

Universidad de las Ciencias Informáticas

Facultad 1



Título: Propuesta de indicadores para priorizar los requisitos funcionales de los proyectos de software de la Universidad de las Ciencias Informáticas.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

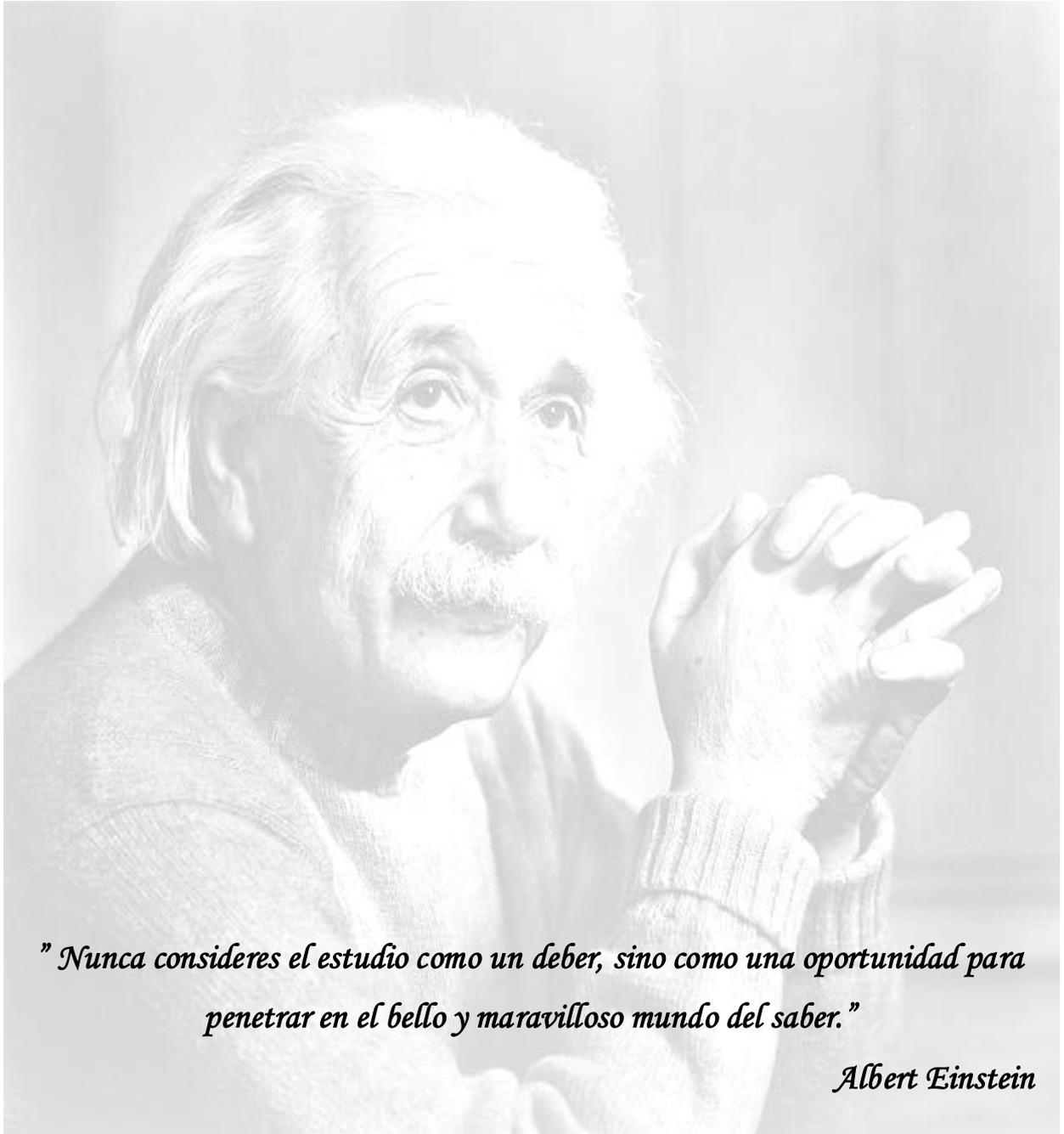
Yaité Armero Cardona
Medaimy Rivera Valiente

Tutor:

Ing. Maidileydys Castellano Báez

Ciudad de La Habana, Cuba.

Junio, 2010



"Nunca consideres el estudio como un deber, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: “Propuesta de indicadores para priorizar los requisitos funcionales en los proyectos de software de la Universidad de las Ciencias Informáticas”, y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Medaimy Rivera Valiente

Yaité Armero Cardona

Firma del Autor

Firma del Autor

Maidileydys Castellano Báez

Firma del Tutor

DATOS DE CONTACTO

DATOS DE CONTACTO

Medaimy Rivera Valiente

Correo: mvaliente@estudiantes.uci.cu.

Ciudad de La Habana, Cuba

Yaité Armero Cardona

Correo: yarmero@estudiantes.uci.cu.

Ciudad de La Habana, Cuba

Ing. Maidileydys Castellano Báez.

Ingeniera en Ciencias Informáticas.

Correo: mcastellano@uci.cu, teléfono 835-8863

Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba.

AGRADECIMIENTOS

Agradecimientos generales:

Los agradecimientos especiales son para nuestra tutora que ha estado con nosotras en todo momento alentándonos para llegar al final. Ella ha sido el motor impulsor de este trabajo. Más que una tutora ha sido una guía.

Agradecemos a todas esas personas que han dedicado un minuto de su preciado tiempo para aportar un granito de arena a este trabajo, en especial a Geidis y Eilén que también han sido como tutoras y a esos expertos que a pesar de ser personas ocupadas nos brindaron su ayuda Febe Ángel, Karina, Kenia, Tito, Aracelis, Sacha, Yudenia y Héctor.

AGRADECIMIENTOS

Deseo expresar mis más sinceras muestras de agradecimiento, a todas las personas que de una forma u otra han colaborado en la realización de este sueño:

A mi mamá, por ser mi amiga, por apoyarme en todas las decisiones que he tomado en la vida, por confiar en mí, por darme su cariño, por su apoyo y por ayudarme a enfrentar los desafíos más grandes de mi vida. Espero que estés orgullosa de mí como yo lo estoy de ti. Mami te quiero mucho.

A mi hermano Medardito, a quien quiero con la vida, quien ha puesto siempre mis sueños por encima de los suyos propios, sin él no hubiese sido posible este sueño.

A Jose, mi incansable segundo papá, muchas gracias por el amor y la dedicación que tienes para con mi madre, gracias por el cariño que me has dado en estos años.

A mi novio, porque sin él estos cinco años hubiesen sido muy difíciles, gracias por confiar en mí, por ayudarme y apoyarme siempre que lo necesito, gracias por ser incondicional. Te amo mi amor.

A mi familia, en especial a mis primos Alex y Eskarly por estar siempre ahí, por cuidarme, por enseñarme las cosas buenas de la vida, por apoyarme siempre sin esperar nada a cambio, gracias por todas las cosas bellas que me han dado, este sueño es de todos, espero que se sientan orgullosos de mí.

A la maravillosa familia de Dario por haberme acogido como un miembro más, por el cariño y la atención que me han dado.

A mis amigos Ailedys, Javier, Claribel, Mónica y Alain por demostrarme que la vida es hermosa, cuando se tienen buenos amigos, gracias por hacer de estos cinco años una experiencia inolvidable, aunque el destino nos separe nunca los olvidaré.

Gracias a mi compañera de tesis, a mi tutora por su paciencia y dedicación, por su guía y apoyo en la elaboración de este trabajo.

Medaimy Rivera Valiente

AGRADECIMIENTOS

Realmente no sé por dónde comenzar, porque son tantas a las personas que tengo que agradecerles por haberme ayudado a llegar hasta aquí, hasta este momento tan especial.

Quiero agradecerle en primer lugar a la persona que más amo en la vida, mi madre que más que eso ha sido mi padre, hermana, amiga y compañera. Que supo guiarme por el mejor camino y siempre me dio de ella lo mejor. Gracias mamita por estar a mi lado.

A los demás miembros de mi familia que tanto me han apoyado en mis decisiones y que tanto me quieren.

A mi padrastro que ha sido el mejor padre que he tenido y me ha dado todo su amor y cariño.

A mis grandes amigos que han estado conmigo en los momentos difíciles y en los felices también, muchas gracias a ustedes que forman parte de mi gran familia.

A todos los que han sido compañeros míos de aula desde que comencé mis estudios, son muy especiales para mí, porque se que me tienen presente siempre, en especial a los que compartieron tres años conmigo en el pre y al primer grupo de la universidad, incluyendo a Iviannis.

Agradecerle muchísimo a dos personas que cuando entré a la universidad me guiaron y fueron como hermanas para mí: Sula y Neyo.

A Miguel por darme tantas fuerzas para seguir adelante cuando menos yo creía que podía.

A mis vecinos que tanto me han malcriado y consentido.

Gracias a mi novio por aparecer en mi vida cuando más lo necesitaba y a su familia por acogerme de la forma que lo han hecho.

Muchas gracias a mi compañera por ayudarme en este trabajo y a la revolución cubana por darme esta oportunidad.

Gracias a todas las personas que conozco y que me quieren, personas que han confiado en mí más que yo misma. En fin, gracias a TODAS esas personas lindas que han hecho de mí esta mujer que soy ahora.

Yaité Armero Cardona

DEDICATORIA

Dedicatoria de Medaimy

*A Yaya y a Mima, porque aunque no estén aquí sé que me acompañan espiritualmente,
ojalá y me hubiesen visto convertirme en la persona que soy,
ojalá y estén orgullosas de mí.
Les dedico mi tesis con todo mi corazón.
Nunca las olvidaré.....*

Dedicatoria de Yaité

*Dedico este trabajo de diploma a mi mamá
que tanto se esforzó y luchó porque yo llegara hasta aquí
y ha sido la fuente de toda mi inspiración.
A mi abuelita del alma que ha estado siempre conmigo
dándome apoyo en los momentos más difíciles.
A mi padrastro que ha sido como un padre y me ha guiado por el mejor camino.*

RESUMEN

La Universidad de las Ciencias Informáticas desarrolla dentro de sus procesos fundamentales la producción de software. Una de las etapas de este proceso es la ingeniería de requisitos, esta contiene cuatro fases. La fase de análisis y obtención de requisitos incluye la priorización de requisitos de software que permite obtener las categorías de los requisitos teniendo en cuenta indicadores.

Actualmente en la Universidad existen proyectos que presentan problemas debido a la incorrecta priorización de requisitos, lo que deriva la necesidad de mejorar la asignación de prioridades a los mismos. A partir de esta situación el presente trabajo se propuso obtener un conjunto de indicadores para priorizar los requisitos funcionales de los proyectos de software de la Universidad de las Ciencias Informáticas.

Los resultados obtenidos del trabajo realizado que se recogen en este documento muestran los problemas actuales en la priorización de requisitos funcionales en los proyectos de software de la Universidad de las Ciencias Informáticas. Se realiza un estudio de todos los indicadores existentes para realizar la priorización de requisitos y se obtiene una propuesta de los más adecuados para resolver el problema planteado. Además, se proponen un conjunto de pasos para medir los indicadores propuestos de manera cualitativa y cuantitativa. Para finalizar se realizó la validación de la propuesta de solución con expertos en el tema, la cual mostró que dicha propuesta garantiza la mejora de la priorización de requisitos funcionales en los proyectos de software de la Universidad de las Ciencias Informáticas.

Palabras claves: priorización, requisitos funcionales, indicadores.

ÍNDICE

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1. Introducción	6
1.2. Ingeniería de requisitos en los proyectos de software	6
1.2.1. Fases de la ingeniería de requisitos	7
1.3. Priorización de requisitos	11
1.3.1. Indicadores	12
1.3.2. Tipos de indicadores	12
1.4. Métodos para priorizar requisitos	13
1.4.1. Decisión basada en Costo-Valor usando el Proceso de Análisis Jerárquico (AHP)	14
1.4.2. El despliegue de la función de la calidad del software (<i>Software Quality Function Deployment, SQFD</i>)	15
1.4.3. Wieggers	15
1.4.4. Teoría-W	16
1.5. Metodologías en la priorización de requisitos	16
1.5.1. Proceso Unificado de Desarrollo (<i>Rational Unified Process, RUP</i>)	16
1.5.2. Scrum	18
1.5.3. Programación Extrema (<i>Extreme Programming, XP</i>)	18
1.5.4. Desarrollo Basado en Funcionalidades (<i>Feature Driven Development, FDD</i>)	19
1.5.5. Elicitación de requisitos- Desarrollo Global de Software (<i>RE -GSD</i>)	20
1.5.6. MSF for CMMI	20
1.5.7. Método de Desarrollo de Sistemas Dinámicos (<i>Dynamic Systems Development Method, DSDM</i>)	21
1.6. Herramientas para priorizar requisitos	22
1.7. Limitantes de los enfoques estudiados	23
1.8. Conclusiones parciales	25
CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN	26
2.1. Introducción	26
2.2. Características de la organización	26
2.2.1. Producción en la UCI	26
2.2.2. Programa de mejora	28

ÍNDICE

2.3. Población, muestra y técnicas utilizadas en la investigación	28
2.4. Análisis e interpretación de los resultados de la entrevista y la encuesta.	29
2.5. Conclusiones del diagnóstico	40
2.6. Propuesta de solución	40
2.7. Conclusiones parciales	51
CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA	52
3.1. Introducción	52
3.2. Juicios de expertos	52
3.3. Proceso de selección de expertos	53
3.3.1. Determinar la cantidad de expertos	54
3.3.2. Conformar el listado de los expertos	54
3.3.3. Confirmar la participación de los expertos	54
3.4. Elaboración de la encuesta	54
3.5. Resultado de la validación	55
3.6. Conclusiones parciales	59
CONCLUSIONES	60
RECOMENDACIONES	61
REFERENCIAS BIBLIOGRÁFICAS	62
BIBLIOGRAFÍA	64
GLOSARIO DE TÉRMINOS	67

INTRODUCCIÓN

INTRODUCCIÓN

El software es un componente imprescindible de una amplia gama de dispositivos que se utilizan continuamente en diferentes esferas de la vida social. El desarrollo de este no es un proceso sencillo, presenta mucha incertidumbre en su realización lo que ha conllevado al surgimiento de un gran número de modelos, metodologías, teorías y normas a seguir en busca de obtener mejoras en la productividad y calidad del mismo.

Una metodología de desarrollo no es más que una colección de documentos formales referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo de software, que tienen como finalidad garantizar la eficacia y la eficiencia en el proceso de generación de software. Estas se dividen en dos grupos, las metodologías ágiles y las metodologías pesadas, ambas desarrollan el proceso de ingeniería de requisitos.

Uno de los aspectos esenciales que intervienen en la obtención de un software con calidad es el proceso de levantamiento de requisitos. De acuerdo con el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) un requisito es una condición o capacidad que debe tener un sistema para satisfacer las necesidades de un cliente, con el fin de resolver un problema planteado por el mismo en forma de un documento formal.

Con el objetivo de obtener resultados satisfactorios en el desarrollo de software, se lleva a cabo la ingeniería de requisitos y dentro de ella la priorización de requisitos, la cual debe ser simple y generar resultados confiables y exactos, garantizando la satisfacción de clientes y desarrolladores. La priorización de los requisitos facilita la toma de decisiones dentro de los proyectos de software, así como el poder determinar aquellas funcionalidades importantes que deben ser desarrolladas de forma inmediata.

El modelo de guía para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software (CMMI), propone entre sus mejores prácticas relacionadas con la gestión de requisitos, que la priorización de los mismos sea utilizada para determinar aquellos que se deberían cumplir en la primera versión y aquellos que pueden llevarse a cabo en sucesivas versiones. La incorrecta aplicación de indicadores para priorizar los requisitos funcionales puede acarrear problemas en la definición de los objetivos de los proyectos de software, en la asignación de responsabilidades, recursos y en la flexibilidad del producto, e incluso provocar, en algunos casos la cancelación de un producto determinado debido a la insatisfacción del cliente.

En Cuba, la Universidad de las Ciencias Informáticas (UCI) ha trascendido en la producción de software, haciendo grandes aportes a la industria nacional del mismo. Con el fin de lograr una

INTRODUCCIÓN

organización en dicho proceso, se ha tomado como estándar la aplicación de técnicas para la Ingeniería de Software. En la UCI se utilizan diferentes metodologías de desarrollo que permiten realizar la disciplina de levantamiento de requisitos, entre las más usadas se encuentran: Programación Extrema (XP), Scrum, Desarrollo Basado en Funcionalidades (FDD), Proceso Unificado de Desarrollo (RUP) y Scrum –XP.

Actualmente la Universidad utiliza metodologías como RUP y Scrum que plantean criterios e indicadores para llevar a cabo la priorización de requisitos. RUP propone cuatro criterios generales y Scrum se basa en el método Retorno de Inversión (ROI) haciendo uso de los indicadores costo y beneficio. Pese a esto, la gran mayoría de los proyectos no tienen indicadores definidos a la hora de determinar la prioridad de los requisitos y en la gran mayoría de los casos, hacen uso de sólo aquellos indicadores que propone la metodología que utilizan, los cuales no son suficientes. Debido a esto, las funcionalidades implementadas no son las más necesarias o las más importantes, por lo que se incurre en gastos de tiempo y recursos, variando el tiempo estimado del proyecto, provocando consigo la insatisfacción del cliente.

Por lo anteriormente expuesto se define como **problema científico** de la investigación: ¿Cómo mejorar la priorización de los requisitos funcionales en los proyectos de software desarrollados en la Universidad de las Ciencias Informáticas?

El **objeto de estudio** en el que se enmarca este trabajo de diploma es la ingeniería de requisitos en los proyectos de software; cuyo **campo de acción** abarca la priorización de los requisitos funcionales en los proyectos de software desarrollados en la Universidad de las Ciencias Informáticas.

Para dar solución al problema planteado anteriormente se traza el siguiente **objetivo general**: Proponer indicadores para priorizar los requisitos funcionales en los proyectos de software desarrollados en la Universidad de las Ciencias Informáticas.

Del objetivo general se derivan los siguientes **objetivos específicos**:

- ✓ Analizar el marco teórico de la investigación.
- ✓ Realizar un diagnóstico sobre la priorización de los requisitos funcionales en los proyectos de la Universidad de las Ciencias Informáticas.
- ✓ Obtener los indicadores para priorizar los requisitos funcionales de los proyectos de la Universidad de las Ciencias Informáticas.
- ✓ Validar la propuesta de solución.

La investigación se sustenta en el planteamiento de la siguiente **hipótesis**: Si se proponen indicadores para priorizar los requisitos funcionales en los proyectos de software desarrollados en la Universidad de las Ciencias Informáticas se logrará mejorar la priorización de dichos requisitos.

INTRODUCCIÓN

Tomando en cuenta la hipótesis anterior se define como **variable independiente**: Propuesta de indicadores para priorizar los requisitos funcionales y como **dependiente**: Mejorar la priorización de los requisitos.

A continuación se muestra en la tabla 1 la **operacionalización de dichas variables**:

Variable	Dimensión	Indicadores	Sub-indicadores	Unidad de Medida
Propuesta de indicadores para priorizar los requisitos funcionales.	Proyectos de software desarrollados en la Universidad de las Ciencias Informáticas.	Posibilidad de aplicación	Bajo	1
			Regular	2
			Bien	3
			Muy Bien	4
			Excelente	5
		Adaptabilidad	Bajo	1
			Regular	2
			Bien	3
			Muy Bien	4
			Excelente	5
		Facilidad de uso	Muy difícil	1
			Difícil	2
			Aceptable	3
			Fácil	4
			Muy fácil	5
Mejorar la priorización de los requisitos.	Proyectos de software desarrollados en la Universidad de las Ciencias Informáticas.	Mejora de la priorización de los requisitos funcionales	Bajo	1
			Regular	2
			Bien	3
			Muy Bien	4
			Excelente	5

Tabla 1: Operacionalización de las variables.

