterizan a los productos obtenidos en cada fase del ciclo de vida de un proyecto, es decir: información adicional y asociada a un requisito, detalles importantes. Deben ser especificados para que su logro pueda verificarse objetivamente. Se utilizan mayormente para la gestión de los artefactos. Se

seleccionaron en la entrevista los que se proponen en la metodología RUP, pues son los más completos para ejecutar una correcta ingeniería de requisitos, además de ser la metodología que más se utiliza en los proyectos de la universidad (Ver Fig. 5)

A continuación se muestra una lista parcial de dichos atributos:

◆ Grado de estabilidad

Puede expresarse la estabilidad por lo que refiere al número de cambios esperados a cualquier requisito basado en experiencia o conocimiento de eventos venideros que afectan la organización, funciones y a las personas que apoyan el sistema del software.

◆ Grado de necesidad

La necesidad de un requisito hace referencia al interés de los usuarios/clientes en que la aplicación lo realice, y hasta qué punto estarían dispuestos a prescindir de él.

Para negociar entre funcionalidad, planificación, presupuesto y nivel de calidad, es conveniente que los requisitos funcionales tengan asignado un nivel de necesidad (tres o cuatro categorías: esencial, deseable, opcional...) Siendo otra manera de alinear los requisitos

Esencial

Implica que el software no será aceptable a menos que estos requisitos se proporcionan de una manera convenida.

Condicional

Implica que éstos son requisitos que reforzarían el producto del software, pero no lo haría inaceptable si ellos están ausentes.

Optativo

Implica una clase de funciones que pueden o no pueden valer la pena.

◆ **Prioridad**

La prioridad de un requisito hace referencia al orden temporal: indica en qué fase de construcción del sistema se incluirá la funcionalidad que realice el requisito, con un nivel de prioridad temporal (dos o tres categorías: alta, baja...)

◆ **Riesgo**

Cada requisito debería ir acompañado de una estimación del riesgo que supone realizarlo (relacionado con la dificultad de implementarlo). Ej.: alto, moderado, bajo.

En (Pressman, 2002) se presenta la definición de riesgo dada por Robert Charette en [Charette, 1989] donde plantea que en primer lugar, el riesgo afecta a los futuros acontecimientos. En segundo lugar, el riesgo implica cambios. En tercer lugar, el riesgo implica elección, y la incertidumbre que entraña esta.

Cuando se considera el riesgo es indiscutible que están presentes permanentemente las características de incertidumbre (acontecimiento que caracteriza al riesgo y que puede o no ocurrir) y de pérdida (si el riesgo se convierte en una realidad ocurrirán consecuencias no deseables o pérdidas).

Están definidas las categorías de riesgos: los riesgos del proyecto, que amenazan el plan; los riesgos técnicos, que amenazan la calidad y la planificación temporal; y los riesgos del negocio, que amenazan la viabilidad del proyecto o del producto. Otra categorización a considerar son los riesgos conocidos (los que se descubren en las evaluaciones); los riesgos predecibles (se extrapolan de la experiencia) y los riesgos impredecibles (pueden ocurrir, aunque es muy difícil identificarlos de antemano).

◆ **Asignado a**

En muchos proyectos las características pueden ser asignadas a sus diferentes miembros, y descritas en los requerimientos del software. Este atributo permite entender qué responsabilidades están concedidas a cada miembro del equipo para que los requerimientos sean satisfechos.

◆ Estado

Permite determinar el progreso de las tareas durante la definición de la línea base del proyecto. Se clasifica en:

Propuesto: Usado para describir las modificaciones que están bajo discusión pero que no han sido revisadas ni aceptadas.

Aprobado: Capacidades son juzgadas, útiles y factibles y deben ser aprobadas para su implementación.

Incorporadas: Características incorporadas en la línea base del producto como un punto específico en tiempo.

◆ Esfuerzo

Algunas características requieren más tiempo y recursos que otras, teniendo en cuenta la cantidad de personas por semanas, líneas de código o puntos de función. Este atributo muestra la mejor forma de equilibrar las expectativas y definir lo que deben o no hacer en dependencia del tiempo asignado. Es usado en el manejo del alcance y determina el desarrollo de la prioridad

◆ Versión que lo incorporó

Registra la versión prevista del producto en la cual el requisito es añadido para su implementación. Este atributo puede ser usado para asignar las características en el documento Visión como parte de la línea base particular del lanzamiento.

◆ Beneficio

Todos los requerimientos no son creados con el mismo propósito. Alinear los requerimientos por los relativos beneficios que aportan al usuario final puede ser motivo de diálogo entre los analistas y miembros del equipo de desarrollo. Es usado en el manejo del alcance y determina el desarrollo de la prioridad.

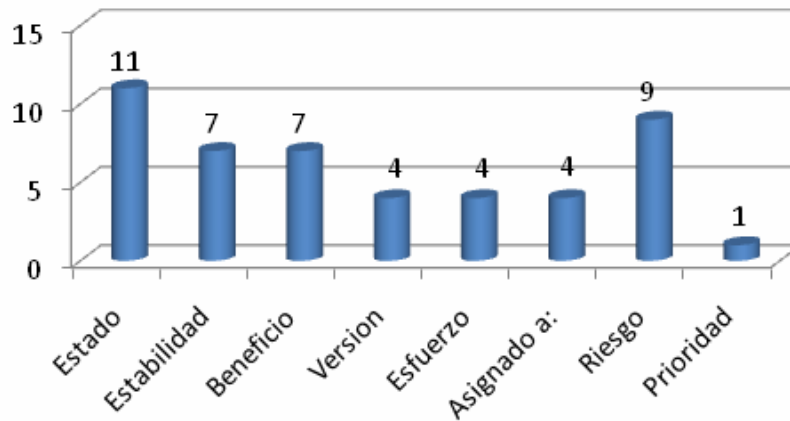


Fig. 5 Relación de atributos definidos por proyectos.

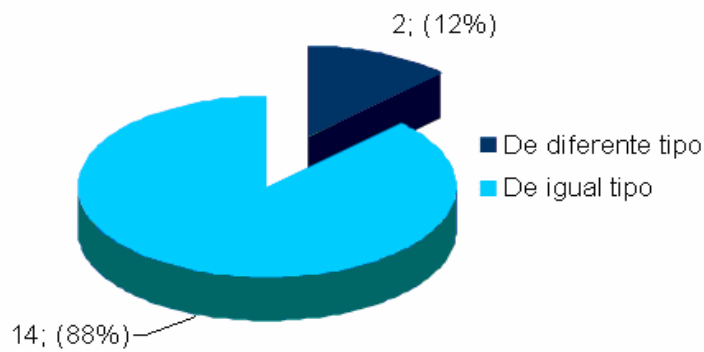
### 2.3.6 Trazabilidad de los requisitos.

En el desarrollo del software luego de la identificación de los requisitos, estos pueden ser susceptibles a cambios. Estos cambios pueden ocurrir, no solo después de su entrega sino durante el desarrollo iterativo de su proceso. Con el objetivo de establecer las dependencias entre los requisitos aparece el término trazabilidad que no es más que la habilidad para describir y seguir la vida de un requisito en ambos sentidos, hacia sus orígenes o hacia su implementación, mostrando el impacto de un cambio a través de todas las especificaciones generadas durante el proceso de desarrollo de software.

En (COMPUTING Agosto 1995 ) se introduce una mayor precisión de las ideas de trazabilidad hacia adelante o hacia atrás con los conceptos de trazabilidad horizontal y vertical. La horizontal es la trazabilidad entre las versiones de los requerimientos, en tanto que la vertical es la que establece las dependencias entre las fases previas y posteriores del ciclo de vida del desarrollo del software (SDLC).

Estas dependencias se pueden realizar entre los requisitos de igual y de diferente tipo mostrando una visión del comportamiento de las relaciones que existen entre ellos. El análisis de este punto en los proyectos productivos de la UCI no se encuentra muy bien precisado, pues casi la totalidad realiza la trazabilidad entre igual tipo de requisitos, destacándose los proyectos de mayor envergadura, mientras

que solo 2 proyectos tienen concebidas las dependencias entre diferentes tipos de requisitos de software, quedando 6 proyectos que no la definen. Los resultados se pueden ver en la Fig 6



**Fig.6 Relación de trazabilidad definida por proyectos**

### 2.3.7 Herramientas en la gestión de requisitos.

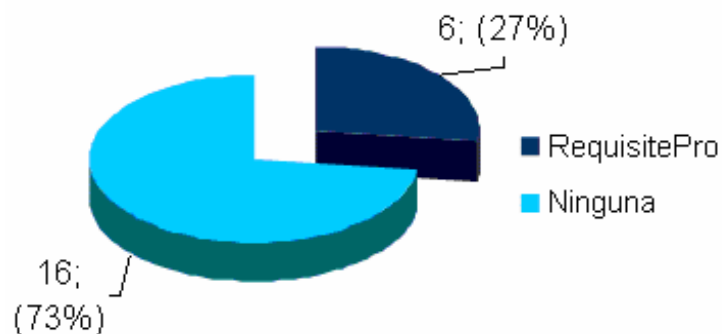
La gestión de requisitos es un componente vital en el desarrollo de un proyecto software ya que provee la dirección y alcance del proyecto. El uso de herramientas para auxiliar la gestión de requisitos se ha convertido en un aspecto importante en la ingeniería de sistemas y el diseño.

Considerando el tamaño y la complejidad del desarrollo, las herramientas vienen siendo algo esencial. Su utilización ayuda a gestionar el proyecto disminuyendo el arduo trabajo en el mantenimiento de los requisitos, de forma tal que proporcione grandes beneficios al reducir errores y mejorar tanto la productividad como la calidad en el desarrollo de un proyecto software. Además de permitir:

- ◆ Mejor comunicación interna de los aspectos referentes al proyecto.
- ◆ Documentación del proyecto siempre actualizada.
- ◆ Posibilidad de realizar análisis confiables sobre el progreso del proyecto, la estimación de los recursos, proyecciones futuras.

Lo que ha motivado a utilizar herramientas en la gestión de requisitos fue la complejidad en gestionarlos. Un estudio realizado por el grupo Meta, en marzo del 2003 detectó que aproximadamente el 60-70% de los proyectos de ingeniería de sistemas tenían fallos debido a la pobre recopilación, análisis y gestión de requisitos, debido a que la causa fundamental del éxito de un proyecto software, es la satisfacción de las necesidades de los usuarios, que se logra si sus requisitos son completamente definidos y documentados, de no ser así podría afectar a todo el proyecto.

En la UCI, de los proyectos entrevistados solo 6 proyectos productivos, lo que representa el 27 % tienen incluido en su desarrollo el empleo de herramientas en la gestión de requisitos siendo esta fundamentalmente Rational Requisite Pro (Ver Fig. 7). Las principales causas de no tener definida una herramienta, se debe a que en la concepción del proyecto no se gestionan los requisitos, por lo que no es necesario contar con la misma.



**Fig. 7 Porcentaje de proyectos que utilizan herramientas para gestionar los requisitos**

### **2.3.8 Proceso de gestión de requisitos.**

Se pudo determinar que la gestión de requisitos no es bien aplicada por ningún proyecto informático a pesar de que se pueda hacer de diferentes formas como se muestra en diversas bibliografías, es una actividad que consume tiempo y debería ser desarrollada por aquellos con un entrenamiento y/o experiencia previa. En la universidad ninguno de los proyectos entrevistados se guía por un proceso para gestionar los requisitos, siendo uno de los eslabones fundamentales para lograr la calidad y el factor fundamental a investigar.

A lo largo de este capítulo se pudo apreciar cómo se manifiesta el tema de la gestión de requisitos en los diferentes proyectos productivos de la universidad, generando como premisas fundamentales que en la mayoría no se realiza de la manera correcta. Ya que en muchos de ellos no se ha definido bien si se realizará o no, pues no se tiene bien definido el concepto de gestión de requisitos y por lo tanto no se llega a un completo análisis y desarrollo del tema.



## **CAPÍTULO 3**

## CAPÍTULO 3: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

El éxito o no de un proyecto requiere entre otras características de la utilización de la gestión de requisitos. Cuanto mejor sea la comunicación y gestión de requisitos, mayor será la oportunidad de que los proyectos se entreguen en tiempo y dentro del presupuesto establecido.

El modelo CMMI del SEI define para el nivel 2 el área clave de proceso “Gestión de requisitos” cuyo propósito es gestionar los requisitos del proyecto e identificar las inconsistencias entre dichos requisitos y los planes del proyecto así como en sus productos intermedios

¿Por qué esta tarea es tan importante? “El factor más importante en el desarrollo del software no son los problemas técnicos, sino los problemas de gestión” Paul Mark “Si todas las partes están de acuerdo en que se tiene que hacer una mejora al software, lo primero que hay que hacer es actualizar la especificación de requisitos y conseguir su aprobación” Arthur J. , Software Evolution

Debido a que todo proyecto informático es susceptible a cambios y según las estadísticas mostradas en el capítulo anterior se hace necesario que se defina un proceso que permita proceder a la incorporación, actualización, o eliminación de nuevas funcionalidades. En el presente capítulo se definirá un proceso que cumpla con lo anteriormente explicado y se estudiarán herramientas que se pueden aplicar al proceso definido para gestionar los requisitos.

### 3.1 Procesos. Elementos fundamentales.

¿Qué es un proceso?

No es más que el conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman entradas en salidas.

Los procesos se centran, por tanto, en obtener resultados como consecuencia de la transformación que producen las actividades que lo componen. Por ello, los procesos tienen objetivos relacionados con los resultados que obtienen, y para ello se debe ejercer un control para que las entradas sean las adecuadas,

las actividades se realicen de manera conforme y los recursos se empleen en el momento apropiado y de forma correcta, para obtener la salidas.

Todo proceso trae consigo diferentes conceptos que se definen a continuación para lograr un mejor entendimiento.

<b>Concepto</b>	<b>Descripción</b>
Rol	Es responsable por un conjunto de actividades de uno o más procesos. Un rol puede ser asumido por una o más personas de tiempo parcial o completo.
Artefactos	Cualquier elemento que se genera en un proceso.
Actividad	<p>Conjunto de tareas específicas asignadas para su realización a uno o más roles. Representa las operaciones individuales de un proceso. Puede ser una acción manual o automatizada que al realizarse contribuye a la realización del proceso al que pertenece. También puede desencadenar la creación de otro proceso.</p> <p>Las actividades describen lo que debe ser hecho.</p>
Estado	<p>Condición o situación durante el ciclo de vida de un proceso donde cumple determinadas condiciones que lo distinguen.</p> <p>Las estados describen “como está” el proceso en un instante determinado.</p> <p>Existen tres tipos de estados:</p> <ul style="list-style-type: none"> <li>◆ Listo: son aquellos procesos que pueden pasar a la ejecución, pues ya tienen asegurados todos los recursos necesarios.</li> <li>◆ En ejecución: se están realizando en el momento dado.</li> <li>◆ Bloqueado: esperan la respuesta de algún otro proceso para poder continuar con su ejecución.</li> </ul>

Evento	Mensajes o sucesos que disparan un cambio de estado o la ejecución de una actividad determinada en un flujo de proceso. Las actividades al concluir generan un evento que conlleva a un nuevo estado o las enlaza con la actividad siguiente. Por ejemplo:  Criterios de inicio/finalización  Secuencia antes/después.
Actor	Cualquier trabajador de la organización, responsable de una tarea en un proceso.  Los actores “hacen” las actividades.
Objetos de información	Dato o conjunto de datos que necesita una actividad para poder llevarse a cabo. Ejemplo:  Artefactos de entrada/salida.

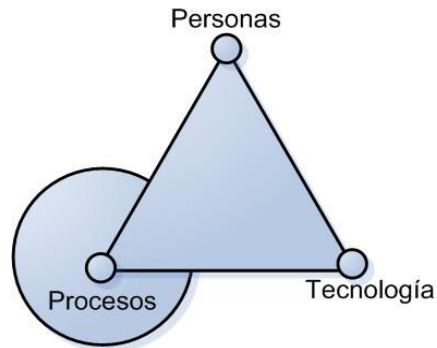
### **3.2 Concepto de gestión.**

El término de gestión es usado para describir un conjunto de técnicas y experiencias de una eficiente organización, planificación, dirección y control de las operaciones de un negocio.

### **3.3 La utilización de procesos en los proyectos productivos.**

La calidad de un sistema está altamente influenciada por la calidad de los procesos usados para adquirirlo, desarrollarlo y mantenerlo.

Los procesos, las personas y las tecnologías son los factores determinantes en los costos de los productos, el calendario y la calidad, son precisamente los procesos los que mantienen unidos el triángulo de proceso-persona-tecnología (Fig. 8)



**Fig.8.Triángulo Proceso-Persona-Tecnología.**

Es importante contar en las organizaciones con personal altamente calificado y motivado, pero estas personas no podrán ejecutar sus actividades sino comprenden cómo es el proceso. Lo importante es: además de contar con este personal altamente calificado poseer procesos bien definidos para lograr un desempeño eficiente.

Al utilizar procesos en un proyecto se consigue que todos los desarrolladores ejecuten el trabajo siguiendo las mismas pautas, y que los parámetros de ejecución y los resultados puedan ser medidos de forma homogénea en toda la organización.

De esta forma se reduce drásticamente la heterogeneidad de los resultados, tanto en la eficiencia de los tiempos invertidos, como en la calidad obtenida.