Para lograr el cumplimiento de los objetivos y así darle solución al problema planteado, se proponen las siguientes **tareas científicas**:

- ✓ Realización de un estudio sobre la utilización de indicadores para priorizar los requisitos funcionales de los proyectos de software.

INTRODUCCIÓN

- ✓ Identificación de los problemas existentes en la priorización de los requisitos funcionales de los proyectos de software de la Universidad de las Ciencias Informáticas.
- ✓ Descripción de los indicadores para priorizar los requisitos funcionales de los proyectos de la Universidad de las Ciencias Informáticas.
- ✓ Definición de las métricas necesarias para medir los indicadores establecidos.
- ✓ Validación de los indicadores establecidos con la técnica juicio de experto aplicada a un grupo de expertos.

Para llevar a cabo estas tareas se emplearon métodos teóricos y empíricos de la investigación científica.

Los **métodos teóricos** empleados para cumplir con las tareas a desarrollar son:

- ✓ Analítico sintético: Se realiza un análisis de la bibliografía referente a la priorización de los requisitos funcionales para sintetizar los aspectos más importantes en la investigación.
- ✓ Histórico lógico: Se utiliza para estudiar los antecedentes de la priorización de los requisitos funcionales.

Los **métodos empíricos** utilizados para obtener información del objeto de estudio son:

- ✓ Entrevista: Se utiliza para obtener la situación real de los proyectos de software desarrollados en la Universidad de las Ciencias Informáticas en cuanto a la priorización de requisitos funcionales y entender sus especificidades para poder elaborar la propuesta de solución.
- ✓ Encuesta: Se utiliza para obtener la situación real de los proyectos de software desarrollados en la Universidad de las Ciencias Informáticas en cuanto a la priorización de requisitos funcionales y para comprobar la factibilidad de la propuesta que sea planteada a través del juicio de expertos.

El trabajo de diploma quedará estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica

En este capítulo se realiza un estudio de la ingeniería de requisitos y en específico la priorización de requisitos a nivel mundial. Para ello se lleva a cabo un análisis profundo sobre los diferentes métodos, metodologías y herramientas, haciendo énfasis en el estudio de los indicadores que utilizan para priorizar requisitos funcionales.

Capítulo 2: Diagnóstico del estado actual del problema y propuesta de solución

En este capítulo se presentan los resultados obtenidos del estudio realizado a la priorización de requisitos en los proyectos de software de la Universidad de las Ciencias Informáticas a través de las técnicas: encuesta y entrevista. Se propone un conjunto de indicadores para priorizar los requisitos

INTRODUCCIÓN

funcionales en los proyectos de software que se desarrollan en la Universidad de las Ciencias Informáticas y un conjunto de pasos para evaluar dichos indicadores.

Capítulo 3: Validación de la propuesta

En este capítulo se valida el resultado de la investigación mediante el criterio de expertos a través de una encuesta. Se detalla el proceso de selección del panel de expertos, los aspectos esenciales que se tuvo en cuenta para la elaboración de la encuesta y los resultados obtenidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se abordan conceptos relacionados con el proceso de ingeniería de requisitos tales como: qué es un requisito y las diferentes fases que lo componen. Se describen aspectos fundamentales dentro de este proceso, como es el caso de la priorización de los requisitos y el uso de indicadores para determinar la misma. También se realiza un análisis de algunas de las metodologías de desarrollo más utilizadas a nivel mundial como RUP¹, FDD², Scrum, MSF³ for CMMI⁴, XP⁵ y DSDM⁶, haciendo énfasis en los indicadores que utilizan. Por último, se procede al estudio de métodos y herramientas, haciendo una breve descripción y analizando detalladamente los indicadores que utilizan para definir la prioridad de los requisitos.

1.2. Ingeniería de requisitos en los proyectos de software

La ingeniería de requisitos fue definida por Bradley Thayer en el año 1995 como “la ciencia y disciplina concerniente al análisis y documentación de los requisitos de software, a través de un proceso iterativo de definición, análisis, estudio de balance, elaboración de prototipos y simulación.” (Thayer, 1995)

Fue reconocida también como "el desarrollo sistemático de los requisitos a través de un proceso iterativo y cooperativo en el que se analiza el problema, se documenta el resultado en diversos formatos de representación, y se comprueba la exactitud de la comprensión alcanzada." (Karakostas, 1995)

Ian Sommerville plantea que la ingeniería de requisitos es una “etapa particularmente crítica en el proceso de software ya que los errores en esta etapa originan inevitables problemas posteriores en el diseño e implementación del sistema.” (Sommerville, 2005)

El éxito del desarrollo de software requiere colaboración y comunicación fluida entre clientes, usuarios y los desarrolladores del mismo, cuanto más completo y menos ambiguo sea el conjunto de requisitos, menor será la probabilidad del fracaso. La ingeniería de requisitos facilita el mecanismo apropiado para

¹ RUP: Proceso Unificado de Desarrollo (Rational Unified Process)

² FDD: Desarrollo Basado en Funcionalidades (Feature Driven Development)

³ MSF: Solución de Microsoft del Marco de trabajo (Microsoft Solution Framework)

⁴ CMMI: Modelo de guía para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software

⁵ XP: Programación Extrema (Extreme Programming)

⁶ DSDM: Método de Desarrollo de Sistemas Dinámicos (Dynamic Systems Development Method)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

comprender lo que desea el cliente. Es considerada además el fundamento básico de cualquier software.

Según el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) un requisito es: “Una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. Una condición o capacidad que debe estar presente en un sistema o componentes de un sistema para satisfacer un contrato, estándar, especificación u otro documento formal.” (Chaves, 2005)

Se puede entender por requisito algo que el producto debe hacer o una característica que debe tener o simplemente objetivos y especificaciones señaladas por el cliente para un sistema. La tarea de todo analista de requisitos es conversar con las personas involucradas en el proceso, entenderlas y escuchar lo que dicen, para entender lo que necesitan.

1.2.1. Fases de la ingeniería de requisitos

La ingeniería de requisitos es la fase de la ingeniería de software donde se definen las propiedades y la estructura del software. El modelo de guía para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software (CMMI), divide el proceso en cinco fases:

- ✓ “Obtención (elicitación) de requisitos
- ✓ Análisis de requisitos
- ✓ Especificación de requisitos
- ✓ Validación de requisitos
- ✓ Administración de requisitos” (CMMI, 2006)

Otros autores, como Pressman, describen el proceso en 6 fases distintas:

- ✓ “Identificación de Requisitos
- ✓ Análisis de Requisitos y Negociación
- ✓ Especificación de Requisitos
- ✓ Modelado del Sistema
- ✓ Validación de Requisitos
- ✓ Gestión de Requisitos” (Pressman, 2006)

Ian Sommerville en el libro Ingeniería del Software plantea cuatro fases principales, como se muestra en la Figura 1, las cuales están dentro del proceso de ingeniería de requisitos, estas son:

- ✓ “Estudio de viabilidad
- ✓ Obtención y análisis de requisitos
- ✓ Especificación de requisitos

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

✓ Validación de requisitos” (Sommerville, 2005)

Es válido aclarar que muchos autores nombran y dividen de formas diferentes este proceso pero independientemente de ello siguen el mismo procedimiento para analizar y documentar los requisitos de software. A partir del análisis realizado las autoras del presente trabajo consideran que las cuatro fases que plantea Ian Sommerville recogen dentro de ellas las seis fases que define Pressman y las cinco fases definidas por CMMI, por tanto el estudio de la presente investigación se basa en estas cuatro fases que se muestran en la Figura 1.

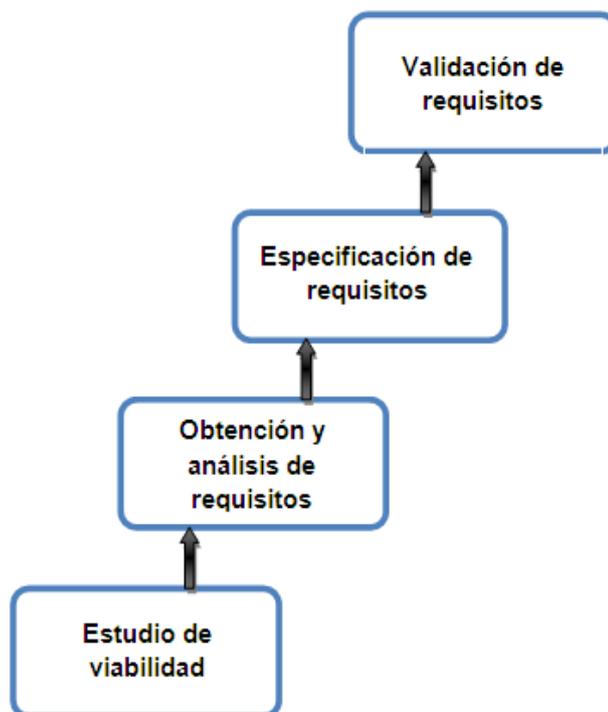


Figura 1 : Fases de la ingeniería de requisitos.

1.2.1.1. Estudio de viabilidad

El estudio de viabilidad es una de las fases del proceso de ingeniería de requisitos donde “se estima si las necesidades del usuario se pueden satisfacer con las tecnologías actuales del software y hardware. Este estudio analiza si el sistema propuesto será rentable desde un punto de vista de negocios y si se puede desarrollar dentro de las restricciones de presupuesto existentes.” (Sommerville, 2005)

El estudio de la viabilidad no es más que el análisis que se debe realizar en cuanto a las necesidades del mercado para ver si es conveniente la realización del software que se quiere desarrollar y saber si

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

se cuenta con el presupuesto para desarrollar el mismo, al igual que conocer si las tecnologías que posee el equipo de desarrollo son las que resolverán el problema que el cliente plantea.

1.2.1.2. Obtención y análisis de requisitos

La obtención y análisis de requisitos es la fase de la ingeniería de requisitos en la cual se define un proceso para “obtener los requisitos del sistema por medio de la observación de los sistemas existentes, discusión con los usuarios potenciales y proveedores, el análisis de tareas, etcétera. Esto puede implicar el desarrollo de uno o más modelos y prototipos del sistema que ayudan al analista a comprender el sistema a especificar.” (Sommerville, 2005) Es un proceso iterativo.

Según CMMI es el “uso de técnicas sistemáticas, como prototipos y entrevistas estructuradas, para proactivamente identificar y documentar las necesidades de los clientes y de los usuarios finales.” (CMMI, 2006)

Dentro de esta fase se definen un conjunto de actividades:

- ✓ “Descubrimiento de requisitos: Es el proceso de interactuar con los *stakeholders*⁷ del sistema para recopilar sus requisitos.
- ✓ Clasificación y organización de los requisitos: Esta actividad toma la recopilación no estructurada de requisitos, grupos relacionados de requisitos y los organiza en grupos coherentes.
- ✓ Ordenación por prioridades y negociación de requisitos: Inevitablemente, cuando existen muchos *stakeholders* involucrados, los requisitos entrarán en conflicto. Esta actividad se refiere a ordenar según las prioridades de los requisitos, y a encontrar y resolver los requisitos en conflicto a través de la negociación.
- ✓ Documentación de requisitos: Se documentan los requisitos.” (Sommerville, 2005)

El gran desafío es averiguar y expresar claramente lo que los clientes necesitan. A veces bastan las palabras, pero otras veces las figuras o tablas son de gran ayuda. El cliente suele tener una idea vaga, inconsciente e incompleta de lo que espera de su aplicación. Un buen proceso de obtención de requisitos debe tener en cuenta una serie de atributos:

- ✓ “Deben ser completos, consistentes y han de estar dentro del alcance del proyecto
- ✓ Deben tener un único identificador
- ✓ Cumplen con los objetivos de los clientes

⁷ Stakeholders: Clientes y usuarios que intervienen en el proceso de desarrollo de software.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Son viables y apropiados para el desarrollo
- ✓ Los requisitos han de ser *testeables*⁸ (INTECO, 2008)

Existen múltiples técnicas que pueden ayudar a la hora de recoger los requisitos de un producto. Algunas son bastante conocidas como por ejemplo: realizar entrevistas, reuniones, cuestionarios y otras como la utilización de casos de uso, escenarios o prototipos que son menos conocidas.

1.2.1.3. Especificación de requisito

La especificación de requisitos es la fase de la ingeniería de requisitos que se encarga de “traducir la información recopilada durante la actividad de análisis en un documento que define un conjunto de requisitos. En este documento se pueden incluir dos tipos de requisitos: los requisitos del usuario, que son aclaraciones abstractas de los requisitos del cliente y del usuario final del sistema, y los requisitos del sistema, que son una descripción más detallada de la funcionalidad a proporcionar.” (Sommerville, 2005)

Para una correcta especificación se hace necesario realizar una definición de los requisitos, en la que se tengan en cuenta las perspectivas del usuario. También se deben documentar los requisitos, a pesar de que esto pueda parecer una tarea tediosa, es la única manera de asegurar que la esencia de los requisitos ha sido capturada correctamente.

1.2.1.4. Validación de requisitos

La validación de requisitos es la fase que “comprueba la veracidad, consistencia y completitud de los requisitos. Durante este proceso, inevitablemente se descubren errores en el documento de requisitos. Se debe modificar entonces para corregir estos problemas.” (Sommerville, 2005) Se examinan todos los requisitos registrados en documentos para asegurarse de que el sistema posea todo lo que el usuario espera. Otra razón por la que el proyecto tiene que ser validado es para prevenir posibles fugas de requisitos. Los requisitos, en ocasiones, aparecen en las especificaciones sin que nadie realmente sepa de donde vienen o qué valor añaden al producto.

Otro punto muy importante a tener en cuenta es asegurarse de que los requisitos tengan consistencia, y en caso contrario, que cualquier conflicto entre los requisitos haya sido resuelto. Dos requisitos están

⁸ *Testeables*: Deben tener capacidad de prueba.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

en conflicto cuando no es posible que puedan implementarse juntos, es decir, si la solución a un requisito impide la implementación de otro.

1.3. Priorización de requisitos

La priorización de los requisitos es una de las actividades que se desarrolla en la fase de obtención y análisis de requisitos la cual forma parte del proceso de ingeniería de requisitos. Dicha priorización es conveniente para planificar correctamente el alcance de los proyectos en cada fase o iteración de su desarrollo.

“Esta actividad tiene como objetivo:

- ✓ Asignar categorías a los requisitos en función de indicadores.
- ✓ Definir las versiones entregables del proyecto.

Esta práctica permite realizar una organización de la construcción o desarrollo de los requisitos de acuerdo con las necesidades del cliente o la importancia que tienen los requisitos para él.

Es importante realizar la priorización de requisitos debido a que se obtendrán claros objetivos de desarrollo, se definirá la entrada y salida de cada una de las etapas del proyecto, se podrá hacer una organización y distribución de actividades para el desarrollo incremental que permitirá por tanto negociar el cómo se liberará el producto.” (Merchán, y otros, 2008)

La tarea de asignar prioridades requiere de la participación de clientes y usuarios con cierto nivel de decisión y puede realizarse de diversas maneras, tales como reuniones, cuestionarios y otras. Se debe determinar la importancia relativa que tiene un requisito para los clientes y usuarios, y organizar aquellos requisitos que deben implementarse inicialmente frente a aquellos que pueden postergarse.

Al asignar prioridades, se deben tener en cuenta la dependencia entre requisitos, la multiplicidad de intereses de los clientes y usuarios, las limitaciones de recursos, entre otros aspectos. Por ende, los ingenieros de software deberán orientar a los clientes y usuarios respecto a estas contingencias.

Este primer objetivo de la priorización de requisitos es precisamente donde se enmarca la presente investigación. El propósito de la misma es determinar el conjunto de indicadores que permitan establecer la prioridad de los requisitos definidos en un proyecto de software. Lo anteriormente expuesto posibilitará que en un segundo momento el equipo de trabajo pueda continuar con el segundo objetivo de la priorización, el cual debe generar la planificación de las iteraciones de desarrollo y los entregables de estas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3.1. Indicadores

Según la norma ISO 9126 “un indicador es una medición que se puede realizar para estimar o pronosticar otra medida.” (Calisoft Centro de Calidad para Soluciones Tecnológicas, 2009) Se refiere a datos esencialmente cuantitativos pero que también pueden ser cualitativos, que permiten darse cuenta de cómo se encuentran las cosas en relación con algún aspecto de la realidad que interesa conocer. Los indicadores pueden ser medidas, números, hechos, opiniones o percepciones que señalen condiciones o situaciones específicas.

Se deben caracterizar por ser estables y comprensibles, por tanto, no es suficiente con uno solo de ellos para medir la prioridad de un requisito sino que se impone la necesidad de considerar un sistema de indicadores, es decir, un conjunto interrelacionado de ellos que abarquen la mayor cantidad posible de magnitudes a medir, pero el número de estos que se decidan establecer tampoco puede ser muy elevado debido a que se necesita mantener el control de la información, para poder hacer una mejor interpretación de los mismos.

Los indicadores son de gran importancia ya que permiten medir cambios en la condición o situación analizada a través del tiempo. Facilitan mirar de cerca los resultados de iniciativas o acciones. Son instrumentos valiosos para orientar en la obtención de mejores resultados dentro de los proyectos de desarrollo de software.

1.3.2. Tipos de indicadores

A la hora de definir indicadores para determinar la prioridad de los requisitos funcionales es necesario dominar conceptos referentes al carácter cuantitativo y cualitativo de los mismos. Dicho carácter establece que los indicadores pueden ser de dos tipos.

- ✓ Indicadores cuantitativos: Son aquellos que se refieren directamente a medidas en números o cantidades.
- ✓ Indicadores cualitativos: Son los que se refieren a cualidades. Se trata de opiniones, percepciones o juicio de parte de las personas sobre algo, no son cuantificados directamente.

Existen diferentes indicadores que se establecen en el proceso de priorización de requisitos, los mismos pueden estar incluidos en los métodos utilizados por los proyectos de desarrollo de software o en las metodologías que incluyen dentro de sus actividades la priorización. Como ejemplos de estos indicadores se muestran los siguientes:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Indicadores	Descripción
Complejidad	Se identifican criterios para estimar el esfuerzo de implementación de cada uno de los requisitos tales como: complejidad por número de transacciones, complejidad por entidades candidatas, complejidad por interfaces de comunicación con actores, complejidad por número de requisitos no funcionales, complejidad por tipo de tecnología y complejidad por reutilización.
Beneficios	Determina la importancia de acuerdo con las necesidades que se identifican en el proceso de negocio.
Dependencia	Se establece a través de las relaciones que tienen unos requisitos con otros.
Estabilidad	Depende del criterio de estabilidad del requisito ante los procesos de cambio que ocurren en la organización.
Frecuencia	Identifica el nivel de ocurrencia de los requisitos para poder determinar cuáles son los que tienen que ser implementados de forma inmediata.
Costo	Es el impacto monetario que implica la implementación de un software.

Tabla 1.1: Indicadores.

1.4. Métodos para priorizar requisitos

Existen disímiles métodos que pueden ser utilizados por los equipos de proyecto en el proceso de desarrollo de software que permiten asignar prioridades a los requisitos a través del uso de diferentes indicadores. Estos métodos pueden llegar a ser simples o complejos involucrando directamente a los desarrolladores, a los clientes y usuarios del sistema. Algunos autores clasifican los requisitos como importantes o no, obligatorios o postergables dentro de una escala de importancia de los mismos, utilizando estos dentro de los métodos para facilitar la toma de decisiones dentro de los proyectos de desarrollo de software. A continuación se muestran algunos métodos de asignación de prioridades a requisitos, que establecen diferentes indicadores.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4.1. Decisión basada en Costo-Valor usando el Proceso de Análisis Jerárquico (AHP)

El método Decisión basada en Costo-Valor usando el Proceso de Análisis Jerárquico (AHP) es utilizado para obtener la priorización de los requisitos de software. Este método que se realiza de manera simple, incluye cinco etapas donde intervienen los miembros designados del equipo de proyecto junto con los *stakeholders*.