El objetivo fundamental de la ingeniería de requisitos es obtener todas las necesidades de los clientes para satisfacerlas con la implementación del producto de software, la norma ISO9001 como norma de gestión de la calidad, exige definir una serie de procesos de la organización que permitan conseguir los objetivos de calidad y tiende el puente entre los requerimientos del cliente y la satisfacción del mismo, definiendo para ello procesos estratégicos, claves y de soporte. Así mismo, el modelo CMMI tiene como propósito proveer una guía para mejorar los procesos de la organización y la capacidad para gestionar el desarrollo, la adquisición y el mantenimiento de productos y servicios. Planteando ambos que se deben definir procesos para la gestión de requisitos.

### **3.4 El proceso de gestión de requisitos**

El proceso de gestión que se define a continuación es independiente de cualquier metodología, es decir que se puede utilizar en cualquier proyecto. Fue definido a partir de un estudio del comportamiento de la gestión de requisitos en las diferentes metodologías actuales tales como: XP, MSF, Scrum, RUP; tomando sus mejores prácticas.

Dentro de este proceso propuesto los pasos a seguir son:

- ◆ Identificar las principales actividades que formarán parte del proceso.
- ◆ Construir el mapa de procesos (Fig.9) En él se representan gráficamente las actividades que forman parte del proceso así como sus relaciones. Facilitan el entendimiento de la secuencia e interrelación de las actividades y de cómo estas aportan valor y contribuyen a los resultados.
- ◆ Describir cada una de las actividades del proceso.

### **3.5 Principales actividades del proceso de gestión de requisitos.**

- ◆ Definir artefactos y tipos de requisitos relevantes para la trazabilidad.
- ◆ Definir las relaciones entre los artefactos y los tipos de requisitos.
- ◆ Definir atributos de los requisitos
- ◆ Elaborar matrices de trazabilidad
- ◆ Mantener y actualizar documentos.

### 3.6 Mapa del proceso.

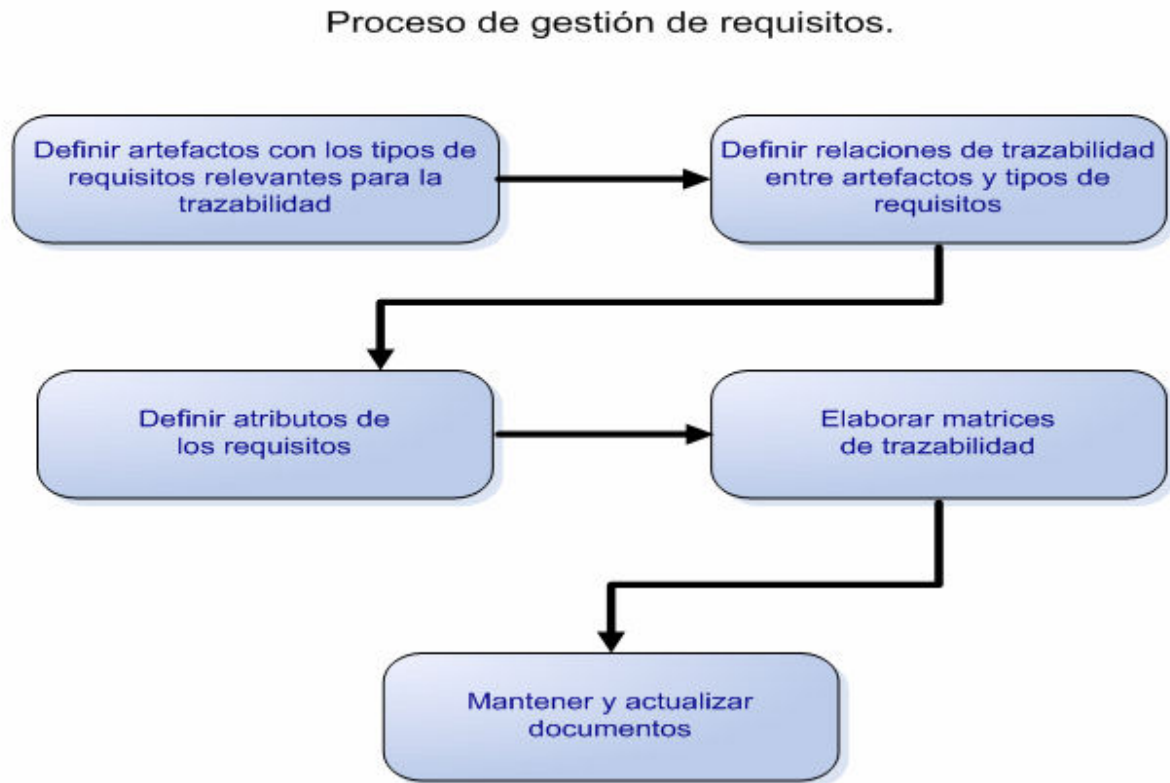


Fig. 9 Mapa del proceso.

### 3.7 Descripción del proceso de gestión de requisitos

El proceso comienza con la de definición de los artefactos y tipos de requisitos relevantes para la trazabilidad. Se inicia cuando se han determinado las características con las que contará el proyecto y el alcance que tendrá, lo que permite conocer el problema a resolver. Su objetivo fundamental está encaminado al análisis de los artefactos generados en la ingeniería de requisitos que pueden estar sujetos a cambios en cualquier momento del ciclo de vida del software y dentro de estos los tipos de requisitos que deben ser seguidos y controlados. Brinda una visión de la organización de los requerimientos, al relacionarlos con el artefacto del cual forma parte.

Las personas capacitadas para cumplir esta actividad son el jefe del proyecto y el analista del sistema, porque poseen los conocimientos necesarios, tienen dominio de lo que se va a realizar, cómo se hará y han trabajado anteriormente con los artefactos generados a lo largo del proyecto. El analista, al mismo tiempo dirige el modelado y coordina la captura de requisitos.

Para analizar aquellos artefactos que son importantes para la trazabilidad se necesita realizar primeramente las entrevistas con el cliente y poseer el plan del proyecto, pues en él están recogidos todos los documentos que serán creados en el desarrollo del proyecto.

El cumplimiento de esta actividad está fundamentado con la creación de subactividades cuyo propósito es descomponer las tareas a realizar, para una mejor solución del proceso descrito. Es necesario revisar el plan del proyecto para identificar aquellos artefactos, relacionados con la ingeniería de requisitos, que formarán parte del proceso. Posteriormente de cada artefacto seleccionado se definirán los tipos de requisitos comprendidos, que serán descritos en la plantilla propuesta (Fig. 10). Esta tarea brinda una medida de la organización de los tipos de requisitos en cada uno de los artefactos.

Al concluir esta actividad se crea un documento llamado gestión de requisitos el cual consta de dos capítulos.

**Capítulo 1:** Trazabilidad, artefactos y requisitos.

**Capítulo 2:** Gestión de proyecto.

Cada capítulo contará de diferentes epígrafes los cuales se incluirán a medida que se avance en el proceso. Con esta actividad se adiciona el primer epígrafe del capítulo 1 cuyo nombre será:

- ◆ Lista de artefactos con los tipos de requisitos definidos.

Esta actividad es realizada a lo largo del ciclo de desarrollo del software mientras se incorporen nuevos requisitos, que es una labor que se puede hacer en cualquier momento. Luego se definen las relaciones entre los artefactos y los tipos de requisitos seleccionados anteriormente, próxima actividad del proceso, se inicia cuando se tiene un análisis preliminar del sistema que se va a implementar. Su propósito es



construir las posibles dependencias entre los requisitos, desde sus orígenes, o hacia su implementación. Además de establecer la trazabilidad entre las versiones de los requisitos (horizontal), y las dependencias entre las fases previas y posteriores del ciclo de vida del software (vertical), para permitirle a cada requerimiento remontarse hacia atrás en el origen de los requerimientos y referenciar a cada uno en la documentación de desarrollo.

El responsable de esta actividad es el analista, que cuenta con el dominio de los requisitos del sistema y de las posibles relaciones entre los mismos, así como su comportamiento a lo largo del desarrollo del producto. Es el encargado de examinar el documento gestión de requisitos creado anteriormente que constituye un elemento de entrada en este momento del proceso, motivando el avance del mismo.

Dentro de las tareas previstas, se debe hacer un razonamiento previo de los requisitos con el objetivo de dividirlos en dos grupos: de igual y diferente tipo, para su posterior análisis. Primeramente se seleccionan los requisitos de un mismo tipo y se le aplica la estructura de dependencia definida en la figura 11. Este tipo de dependencia se determina fácilmente pues está ligado fuertemente a las existentes en el negocio

Al finalizar el trabajo con los requisitos de igual tipo se procede al análisis de los de diferente tipo. El propósito es seguir la traza de un requisito desde su origen hacia su implementación o viceversa mediante el establecimiento de la trazabilidad vertical, estableciendo las dependencias. El modelo de trazabilidad que se propone usar para este proceso es el mostrado en la figura 12.

Con el fin de guiar el proceso de trazabilidad los gráficos realizados para el seguimiento de los requisitos de diferente tipo son incorporados al documento gestión de requisitos, como otro punto dentro del epígrafe 2 denominado:

- ◆ Gráficos de las relaciones entre los requisitos.
  - ✓ Para requisitos de igual tipo.
  - ✓ Para requisitos de diferente tipo.

La definición de la trazabilidad es de suma importancia para el desarrollo del proceso porque muestra una medida de cómo se está gestionando el mismo, observándose el desarrollo de los requerimientos. Permite

si se está implementando, evaluar el impacto en los requerimientos, diseño e implementación. Además si se desea diseñar o probar cualquier componente del sistema es necesario saber los requerimientos que satisface, y cuáles están siendo validados en cada prueba.

Esta es una actividad que se desarrolla en cualquier momento del proceso de desarrollo del software, cuando pueden aparecer nuevos requerimientos que necesitan ser traceados.

Se continúa al definirle atributos a los requisitos lo que facilita la definición del alcance de cada una de las iteraciones del proyecto, y contribuye a manejar los cambios en los requisitos. Esta actividad es iniciada al identificarse los requisitos que debe cumplir el sistema para satisfacer las necesidades de los clientes.

El rol encargado de desarrollar esta actividad al igual que la anterior es el analista del sistema, persona encargada de determinar las relaciones entre los requisitos. Para realizar esta tarea es necesario tener actualizado el documento gestión de requisitos.

Las subactividades que guían esta actividad están relacionadas con la identificación de los posibles atributos que le asignarán a los requisitos según su tipo. Dentro de estos atributos se consideran más importantes: beneficio como perspectivas de los clientes, el esfuerzo a implementar, el riesgo en el desarrollo del esfuerzo, la estabilidad (probabilidad de permanecer sin cambiar), y el impacto en la arquitectura (si es arquitectónicamente significativa) de cada requerimiento. Aunque existen otros que en dependencia de la información que se desee obtener se pueden incorporar o no.

Posteriormente a los atributos se le definen los posibles valores que puedan tomar en caso de que lo posean, que ayuda al proceso de gestión del proyecto, mostrando una serie de datos como:

¿Cuáles requisitos se implementarán en la próxima iteración?

¿Cuáles requisitos han sido asignados?

¿En qué versión del proyecto se incorporó?

¿En qué estado se encuentra el requisito?

Toda esta información se debe recopilar en la plantilla definida en la figura 13 como relación de atributos por requisito y debe ser agregada al documento gestión de requisitos en el capítulo 2 como un nuevo epígrafe, para actualizar el documento.

- ◆ Lista de tipos de requisitos con sus atributos.

La terminación de esta actividad está vinculada con la de la especificación de los requisitos.

Otra de las actividades del proceso es la elaboración de las matrices de trazabilidad, necesarias para poder analizar el impacto de un cambio en un requisito pues mediante ellas se obtiene todo el alcance del cambio a partir del cual se puede estimar el esfuerzo y el costo del mismo.

El jefe del proyecto y el analista del sistema son los que realizan esta actividad. A partir del documento de gestión de requisitos se elabora la matriz según el cambio solicitado, en ella se refleja todos los requisitos que dependen del requisito a modificar y a partir de ello se calcula el esfuerzo y el costo de todos los que sufren cambios, pudiéndose realizar la planificación con los recursos disponibles.

Las matrices que se pueden elaborar son las siguientes.

- ◆ Matriz de seguimiento de orígenes: Identifica el origen de cada requisito.
- ◆ Matriz de seguimiento de características: Es definida por el cliente.
- ◆ Matriz de seguimiento de dependencias: Indica cómo se relacionan los requisitos entre sí.
- ◆ Matriz de seguimiento de artefactos: Vincula a los requisitos con los artefactos que los manejan.

Para una mayor comprensión, la actividad se ha dividido en dos segmentos, en dependencia de lo que se necesite realizar, una matriz, o un informe. En caso de ser una matriz se seleccionan los requisitos que se deban relacionar en la matriz y los artefactos que contienen estos requisitos y se procede a elaborar la matriz de seguimiento de artefactos que mostrará una idea de la organización de los requisitos por artefactos. En caso de que sea un informe, se escogerá el requisito al cual se le desea elaborar el informe, tomando la información del documento de gestión de requisitos. Cuando finalicen estas actividades, se actualizará este documento en el capítulo 1 con los siguientes epígrafes:

- ◆ Matrices de trazabilidad.
- ◆ Informes del estado de los atributos.

Esta actividad termina cuando finalice el proyecto, porque en cualquier momento del desarrollo del software pueden aparecer nuevos cambios y se deberán actualizar las matrices de trazabilidad.

Los requisitos pueden ser modificados por diversas razones entre las que se encuentran:

- ◆ Cambió el problema que se estaba resolviendo.
- ◆ Los usuarios cambiaron su forma de pensar o sus percepciones.
- ◆ Cambió el ambiente de negocios.
- ◆ Cambió el mercado en el cual se desenvuelve el negocio.

Debido a que los cambios persisten a lo largo de la vida del sistema es necesario definir dentro del proceso de gestión de requisitos una actividad para mantener y actualizar los documentos, que permita modificar y dejar constancia de los cambios realizados a los diferentes documentos creados.

El encargado de efectuar estas actualizaciones en los artefactos creados es el jefe del proyecto, que en conjunto con los desarrolladores aprueban o no el cambio solicitado según el proceso de gestión de la configuración designado.

El documento gestión de requisitos constituye la fuente principal para gestionar los requisitos y sus cambios. Al solicitarse un cambio, se elaboran las matrices necesarias para identificar a los requisitos afectados. Una vez realizado el cambio se actualizan las dependencias para el caso de que se eliminen o se creen nuevas. Cuando sean seleccionadas, se procede a ejecutar el cambio y se actualiza el epígrafe involucrado dentro del documento gestión de requisitos, quedando actualizado el artefacto principal del proceso.

Todas estas acciones se estarán realizando durante todo el ciclo de vida del proyecto, porque en cualquier momento de su ejecución pueden generarse nuevos cambios que necesitarán ser gestionados.

### 3.8 Descripción de las actividades propuestas.

**Actividad:** Definir artefactos y tipos de requisitos relevantes para la trazabilidad.

**Propósito:** Analizar los artefactos de la ingeniería de requisitos y/o tipos de requisitos que estarán sujetos al control de cambio y que requisitos son de interés para su seguimiento y control.

**Actores:** Analista del sistema y Jefe de Proyecto.

**Artefactos de entrada:** Entrevistas con el cliente y el plan de proyecto.

**Artefacto de salida:**

Documento gestión de requisitos con el epígrafe.

- ◆ Lista de artefactos con los tipos de requisitos definidos. Ver (Fig. 10)

Nombre del artefacto	Nombre del tipo de requisito

**Fig. 10 Lista de artefactos con los tipos de requisitos.**

**Criterios de inicio:** Esta actividad comienza cuando se tiene un conocimiento del alcance que tendrá el proyecto y los problemas a los que se les dará solución.

**Criterios de finalización:** Cuando finalice el proyecto, porque en cualquier momento del desarrollo del software pueden aparecer nuevos requisitos.

**Subactividades:**

- ◆ Revisar el plan de proyecto, en el están reflejados todos los artefactos que serán desarrollados.
- ◆ Analizar los artefactos relacionados con la gestión de requisitos que se generan en la metodología utilizada

- ◆ Definir los tipos de requisitos contenidos en cada uno de los artefactos seleccionados.
- ◆ Actualizar documento de gestión de requisitos.

**Secuencia:**

**Antes:** A esta actividad no la antecede ninguna, porque es la primera actividad del proceso.

**Después:** Definir las relaciones entre los artefactos y los tipos de requisitos

**Actividad:** Definir las relaciones entre los artefactos y los tipos de requisitos.

**Propósito:** Precisar las posibles dependencias entre los requisitos, desde sus orígenes, o hacia su implementación. Además de establecer la trazabilidad entre las versiones de los requisitos (horizontal), y las dependencias entre las fases previas y posteriores del ciclo de vida del software (vertical).

**Actor:** Analista del sistema

**Artefacto de entrada:** Documento gestión de requisitos actualizado.

**Artefacto de salida:**

Documento gestión de requisitos con los epígrafes.

- ◆ Gráficos de las relaciones entre los requisitos.

Para requisitos de igual tipo

La estructura de dependencia que se propone es la siguiente.

Sean A, B, C, D, E y F requisitos del sistema, de los cuales se definen las siguientes dependencias.

A es padre de B

B es padre de D

C, E, y F no tienen hijos y solo pueden ser padres.

La estructura de dependencia quedará de la siguiente manera.



**Fig. 11 Estructura de dependencia para requisitos de igual tipo.**

Según la descripción anterior, a la izquierda del diagrama se colocan aquellos requisitos que sean solo padres y a la derecha se van colocando sus hijos hasta terminar con todas las dependencias

Para requisitos de diferente tipo.

El tipo de relación definida entre los requisitos de diferente tipo será (Desde/ Para).

- ◆ Desde: Se realiza de un determinado elemento al seleccionado.
- ◆ Para: Se realiza desde un elemento seleccionado hacia otros elementos.