Los indicadores de costo, valor y calidad son el fundamento del método, según los autores del mismo. La calidad del producto es reflejada en la contribución de un requisito a satisfacción del cliente y el par costo-valor en el valor que tiene un requisito para un cliente y el costo que implica para los desarrolladores su implementación.

Para medir los indicadores de costo y valor los clientes y desarrolladores utilizan el método de Proceso de Análisis Jerárquico (AHP). El AHP fue desarrollado por Thomas L. Saaty en el año 1980 con el objetivo de resolver problemas de criterios múltiples. Como dijera su creador el AHP “se trata de desmenuzar un problema y luego unir todas las soluciones de los subproblemas en una solución.” (Hurtado & Bruno) Este método utilizado en la toma de decisiones se fundamenta en:

- ✓ “La estructuración del modelo jerárquico (representación del problema mediante identificación de metas, criterios, subcriterios y alternativas).
- ✓ Priorización de los elementos del modelo jerárquico.
- ✓ Comparaciones binarias entre los elementos.
- ✓ Evaluación de los elementos mediante asignación de “pesos”.
- ✓ *Ranking* de las alternativas de acuerdo con los pesos dados.
- ✓ Síntesis.
- ✓ Análisis de Sensibilidad.” (Hurtado & Bruno)

Con el método de Decisión basada en Costo-Valor se obtiene a partir de la aplicación del AHP y mediante un mapa conceptual, un análisis que permite decidir qué requisito será implementado en el sistema y realizar estrategias para planificar las diferentes entregas del producto. A pesar de que este método tiene un sustento matemático para determinar la medida cuantitativa y el análisis cualitativo de los indicadores propuestos, no cuenta con un sistema amplio de indicadores que midan otros aspectos involucrados en la priorización de requisitos como puede ser la dependencia, frecuencia, complejidad, entre otros.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4.2. El despliegue de la función de la calidad del software (*Software Quality Function Deployment, SQFD*)

El despliegue de la función de la calidad (QFD) nacido en el mundo de la gestión, es un método utilizado por múltiples empresas que generan diferentes productos, entre ellos se encuentran los productos de software, de ahí que se aplique a la ingeniería de requisitos y se le conozca como SQFD. El método es también llamado como “la voz de los clientes” o “la casa de la calidad” por la forma de su diagrama resultante, semejante a una casa. Utiliza representaciones gráficas que permiten analizar las prioridades de los requisitos, asegurando que los deseos y las necesidades de los clientes sean traducidos en características técnicas.

Se realiza en cinco pasos, los cuales tienen como objetivo fundamental colocar las necesidades de los clientes y las especificaciones técnicas del producto dentro de una matriz. Luego se determina a través de sondeos o incluso aplicando el método AHP la prioridad de cada uno de sus componentes de la matriz y las interrelaciones de los mismos. Los indicadores que establece el método están a consideración de clientes y desarrolladores, en dependencia de las necesidades de los mismos, ejemplo de ellos son: para el caso del equipo de desarrollo el costo y la dificultad de implementación y para el caso de los clientes o usuarios las especificaciones técnicas y medibles del producto de software.

Las garantías que proporciona el método están relacionadas con la definición cuantificable de los requisitos críticos del cliente, la obtención de análisis que justifiquen las decisiones a tomar y la elaboración de productos con calidad que satisfagan las necesidades de los usuarios. A pesar de esto se muestran como desventajas la engorrosa ejecución del método, la necesidad de tiempo para su ejecución, el esfuerzo y la paciencia que deben dedicar los clientes y desarrolladores.

1.4.3. Wiegiers

Wiegiers es un método que selecciona los requisitos más relevantes sin tener en cuenta los requisitos que se derivan de estos. Para realizar la priorización de los requisitos relevantes tiene en cuenta los valores de importancia que le asignan los clientes a los requisitos, la penalidad si un requisito no se incluye y los riesgos técnicos asociados a la implementación. A cada atributo se le puede dar un valor cuantitativo del 1 al 9 por los involucrados en el proceso. Uno de los inconvenientes de este método es

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

que solo tiene en cuenta los requisitos principales, descartando la posibilidad de análisis del indicador dependencia. Además, no contempla la frecuencia como indicador.

1.4.4. Teoría-W

La Teoría-W fue desarrollada en la universidad del Sur de California en el año 1989, su objetivo es establecer una situación satisfactoria entre los *stakeholders* y los desarrolladores. Para la priorización de los requisitos se basa en las condiciones de ganancia, clasificándolas en dos perspectivas: “importancia de negocio y facilidad de realización. La primera es la relevancia de la condición para cliente y usuarios. La segunda corresponde a las dificultades percibidas y esperadas de implementar una condición de ganancia. Los *stakeholders* asignan un indicador de importancia creciente de 1 a 10. En este proceso los clientes y usuarios asignan el indicador a la perspectiva de importancia de negocio y los desarrolladores se la asignan a la facilidad de realización.” (Poblete) Este método emplea solo los indicadores beneficio y complejidad sin tener en cuenta otros indicadores importantes como la frecuencia y la dependencia.

1.5. Metodologías en la priorización de requisitos

Con el transcurso del tiempo el proceso de desarrollo de software ha ido evolucionando, trayendo consigo el surgimiento y perfeccionamiento de diferentes metodologías con el propósito de lograr la eficacia y la eficiencia en el proceso de generación de software. Las metodologías desarrollan el proceso de ingeniería de requisitos y dentro de este de una forma u otra llevan a cabo la priorización de los mismos, para ello establecen indicadores. En la gran mayoría de los casos esta prioridad está determinada por los clientes y usuarios. A continuación se analizan algunas metodologías, profundizando en los diferentes indicadores que utilizan para determinar la prioridad de los requisitos.

1.5.1. Proceso Unificado de Desarrollo (*Rational Unified Process, RUP*)

La metodología Proceso Unificado de Desarrollo fue creada por Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998, con el fin de realizar un conjunto de actividades para transformar las

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

necesidades del cliente en un sistema de software. Cuenta con nueve flujos de trabajo, seis ingenieriles y tres de apoyo, los cuales se van desarrollando por iteraciones en las diferentes fases que posee, como refleja la Figura 2.

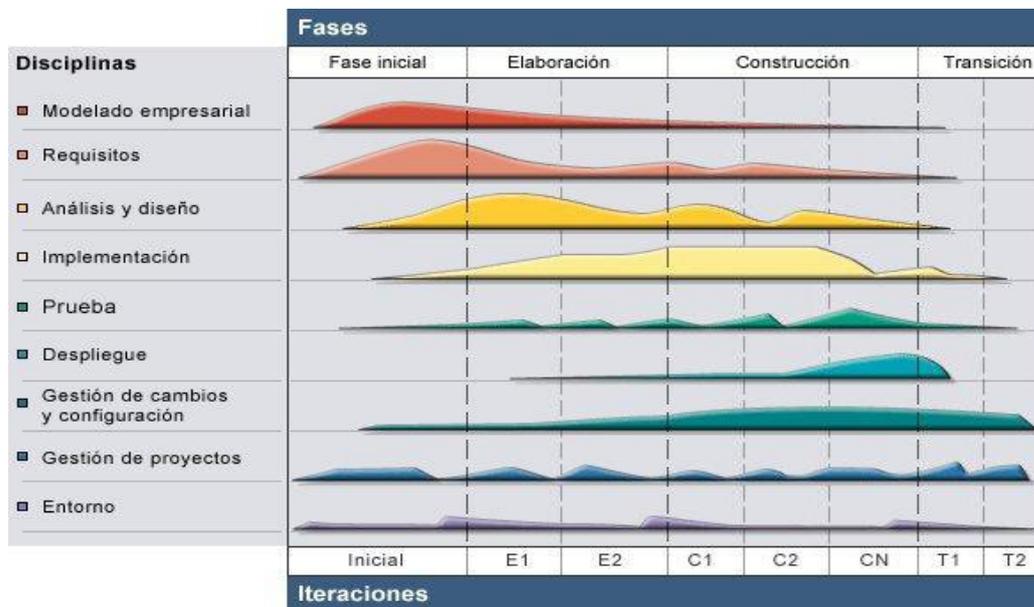


Figura 2: RUP en dos dimensiones. Fuente: (Díaz, y otros, 2009).

Emplea la priorización con el fin de mitigar riesgos, esta es efectuada por el arquitecto. El cual clasifica los casos de uso en función del impacto que tendrán en la arquitectura, en críticos, secundarios, auxiliares y opcionales. Los casos de uso señalados como críticos son los más importantes para el arquitecto porque cubren las principales tareas o funciones que el sistema debe realizar. Los secundarios son aquellos que sirven de apoyo a los críticos pero con un impacto menor, aunque deben implementarse pronto porque responden a casos de uso importantes. Los auxiliares no son claves pero completan a los críticos y secundarios y los opcionales son aquellos que pueden estar o no en la aplicación pero en caso de estar no son necesarios realizarlos en las primeras iteraciones. Se debe tener en cuenta que estas clasificaciones son propias del sistema, es decir, que en dependencia de las características del software a implementar será la clasificación dada.

La priorización de los requisitos según esta metodología está basada en indicadores seleccionados por el arquitecto del proyecto, dejando toda la responsabilidad del proceso al conocimiento y la experiencia personal.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.5.2. Scrum

Scrum es una metodología que más que tener un carácter adaptable tiene un carácter predictivo, debido a que está más orientada a las personas que a los procesos. El trabajo con ella es un poco difícil porque no sigue un plan, ni las prácticas de la Ingeniería de Software sino que se va adaptando a las circunstancias en las que se vaya realizando el proyecto. Cuenta con tres fases y cuatro flujos de trabajo.

Trabaja con una lista de requisitos priorizadas que es elaborada por el cliente con la ayuda del facilitador y del equipo de desarrollo y actúa como plan del proyecto. Esta lista es priorizada balanceando el valor que cada requisito aporta al negocio frente al costo estimado para su desarrollo, es decir, basándose en el método de Retorno de Inversión (ROI). Este método matemático que es calculado como un porcentaje teniendo en cuenta el beneficio obtenido de una inversión en relación con el costo que esta representa, no implica solamente dinero sino que también se puede medir en términos de tiempo.

A pesar de ser una metodología que emplea un método matemático pone en manos del cliente la realización de la priorización. Esta no es apta para todos los proyectos de software y requiere que todos los procesos que se desarrollen se realicen de una manera rápida, por ello no es la más factible. La principal desventaja de esta metodología radica en el proceso de replanificación de las actividades después de cada iteración, debido a que una vez concluida cada una de ellas el propietario de planificación del proyecto tiene que volver a estimar los costos de los requisitos que hayan sido modificados por el cliente en la lista de priorización.

1.5.3. Programación Extrema (*Extreme Programming, XP*)

Esta metodología es utilizada generalmente por equipos pequeños o medianos, en los que se quiera obtener una elevada calidad y que el gasto fijo sea mínimo. Su ciclo de vida se basa en el desarrollo de seis fases: exploración, planeación, iteración, producción, mantenimiento y muerte. Es muy adecuada para proyectos en los cuales los requisitos sean imprecisos y muy cambiantes y donde el riesgo técnico suele ser elevado. Utiliza la técnica de Historias de Usuario para especificar los requisitos de software, estas historias de usuario describen las características y las funcionalidades requeridas para el software que se construirá. Cada historia es escrita por el cliente, el cual le asigna la prioridad basándose en los valores del negocio que a él le atañen, o sea, la prioridad es determinada por el cliente teniendo en cuenta el grado de importancia que representa para el negocio el desarrollo

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de la funcionalidad. Después de esto el equipo de desarrollo evalúa cada historia y le asigna un costo, el cual es medido según las semanas de desarrollo. En caso de que la historia se pase de tres semanas de desarrollo, se le pide al cliente que la divida en historias más pequeñas y entonces se le hace de nuevo la asignación de costo y valor. La principal problemática en cuanto al tema de asignar prioridades es que esta metodología pone este proceso tan importante en manos de los clientes quienes saben lo que quieren pero no tienen mucha idea de cómo es el trabajo del equipo de desarrollo.

1.5.4. Desarrollo Basado en Funcionalidades (*Feature Driven Development, FDD*)

FDD fue diseñado por Peter Coad, Erich Lefebvre y Jeff De Luca. Se puede considerar como una metodología que está entre RUP y XP. Consta de cinco fases las cuales son: desarrollo de un modelo general, construcción de la lista de funcionalidades, plan de releases⁹ en base a las funcionalidades a implementar, diseñar e implementar en base a las funcionalidades. Se emplea en proyectos de corto tiempo de duración que pueden ser monitoreados y visualizados por el cliente y la dirección de la empresa. Es un proceso iterativo que posee iteraciones de corta duración. Las funcionalidades son definidas por los integrantes del equipo de desarrollo, estos presentan al cliente las posibles funcionalidades con un orden de prioridad, pero para que sea implementada tiene que ser aprobada por el mismo.

Este método no se puede aplicar a grandes proyectos de software y de larga duración. No posee indicadores definidos para determinar la prioridad de los requisitos, esta es determinada por apreciación de los desarrolladores y presentada al cliente. El equipo de trabajo no posee la libertad necesaria para determinar las funcionalidades que serán implementadas dentro del sistema debido a que tiene que estar sujeto a la opinión del cliente.

⁹ Releases: Versiones

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.5.5. Elicitación de requisitos- Desarrollo Global de Software (RE-GSD)

La metodología RE-GSD cuenta con seis etapas, donde una de ellas es la de priorizar requisitos, como se muestra en la Figura 3. En la etapa de priorización de requisitos se le asigna una importancia relativa a cada requisito en relación con los demás. Para calcular esta importancia se tiene en cuenta a los clientes y usuarios, quienes brindarán el grado de satisfacción con respecto a la implementación o no de los requisitos. La principal problemática dentro de este proceso es el uso de un único indicador para determinar la prioridad de los requisitos de software, basándose sólo en la opinión del cliente.

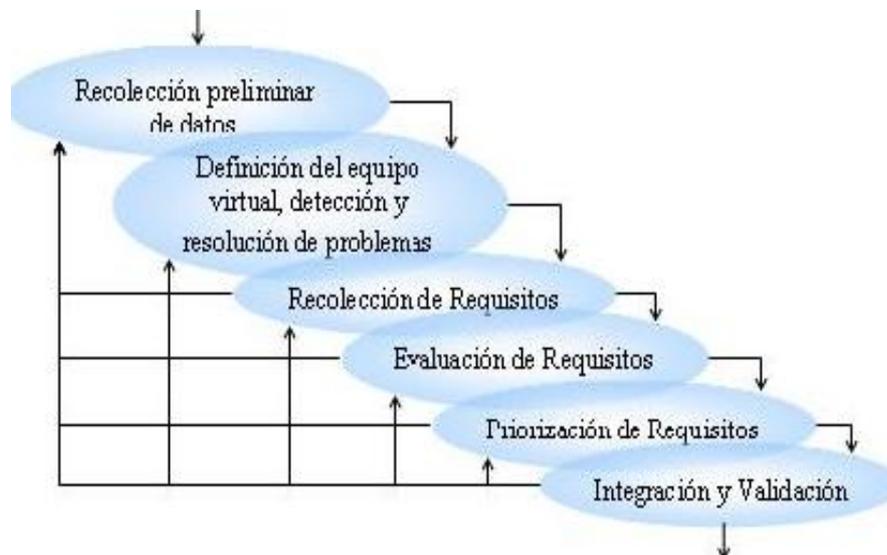


Figura 3: Etapas de RE-GSD. Fuente: (Aranda, Vizcaíno, Cechich, Piattini, & Soto)

1.5.6. MSF for CMMI

La metodología de Solución de *Microsoft* del Marco de trabajo para el Proceso de mejora CMMI es la guía de procesos para el equipo actual del proyecto de software con el propósito fundamental de que el trabajo esté organizado, haciendo que el mismo posea la calidad requerida y que los clientes queden satisfechos con el producto final. Consta de cinco fases las cuales son: previsión, planificación, desarrollo, estabilización e implementación.

La priorización es realizada por el analista del negocio, el cual se basa en la importancia que poseen los escenarios para los usuarios. Cuando un escenario se excede del costo que se le planificó para una determinada iteración, el analista determina si este escenario puede ser dividido en dos más

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

pequeños. Dicho análisis llevado a cabo teniendo en cuenta el costo proporciona grandes desventajas debido a que requisitos importantes pueden ser postergados.

La prioridad está basada en diferentes categorías según la importancia que tengan los escenarios para los flujos de trabajo en los que se desarrolla la metodología a los cuales se les da un *ranking* determinado. Los escenarios que se clasifican como muy altos son los más convenientes para el desarrollo del producto de software. Los altos son los deseados y clasificados por el cliente en importantes. Los medios son los evidentes pero no mejoran o dificultan la necesidad del cliente, es decir, pueden o no ser implementados. Los bajos son aquellos que de manera automática son eliminados y abandonados, por lo que no se realiza su priorización. En fin, todas estas categorías expresan que prioridad tiene cada escenario para el cliente para poder determinar la implementación de dicho escenario.

1.5.7. Método de Desarrollo de Sistemas Dinámicos (*Dynamic Systems Development Method, DSDM*)

Esta metodología fue creada en 1994 en el Reino Unido por el consorcio de DSDM con el objetivo de crear una metodología de Desarrollo Rápido de Aplicaciones (RAD) unificada, que se esforzara por prestar un servicio más eficiente en el proceso de desarrollo de y que respondiera a plazos de entrega cortos con recursos limitados. Se centra fundamentalmente en los requisitos del negocio. Al igual que RUP es una metodología iterativa e incremental, donde el equipo de desarrollo y los usuarios trabajan unidos.

Su ciclo de desarrollo está compuesto por las siguientes fases:

- ✓ “Estudio de viabilidad
- ✓ Estudio de negocio
- ✓ Iteración de modelado funcional
- ✓ Iteración de diseño y desarrollo
- ✓ Implementación” (Metodologías)

Siendo las tres últimas fases iterativas, a pesar de que todas se retroalimentan. Dicha metodología para la priorización de requisitos emplea la técnica de MoSCoW, la cual tiene en cuenta el costo de los requisitos y hace posible que se descubra con tiempo las limitaciones que puedan existir en el sistema.

A continuación se representa el significado de sus siglas M, S, C, W sucesivamente:

- ✓ “Must, debe de tener esta característica

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Should, debería de tener esta característica
- ✓ Could, podría tener esta característica
- ✓ Won't, no por lo pronto, pero si en el futuro" (Rascón, 2010)

El principal problema dentro de este proceso que es realizado por esta metodología es el uso de un único indicador para determinar la prioridad de los requisitos de software, basándose solamente en el costo que puede tener la implementación de cada requisito.

1.6. Herramientas para priorizar requisitos

En la actualidad existen herramientas que soportan todas las etapas de la Ingeniería de Software incluyendo la priorización de requisitos con el propósito de lograr efectividad y consistencia. Entre ellas se pueden mencionar:

- ✓ **SREPTOOL:** Esta herramienta va guiando el trabajo de los diferentes roles y responsables, así como ayuda en la generación de la documentación necesaria en el proceso de ingeniería. Cuenta con nueve actividades y cada una tiene una pestaña diferente. La actividad número siete es la que corresponde a la priorización de requisitos de seguridad del software. Utiliza un único indicador para determinar el nivel de seguridad del requisito, este toma valores en función del riesgo de las amenazas que mitigan, los cuales pueden ser crítico, estándar u óptimo. Esta prioridad es determinada por el cliente y no se tiene en cuenta la experiencia de los trabajadores o desarrolladores del proyecto en temas relacionados con la seguridad del software incurriendo en fallas o errores dentro de este proceso. La seguridad del software es un aspecto sumamente importante en el desarrollo del mismo, es por ello que a la hora de determinar la prioridad de los requisitos de seguridad es necesario utilizar un mayor grupo de indicadores con el fin de obtener los resultados esperados.
- ✓ **SREPPLineTool:** Esta herramienta guía intuitivamente a la persona que la utilice, llevando una secuencia de ejecución con una representación visual de los artefactos que se manejan en cada etapa. Consta con nueve actividades, definiendo entre ellas una actividad para priorizar los requisitos de seguridad, los cuales se ordenan a partir de valores numéricos de 0 a 10. Es necesario aclarar que esta herramienta se enfoca solamente en la priorización de los requisitos de seguridad mediante indicadores cuantitativos obviando el resto de los indicadores que existen.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.7. Limitantes de los enfoques estudiados

A partir del análisis de algunos métodos y metodologías existentes en el marco de la ingeniería de requisitos a continuación se presenta un estudio a través de tablas que muestran los principales indicadores que se ponen de manifiesto para la obtención de la prioridad de los requisitos en dichos métodos y metodologías:

INDICADOR/MÉTODOS	Decisión basada en Costo-Valor usando AHP	SQFD	Wieggers	Teoría-W
Complejidad		X		X
Beneficios	X	X	X	X
Dependencia				
Estabilidad				
Frecuencia			X	
Costo	X	X	X	

Tabla 1.2: Indicadores reflejados en los métodos.

Según los valores obtenidos en la Tabla 1.2 el indicador más utilizado en los métodos analizados es el beneficio. Este indicador muestra el nivel de importancia que representan los requisitos para los clientes y cuales son necesarios e imprescindibles en el desarrollo de software. El otro más importante es el costo, el cual representa el valor monetario, el tiempo y los recursos de implementación de un requisito para los clientes y desarrolladores. La complejidad es otro indicador frecuentemente utilizado, el cual permite estimar los esfuerzos necesarios para desarrollar los requisitos.

INDICADOR	RUP	Scrum	XP	FDD	RE-GSD	MSF for CMMI	DSDM
Complejidad							
Beneficios	X	X	X	X	X	X	
Dependencia	X						
Estabilidad							
Frecuencia							
Costo		X				X	X

Tabla 1.3: Indicadores reflejados en las metodologías.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Se plantea que la gran mayoría de las metodologías de desarrollo utilizan el indicador beneficio para determinar la prioridad de los requisitos de software, según se muestra en la Tabla 1.3 es evidente que muchas metodologías dejan la priorización de requisitos en manos de los clientes, basándose únicamente en el valor que tiene una determinada funcionalidad para resolver las necesidades de estos. Al igual que en los métodos, el costo es otro indicador muy usado por las metodologías.

INDICADOR	TIPO DE INDICADOR	
	Cualitativo	Cuantitativo
Complejidad	X	X
Beneficios	X	
Dependencia	X	X
Estabilidad	X	X
Frecuencia	X	
Costo		X

Tabla 1.4: Clasificación de indicadores.

La mayoría de los indicadores utilizados en la priorización de requisitos se miden de forma cualitativa y/o cuantitativa como se muestra en la Tabla 1.4. Generalmente todos estos indicadores cualitativos son evaluados de forma cuantitativa para poder obtener un mejor resultado.

Métodos y Metodologías/ Indicadores	Complejidad	Beneficio	Dependencia	Frecuencia	Costo
Costo-Valor (AHP)		C			D
SQFD	D	C			D
Wiegiers		C		D	D
Teoría-W	D	C			
RUP		C	D		
Scrum		C			D
XP		C			
FDD		C			
RE-GSD		C			
MSF for CMMI		C			D
DSDM					D

Tabla 1.5: Responsables por métodos y metodologías de indicadores.

Legenda C: cliente, D: desarrollador

Como se muestra en la Tabla 1.5 todas las metodologías y métodos estudiados usan el indicador beneficio para determinar la prioridad de los requisitos siendo evaluados por el cliente. Sin embargo, existen metodologías como RUP que aunque tienen en cuenta la opinión del cliente para medir el

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

indicador beneficio quien realmente asigna la prioridad es el desarrollador. Los desarrolladores al igual que los clientes se apoyan en métodos cualitativos y cuantitativos para determinar el valor de los indicadores. Los indicadores más evaluados por los desarrolladores son: complejidad, costo, frecuencia y dependencia.

Mediante el estudio del uso de los indicadores para priorizar los requisitos y su aparición en métodos y metodologías se puede concluir que no existe un conjunto de indicadores lo suficientemente robusto y completo para aplicar en los procesos de desarrollo de software. Además, los indicadores reflejados abarcan amplios conceptos relacionados con la elaboración de software que pudieran ser desglosados de forma minuciosa en nuevos indicadores que ayuden a establecer con más exactitud la prioridad de un requisito.

1.8. Conclusiones parciales

A través del estudio de la ingeniería de requisitos se demuestra la existencia de la fase de obtención y análisis de requisitos, en la cual se pone de manifiesto la priorización de los requisitos como parte de la etapa de evaluación. Uno de los factores de éxito para priorizar las funcionalidades del sistema a desarrollar son los indicadores, establecidos por los clientes y desarrolladores, los cuales se evalúan de forma cualitativa y/o cuantitativa. A pesar de la cultura alcanzada en el uso de indicadores para priorizar requisitos, el análisis desarrollado durante el capítulo demuestra la insuficiente variedad de los mismos y la necesidad de establecer un conjunto de estos que ayuden a los equipos de desarrollo a determinar posteriormente de organizar los requisitos cuales serán implementados en las primeras iteraciones teniendo en cuenta la prioridad de los mismos.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2.1. Introducción

Con el fin de lograr los objetivos trazados en la investigación científica se hace necesario el estudio del estado actual de la priorización de requisitos funcionales en los proyectos productivos que se desarrollan en la Universidad de las Ciencias Informáticas. En el presente capítulo se analiza una muestra que representa el 10,46% de la población a través de las técnicas: encuesta y entrevista, se arriba a conclusiones sobre el estado actual referente al uso de indicadores en la priorización de requisitos y se establecen un conjunto de indicadores para darle solución al problema planteado en este trabajo científico.

2.2. Características de la organización

La Universidad de las Ciencias Informáticas surge al calor de la Batalla de Ideas, comenzando su curso escolar el 23 de septiembre del 2002. Esta Universidad tiene como misión fundamental “formar profesionales, comprometidos con su Patria, calificados en la rama de la Informática, a partir de un modelo pedagógico flexible, que vincula dinámicamente y coherentemente el estudio con la producción y la investigación, acorde con las necesidades sociales del país y de otros pueblos hermanos.” (Hernández & otros)

Desde que se decide crear la Universidad se toma como perspectiva que esta llegue a ser una empresa productora de software, cuya fuerza de trabajo fundamental está compuesta por estudiantes y profesores, al igual que recién graduados provenientes de las diferentes universidades del país. La UCI es un modelo de universidad docente-productiva que cuenta con más de doce mil estudiantes, profesores y técnicos.

2.2.1. Producción en la UCI

La Universidad cuenta hoy con “más del 85% de los estudiantes de 3ro a 5to años, vinculados a proyectos productivos. Además, existen más de 11 000 estudiantes de pregrado y alrededor de 2 000 profesores que se integran en las 10 Facultades ya existentes, con más de 1000 estudiantes cada

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

una.” (albet) En estos momentos, se han realizado tres graduaciones donde se han formado Ingenieros en Ciencias Informáticas.

A partir del año 2003 la UCI se abre camino como una empresa de software proveedora de Soluciones Tecnológicas Integrales. En el transcurso del curso 2006-2007 más de 150 proyectos se encontraban elaborando software y es aquí cuando se propone el modelo de Polo Productivo que no es más que el “espacio natural para ejecutar proyectos temáticos a partir de la integración de los procesos fundamentales de formación, investigación, producción y comercialización.” (La UCI Portal UCI - Universidad de las Ciencias Informáticas) También surge una nueva idea en el proceso de producción que es la creación de los Centros de Desarrollo, con el objetivo fundamental de fortalecer el proceso de software en la UCI en temas tales como la seguridad, la calidad, el soporte, la arquitectura, entre otros. Los Centros están vinculados a cada una de las facultades de la Universidad y se encargan del desarrollo productivo de los estudiantes y profesores que se vinculan a los mismos.

En la Universidad se construyen y desarrollan productos y servicios de software en dependencia de las necesidades. “La fabricación de productos tiene ciertas características en común: se requiere contar con un capital previo, primero se diseña el producto, luego se replica ininidad de veces, mas tarde se reparte a través de una cadena de distribución y finalmente el cliente paga por obtener una copia del producto.” (Stankevicius, 2006) Estos proyectos que fabrican productos no tienen un cliente definido, desarrollan el producto y luego es vendido a diferentes clientes y personalizado por los mismos. También se proveen servicios y productos de software. “El proveedor de servicio también presenta ciertas características en común: no se requiere contar con un capital previo, se debe encontrar una necesidad insatisfecha que tengan los clientes, no se requiere una cadena de distribución y el cliente paga por tener su necesidad debidamente atendida. El software atraviesa las etapas propias de la fabricación de productos: es diseñado aplicando técnicas de ingeniería de software, luego es replicado, es eventualmente distribuido y finalmente es vendido.” (Stankevicius, 2006) En este proceso se desarrolla la solución pactada con el cliente, para ello se realiza un levantamiento de requisitos el cual contempla desde la comprensión de los procesos a informatizar hasta la obtención de las funcionalidades del futuro sistema.

Actualmente la Universidad cuenta con varios proyectos productivos que abarcan diferentes ramas dentro del campo de la informática y tienen como tarea no solo la informatización de la Universidad y el país, sino también el deber de ingresar al país importantes sumas de dinero que colaboren en el desarrollo de la sociedad.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2.2.2. Programa de mejora

Actualmente la UCI se considera una de las mayores organizaciones que llevan a cabo la producción de software. En estos momentos “el centro está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI (*Capability Maturity Model Integration*) y con la contratación de los servicios de consultoría del SIE Center (*Software Industry Excellence Center*) del Tecnológico de Monterrey.” (Calisoft Centro de Calidad para Soluciones Tecnológicas, 2009) Dicho proceso de mejora está encaminado a que la Universidad en el año 2010 alcance “una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo.” (Calisoft Centro de Calidad para Soluciones Tecnológicas, 2009)

En la actualidad hay definidos y en piloto cinco procesos: “Aseguramiento de la Calidad a Procesos y Productos (PPQA), Administración de Requisitos (REQM), Planeación del Proyecto (PP), Monitoreo y Control del Proyecto (PMC) y Medición y Análisis (MA). En definición se encuentran los procesos de Administración de Acuerdos con Proveedores (SAM) y Administración de la Configuración (CM).” (Calisoft Centro de Calidad para Soluciones Tecnológicas, 2009)

El proyecto de mejora posee una guía conocida mayormente con el nombre de: Guía del Proceso de Mejora IPP3510:2009, la cual pertenece al proceso de Administración de Requisitos (REQM). Establece un grupo de criterios con el fin de obtener la complejidad de los casos de uso, estos se analizan individualmente para poder clasificarlos en Alta, Media o Baja con el fin de estimar el esfuerzo de implementación del caso de uso y contribuir a la decisión sobre la inclusión o no de cada caso de uso en cada una de las iteraciones de desarrollo de software.

Esta guía también establece un grupo de indicadores para determinar la prioridad de los casos de uso a implementarse y así poder elaborar un cronograma que responda a la planificación de cada una de las iteraciones priorizando los casos de uso arquitectónicamente significativos. Estos casos de uso son evaluados de acuerdo a los indicadores: beneficio, dependencia, estabilidad y frecuencia, evaluados de forma cualitativa a través de resultados cuantitativos.

2.3. Población, muestra y técnicas utilizadas en la investigación

A partir del libro de Diagnóstico emitido por el Centro para la excelencia en el desarrollo de proyectos tecnológicos (*Calisoft*) en el año 2008, se pudo conocer el número de proyectos que se desarrollan en la Universidad de las Ciencias Informáticas. En este libro se diagnosticaron un total de 153 proyectos

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

los cuales conforman la población de esta investigación. De dicha población se tomó una muestra de 16 proyectos productivos la cual representa un 10,45% del total.

Para la recogida de la información se realizó una encuesta (ver Anexo 2) que fue aplicada a diferentes integrantes que tenían una participación activa dentro de la priorización y una entrevista (ver Anexo 1) al principal representante dentro de dicha priorización en cada uno de los proyectos seleccionados.

La encuesta elaborada tenía como propósito diagnosticar los siguientes aspectos:

- ✓ La priorización de requisitos teniendo en cuenta el uso de indicadores.
- ✓ La experiencia en la priorización de requisitos.
- ✓ La frecuencia con que se realiza la priorización de requisitos.
- ✓ Quiénes son las personas responsables de realizar la priorización de requisitos.
- ✓ Si se tiene en cuenta el criterio del cliente en el momento de priorizar requisitos.
- ✓ Cuán importante suele ser la priorización de requisitos.
- ✓ La metodología empleada y conocer si usaban los indicadores que propone esta u otros.
- ✓ Que indicadores se utilizan en cada proyecto.
- ✓ Cuán importante es el uso de indicadores.
- ✓ De qué forma se le asignan los valores a los indicadores.
- ✓ Ventajas del uso de indicadores.

La entrevista llevada a cabo tenía los siguientes objetivos:

- ✓ Conocer la importancia de la priorización de requisitos.
- ✓ Identificar las dificultades dentro de la priorización de requisitos.
- ✓ Conocer los involucrados dentro de la priorización de requisitos.
- ✓ Conocer cómo es realizada la priorización de los requisitos.
- ✓ Conocer el uso de herramientas, métodos o metodologías para priorizar y como se emplean.
- ✓ Conocer si se usan indicadores para priorizar y cuáles.
- ✓ Saber cómo le son asignados los valores a los indicadores y que aspectos tienen en cuenta para asignar dichos valores.
- ✓ Conocer de quién es la decisión más importante para priorizar.

2.4. Análisis e interpretación de los resultados de la entrevista y la encuesta.

Para realizar el diagnóstico interno de la priorización y el uso de indicadores en dicha priorización en la Universidad de las Ciencias Informáticas se aplicaron diferentes técnicas como: encuesta y entrevista.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

A continuación se describen los resultados obtenidos y analizados a través de los puntos de vista de las personas vinculadas a la priorización de requisitos.

Entrevista:

Se realizó una entrevista a diferentes personas que están vinculadas al proceso de producción, específicamente en la priorización de requisitos. La mayoría de estas fueron realizadas a los líderes de proyectos, máximos responsables del trabajo del proyecto y analistas quienes son los principales responsables de la priorización de requisitos. De estas se obtuvo los indicadores y las personas que participan en la priorización. Como resultados de las entrevistas se identificaron además los problemas siguientes:

- ✓ No se utilizan herramientas y métodos matemáticos para la realización de la priorización de los requisitos funcionales.
- ✓ No existe un conjunto de indicadores que se pueda emplear en los proyectos sin tener en cuenta la metodología que emplean. Es decir, que generalmente emplean sólo los indicadores que brinda la metodología que utilicen.

A raíz de la pregunta 1 de la entrevista se realizó el Gráfico 1 en el cual se puede observar que estas entrevistas estuvieron generalmente enfocadas a líderes de proyectos los cuales representan un 63% de un total de 16 entrevistados y el 13% de analistas principales. También fueron entrevistados arquitectos, jefes de módulos, otros analistas y sublíderes de módulos, los mismos tenían conocimiento y participación en la priorización con una representación de un 6% del total.

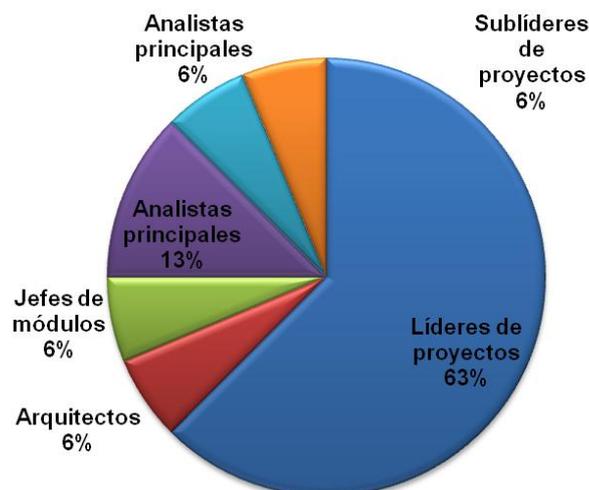


Gráfico 1 : Personal entrevistado.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

De la pregunta 2 de la entrevista se obtuvo que el 100% de los entrevistados garantizaron que realizaban la priorización de requisitos expresando que era de vital importancia para el desarrollo de software debido a que contribuye a facilitar el trabajo y la calidad del mismo.

Con la realización de la pregunta 3 de la entrevista que se puso en práctica en una muestra de 16 personas como se observa en el Gráfico 2, 15 de ellas coincidieron en que el personal mayormente involucrado en la asignación de prioridades son los arquitectos, los clientes y los líderes de proyectos, seguido de estos y con un valor de 14 se encuentran los analistas principales. Otros analistas que estén vinculados al proyecto también pueden participar en la priorización y como últimas opciones están los sublíderes de proyecto y especialistas de base de datos.

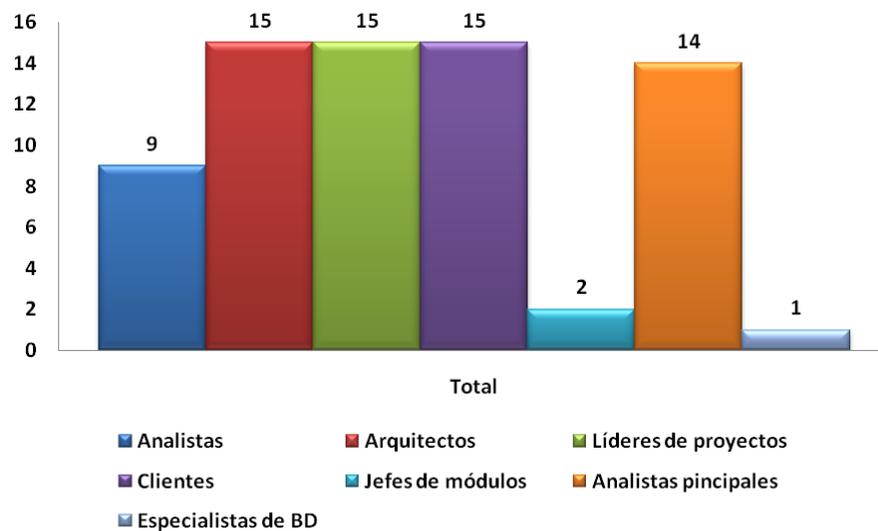


Gráfico 2: Personal involucrado en la priorización.

A través de las respuestas dadas a la pregunta 4 se obtuvo que el 93,75% de los proyectos tienen en cuenta el criterio del cliente en la priorización de requisitos. En un gran número de casos la priorización es un trabajo en conjunto, el equipo debe seguir las necesidades expresadas por el cliente y definir las funcionalidades que serán implementadas, asumiendo que el criterio del cliente es sumamente importante para determinar la prioridad. En aquellos proyectos que no tienen clientes definidos simplemente el grupo de desarrollo toma sus propias decisiones para priorizar.

De un total de 16 proyectos entrevistados el 87,5% de estos hacen uso de metodologías para priorizar requisitos funcionales y solo el 18,8% hacen uso además de la Guía del Proceso de Mejora IPP3510:2009 Administración de Requisitos. A pesar de que existen herramientas y métodos para priorizar estos no son muy utilizados, pudiendo hacer más fácil el trabajo a la hora de priorizar los requisitos.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Teniendo en cuenta las respuestas dadas a la pregunta 5 se realizó el Gráfico 3 donde se muestra que el 69% de los proyectos entrevistados utilizan la metodología RUP, el 19% utilizan la metodología FDD, el 6% hacen uso de la metodología Scrum-XP y de la metodología enfocada a procesos de negocio alineados al proceso de mejora de la Universidad, demostrando que la gran mayoría de los proyectos que se desarrollan en la Universidad hacen uso de metodologías pesadas ya que son proyectos muy grandes y de largo tiempo de duración.