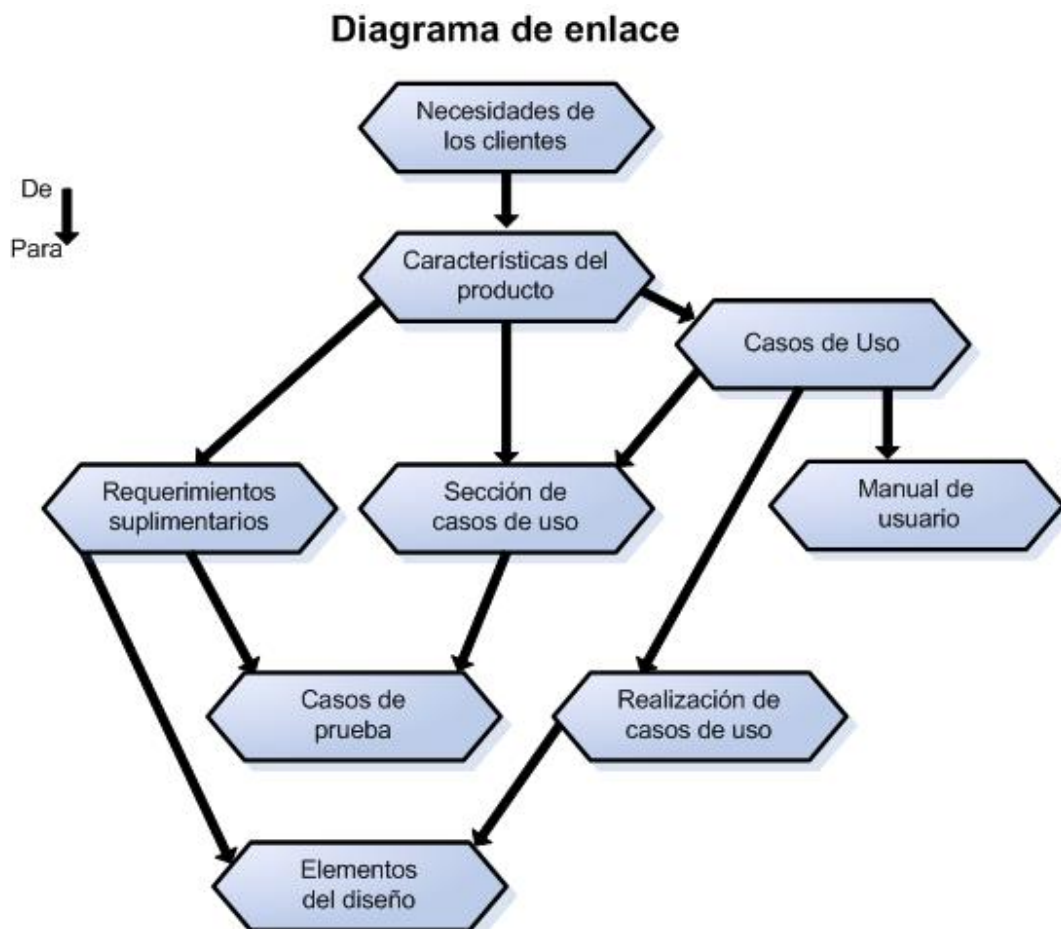


Fig. 12 Diagrama de enlace para requisitos de diferentes tipos.

**Criterios de inicio:** Cuando se inicie el análisis del sistema que se va a construir.

**Criterios de finalización:** Cuando finalice el proyecto, porque en cualquier momento del desarrollo del software pueden aparecer nuevos requisitos.

**Subactividades:**

- ◆ Analizar los requisitos desde sus orígenes o hacia su implementación para definir la trazabilidad vertical
- ◆ Establecer la dirección de dependencia entre los diferentes requisitos.



- ◆ Analizar los requisitos de un mismo tipo y estructurar sus dependencias.
- ◆ Analizar a que tipos de requisitos se le aplicará una estructura de dependencia.
- ◆ Actualizar documento de gestión de requisitos.
- ◆ Elaborar los artefactos donde se realice la especificación de los requisitos definidos en la subactividad anterior.

**Secuencia:**

**Antes:** Definir artefactos y tipos de requisitos relevantes para la trazabilidad.

**Después:** Definirle atributos a los requisitos.

**Actividad:** Definir atributos de los requisitos.

**Propósito:** Facilitar la definición del alcance de cada una de las iteraciones del proyecto.

**Actor:** Jefe de Proyecto

**Artefacto de entrada:** Documento gestión de requisitos actualizado.

**Artefacto de salida:**

Documento gestión de requisitos con los epígrafes.

- ◆ Relación de atributos por requisitos.

Artefacto	Tipo de requisito	Atributos			
		Atributo 1		Atributo 2	
Artefacto 1		Valor 1	Valor 2	Valor 1	Valor 2
	Requisito 1				
	Requisito 2				

**Fig. 13 Relación de atributos por requisito**

**Criterios de inicio:** Una vez identificados los requisitos que debe cumplir el sistema para satisfacer las necesidades de los clientes.

**Criterios de finalización:** Cuando se concluya la especificación de los requisitos.

**Subactividades:**

- ◆ Analizar para cada tipo de requisitos los atributos que estos pueden tener.
- ◆ Definir los posibles valores de cada atributo, en caso de que los posea.
- ◆ Actualizar documento de gestión de requisitos.

**Secuencia:**

**Antes:** Definir las relaciones entre los artefactos y tipos de requisitos

**Después:** Elaborar matrices de trazabilidad.

**Actividad:** Elaborar matrices de trazabilidad.

**Propósito:** Crear las matrices de trazabilidad para poder analizar el impacto de un cambio en un requisito y la información que se puede obtener, dado un tipo de requisito, su valor y viceversa.

Propuesta de los tipos de matrices que se pueden crear.

- ◆ Matriz de seguimiento de orígenes: Identifica el origen de cada requisito.
- ◆ Matriz de seguimiento de características: Definidas por el cliente.
- ◆ Matriz de seguimiento de dependencias: Indica cómo se relacionan los requisitos entre sí.
- ◆ Matriz de seguimiento de artefactos: Vincula a los requisitos con los artefactos que los manejan.

**Actores:** Analista del sistema y Jefe de Proyecto

**Artefacto de entrada:** Documento de gestión de requisitos actualizado.

**Artefacto de salida:**

Documento de gestión de requisitos agregando epígrafes.

- ◆ Matrices de trazabilidad
- ◆ Informes del estado de los atributos.

**Criterios de Inicio:** Cuando se determinen aquellos artefactos relacionados con la ingeniería de requisitos y los tipos de requisitos contenidos, y se actualizan las matrices ante una solicitud de cambio en un requisito o una planeación de una iteración.

**Criterios de finalización:** Cuando finalice el proyecto, porque en cualquier momento del desarrollo del software pueden aparecer nuevos cambios.

**Subactividades:**

Las subactividades se dividen en dos grupos en dependencia de lo que se desee generar.

Si es una matriz de dependencia.

- ◆ Seleccionar los tipos de requisitos para hacer la matriz.
- ◆ Buscar los artefactos que contienen a estos tipos de requisitos.
- ◆ Elaborar la matriz.

Si es un informe

- ◆ Seleccionar el tipo de requisito al cual se le va a elaborar el informe.
- ◆ Seleccionar del documento el estado de estos requisitos.

**Secuencia:**

**Antes:** Definir atributos a los requisitos.

**Después:** Mantener y actualizar documentos.

**Actividad:** Mantener y actualizar documentos.

**Propósito:** Actualizar los documentos con los cambios que puedan ocurrir.

**Actor:** Jefe de Proyecto

**Artefacto de entrada:** Documento de gestión de requisitos actualizado.

**Artefacto de salida:** Documento de gestión de requisitos actualizado con los nuevos cambios.

**Criterios de inicio:** Cuando se analice una solicitud de cambio en un requisito o en un documento.

**Criterios de finalización:** Cuando finalice el proyecto, porque en cualquier momento del desarrollo del software pueden generarse nuevos cambios.

**Subactividades:**

- ◆ Identificar qué actividades y plantillas están relacionadas con el cambio.
- ◆ Actualizar las plantillas y el epígrafe referente a la actividad.
- ◆ Actualizar documento de gestión de requisitos

**Secuencia:**

**Antes:** Elaborar matrices de trazabilidad.

**Después:** Ninguna, es la última actividad del proceso.

El artefacto de salida de este proceso es el documento Gestión de requisitos cuya composición se muestra en el Anexo 3.

### 3.9 La gestión de requisitos en la práctica.

La gestión de requisitos es una actividad que consume tiempo y debe ser desarrollada por aquellos que posean conocimientos de este proceso. Antes de adentrarse en el proceso, se debe de considerar lo siguiente:

- ◆ El equipo del proyecto entiende y sigue un apropiado proceso de desarrollo en el cual están descritos los procedimientos y actividades a realizar,
- ◆ La buena comunicación es importante durante el proceso de gestionar requerimientos.
- ◆ Se debe mantener una lista de requerimientos agrupados por tipos y relacionados entre ellos en una base de datos donde se puedan manejar fácilmente,
- ◆ Es necesario implementar un proceso de gestión de la configuración de requerimientos,
- ◆ No añadir ni cambiar requerimientos sin llevar un análisis de riesgos que determine el impacto en un plan de proyecto y reestimación del costo y planeación del proyecto.