En el caso de RUP, esta metodología no posee indicadores definidos para llevar a cabo la priorización sino que establece 4 criterios para clasificar los casos de uso en críticos, secundarios, auxiliares y opcionales. El 19% está representado por aquellos proyectos que utilizan la metodología FDD, la misma no define indicadores para determinar la prioridad de los requisitos, esta es determinada por apreciación de los desarrolladores y presentada al cliente. Otros sin embargo utilizan la metodología enfocada a procesos de negocio alineados al proceso de mejora de la Universidad utilizando una guía para la realización de la priorización, conocida como Guía del Proceso de Mejora IPP3510:2009 Administración de Requisitos la cual establece un grupo de indicadores asignándole valores cualitativos a través de resultados cuantitativos.

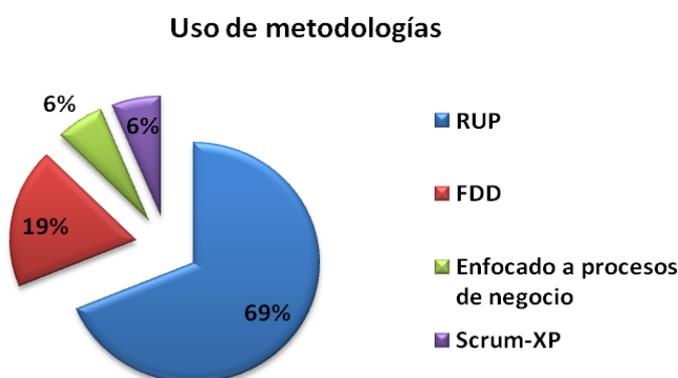


Gráfico 3: Uso de metodologías.

A pesar de que el 93,75% de los proyectos entrevistados utilizan indicadores se puede reafirmar que no hacen uso de un conjunto de ellos para llevar a cabo la priorización. A través de los resultados de la pregunta 6 se puede decir que de una muestra de 15 proyectos que hacen uso de indicadores para priorizar requisitos 13 de ellos emplean el beneficio y la dependencia y solo 2 de ellos utilizan los indicadores estabilidad y frecuencia. A pesar de tener en cuenta otros indicadores el 62,5% de los proyectos se enfocan nada más en los que proponen la metodología que emplean y el 37,5% utilizan otros que no son propuestos por la metodología que desarrollan estos valores se pudieron determinar con la realización de la pregunta 7 de la entrevista.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

En el Gráfico 4 que fue elaborado según las respuestas dadas a la pregunta 8 se puede observar que 15 de los proyectos entrevistados miden los indicadores de forma cualitativa y solo 4 de ellos se apoyan en la experiencia de las personas que están al frente de la priorización. Solo 2 de ellos miden de forma cuantitativa lo que demuestra que la priorización no es realizada con el rigor y la exactitud que merece.

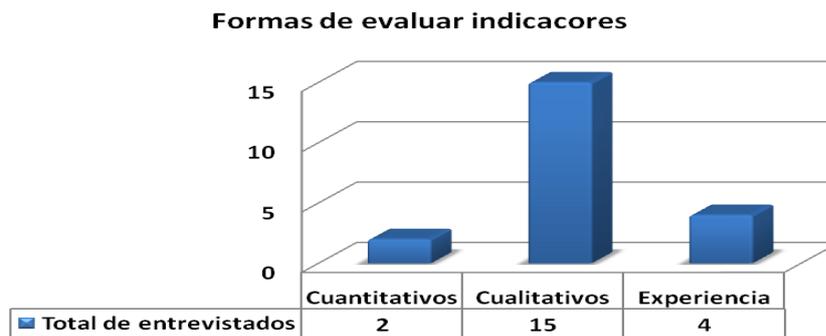


Gráfico 4: Formas de evaluar indicadores.

En las respuestas dadas a la pregunta 9 se demuestra que el criterio del analista principal es el que más predomina por parte del equipo de desarrollo a la hora de darle valores a los indicadores y por tanto en la mayoría de los casos es la persona encargada de asignar este valor, a partir de las entrevistas realizadas se obtuvo que 10 de las personas coincidieron en que es el analista principal la persona encargada de realizar esta tarea, 3 por el arquitecto, 2 por el líder de proyecto y solo 1 porque sea realizada por algún analista vinculado a la priorización. Estas personas para darle valor a dichos indicadores tienen en cuenta el criterio del cliente y de los demás involucrados en la priorización, así como la experiencia, el estudio del arte en relación a software similar y los criterios planteados en la metodología que utilizan.

El trabajo dentro de la priorización de requisitos es entre el equipo de desarrollo y clientes. Estas personas tienen que estar mutuamente de acuerdo en cuanto a la decisión final que se tomará para ver qué requisitos serán priorizados en cada iteración del trabajo en el proyecto. Sin embargo, son los desarrolladores los que tomarán la decisión final, porque ellos son los que saben cómo se debe hacer el trabajo pero teniendo siempre en cuenta que es lo que el cliente desea y necesita en cada fase o iteración del proyecto. Con la realización de la pregunta 10 en la muestra escogida de los proyectos de la Universidad el 62,75% de los proyectos entrevistados plantea que la prioridad está determinada por desarrolladores teniendo en cuenta el criterio del cliente y solo el 18,75% es determinado por el cliente. Otros sin embargo plantean que es un trabajo en conjunto, este criterio está representado por un 18,75% del total.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

A partir de la pregunta 11 de la entrevista se pudo determinar la importancia que representa la priorización de requisitos. El 100% de ellos consideran que es de vital importancia para el desarrollo de software expresando que facilita el trabajo, debido a que sirve como una guía para poder determinar el orden en que serán implementadas las funcionalidades.

A continuación se presentan algunas de las dificultades encontradas en la priorización que fueron obtenidas a partir de las respuestas dadas a la pregunta 12:

- ✓ Desconocimiento por parte del cliente en cuanto al tema, lo que trae consigo desacuerdos entre clientes y desarrolladores.
- ✓ Poca comunicación con el cliente, en muchos casos los encuentros planificados no se dan lo que dificulta el trabajo.
- ✓ Desorganización en la priorización y las personas involucradas no juegan un buen papel como es el caso de arquitectos y analistas.
- ✓ Clasificación de muchos casos de uso como críticos, trayendo consigo que no se entreguen las versiones en el tiempo establecido.

Con la realización de la última pregunta de la entrevista se pudo obtener aquellos indicadores que son considerados importantes en el momento de asignarle la prioridad a los requisitos. De 16 personas entrevistadas 13 de ellas consideran en primer lugar el uso del indicador beneficio y en segundo lugar con un número de 7 personas el indicador dependencia. Algunos proyectos pero en un menor número 2 y 1 hacen referencia a la necesidad de utilizar el indicador costo y frecuencia respectivamente, señalan que en ocasiones han tenido que aplazar la implementación de requisitos debido a la necesidad de equipos costosos y que la reutilización de funcionalidades a implementar reduce en gran proporción el tiempo de implementación por lo que se hace necesario tener en cuenta las funcionalidades más utilizadas.

Encuesta:

Para conocer la realización de la priorización de requisitos con el uso de indicadores en la Universidad se realizó una encuesta, la cual fue entregada a las personas entrevistadas para que fuera respondida por algún integrante del proyecto. Fue aplicada a 16 personas de las cuales el 37,5% estaba representado por analistas principales y el 18,75% ocupaban el cargo de jefes de proyectos y analistas. También fueron encuestados aunque con solo un 6,25% del total arquitectos, diseñadores de base de datos, jefes de módulos y sublíderes de proyectos. De dicha encuesta se obtuvo los resultados que se describen a continuación.

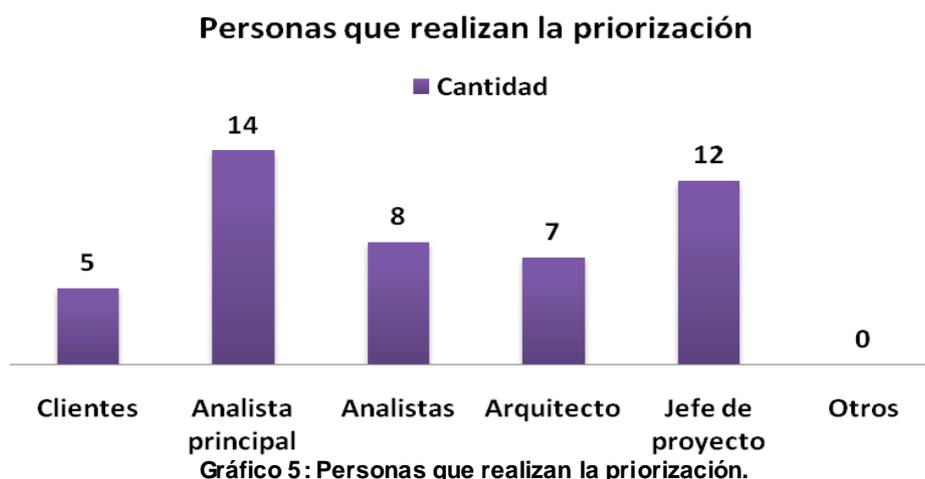
De 16 personas encuestadas un 93,75% y un 6,25% respondió que tenían experiencia en la priorización de requisitos: poca y mucha respectivamente. Según los valores obtenidos en la pregunta

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2 se puede decir que a pesar de que la mayoría tiene poca experiencia en este importante trabajo es válido aclarar que todos tienen aunque sea un conocimiento ínfimo y han trabajado en la priorización.

Un 75% de las personas encuestadas respondió que realizaban siempre la priorización de requisitos en su proyecto y solo un 25% lo realiza a veces. De la información obtenida a partir de la pregunta 3 se puede concluir que no se le da a la priorización toda la importancia requerida.

A partir de la pregunta 4 de la encuesta para obtener por qué personas es realizada la priorización, se obtuvo como resultado que 14 personas de un total de 16 encuestados consideran que los analistas principales son las personas que más influyen en la priorización. Son los analistas los que tienen la mayor comunicación con el cliente y los que mejor entienden las necesidades y expectativas de los mismos. A pesar de esto 12 personas consideran que los jefes de proyectos tienen buena participación en la priorización de requisitos, debido a que son los de mayor autoridad en el proyecto y deben conocer todas las decisiones que sean tomadas en el mismo. Los analistas y arquitectos con un número de 8 y 7 personas consideran también que tienen una buena participación, estos últimos debido a que son los que integran todos los componentes y capas de implementación para el despliegue final. A pesar de ello también hay otras personas en el proyecto que se vinculan a la hora de tomar decisiones para darles prioridad a los requisitos al igual que la opinión de los clientes que no son más que las personas que saben lo que quieren y como lo quieren. En el caso de los clientes se puede decir que no siempre realizan directamente la priorización pero sin embargo la mayoría de las veces su criterio juega un papel importante para poder determinar la prioridad. En el Gráfico 5 se pueden observar datos estadísticos que demuestran el análisis realizado.



CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Con la realización de la pregunta 5 de la encuesta, se pudo obtener los criterios de mayor importancia que se tienen en cuenta en el momento de darle prioridad a los requisitos. Según los resultados se obtiene como criterio más importante el del cliente con un 93,75% de un total de 16 encuestas. Es válido aclarar que el cliente no siempre forma parte de las personas que llevan a cabo la priorización dentro del proyecto pero su criterio influye directamente. Otros criterios que son sumamente importantes es el del analista principal el cual está representado por un 81,25% del total y el del líder de proyecto con un 75%. De acuerdo con los resultados obtenidos se puede decir que el 56,25% de las personas encuestadas coinciden en que el criterio de los analistas también es importante dentro de la priorización al igual que el del arquitecto con una representación del 50%. Se puede decir que la priorización es un trabajo en conjunto y que para poder obtener resultados satisfactorios es necesario tener en cuenta el criterio de todos los involucrados de una forma u otra.

En las respuestas dadas a la pregunta 6 queda demostrado que a pesar de que la priorización no es realizada con todo el rigor y la exactitud que merece de las 16 personas encuestadas, 10 de ellas opinan que es sumamente importante y 6 que es simplemente importante.

La pregunta 7 de la encuesta responde al papel importante que tiene el cliente en la priorización de requisitos. Para evaluar esta pregunta se puso de manifiesto que el entrevistado diera su opinión de por qué el papel del cliente era o no importante en la priorización. Como resultado se obtuvo que el 100% estuvo de acuerdo en que el cliente juega un papel importante dentro de la priorización. A continuación se presentan algunas de las razones dadas por las personas entrevistadas.

- ✓ El cliente es el destinatario final de lo que se va a desarrollar en el proyecto y por lo general él decide cuándo, cómo o qué es lo que se le va a entregar en cada iteración.
- ✓ A partir de lo que el cliente desea y necesita todo el equipo de desarrollo analiza y determina la prioridad incluyéndolo a él en la determinación final.
- ✓ El cliente según lo que le hace falta precisa el orden de entrega de cada requisito ya desarrollado.

En el Gráfico 6 se muestra el porcentaje que representa para la muestra analizada cada una de las metodologías utilizadas a partir de la pregunta 8 de la encuesta. Como se puede observar la metodología más utilizada es RUP, metodología pesada la cual representa el 69% del total y seguido de ella con una representación del 19% se encuentra la metodología FDD, estas no definen un amplio número de indicadores para realizar la priorización. Por último, con solo un 6% del total se encuentra la metodología Scrum-XP y la metodología enfocada a procesos de negocio. La metodología Scrum-XP es una mezcla de las metodologías Scrum y XP, estas trabajan con una lista de requisitos priorizada

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

que es elaborada por el cliente con la ayuda del facilitador y del equipo de desarrollo, balanceando el valor que cada requisito aporta al negocio frente al costo estimado para su desarrollo.

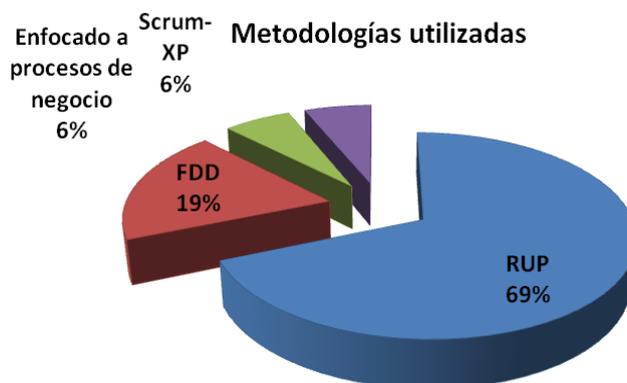


Gráfico 6: Metodologías utilizadas.

A partir de los resultados de la pregunta 9 de la encuesta se pudo determinar que el 50% de los proyectos encuestados hacen uso solamente de los indicadores que propone la metodología. El 31,25% de ellos hacen uso de otros indicadores además de los que propone la metodología. El 18,75% no utilizan los indicadores que son propuestos por la metodología.

De la pregunta 10 para identificar los indicadores que más son utilizados en los proyectos de la Universidad, se obtuvo que la mayoría ponen en práctica la dependencia y el beneficio representados por un 75% para un total de 16 encuestados. Otro de los más utilizados es el costo el cual representa un 62,5% del total. También son empleados pero en menor escala la estabilidad y la frecuencia con un 50% y un 31,25% respectivamente.

A pesar de estos resultados la moda obtenida demuestra que el beneficio es el indicador que consideran sumamente importante, mientras que la dependencia y la frecuencia con una moda de 3 son consideradas como importantes. La estabilidad y el costo son considerados pocos importantes con una moda de 2. Esto demuestra el gran desequilibrio y desconocimiento que existe en el empleo e importancia de los indicadores por parte de las personas que lo utilizan. Para obtener estas respuestas se realizó la pregunta 11 donde había que evaluar el nivel de importancia que representaban un conjunto de indicadores a través de una escala del 1 al 4 de la siguiente manera:

- ✓ 4: sumamente importante
- ✓ 3: importante
- ✓ 2: poco importante
- ✓ 1: nada importante

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

En el gráfico 7 se muestran las formas de asignar valores a los indicadores determinados a partir de los resultados obtenidos de la pregunta 12. Como se puede observar de un total de 16 encuestados 13 de ellos plantean que asignan estos valores de acuerdo con la experiencia de los involucrados en proyectos anteriores y sólo 3 tienen en cuenta el criterio de especialistas. Por lo que se puede afirmar que existen problemas en cuanto a demostrar que tan eficiente puede ser la priorización, pues no utilizan ningún método matemático que demuestre que lo que se está realizando es totalmente correcto y factible. Esto también es demostrado cuando reafirman al igual que en las entrevistas que los valores que se les dan a los indicadores por lo general son valores cualitativos.

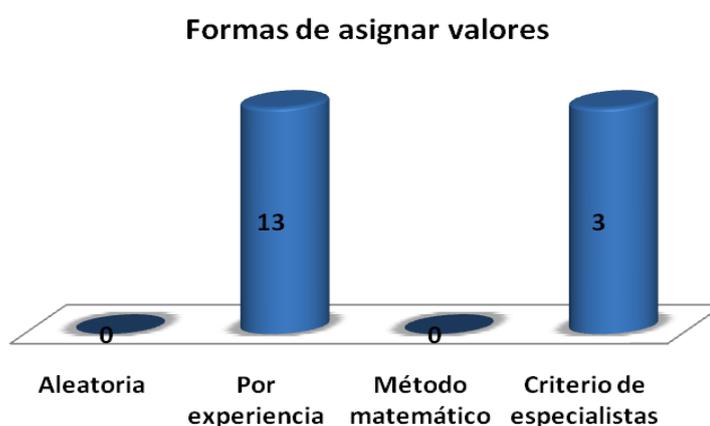


Gráfico 7: Formas de asignar valores.

Con la aplicación de la pregunta 13 de la encuesta se puede decir que la gran mayoría de los proyectos les asignan valores cualitativos a los indicadores a través de opiniones, percepciones o juicio de parte de las personas involucradas en la priorización. En el Gráfico 8 se pueden observar datos numéricos que representan los tipos de valores asignados.



Gráfico 8: Tipos de valores.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

También se probó a partir del resultado de la pregunta 14 que a pesar de que el 56,25% hace uso permanente de indicadores hay un 43,75% que no los emplean, pudiendo reafirmar el desconocimiento acerca de las ventajas y facilidades del uso de indicadores para priorizar.

El 93,75% de las personas encuestadas consideran que la priorización de requisitos suele ser necesaria a pesar de que no se ha realizado con el rigor que amerita. El 25% del total considera que por la falta de indicadores puede hasta llegar a ser un poco complicado, por estas mismas razones el 18,75% de ellos considera que es tedioso y engorroso.

De un total de 16 encuestados 6 de ellos consideran que la priorización les ha ayudado siempre a facilitar el desarrollo de las actividades y 4 de ellos en que casi siempre les ha facilitado ese desarrollo. También 3 y 2 de ellos consideran que la priorización les ha permitido siempre y casi siempre respectivamente reducir el costo y tiempo de las implementaciones, lo que demuestra que si se logra hacer de la priorización un proceso sumamente importante y correcto los resultados del desarrollo de los productos serían mejores. En la Tabla 2.1 se puede observar el porcentaje a partir del resultado obtenido de la pregunta 16 de la encuesta en la que los usuarios encuestados marcaron las ventajas que les ha proporcionado el uso de indicadores para priorizar requisitos funcionales en cada uno de los proyectos.

Ventajas	Siempre	Casi siempre	Algunas veces	Nunca
Facilitar el desarrollo de actividades	42,86 %	28,57 %	0 %	0 %
Reducir el costo de implementación	21,43 %	14,29 %	7,14 %	7,14 %
Reducir el tiempo de implementación	21,43 %	28,57 %	7,14 %	0 %

Tabla 2.1: Ventajas del uso de indicadores.

A continuación se presentan algunas de las personas que según el criterio expresado por los encuestados deben ser las personas que realicen la priorización de requisitos:

- ✓ Clientes con mayor conocimiento sobre lo que se quiere realmente que se lleve a cabo por el equipo de desarrollo.
- ✓ Analistas
- ✓ Jefe de proyecto
- ✓ Arquitecto
- ✓ Analista principal
- ✓ Especialista de Base de Datos

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

2.5. Conclusiones del diagnóstico

El árbol de problemas “es un conjunto de los principales problemas de una situación, con relaciones de causa y efecto establecidas entre ellos.” (Castro, 2008) Con el propósito de representar los resultados del diagnóstico interno realizado se resumen los problemas identificados en el siguiente árbol de problemas (ver Figura 4).

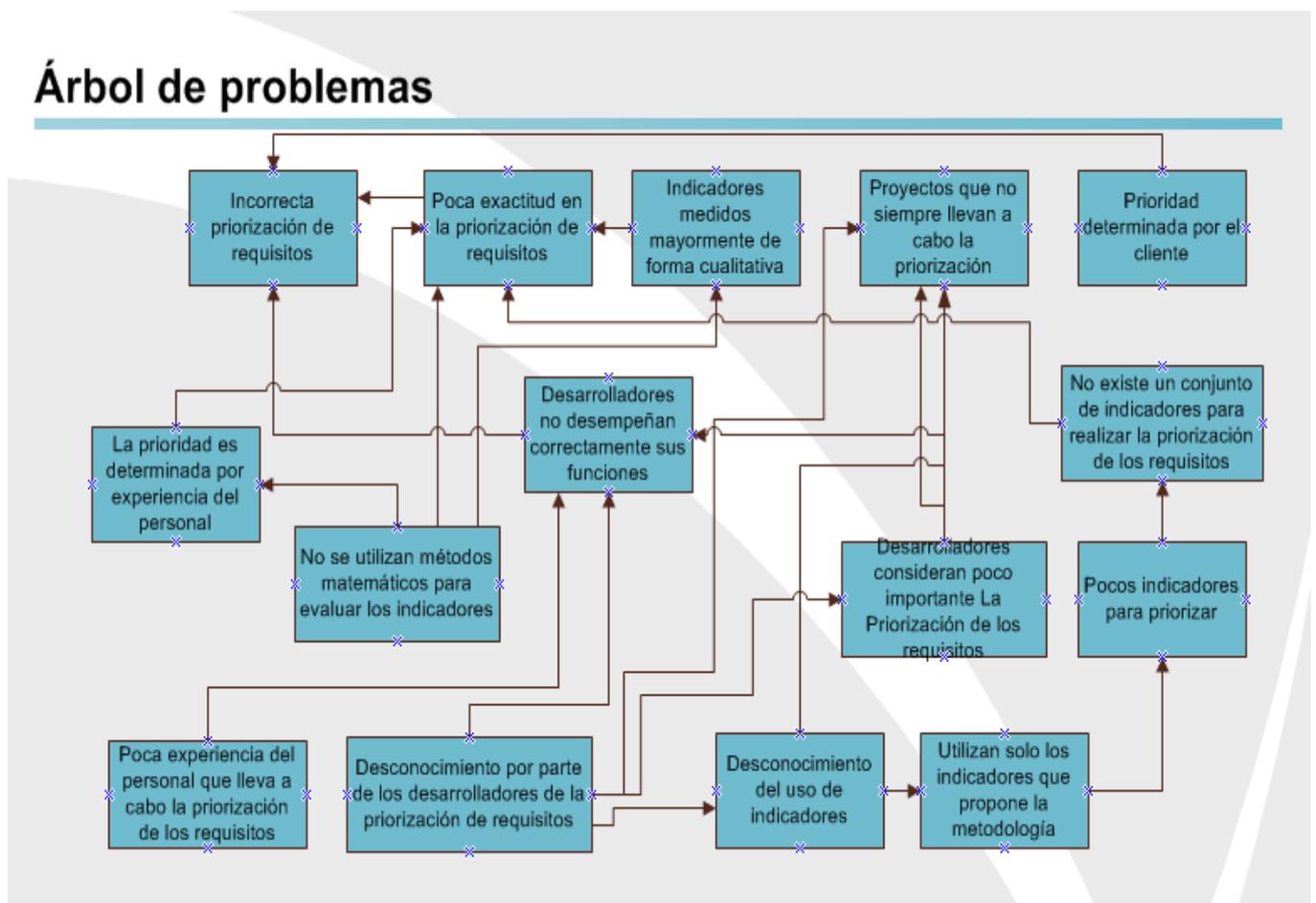


Figura 4: Árbol de problemas.

2.6. Propuesta de solución

Con el objetivo de darle solución a los problemas encontrados en los proyectos de software de la Universidad de la Ciencias Informáticas en cuanto a la priorización de requisitos se define un conjunto de indicadores. Se propone un conjunto de pasos a seguir para medir estos indicadores de forma

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

cuantitativa y cualitativa, y así poder obtener una lista de requisitos priorizados teniendo en cuenta los indicadores propuestos.

Nombre del indicador: Influencia de riesgo.

Descripción: Este indicador permite identificar qué requisitos serán los más importantes, según la influencia de los riesgos identificados en el proyecto sobre los mismos y el nivel de importancia que representa la mitigación de estos riesgos para el desarrollo de un requisito.

Roles que se ocupan de medir el indicador:

- ✓ Líder del proyecto: Gestiona los riesgos asociados al proyecto por tanto está capacitado para priorizar teniendo en cuenta los riesgos asociados a cada requisito.
- ✓ Analista: Gestiona los cambios asociados a los requisitos establecidos, los problemas en la definición de los requisitos y los riesgos asociados a la gestión de requisitos.

Forma de medirlo:

Paso 1

Listar en una tabla todos los requisitos (RF1, RF2,..., RFn) y riesgos (R1, R2,..., Rn) del software como se muestra en la Tabla 2.2.

Requisitos	R1	R2	R3	R4	R5	Rn
RF1						
RF2						
RF3						
RFn						

Tabla 2.2: Requisitos y riesgos del software.

Paso 2:

Identificar por cada requisito (RF1, RF2,..., RFn) que riesgos (R1, R2,..., Rn) se encuentran asociados a él. Determinar los valores (V_{ij}) reflejados en la Tabla 2.3, que representan el nivel de importancia que posee mitigar un riesgo para la realización del requisito satisfactoriamente. En caso de que el requisito (RFn) no esté asociado al riesgo (Rn) no se asignan niveles de importancia y el valor V_{ij} es igual a 0. Estos valores quedarán registrados en la Tabla 2.4:

Nivel de importancia	Valor (V_{ij})
Nada importante	1
Poco importante	2
Importante	3

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Muy Importante	4
----------------	---

Tabla 2.3: Importancia para influencia de riesgo.

Requisitos	R1	R2	R3	R4	R5	Rn
RF1	V ₁₁	V ₁₂	V ₁₃	V ₁₄	V ₁₅	V _{1n}
RF2	V ₂₁	V ₂₂	V ₂₃	V ₂₄	V ₂₅	V _{2n}
RF3	V ₃₁	V ₃₂	V ₃₃	V ₃₄	V ₃₅	V _{3n}
RF4	V ₄₁	V ₄₂	V ₄₃	V ₄₄	V ₄₅	V _{4n}
RF5	V ₅₁	V ₅₂	V ₅₃	V ₅₄	V ₅₅	V _{5n}
RFn	V _{n1}	V _{n2}	V _{n3}	V _{n4}	V _{n5}	V _{nn}

Tabla 2.4: Valores de influencia de riesgo.

Paso 3:

Calcular el total de riesgos (Tr) asociado a cada uno de los requisitos. Asumiendo que para cada RFn el total de riesgo Tr: es la cantidad de riesgos asociados a cada requisito representado en la tabla por un valor (V_{ij}) no nulo. Ver ejemplo Tabla 2.5

Requisitos	R1	R2	R3	R4	R5	R6	Total de riesgo(Tr)
RF1	0	1	3	2	0	0	3
RF2	2	0	0	3	0	0	2
RF3	0	0	0	2	0	0	1
RF4	1	0	4	3	0	0	3
RF5	0	0	2	4	2	1	4

Tabla 2.5: Total de riesgos asociados.

Paso 4:

Calcular el promedio de importancia de riesgo (Pi) que influye en cada requisito a partir de la fórmula siguiente:

$$P_i = \sum V_{ij} / n$$

Donde n es el número de riesgos asociados al software y V_{ij} es el valor de importancia de todos los riesgos asociados al requisito que está siendo analizado. Ver ejemplo Tabla 2.6

Requisitos	R1	R2	R3	R4	R5	R6	Total de riesgo (Tr)	Promedio de Importancia (Pi)
RF1	0	1	3	2	0	0	3	1
RF2	2	0	0	3	0	0	2	0,83

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

RF3	0	0	0	2	0	0	1	0,33
RF4	1	0	4	3	0	0	3	1,33
RF5	0	0	2	4	2	1	4	1,5

Tabla 2.6: Importancia de riesgo.

Paso 5:

Graficar los valores del promedio de importancia de riesgo (P_i) en el eje X y el total de riesgos (T_r) asociado a cada requisito en el eje Y como se muestra en el Gráfico 9.

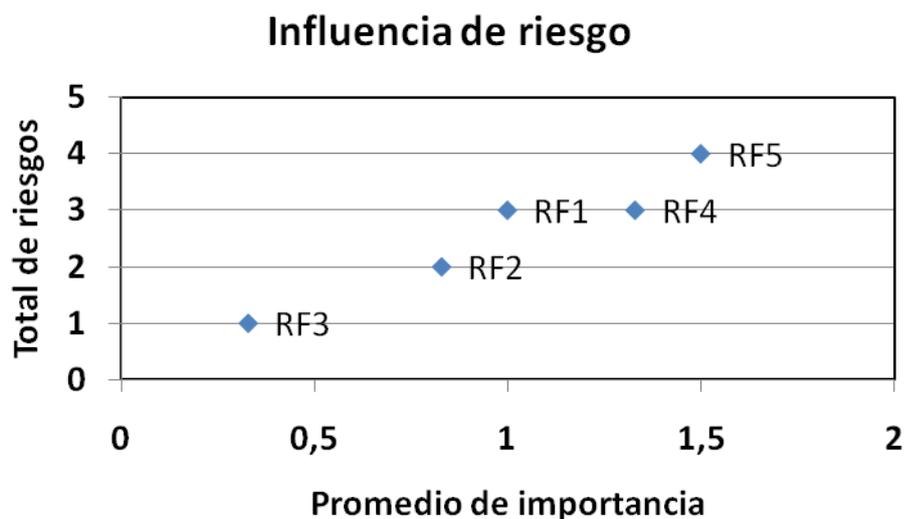


Gráfico 9: Influencia de riesgos.

Paso 6:

A partir del gráfico obtenido crear la lista de requisitos priorizados teniendo en cuenta que mientras menor sea el valor del promedio de importancia (P_i) y el del total de riesgos (T_r) asociados menor será la influencia de riesgo (I_r). Para la creación de la lista primeramente debe tenerse en cuenta el promedio de importancia obtenido de cada requisito y luego analizar el valor del total de riesgos asociados. En caso de coincidencia del promedio de importancia entre requisitos se tomará como el de mayor prioridad de realización aquel requisito que menor número de riesgos tenga asociado. Si coincide el total de riesgos para varios requisitos se tomará el requisito que posea menor promedio de importancia como el de mayor prioridad de realización. Basándose en el ejemplo del Gráfico 9 quedaría organizado de la siguiente forma: R/: RF3, RF2, RF1, RF4, RF5

Nombre del indicador: Beneficio.

Descripción: Está dado por el nivel de importancia que representa una funcionalidad del sistema para el correcto desarrollo del negocio.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Roles que se ocupan de medir el indicador:

En el caso de ser un servicio de software posee un cliente definido y la prioridad es determinada por:

- ✓ Experto funcional del negocio o cliente: Es la persona que conoce las funcionalidades del negocio y cuán importantes pueden ser para el desarrollo del mismo.
- ✓ Analista: Se encarga de trabajar con los proveedores de requisitos (funcionales, directivos, expertos) en función de analizar y obtener la visión del producto y las necesidades del cliente. Además, es el encargado de obtener los requisitos del sistema a partir de un conocimiento previo del negocio.

En el caso de un producto sin cliente definido:

- ✓ Experto del mercado: Este es el encargado de realizar un estudio de mercado de software similar.
- ✓ Analista: Tiene como responsabilidad obtener los requisitos a partir del estudio de mercado realizado.

Forma de medirlo:

Paso 1:

Asignar los valores numéricos mostrados en la Tabla 2.7 a partir de los siguientes criterios:

- ✓ Muy importante: Funcionalidad vital para el sistema, sin esta la aplicación no puede ser liberada.
- ✓ Importante: Funcionalidad importante que sirve de apoyo a las funcionalidades muy importantes.
- ✓ Poco importante: La aplicación tiene fuertes deficiencias si no se implementan. Complementan aquellos requisitos que son considerados importantes, ejemplo de ellas: generación de reportes, datos estadísticos y supervisión.
- ✓ Nada importante: Funcionalidad que sería conveniente desarrollar, no intervienen en el logro de los objetivos trazados con el sistema.

Niveles de importancia	Valor (Vx)
Muy importante	1
Importante	2
Poco importante	3
Nada importante	4

Tabla 2.7: Niveles de importancia.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Paso 2:

Registrar los valores de importancia y crear una lista ordenada ascendentemente de acuerdo con estos valores como se muestra en el ejemplo de la Tabla 2.8: RF2, RF4, RF5, RF1, RF3.

Requisitos	Nivel de importancia(Vx)
RF1	3
RF2	1
RF3	4
RF4	2
RF5	2

Tabla 2.8: Ejemplo indicador beneficio.

Nombre del Indicador: Frecuencia.

Descripción: Identifica las funcionalidades con mayor nivel de ocurrencia en el tiempo dentro de los procesos que se desarrollan en el negocio.

Roles que se ocupan de medir el indicador:

- ✓ Experto del mercado: Conoce a partir del estudio de mercado realizado cuales funcionalidades pueden tener menor o mayor nivel de concurrencia.
- ✓ Experto funcional del negocio: Conoce el desarrollo de los procesos del negocio, lo cual le permite identificar el nivel de concurrencia de las mismas.
- ✓ Analista: Identifica las funcionalidades del sistema a partir de los conocimientos del negocio que le permite conocer el nivel de concurrencia de las funcionalidades dentro del negocio.

Forma de medirlo:

Paso 1:

Teniendo en cuenta los siguientes criterios de frecuencia y los criterios de ocurrencia ubican cada requisito de acuerdo con dichos criterios en la Tabla 2.9.

Criterios de frecuencia:

- ✓ Muy frecuente: Funcionalidad que es utilizada diariamente
- ✓ Frecuente: Funcionalidad importante que es utilizada semanalmente.
- ✓ Poco frecuente: Aquellos requisitos que son utilizados mensualmente.
- ✓ Nada frecuente: Funcionalidad que es utilizada rara vez.

Criterios de ocurrencia:

- ✓ Muy ocurrente: Funcionalidad que es utilizada por más de 10 personas.
- ✓ Ocurrente: Funcionalidad que es utilizada de 4 a 10 personas.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

- ✓ Poco ocurrente: Funcionalidad que es utilizada de 1 a 3 personas

Niveles de frecuencia	Muy ocurrente	Ocurrente	Poco ocurrente
Muy frecuente	RFn	RFn	RFn
Frecuente	RFn	RFn	RFn
Poco frecuente	RFn	RFn	RFn
Nada frecuente	RFn	RFn	RFn

Tabla 2.9: Niveles de frecuencia y ocurrencia.

Paso 2:

Crear una lista priorizada por cada nivel de frecuencia teniendo en cuenta que mientras mayor sea la ocurrencia mayor será la prioridad. Ver ejemplo Tabla 2.10.

Niveles de frecuencia	Muy ocurrente	Ocurrente	Poco ocurrente
Muy frecuente	RF2	RF4	
Frecuente	RF1	RF5	
Poco frecuente			
Nada frecuente		RF3	

Tabla 2.10: Ejemplo de frecuencia.

Muy frecuente: R/: RF2, RF4

Frecuente: R/: RF1, RF5

Poco frecuente: -

Nada frecuente: R/: RF3

Paso 3:

Crear la lista priorizada final de frecuencia teniendo en cuenta que mientras mayor sea la frecuencia y la ocurrencia mayor será la prioridad.

R/: RF2, RF4, RF1, RF5, RF3

Indicador: Dependencia.

Descripción: Está dado por la necesidad de implementación de aquellos requisitos de los cuales dependen otros requisitos.

Roles que se ocupan de medir el indicador:

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

- ✓ Analista: Crea los requisitos de los productos y de calidad de servicios, sus especificaciones y prototipos no funcionales por tanto es la persona que más conoce acerca de las dependencias que se establecen.

Forma de medir: El indicador dependencia tiene como particularidad que no genera una lista priorizada de requisitos, su resultado final está dado por un grafo que contiene las dependencias entre los requisitos analizados.

Paso 1:

Asignar valores de dependencia (D_{ij}). Donde $D_{ij}=1$: existe dependencia, $D_{ij}=0$: no hay dependencia, analizando cada requisito (RFn) de la columna. Registrar en la Tabla 2.11 estos valores.

Requisitos	RF1	RF2	RF3	RF4	RF5	RFm
RF1	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{1m}
RF2	D_{21}	D_{22}	D_{23}	D_{24}	D_{25}	D_{2m}
RF3	D_{31}	D_{32}	D_{33}	D_{34}	D_{35}	D_{3m}
RF4	D_{41}	D_{42}	D_{43}	D_{44}	D_{45}	D_{4m}
RF5	D_{51}	D_{52}	D_{53}	D_{54}	D_{55}	D_{5m}
RFn	D_{n1}	D_{n2}	D_{n3}	D_{n4}	D_{n5}	D_{nm}

Tabla 2.11: Relación de dependencia.

Paso 2:

Calcular el total de hijos (Th) de cada uno de los requisitos para poder determinar cuántos requisitos dependen de él (valores de la columna) y calcular el total de padres (Tp) con el objetivo de saber el número de dependencias de cada requisito (valores de la fila). Ver ejemplo Tabla 2.12:

Requisitos	RF1	RF2	RF3	RF4	RF5	Total (Tp)
RF1	0	0	0	0	0	0
RF2	1	0	1	0	0	2
RF3	1	0	0	0	0	1
RF4	0	1	0	0	0	1
RF5	0	1	1	1	0	3
Total (Th)	2	2	2	1	0	

Tabla 2.12: Total de hijos y padres por requisitos.

Paso 3:

Crear un grafo como el que se muestra en la Figura 5 a partir de la Tabla 2.12.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

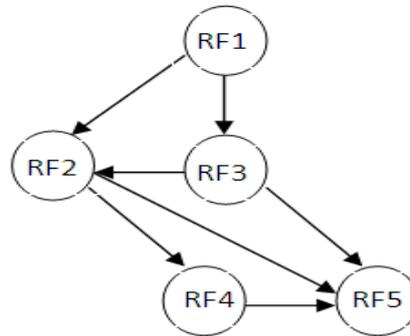


Figura 5: Grafo de dependencia.

Una vez analizados los indicadores influencia de riesgo, beneficio y frecuencia, se conforma una lista a partir del resultado de los mismos, luego se analiza dicha lista teniendo en cuenta el grafo obtenido al analizar el indicador dependencia con el fin de obtener la lista de prioridad final.

Crear la lista de prioridad final teniendo en cuenta el resultado del conjunto de indicadores planteados.

Paso 1:

Llenar los valores de la Tabla 2.13, donde P_{ij} es la posición del requisito en cada una de las listas formadas al analizar los indicadores influencia de riesgo, beneficio y frecuencia.

Requisitos	Influencia de riesgo	Beneficio	Frecuencia
RF1	P_{11}	P_{12}	P_{13}
RF2	P_{21}	P_{22}	P_{23}
RF3	P_{31}	P_{32}	P_{33}
RF4	P_{41}	P_{42}	P_{43}
RF5	P_{51}	P_{52}	P_{53}
RFn	P_{n1}	P_{n2}	P_{n3}

Tabla 2.13: Resultado de indicadores.