Es necesario utilizar un método de trazabilidad de requerimientos a través del ciclo de vida del producto. Este no es un proceso fácil, por tal motivo, se sugiere que se utilice una herramienta automatizada.

### 3.10 Herramientas para la gestión de requisitos

El uso de herramientas en la gestión de requisitos mejora tanto la productividad como la calidad en el desarrollo de un proyecto software. Existen varias herramientas en el mercado que auxilian a las tareas de gestión.

En cualquier proyecto de ingeniería de sistemas, la gestión de requisitos es una tarea intensiva, debe tener la habilidad de relacionar diferentes documentos, obtener una visión sinóptica de esta relación, crear

reportes especiales de estos documentos, controlar los cambios hechos a través del conjunto de documentos de una manera consistente, acomodar los requisitos estructurados de documentos diversos y tipos de documentos.

El uso de una herramienta de gestión de requisitos proporciona a la organización:

- ◆ El ahorro en costos de especificación y de desarrollo, minimizando el impacto de errores.
- ◆ Mejora la calidad mediante un adecuado análisis y gestión de los requisitos,
- ◆ Mejora la productividad facilitando la reutilización real de los requisitos desde la especificación.
- ◆ Reduce las no-conformidades del sistema.
- ◆ Permite controlar la especificación.
- ◆ Ayuda a cumplir con estándares de calidad.
- ◆ Proporciona un repositorio no propietario de especificación.
- ◆ Permite centralizar toda la información del problema.
- ◆ Permite especificar sistemas de una forma estructurada y gráfica.
- ◆ Proporciona una trazabilidad completa de los requisitos.
- ◆ Brinda soporte a la planificación del proyecto.

### **3.11 Funcionalidades de las herramientas de gestión de requisitos.**

La mayoría de las herramientas de gestión de requisitos en el mercado realizan principalmente las mismas funciones. Permiten a los desarrolladores del sistema a importar grandes documentos de una variedad de formatos estándar de procesadores de palabras. Los elementos del documento están sujetos a rigurosos cambios y a un régimen de control de versiones. Se puede establecer una relación entre los elementos del documento, y los atributos pueden ser asociados con los elementos del sistema y a menudo relacionados. Pueden ser generados una variedad de vistas de documentos utilizando tanto los atributos como las relaciones, generalmente vistas específicas de trazabilidad tales como matrices de trazabilidad. De la misma manera, plantillas de documentos pueden ser configuradas para crear nuevos documentos compuestos.

Las herramientas de gestión de requisitos son genéricas, esto es que necesitan ser configuradas para soportar ingeniería de requisitos específicos y procesos de desarrollo de sistemas. Dichas configuraciones son soportadas por la creación de plantillas de documentos, esquemas/diseño de atributos, tipos de relación y vistas de documentos.

### 3.12 Herramientas de gestión de requisitos en el mercado

Las herramientas fundamentales para la gestión de requisitos se basan en sistemas centralizados de gestión de bases de datos para almacenar la información correspondiente a los requisitos, que suelen consistir en párrafos de texto libre con una serie de atributos predefinidos y a los que la mayoría de herramientas permiten asociar nuevos tipos de atributos por parte del usuario.

Asumen que la estructura de los requisitos es jerárquica, de forma que un requisito puede estar formado o tener asociados otros requisitos de nivel inferior, y la mayoría permite extraer párrafos de ficheros generados por procesadores de texto comerciales y convertirlos en requisitos.

Otras de las características comunes a la mayor parte de las herramientas es la posibilidad de realizar consultas sobre los requisitos en función de determinados valores de sus atributos.

Se realizó una selección de 3 herramientas que cumplen con la mayoría de las funciones anteriormente expuestas y que están ampliamente difundidas y reconocidas, como demuestra el hecho de que aparecen siempre en todas las comparativas, como por ejemplo en (ALEXANDER) así mismo, tienen un amplio soporte de las empresas que las desarrollan, y lo que es más importante tienen la posibilidad de ampliar su funcionalidad. Estas son:

- ◆ IBM Rational Requisite Pro
- ◆ Caliber RM
- ◆ DOORS ERS

**RequisitePro:** Es una herramienta centrada en documentos, que almacena los requisitos asociándolos a ellos (aunque también permite guardarlos directamente en la base de datos), mientras que las otras herramientas están orientadas a requisitos. Se auxilia especialmente en el control de cambio de requisitos,

con trazabilidad para especificaciones de software y de pruebas. Está muy unido a MS Word ya que es compañero de Microsoft Development. La herramienta permite el uso de Oracle sobre Unix o Windows y también soporta SQL Server sobre Windows.

**CaliberRM:** Es para sistemas grandes y complejos proporcionando una base de datos de requisitos con trazabilidad. La compañía ve a los requisitos como parte del proceso de gestión de la calidad del software, el cual es considerado también, las pruebas y el trazado de defectos. CaliberRM está basado en Internet y maneja referencia de documentos, responsabilidad de usuario, trazabilidad, prioridad y estado entre otras características.

**DOORS:** A diferencia del resto de las herramientas, considera los requisitos como objetos y los documentos como módulos. Tiene una orientación basada en objetos, frente a Requisite Pro y Caliber-RM, que manejan solamente requisitos y sus atributos. Es una herramienta para organizaciones grandes que necesitan controlar complejos conjuntos de usuarios y requisitos de sistemas con una completa trazabilidad. Proporciona buena visualización de tales documentos como jerárquicas, y su lenguaje de extensión permite una gran variedad de soporte de herramientas a ser construidas.

En (INGENIERÍA 2004) se realiza una comparación de estas mismas herramientas dividiendo aspectos técnicos como son ambiente de trabajo, captura e identificación de requisitos, modelo del dominio del problema, captura de la estructura del sistema, trazabilidad, gestión de la configuración, etc., Para esta investigación se toman como referencias estos datos y se analizan las actividades del proceso propuesto que se llevan a ejecución en cada una de estas herramientas.

<b>Características</b>	<b>DOORS</b>	<b>Caliber-RM</b>	<b>RequisitePro</b>
Importa requerimientos desde una tabla Word a una base de datos	X	X	
Incorpora objetos no textos e imágenes en la base de datos.	X	X	



Sincroniza textualmente el documento de Especificación de Requisitos del Software con el contenido de la base de datos.			X
Define diferentes atributos para diferentes tipos de requerimientos y le asigna de forma individual un valor al atributo del requerimiento.	X	X	X
Define la línea base del requerimiento.	X		
Notifica por correo electrónico a los participantes del proyecto sobre los cambios en los requerimientos	X	X	
Define las relaciones de trazabilidad entre los requerimientos individuales y entre los requerimientos y otros elementos del sistema.	X	X	X
Incluye una ayuda de aprendizaje, como un tutorial.	X	X	X
Integra otras herramientas, como prueba, diseño, y administración del proyecto.	X	X	X
Define usuarios, grupos con sus privilegios de			

accesos.	X	X	X
Incluye interfaces Web para las consultas y actualizaciones de los atributos de los requerimientos en la base de datos.	X	X	X
Incluye en el sistema propuesta de cambio.	X		
Incluye manuales de usuarios de impresión.	X	X	X

Con esta comparación se pretende resaltar la diferencia entre lo teórico propuesto en el proceso y lo práctico del proceso de gestión de requisitos, en cada una de las herramientas. Como resultado se aprecia que las herramientas expuestas, en su concepción aplican de una y otra manera cada una de las actividades descritas en el proceso propuesto, de manera tal que una vez que se halla estudiado el proceso anteriormente descrito se pueda utilizar cualquiera de estas herramientas para auxiliarse en el trabajo de gestionar los requisitos.

Valorando los beneficios contra las pérdidas de tiempo, costo y fracasos en proyectos de software, se está adquiriendo conciencia en la utilización de herramientas de ingeniería de requisitos que en un futuro próximo habrá de incorporarse nuevas funcionalidades que se detecten como necesarias con la experiencia de su utilización.

En este capítulo se ha mostrado detalladamente el proceso de gestión de requisitos, así como las diferentes herramientas, que se pueden utilizar para facilitar el mismo.

Además de queda demostrado que todas las actividades definidas en el proceso son implementadas por las herramientas anteriormente descritas.

**CONCLUSIONES.**

Con la definición de un proceso que permita gestionar los requisitos del software, se da cumplimiento al objetivo de este trabajo, pues se obtuvo un proceso en el que se aplican los resultados de la investigación realizada. Los logros más relevantes se mencionan a continuación:

A través del estudio realizado se detectaron deficiencias en la gestión de requisitos en los proyectos productivos de la UCI, deficiencia que podría superarse con la definición de un proceso que permita su seguimiento y control.