Paso 2:

Calcular el número de prioridad (NP_i) de cada requisito y registrarlo en la Tabla 2.14 a partir de la fórmula siguiente:

$$NP = \sum P_{ij} / 3$$

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Requisitos	Influencia de riesgo	Beneficio	Frecuencia	Número de prioridad
RF1	P ₁₁	P ₁₂	P ₁₃	NP ₁
RF2	P ₂₁	P ₂₂	P ₂₃	NP ₂
RF3	P ₃₁	P ₃₂	P ₃₃	NP ₃
RF4	P ₄₁	P ₄₂	P ₄₃	NP ₄
RF5	P ₅₁	P ₅₂	P ₅₃	NP ₅
RFn	P _{n1}	P _{n2}	P _{n3}	NP _n

Tabla 2.14: Valores de prioridad por requisitos.

Paso 3:

Crear una lista de requisitos ordenada de acuerdo con el número de prioridad (NP_i). En caso de que existan valores del número de prioridad (NP_i) iguales ver los valores obtenidos del indicador que tenga mayor importancia dentro del proyecto y tomar el menor como de mayor prioridad de implementación y situar primero en la lista priorizada. Ver ejemplo Tabla 2.15:

Requisitos	Influencia de riesgo	Beneficio	Frecuencia	Número de prioridad
RF1	2	4	3	3
RF2	4	1	1	2
RF3	1	5	5	3,67
RF4	3	2	2	2,33
RF5	5	3	4	4

Tabla 2.15: Ejemplo lista priorizada.

R/: RF2, RF4, RF1, RF3, RF5

Paso 4:

A partir del grafo obtenido al analizar el indicador dependencia y la lista de prioridad teniendo en cuenta los indicadores influencia de riesgo, beneficio y frecuencia obtenidos en el paso 3 se elabora la **lista final de prioridad**. Asumiendo que para cada requisito obtenido de la lista de indicadores se incluye delante de ellos aquellos requisitos de los cuales depende dicha funcionalidad. Es necesario realizar este paso tantas veces como sea necesario con el fin de establecer el orden de acuerdo con las dependencias existentes.

Por ejemplo para el caso del primer elemento de la lista R/: RF2, RF4, RF1, RF3, RF5.

Se observa en la Figura 5 que RF2 depende del RF1 y RF3 por lo que se hace necesario asignarles mayor prioridad a RF1 y a RF3. R/: RF2, RF4, RF1, RF3, RF5.

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

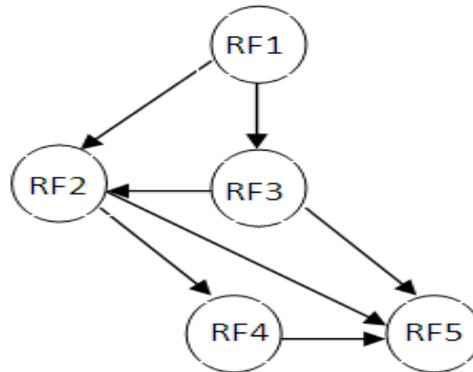


Figura 5: Grafo de dependencia.

Se procede a situar a RF1 en la posición delante del elemento analizado (RF2) y se modifica la posición en que se encontraba obteniendo la lista modificada.

R/: RF1, RF2, RF4, RF1, RF3, RF5

R/: RF1, RF2, RF4, RF3, RF5

Se realiza la misma operación según se encuentren aquellos elementos de los cuales depende RF2.

R/: RF1, RF3, RF2, RF4, RF3, RF5

R/: RF1, RF3, RF2, RF4, RF5

Una vez terminado el análisis del RF2 se procede a analizar nuevamente la lista modificada desde el primer elemento de acuerdo con el grafo.

R/: RF1, RF3, RF2, RF4, RF5

En el ejemplo anterior al analizar nuevamente cada uno de los elementos de la lista todas sus dependencias estaban colocadas correctamente, por tanto, la lista de prioridad final según el ejemplo mostrado quedaría de la siguiente forma: R/: RF1, RF3, RF2, RF4, RF5.

Después del análisis realizado los resultados permiten mostrar los requisitos ordenados teniendo en cuenta su prioridad de ejecución dentro del proyecto, esto facilitará la posterior planificación de cada una de las iteraciones del desarrollo de software. Sin embargo, para la correcta asignación del personal, tiempo y recursos dentro de cada iteración, con el fin de realizar las funcionalidades del

CAPÍTULO 2: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

sistema, es importante tener en cuenta el indicador complejidad que permite estimar el esfuerzo necesario para la realización de un requisito. A pesar de la importancia del papel de la complejidad para la planificación de las iteraciones no es factible tener en cuenta este indicador para asignar prioridades; debido a que generalmente dentro de los proyectos de desarrollo de software los requisitos más importantes para los clientes y desarrolladores, los de mayor impacto en la arquitectura y los de mayor frecuencia, son los que poseen una alta complejidad para su desarrollo. Por tanto, la complejidad no debe influir en el orden de prioridad para la implementación de un requisito.

Al igual que la complejidad el indicador costo no se incluye dentro de la propuesta porque los proyectos que se desarrollan en la Universidad no estiman el costo de cada requisito. A la hora de determinar el costo de un proyecto, lo cual generalmente solo se aplica para proyectos de exportación, se basan en un método de estimación que no tiene en cuenta el valor de implementación de cada requisito de forma independiente e incluso en la mayoría de los casos las estimaciones de este costo se obtiene sin haber realizado un levantamiento de requisitos.

2.7. Conclusiones parciales

Con el estudio del estado actual de la priorización de requisitos en los proyectos de software de la Universidad de las Ciencias Informáticas se obtuvo a través de las técnicas encuesta y entrevista los problemas existentes en el campo de acción. La falta de un conjunto de indicadores para priorizar, el desconocimiento por parte de las personas involucradas, las formas de evaluar los indicadores que utilizan y la necesidad del uso de métodos y herramientas fueron las mayores dificultades que se encontraron después de haber realizado el diagnóstico interno. Con el fin de darle solución a los problemas encontrados, se propuso un conjunto de indicadores estableciendo criterios para que los mismos sean medidos tanto de forma cualitativa como cuantitativa, con el objetivo de mejorar la priorización de requisitos funcionales en los proyectos de software de la Universidad de las Ciencias Informáticas.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

3.1. Introducción

A partir del estudio realizado en cuanto a la priorización de requisitos funcionales en los proyectos de software que se desarrollan en la Universidad de las Ciencias Informáticas se obtuvo como resultado de la investigación científica un conjunto de indicadores para priorizar requisitos funcionales. En el presente capítulo para la validación y aceptación de dicha propuesta de solución se decide utilizar la técnica de juicios de expertos y dentro de ella se seleccionó el método de agregados individuales, donde se pone de manifiesto el criterio de un grupo de expertos en el tema a tratar. Dichos expertos son especialistas que poseen un gran conocimiento en Ingeniería de Software. Se describe el proceso de selección de expertos, la elaboración de la encuesta y los resultados arrojados por la misma.

3.2. Juicios de expertos

El método de juicios de expertos es un método aprobado en la comunidad científica, de una elevada madurez y que proporciona estimaciones de precisión moderada.

“Los juicios de expertos se pueden obtener por métodos grupales o por métodos de experto único. Se pueden seguir, entre otros, el método de agregados individuales, el método Delphi, la técnica grupal nominal y el método de consenso grupal.” (Arquer) Con el fin de validar la propuesta de solución del presente trabajo científico se decidió utilizar el método de agregados individuales. Este método “pide individualmente a cada experto que dé una estimación directa de la probabilidad de éxito o de fracaso en cada una de las tareas descritas. Después se tratan estadísticamente los datos recogidos. Lo habitual es calcular la media aritmética del conjunto de estimaciones individualmente obtenidas, para cada tarea. Esto se hace presuponiendo que el conjunto de los datos posibles tiene una distribución simétrica y, por tanto, la media aritmética es un buen índice de tendencia central.

Éste es un método económico porque, al igual que el método Delphi, no exige que se reúna a los expertos en un lugar determinado. Puede parecer un método limitado porque los expertos no pueden intercambiar sus opiniones, puntos de vista y experiencia, ya que se les requiere individualmente; no obstante, esta limitación puede ser precisamente lo que se esté buscando para evitar los sesgos de los datos ocasionados por conflictos interpersonales, presiones entre los expertos, etc.” (Arquer)

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

3.3. Proceso de selección de expertos

“Para que una persona pueda ser etiquetada como «experta» debe poseer un conocimiento profundo de la tarea o actividad que será objeto de análisis y valoración y tiene que estar familiarizada con el sistema en el que esta se desarrolla.” (Arquer) Se define como experto una persona o grupo de ellas especializadas en algún campo, con reconocida competencia, capaz de emitir un criterio concluyente de un problema en cuestión, emitir valoraciones importantes con un alto nivel de conocimiento y hacer recomendaciones de forma competente.

La veracidad y confiabilidad de los resultados emitidos por el experto está en dependencia de las características que este posea las cuales deben ser: calificación técnica, capacidad de emitir una decisión al respecto, conocimientos específicos sobre el tema a evaluar, disposición a participar, la seriedad, la honestidad, la sinceridad, la responsabilidad, prestigio dentro de su colectivo de trabajo, entre otras.

Actualmente los expertos presentan las siguientes ocupaciones un jefe de Departamento del Centro de Identificación y Seguridad Digital, dos analistas principales, un jefe de proyecto, un subdirector de Investigación y postgrado y un especialista.

En el desarrollo de este proceso de validación de acuerdo con el criterio de expertos se consideran tres etapas fundamentales las cuales son:

- ✓ Determinar la cantidad de expertos.
- ✓ Conformar el listado de los expertos.
- ✓ Confirmar la participación de los expertos.

Ningún experto conoce la identidad de los otros que componen el grupo. Esto proporciona un grupo de aspectos positivos tales como:

- ✓ Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros.
- ✓ Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
- ✓ El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

3.3.1. Determinar la cantidad de expertos

Para mayor veracidad de los resultados se hace necesario tomar una muestra de siete expertos debido a la complejidad del contenido que se trata en este trabajo que no es más que la priorización de requisitos funcionales. Este tema es sumamente importante debido a que a través de él se garantiza que el cliente tenga lo que realmente desea en el tiempo en que lo pide.

3.3.2. Conformar el listado de los expertos

La selección de expertos se realizó de forma minuciosa teniendo en cuenta su especialidad, grado científico y currículum. Además de que fuesen expertos en el tema de ingeniería de requisitos y de priorización de estos. Dentro de la selección de los expertos se encuentran personas que participaron en el proyecto de mejora de procesos basados en el modelo CMMI (*Capability Maturity Model Integration*) que se lleva a cabo en la Universidad, específicamente aquellos que formaron parte del equipo del proceso de Administración de Requisitos.

3.3.3. Confirmar la participación de los expertos

Se elaboró un listado con un grupo de expertos en el tema teniendo en cuenta los criterios planteados anteriormente, luego fueron invitados personalmente para participar en la evaluación. A estos se les explicó en qué consistía el trabajo en general, la propuesta a evaluar y el objetivo de la realización de la encuesta, así como el plazo de entrega. Después de haber recibido la respuesta positiva, se estableció el listado final de los expertos, informando a cada especialista su inclusión en el proceso a evaluar y las instrucciones necesarias para contestar las preguntas. De esta forma, culmina el proceso de selección, logrando la participación de los siete expertos escogidos.

3.4. Elaboración de la encuesta

En la elaboración de la encuesta (ver Anexo 3) se tuvo en cuenta la teoría de la comunicación donde se le hace a los expertos un breve resumen del trabajo científico y se les explica la propuesta de solución elaborada, con mecanismos que reduzcan los sesgos en las respuestas, preguntas claras, precisas e independientes. Se crearon preguntas cuantitativas para calcular medias aritméticas y rangos, y cualitativas para la justificación de sus opiniones, las cuales permitirán visualizar la

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

posibilidad que existe de aplicar la propuesta. La encuesta se llevó a cabo de forma anónima y se les dio un plazo de tiempo para entregarla y que pudieran hacer un análisis profundo de la misma.

3.5. Resultado de la validación

Los resultados de la validación arrojaron que el 100% de los expertos consideran que la propuesta propiciará una mejora en la priorización de los requisitos funcionales dentro de los proyectos que se desarrollan en la Universidad de las Ciencias Informáticas; expresando que será de gran ayuda y que dicha propuesta abarca los aspectos más significativos a tener en cuenta a la hora de asignar una correcta prioridad a los requisitos funcionales de software. También estuvieron de acuerdo en que la propuesta elaborada era de gran interés.

Ante la petición de una evaluación de la propuesta de solución confiriéndole valores numéricos en escala del 1 (bajo) al 5 (excelente), teniendo en cuenta cinco criterios se obtuvo los siguientes resultados:

Como se observa en el Gráfico 10 a partir de los resultados que se obtuvieron en cuanto al criterio de mérito científico, la gran mayoría de los expertos le confieren un valor de 4 a la calidad de la investigación y al valor científico de la investigación. Por lo que el 57.14% de los entrevistados le asignan valor 4 al mérito científico, el 28.57% valor 5 y el por ciento restante le asignan valor 3. Quedando por encima los valores 5 y 4.

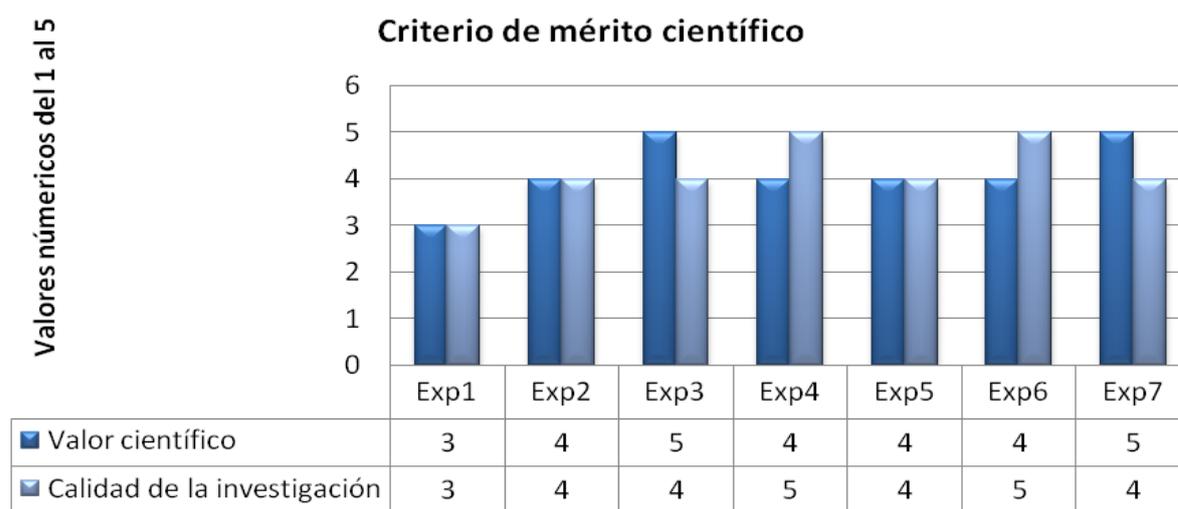


Gráfico 10: Criterio de mérito científico.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

A partir del análisis realizado a los resultados del criterio de implantación expresados en el Gráfico 11 se puede afirmar que la gran mayoría de los expertos consideran que es muy posible la aplicación de la propuesta de solución al igual que es evidente la necesidad de aplicación de dicha propuesta. Esto se demuestra debido a que el 71.43% de los entrevistados plantean la necesidad de empleo que posee la propuesta de solución. A pesar de que un 28.57% de los expertos creen que dicha propuesta tiene una posibilidad de aplicación de valor 3, el 14.28% y el 57.15% le asignan valores de 4 y 5 respectivamente.



Gráfico 11: Criterio de implantación.

Según los resultados reflejados en el Gráfico 12 la gran mayoría de los expertos encuestados consideran que la propuesta de solución tendrá un gran impacto en el desarrollo de software y sobre todo en la priorización de requisitos funcionales en los proyectos de software que se desarrollan en la Universidad de las Ciencias Informáticas. Un aspecto muy positivo que se observa es que el 100% de los expertos consideran que este conjunto de indicadores posee una alta contribución a la priorización de requisitos funcionales de software. También la gran mayoría consideran que la propuesta brinda un alto carácter organizativo, debido a que opinan que les facilitará el trabajo dentro del proyecto a la hora de realizar la priorización de los requisitos funcionales. Esta gráfica demuestra a su vez que la propuesta contribuirá al desarrollo de software representando un 91.71% los valores de los criterios 4 y 5, el 42.86% de los expertos creen que tendrá un buen impacto en los proyectos.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

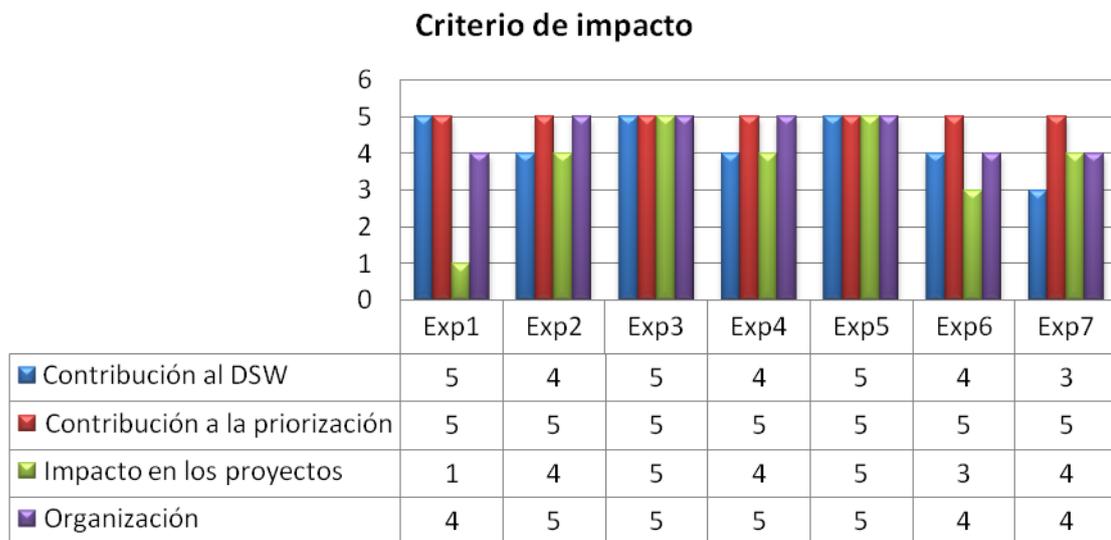


Gráfico 12: Criterio de impacto.

De acuerdo con el criterio de los expertos la propuesta es considerada muy fácil de entender, como se puede observar en el Gráfico 13 el 57% de los encuestados le confirió un valor de 5 a este criterio, el 29% le asignó un valor de 4 y sólo el 14% le otorgó valor 3.

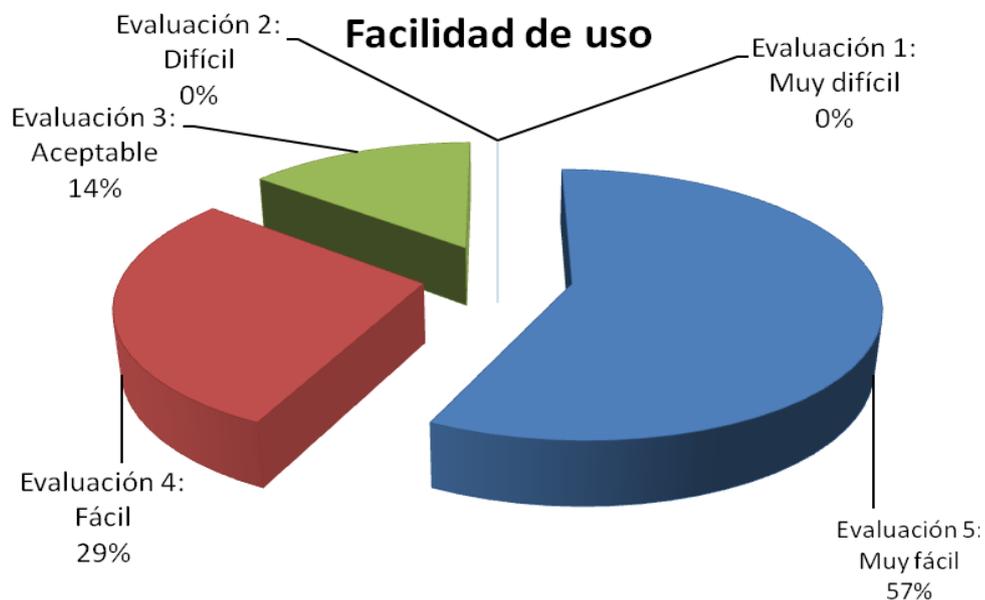


Gráfico 13: Criterio de usabilidad.

Como se puede observar en el Gráfico 14 la mayoría de los expertos consideran excelente y muy buena la propuesta de solución en cuanto a la adaptabilidad de la misma a la priorización de requisitos.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

También consideran que la propuesta posee una excelente capacidad para la admisión de cambios que impliquen la mejora de la misma y de la priorización de requisitos.



Gráfico 14: Criterio de flexibilidad.

De forma general los expertos dieron un alto valor a la propuesta desarrollada ya que en su evaluación los resultados oscilaron generalmente entre 4 y 5 puntos como se puede observar en el Gráfico 15.

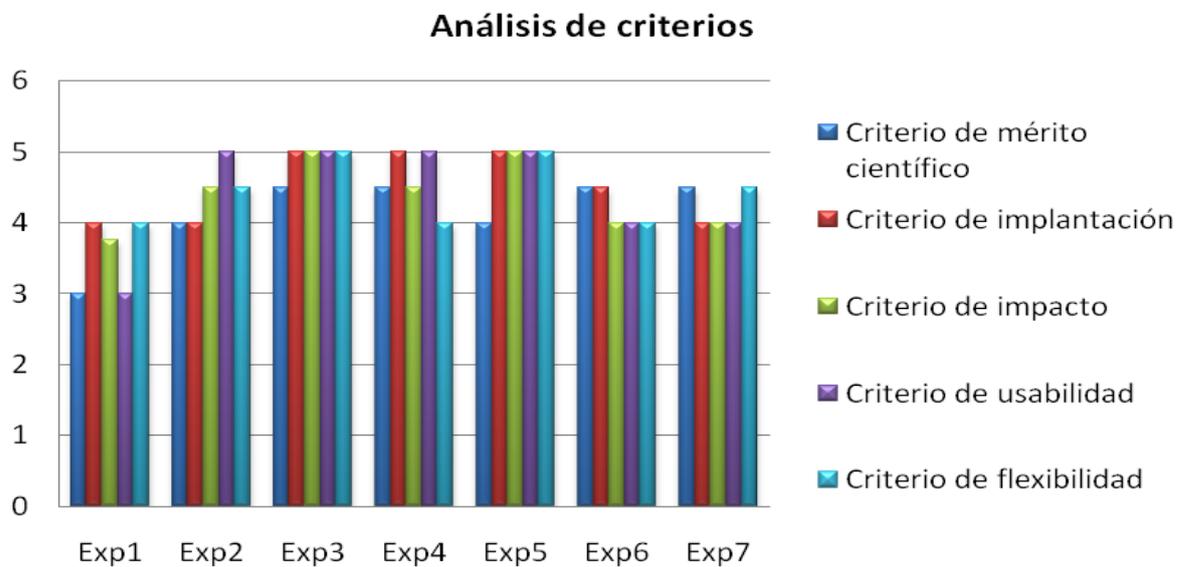


Gráfico 15: Análisis de criterios.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

3.6. Conclusiones parciales

Como parte de este capítulo se realizó la validación y aceptación de la propuesta de solución presentada en el Capítulo 2 mediante la técnica de juicios de expertos específicamente el método de agregados individuales, donde se puso de manifiesto el criterio de un grupo de expertos en el tema. Se realizó una descripción de los pasos utilizados en la selección de los expertos y se efectuó dicha selección. Estos expertos contaban con pleno conocimiento y experiencia en la priorización de requisitos funcionales. Por último, se analizaron los datos obtenidos en las encuestas realizadas a los expertos mediante gráficos. Pudiendo expresar de acuerdo con estos resultados que los objetivos específicos asociados a cada pregunta fueron cumplidos, que todas las preguntas fueron respondidas por parte de los expertos y la gran mayoría de forma positiva. Se puede concluir que la propuesta fue validada por la totalidad de los expertos seleccionados.

CONCLUSIONES

CONCLUSIONES

A partir de un análisis profundo de la ingeniería de requisitos, etapa que forma parte en el desarrollo de software, se conoce la existencia de cuatro fases fundamentales, específicamente la fase de análisis y obtención de requisitos la cual incluye la priorización de requisitos de software. Dicha priorización permite obtener las categorías de los requisitos teniendo en cuenta indicadores y planificar las versiones de los entregables del proyecto. Según el estudio realizado se obtuvo como resultado que diferentes metodologías, métodos y herramientas utilizan indicadores como punto de partida para asignar prioridades a los requisitos funcionales de software. Dentro de los indicadores más utilizados por las diferentes metodologías y métodos estudiados se encuentran el beneficio y el costo, con un menor número y para el caso de los métodos también es utilizada frecuentemente el indicador complejidad.

El diagnóstico realizado a la priorización de requisitos funcionales en los proyectos de software que se desarrollan en la Universidad de las Ciencias Informáticas comprobó la inexistencia de un conjunto de indicadores para priorizar requisitos funcionales que permita obtener una correcta prioridad de dichos requisitos. Además, se evidenció la necesidad de evaluar estos indicadores de forma cuantitativa, así como la necesidad del uso de herramientas y métodos matemáticos con el fin de mejorar la asignación de prioridades a los requisitos funcionales de software.

A partir de los problemas encontrados en el diagnóstico interno realizado se propuso un conjunto de indicadores y se establecieron una serie de pasos para que los mismos sean medidos de forma cualitativa y cuantitativa, con el objetivo de mejorar la priorización de requisitos funcionales en los proyectos de software que se desarrollan en la Universidad de las Ciencias Informáticas. Los indicadores propuestos son: influencia de riesgos, beneficio, frecuencia y dependencia.

A través de la validación de la propuesta se demostró que los indicadores definidos en el presente trabajo se adaptan a las necesidades de la priorización de requisitos y organizan la misma, contribuyen al proceso de desarrollo de software, tienen posibilidades altas de aplicación en los proyectos de la Universidad y son fáciles de entender, posibilitando una mejora en la priorización de los requisitos funcionales de los proyectos de software de la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

RECOMENDACIONES

Como recomendaciones del presente trabajo se tienen:

- ✓ Dar seguimiento al conjunto de indicadores propuestos para priorizar requisitos funcionales, valorando la posibilidad de cambios en los criterios que se establecen en el presente trabajo para evaluar dichos indicadores o estableciendo otros que puedan ser útiles para priorizar requisitos funcionales.
- ✓ Probar la propuesta de solución en diferentes proyectos de software que se desarrollan en la Universidad de las Ciencias Informáticas.
- ✓ Valorar la posibilidad del uso de la media ponderada para el cálculo de la lista de prioridad en los indicadores influencia de riesgo, beneficio y frecuencia en los proyectos que decidan asignar pesos a los indicadores según su nivel de importancia. De igual manera para el caso del cálculo del total de importancia en el indicador influencia de riesgo.
- ✓ A partir de la lista de requisitos que se establecen en los proyectos crear una herramienta que automatice los pasos planteados, para evaluar el conjunto de indicadores, con el objetivo de obtener la lista de requisitos priorizada. Esta herramienta logrará en gran medida disminuir y hacer menos engorroso el trabajo.
- ✓ Profundizar en el estudio de los indicadores complejidad y costo con el propósito de definir las versiones entregables del proyecto, lo cual incidirá en el correcto desarrollo del proceso de priorización de requisitos culminando de esta manera el desarrollo de todos sus objetivos.
- ✓ Continuar el estudio de la propuesta con vista a extender el alcance de la misma a los proyectos que se desarrollan a nivel nacional.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

- ✓ **albet. (s.f.)**. Recuperado el 9 de marzo de 2010, de <http://www.albet.cu/?q=es/node/19>.
- ✓ **Aranda, G. N., Vizcaíno, A., Cechich, A., Piattini, M., & Soto, J. P. (s.f.)**. Una metodología para elicitación de requisitos en proyectos GSD.
- ✓ **Arquer, M. I. (s.f.)**. NTP 401: Fiabilidad humana: métodos de cuantificación, juicio de expertos. España.
- ✓ **Calisoft Centro de Calidad para Soluciones Tecnológicas. (2009)**. Recuperado el 10 de marzo de 2010, de Calisoft Centro de Calidad para Soluciones Tecnológicas:
<http://calisoft.uci.cu/>
- ✓ **Castro, H. M. (noviembre de 2008)**. EL ÁRBOL DE PROBLEMAS. Recuperado el 21 de mayo de 2010, de http://www.iica.int.ni/Estudios_PDF/Arbol_De_Problemas.pdf
- ✓ **Chaves, Michael Arias. 2005**. La ingeniería de requisitos y su importancia en el desarrollo de los proyectos. Costa Rica: s.n., 2005.
- ✓ **Díaz, Yanisleis Pérez y Negret Rodríguez, Grenny. 2009**. Propuesta de guía de desarrollo de software con ajuste a Sistemas de Realidad Virtual. La Habana: s.n., 2009.
- ✓ **Hernández, D. L., & otros. (s.f.)**. Caracterización de la Universidad de las Ciencias Informáticas (UCI). Recuperado el 9 de marzo de 2010, de eumed.net:
<http://www.eumed.net/libros/2009a/514/Caracterizacion%20de%20la%20Universidad%20de%20las%20Ciencias%20Informaticas.html>.
- ✓ **Hurtado, T., & Bruno, G. (s.f.)**. El Proceso de Análisis Jerárquico (AHP) como Herramienta para la Toma de Decisiones en la Selección de Proveedores.
- ✓ **INTECO. 2008**. GUÍA PRÁCTICA DE GESTIÓN DE REQUISITOS. 2008.
- ✓ **Karakostas, Locopoulos y. 1995**. Systems Requirements Engineering. 1995.
- ✓ **La producción en la UCI: una experiencia en marcha. (s.f.)**. Recuperado el 9 de marzo de 2010, de <http://biblioteca.reduc.edu.cu/biblioteca.virtual/cgi/CD-ROM/univerzalizacion.universidad/documentos/21/sala01-La%20producci%C3%B3n%20en%20la%20UCI.pdf>.
- ✓ **La UCI Portal UCI - Universidad de las Ciencias Informáticas. (s.f.)**. Recuperado el 9 de marzo de 2010, de <http://www.uci.cu/?q=node/59>.
- ✓ **McGraw-Hill. 1995**. Systems Requirements Engineering. 1995.

REFERENCIAS BIBLIOGRÁFICAS

- ✓ **Merchán, Luis, Urrea, Alba y Rebollar, Rubén. 2008.** Definición de una metodología ágil de ingeniería de requerimientos para empresas emergentes de desarrollo de software del sur-occidente colombiano. 2008.
- ✓ **Metodologías. (s.f.).** Recuperado el 5 de marzo de 2010, de <http://issi.dsic.upv.es/publications/archives/f-1069167248521/actas.pdf>
- ✓ **Poblete, C. R. (s.f.).** Comunicación: El aspecto humano de la Ingeniería de Requisitos. Chile.
- ✓ **Rascón, E. (10 de febrero de 2010).** Las Prioridades de los Requerimientos de Usuarios. Recuperado el 3 de abril de 2010, de <http://www.eduardorascon.com/>
- ✓ **Sommerville, I. (2005).** La Ingeniería del Software. Séptima Edición. Madrid, España: Addison-Wesley.
- ✓ **Stankevicius, A. G. 2006.** [En línea] 2006. [Citado el: 17 de 3 de 2010.] <http://www.gnu.org/copyleft/fdl.html>.
- ✓ **Team, CMMI Product. 2006.** CMMI® for Development, Version 1.2. 2006.

BIBLIOGRAFÍA

- ✓ **Acuña, K. B. (s.f.).** Caracterización detallada de la metodología de desarrollo de software Scrum. Recuperado el 7 de enero de 2010, de eumed.net: <http://www.eumed.net/libros/2009c/584/Caracterizacion%20detallada%20de%20la%20metodologia%20de%20desarrollo%20de%20software%20Scrum.htm>
- ✓ **Aranda, G. N., Vizcaíno, A., Cechich, A., Piattini, M., & Pablo Soto, J. (s.f.).** Una metodología para elicitación de requisitos en proyectos GSD.
- ✓ **Astudillo, H.** (30 de marzo de 2004). Ingeniería de Software IV: Requerimientos.
- ✓ **Carod, N. M. (s.f.).** Priorización de Requerimientos de Software utilizando una estrategia cognitiva. Buenos Aires, Neuquén, Argentina.
- ✓ **Carvalho, J. P., Franch, X., & Quer, C. (s.f.).** Un Catálogo de Factores de Calidad para la Definición de Requisitos No-Técnicos en la Selección de Componentes COTS. Barcelona, España.
- ✓ **Concepto: Balancear las prioridades en contradicción para maximizar el valor de los stakeholder.** (16 de noviembre de 2008). Recuperado el 27 de octubre de 2009, de http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/concepts/core_principle_balance,_ssG6MMvpEdqukPpotm3DYg.html
- ✓ **Domínguez, L. d.** (mayo de 2006). Sistema de indicadores para evaluar el trabajo de I + D en una empresa de servicios. Estudio de caso. Recuperado el 20 de septiembre de 2009, de <http://www.gestiopolis.com/ads/anuncios/full-leaderboard.htm>
- ✓ **Fernández, D. M., Fernández-Medina Patón, E., & Piattini Velthuis, M.** (2 de junio de 2006). COMPLEMENTANDO A MÉTRICA: PROCESO DE INGENIERÍA DE REQUISITOS DE SEGURIDAD PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN SEGUROS. Sevilla.
- ✓ **Gervás, P.** (octubre de 2004). Capítulo 02 Captura de requisitos.
- ✓ **Hadad, G., Doorn, J., & Kaplan, G. (s.f.).** Explicitar Requisitos del Software usando Escenarios. Argentina.
- ✓ **Hurtado, T., & Bruno, G. (s.f.).** El Proceso de Análisis Jerárquico (AHP) como herramienta para la Toma de Decisiones en la Selección de Proveedores. Proceso de Análisis Jerárquico (AHP).

BIBLIOGRAFÍA

- ✓ **Introducción a la ingeniería de requisitos.** (2005). Recuperado el 9 de enero de 2010, de https://www.u-cursos.cl/ingenieria/2009/1/CC51A/1/material_docente/previsualizar?id_material=213468
- ✓ **Joiz.Net, Itzcoalt Alvarez M.** Desarrollo Ágil con SCRUM.
- ✓ **Korel, George Koutsogiannakis Bogdan.** Experimental Comparison of Code-Based and Model-Based Test. USA: s.n. Vol. 60616.
- ✓ **Mead, Nancy R.** 2006. Requirements Prioritization Introduction. [En línea] septiembre de 2006.
- ✓ **Mellado, D., Rodríguez, M., Fernández-Medina, E., & Piattini, M. (s.f.).** Soporte Automatizado a la Ingeniería de Requisitos de Software. Ciudad Real, Madrid, España.
- ✓ **Merchan, L., Urrea, A., & Rebollar, R.** (marzo de 2008). Definición de una metodología ágil de ingeniería de requerimientos para empresas emergentes de desarrollo de software del sur-occidente colombiano. Revista científica Guillermo de Odcham.
- ✓ **Muslera, R. J. (s.f.).** INGENIERIA DEL SOFTWARE DE GESTION.
- ✓ **Palacio, J.** (2006). Gestión de proyectos ágil: conceptos básicos.
- ✓ **Priorización de los requisitos por valor y coste. (s.f.).** Recuperado el 28 de octubre de 2009, de [proyectosagiles.org: http://www.proyectosagiles.org/priorizacion-requisitos-valor-coste](http://www.proyectosagiles.org/priorizacion-requisitos-valor-coste)
- ✓ **PRIORIZACION DE REQUERIMIENTOS. (s.f.).** Recuperado el 26 de octubre de 2009, de <http://sagesserd.com/Download.html>
- ✓ **Robertson, S., & Robertson, J.** (17 de marzo de 2006). Mastering the Requirements Process, Second Edition. Recuperado el 24 de enero de 2010, de Safari. Book.
- ✓ **Rodríguez, J., Mellado, D., Fernández-Medina, E., & Piattini, M. (s.f.).** Gestión Automatizada de Requisitos de Seguridad para. Ciudad Real, Madrid, España.
- ✓ **Rodríguez, Yanisleis Pérez Díaz y Grenny Negret.** 2009. Propuesta de guía de desarrollo de software con ajuste a Sistemas de Realidad Virtual. 2009.
- ✓ **Rehbein, L. G. (s.f.).** Ingeniería de software: Gestión de Proyectos de Software.
- ✓ **SCALONE, F.** (junio de 2006). ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE. Buenos Aires.
- ✓ **Stoel, Diederick.** 2007. Guía para asesores en formación: Calidad del e-learning y evaluación del ROI. Noviembre de 2007.
- ✓ **Teruel, Karina Pérez.** Proceso, Técnicas y Herramientas para la Ingeniería de requisitos.
- ✓ **ZAMBRANO, A. N.** (junio de 2005). HERRAMIENTA PARA EL ANÁLISIS DE REQUERIMIENTOS DENTRO DE LA PEQUEÑA EMPRESA DESARROLLADORA DE SOFTWARE EN BOGOTÁ. Bogotá D.C.

BIBLIOGRAFÍA

- ✓ **5139_Guía para evaluar impacto de los requisitos del cliente. (s.f.).** Programa de Mejora. La Habana, Cuba.

ANEXOS

GLOSARIO DE TÉRMINOS

G

Grafo: Un grafo es un conjunto de puntos interconectados por un conjunto de líneas. Estos elementos reciben la denominación de puntos (“requisitos”) y aristas (“conexión entre requisitos”) respectivamente.

I

Indicador: Los indicadores son puntos de referencia, que brindan información cuantitativa o cualitativa, conformada por uno o varios datos, constituidos por percepciones, números, hechos, opiniones o medidas, que permiten seguir el desenvolvimiento de un proceso y su evaluación.

Ingeniería de requisitos: La ingeniería de requisitos es la ciencia y disciplina concerniente al análisis y documentación de los requisitos de software.

M

Media ponderada: La media ponderada de un conjunto de números X_1, X_2, \dots, X_n , con las ponderaciones correspondientes w_1, w_2, \dots, w_n , se calcula con la fórmula:

$$\bar{X} = \frac{w_1 X_1 + w_2 X_2 + \dots + w_k X_k}{w_1 + w_2 + \dots + w_k} = \frac{\sum wX}{\sum w}$$

Método: Un método es un conjunto de pasos necesarios a seguir para obtener conocimientos científicos mediante instrumentos confiables.

Metodología de desarrollo: Una metodología de desarrollo es una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo de software.

Métrica: Una métrica es el término que describe muchos y muy variados casos de medición. Es una medida estadística.

Modelo: Un modelo es como un ente que representa de forma precisa algo que será realizado o que ya existe.

ANEXOS

N

Norma: Una norma es una regla que debe ser respetada y que permite ajustar ciertas conductas o actividades.

P

Priorización de requisitos: Consiste en organizar como quedarán los requisitos para su posterior implementación.

Promedio: El promedio de un conjunto de números se calcula sumándolos y luego dividiendo la suma por el número de sumandos.

R

Requisitos: Los requisitos describen las condiciones que deben cumplir o las capacidades que deben tener los productos entregables del proyecto para satisfacer un contrato, norma, especificación o cualquier otro documento formalmente impuesto.

T

Teoría: Una teoría es un conjunto ordenado de conocimientos especulativos, sin aplicación práctica, sobre determinados fenómenos y su comportamiento, que derivan en la enunciación de leyes de validez universal.