El proyecto productivo que emplee este proceso, tendrá un mejor control de los requisitos a lo largo de su ciclo de vida. Permitiendo conocer en todo momento el estado en que se encuentra, que artefacto lo contiene, quién será el responsable de implementarlo, entre otros aspectos, que contribuirán a la gestión del proyecto.

El proceso fue definido de manera general y puede ser aplicado a cualquier proyecto independientemente de la metodología que utilice.

Se estudiaron algunas herramientas que pueden ser utilizadas por los líderes de proyecto para gestionar requisitos una vez que empleen el proceso descrito.

## RECOMENDACIONES

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

- ◆ Aplicarle el proceso de gestión de requisitos propuesto a los diferentes proyectos productivos de la universidad.
- ◆ Divulgar la importancia del uso de herramientas en la gestión de requisitos desde el punto de vista funcional para los desarrolladores de software.
- ◆ Lograr que otros investigadores encaminen su trabajo a la adecuación y/o enriquecimiento del proceso propuesto para su posterior generalización y aplicación.

**BIBLIOGRAFÍA CITADA**

ALEXANDER, I. Requirements Engineering Tool Vendors and Freeware Suppliers.

BECK, K. *“Extreme Programming Explained. Embrace Change”*, Addison Wesley, 2000.

COMPUTING, D. O. Contributions Structure for requirements Traceability, Agosto 1995

DIV, G. A. O. W. D. F. A. G. M. S. I. Contracting for Computer Software Development Serious Problems Require Management Attention to Avoid Wasting Additional Millions., 1979.

FREDERICK P. BROOKS, J. *The Mythical Man-Month*. Addison-Wesley, 1995. p.

INGENIERÍA, T. S. E. COMPARATIVE STUDY BETWEEN REQUIREMENTS MANAGEMENT AND ENGINEERING TOOLS, 2004.

INSTITUTE, E. S. *ESPITI – European User Survey Results.* , 1996.

LAGUNA, M. A. *Ingeniería del Software II* I.T.I.Gestión p.

LETELIER, P. and V. ANAYA Integrando Especificaciones Textuales y Elementos de modelado UML en un Marco de Trabajo para Trazabilidad de Requisitos?

PRESSMAN, R. S. *Ingeniería del software un enfoque práctico*. La Habana, Félix Varela, 2005. p.

SÁNCHEZ, L. F. and L. G. ÁVILA. *Gestión de los requisitos de un proyecto software*

**BIBLIOGRAFÍA CONSULTADA**

BAÑERES, J. P. Visión ejecutiva de procesos y prácticas para desarrollo de software, 2006. Disponible en: [http://www.navegapolis.net/files/presentaciones/iempresa\\_05\\_12\\_17.ppt](http://www.navegapolis.net/files/presentaciones/iempresa_05_12_17.ppt).

CANSECO, E. Metodología MSF, 2005. Disponible en: <http://www.coworker.com.mx/nota.asp?id=28>

CONTRERAS, D. Software Consulting & IBM Rational, 2007. Disponible en:

[http://www.sowre.es/wps/wcm/resources/file/ebcb298ca6e/RAT\\_REFERENCIAS\\_20070130.pdf](http://www.sowre.es/wps/wcm/resources/file/ebcb298ca6e/RAT_REFERENCIAS_20070130.pdf)

De la Disciplina para la administración de proyectos MSF v. 1.1. Disponible en: <http://www.WillyDev.NET>

DUQUE, J. G. Especificación, verificación y mantenimiento de requisitos funcionales con técnicas de descripción formal. Dpto. de Tecnologías de las Comunicaciones ETSI de Telecomunicación. España, UNIVERSIDAD DE VIGO, 2000. p.

EMILIO A. SÁNCHEZ; PATRICIO LETELIER, et al. Mejorando la gestión de historias de usuario en eXtreme Programming, 2005. Disponible en: <http://issi.dsic.upv.es/publications/archives/f-3758/JISBD03SanchezLetelier.pdf>

GRACIA, J. Gestión de proyectos con SCRUM, 2006. Disponible en:

<http://www.ingenierossoftware.com/equipos/scrum.php>

IAN SPENCE and L. PROBASCO. Traceability Strategies for Managing Requirements with Use Cases, 2000. Disponible en: [www.rational.com](http://www.rational.com)

KOCH, N. Requirements Engineering for Web Applications – A Comparative Study, 2004. Disponible en:

<http://www.pst.informatik.uni-muenchen.de/personen/EscalonaJWE-rev.pdf>

KOTONYA; I. G AND SOMMERVILLE, et al. Requirements Engineering. Processes and Techniques, 1998.

LAGUNA, M. A. Ingeniería del Software I 3º I.T.I. Gestión, 2005. Disponible en: <http://www.infor.uva.es/~mlaguna/is1/apuntes/3-requisitos-6.pdf>

- LANDAZURI, B. A. M. Definición de Perfiles en Herramientas de Gestión de Requisitos. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software. España, Universidad Politécnica de Madrid. p.
- LETELIER, P. and V. ANAYA. Integrando Especificaciones Textuales y Elementos de modelado UML en un Marco de Trabajo para Trazabilidad de Requisitos. Departamento de Sistemas Informáticos y Computación. España, Universidad Politécnica de Valencia. p.
- LÓPEZ1, M. D. L. Á. S. Análisis de Requerimientos de Software. Estado del Arte. Instituto Politécnico Nacional Centro de Investigación en Computación 1999.
- LUTOWSKY, R. Software Requirements encapsulation quality and reuse. Estados Unidos, Auerbach Publications, 2005.
- M.J. ESCALONA; J.TORRES, et al. NDT: Navigational Development Techniques. Disponible en: <http://lsi.ugr.es/~gedes/actividades/Dolmen4/a11.pdf>
- MARTÍNEZ, J. C. M. Análisis Comparativo de Técnicas, Metodologías y Herramientas de Ingeniería de Requerimientos. Departamento de Ingeniería Eléctrica Sección de Computación. México, Centro de Investigación y de Estudios Avanzados del IPN CINVESTAV, 2004. p.
- MOLPECERES, A. Procesos de desarrollo: RUP, XP Y FDD, 2002. Disponible en: <http://www.javahispano.org/licencias>
- MORAGA, H. PROGRAMACION AGIL: "SCRUM Y XP", Universidad Mayor, 2005. Disponible en: [http://72.14.209.104/search?q=cache:SlcRRtpKSz8J:www.e-market.cl/dir/umayor/ingsw/Apoyo/GRUPO\\_1\\_PROGRAMACION\\_AGIL.ppt](http://72.14.209.104/search?q=cache:SlcRRtpKSz8J:www.e-market.cl/dir/umayor/ingsw/Apoyo/GRUPO_1_PROGRAMACION_AGIL.ppt)
- PALACIO, J. CIS, Compendio de Ingeniería del Software .Glosario Disponible en: <http://www.navegapolis.net/files/cis/CIS%20Glosario%20004.pdf>
- PALACIO, J. Requisitos del sistema ágil. Disponible en: [www.qualitatis.org](http://www.qualitatis.org)
- PALACIO, J. Gestión y procesos en empresas de software, 2005. Disponible en: [http://www.navegapolis.net/files/articulos/gestion\\_y\\_procesos.pdf](http://www.navegapolis.net/files/articulos/gestion_y_procesos.pdf)



- PALACIO, J. Gestión y modelos para la eficiencia en empresas de desarrollo de software, 2005. Disponible en: <http://www.navegapolis.net>
- PRESSMAN, R. Ingeniería del software. Un enfoque práctico.
- RAMOS., C. A. Aplicación de conceptos de gestión de proyectos y gestión de riesgo en el desarrollo productos nuevos en el campo de tecnología de información., 2005. Disponible en: <http://grad.uprm.edu/tesis/aguilarramos.pdf>
- SÁNCHEZ, E. A. and E. A. P. LETELIER. Mejorando la gestión de historias de usuario en eXtreme Programming. Disponible en: <http://issi.dsic.upv.es/publications/archives/f06878/JISBD03SanchezLetelier.pdf>
- SANCHEZ, M. A. M. Metodologías De Desarrollo De Software, 2004. Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
- SARASTY, M. F. S. AMIR-ST: Propuesta de una Aproximación Metodológica para la Ingeniería de Requisitos de Sistemas Telemáticos. Disponible en: [http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r52\\_art5\\_r.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r52_art5_r.pdf)
- SARASTY, M. F. S. AMIR-ST: Propuesta de una Aproximación Metodológica para la Ingeniería de Requisitos de Sistemas Telemáticos. Disponible en: [http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r52\\_art5\\_r.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r52_art5_r.pdf)
- SOMMERVILLE and I.SAWYER. Requirements Engineering: a good practice guide, 1997. Disponible en: <http://www.amazon.com/Requirements-Engineering-Good-Practice-/0471974447>
- SPENCE, I. and L. PROBASCO. Traceability Strategies for Managing Requirements with Use Cases, 1998. Disponible en: <http://www.rational.com>
- TORO, A. D. Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información. Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla. España, Universidad de Sevilla.

TORO, A. D. and B. B. JIMÉNEZ. Metodología para la Elicitación de Requisitos de Sistemas Software. Sevilla, Departamento de Lenguajes y Sistemas Informáticos Facultad de Informática y Estadística, 2000.

TORO, A. D. and B. B. JIMÉNEZ. Metodología para el Análisis de Requisitos de Sistemas Software Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, 2001.

YOUNG, R. R. The Requeriments Engineering Handbook.

## ANEXOS

## Anexo 1: Entrevista a proyectos productivos vinculados al desarrollo de aplicaciones

## Tema: Gestión de los Requisitos del Software.

Proyecto: \_\_\_\_\_

Facultad: 

Nombre del Entrevistado: \_\_\_\_\_

Rol: \_\_\_\_\_

1. Tamaño del Proyecto: 

2. Tipo de Aplicación que desarrolla:

 Gestión       Web       Procesamiento de Imágenes Otro. DigaCuál: \_\_\_\_\_

3. Metodología que se utiliza:

 RUP       XP       MDA Otra. DigaCuál: \_\_\_\_\_4. ¿En el proyecto realizan de alguna manera la gestión de requisitos? 

Seleccione por qué no la realiza:

 Desconocen que existe No lo consideran importante Otras causas. Mencíonelas: \_\_\_\_\_

5. Seleccione los tipos de requisitos que usted definió en su proyecto:

 Solicitudes de los clientes Casos de Uso Características funcionales Casos de prueba Características no funcionales Otros. Diga cuáles: \_\_\_\_\_

6. Definieron la trazabilidad entre:

- Entre los requisitos de igual tipo
- Entre los requisitos de diferente tipo

7. Definió atributos para los requisitos de su proyecto. Seleccione cuáles:

- Estado
- Estabilidad
- Beneficio
- Versión que lo incorporó
- Esfuerzo
- Asignado a:
- Riesgo
- Otros, Diga cuáles: \_\_\_\_\_

8. ¿Utilizan alguna herramienta para la gestión de requisitos?

En caso de utilizarla, diga cuál: \_\_\_\_\_

En el caso que usted realice la gestión de requisitos, describa brevemente como lo ejecuta.

9. ¿Se basan en algún proceso para realizar la gestión de requisitos?

---

**Anexo 2: Plantilla para describir cada una de las actividades del proceso propuesto.**

**Actividad:** \_\_\_\_\_

**Propósito:** \_\_\_\_\_

**Actor:** \_\_\_\_\_

**Artefacto de entrada:** \_\_\_\_\_

**Artefacto de salida:** \_\_\_\_\_

**Criterios de inicio:** \_\_\_\_\_

**Criterios de finalización:** \_\_\_\_\_

**Subactividades:** \_\_\_\_\_

**Secuencia:** \_\_\_\_\_

**Antes:** \_\_\_\_\_

**Después:** \_\_\_\_\_

## **Anexo 3: Plantilla del documento gestión de requisitos**

### **Capítulo 1:** Trazabilidad, artefactos y requisitos.

#### 1.1 Lista de artefactos con los tipos de requisitos definidos

#### 1.2 Gráficos de las relaciones entre los requisitos.

- ◆ Para requisitos de igual tipo
- ◆ Para requisitos de diferente tipo.

#### 1.3 Matrices de trazabilidad

- ◆ Matriz de seguimiento de orígenes: Identifica el origen de cada requisito.
- ◆ Matriz de seguimiento de características: Definidas por el cliente.
- ◆ Matriz de seguimiento de dependencias: Indica cómo se relacionan los requisitos entre sí.
- ◆ Matriz de seguimiento de artefactos: Vincula a los requisitos con los artefactos que los manejan.

#### 1.4 Informes del estado de los atributos.

### **Capítulo 2:** Gestión de proyectos.

#### 2.1 Relación de atributos por requisitos.

## GLOSARIO DE TÉRMINOS

**Asignación:** Establecimiento de lo que corresponde a algo o alguien para un determinado objetivo.

**Base de datos:** Conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

**Calidad:** Propiedad o conjunto de propiedades inherentes a una persona o cosa que permiten apreciarla con respecto a las restantes de su especie.

**Ciclo de vida:** Periodo de tiempo que comienza con la concepción del producto de software y termina cuando el producto esta disponible para su uso.

**CMMI:** Siglas de “Capability Maturity Model Integration”, modelo desarrollado por el Instituto de Ingeniería de Software (SEI) para medir el nivel de capacidad o madurez de una organización en la ejecución de los procesos relacionados con el desarrollo de un producto de software.

**Crisis del software:** Término utilizado para definir los problemas que existían y existen en el desarrollo de sistemas de software.

**Estándar:** Modelo o patrón que se sigue para realizar un proceso o una guía que se sigue para no desviarnos de un lugar al que se desea llegar.

**Extreme Programming (XP):** Metodología heterodoxa de programación. Es la más popular de las denominadas *metodologías ágiles*. Surgida a partir de la metodología de trabajo empleada por Kent Beck, Wark Cunningham y Martin Fowler en el desarrollo del proyecto C3 para Chrysler.

**Flexibilidad:** Facilidad con la que un sistema o un componente puede modificarse para ser empleado con aplicaciones o en entornos distintos para los que fue construido.

**Gestión de la configuración:** Conjunto de procesos destinados a asegurar la integridad de todo producto obtenido durante cualquiera de las etapas del desarrollo de un Sistema de Información (SI), a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo.

**IBM:** Siglas de Internacional Business Machines. Empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

**Impacto:** Conjunto de posibles efectos que puede ocasionar la modificación o cambio del entorno natural, como consecuencia de obras u otras actividades.

**Internet:** Método de interconexión de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red (lógica) única.

**Línea base:** Conjunto de elementos de configuración, formalmente revisados y aprobados (para su uso interno o para entregar al cliente), que constituyen la base para el desarrollo posterior, y que sólo puede modificarse a través de procedimientos de cambio formales.

**Metodología:** Definición del proceso de investigación que sigue a la iniciación y permite sistematizar los métodos, procedimientos y las técnicas necesarios para llevarla a cabo

**Metodologías ágiles.** Estrategias de desarrollo de software que promueven prácticas que son adaptativas en vez de predictivas; centradas en las personas o los equipos, iterativas, la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa de cliente.

**Metodologías tradicionales:** Metodologías que abordan los problemas basándose en la definición de procesos, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán.

**Microsoft Solutions Framework (MSF):** Marco para desarrollo de sistemas de software basado en principios, modelos, disciplinas, conceptos, prácticas y recomendaciones propias, derivadas de la experiencia de Microsoft. Empleada en entornos que emplean metodologías ágiles.



**Oracle:** Sistema de gestión de base de datos relacional. Es uno de los motores de bases de datos más potentes y utilizados del mercado.

**OTAN:** Siglas utilizadas para la Organización del Tratado del Atlántico Norte

**Plantilla:** Forma de dispositivo que proporciona una separación entre la forma o estructura y el contenido. Es un medio o un instrumento que permite guiar, portar o construir un diseño o esquema predefinido

**Producto de software:** Conjunto de programas, procedimiento y opcionalmente documentación asociada que se entrega al usuario como resultado.

**Rational Unified Process (RUP):** Proceso de Ingeniería del Software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades en las organizaciones de desarrollo de software. Se trata de un proceso integrado en un producto, desarrollado y mantenido por Rational Software, e integrado en su conjunto de herramientas de desarrollo.

**Repositorio:** Sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos

**Reutilización de software:** Proceso de implementar o actualizar sistemas de software usando activos de software existentes.

**Riesgo:** Posibilidad de que un peligro pueda llegar a materializarse

**Scrum:** Metodología ágil, aplicada originalmente por Jeff Sutherland y elaborada más formalmente por Ken Schwaber. Scrum aplica principios de control industrial, junto con experiencias metodológicas de Microsoft, Borland y Hewlet Packard.

**Sistema de software:** Conjunto de programas de ordenador, procedimientos y opcionalmente la documentación y datos asociados, necesarios para el funcionamiento de un sistema.

**Microsoft SQL Server:** Sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

**Trazabilidad:** Conjunto de medidas, acciones y procedimientos que permiten registrar e identificar cada producto desde su origen hasta su destino final.

**Unix:** Sistema operativo portable, multitarea y multiusuario, desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT & T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy

**Validación:** Confirmación mediante examen y aportación de pruebas objetivas de que se cumplen los requisitos concretos para un uso determinado.

**Viabilidad:** Condición que hace posible el funcionamiento del sistema, proyecto o idea al que califica, atendiendo a sus características tecnológicas y a las leyes de la naturaleza involucradas.

**Windows:** Sistema operativo con interfaz gráfica para computadoras personales cuyo propietario es la empresa Microsoft.

**WBS:** Siglas de Work Breakdown Structure. Método para representar jerárquicamente las partes de un proyecto, proceso o producto.