

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/329430043>

ADMINISTRACIÓN DE REQUERIMIENTOS

Chapter · February 2018

DOI: 10.2307/j.ctv86dfv4.7

CITATIONS

0

READS

1,352

4 authors, including:



Miguel Eduardo Torres Moreno
National University of Colombia

45 PUBLICATIONS 60 CITATIONS

[SEE PROFILE](#)



María Patricia Amortegui Vargas
Universidad Nacional Abierta y a Distancia

11 PUBLICATIONS 7 CITATIONS

[SEE PROFILE](#)

CAPÍTULO 4

ADMINISTRACIÓN DE REQUERIMIENTOS

Laura Catalina Zorro Jiménez
Vanessa Carolina Loaiza Carvajal
Miguel Eduardo Torres Moreno
María Patricia Amórtegui Vargas

4.1. ¿Qué es la administración de requerimientos?

Como se muestra en la figura 4.1, la administración de requerimientos es una de las actividades de la ingeniería de requerimientos cuyo objetivo principal es administrar los cambios que pueden surgir en los requerimientos durante su implementación o luego de ella y en todos los lanzamientos del producto, ya que, en ciertos casos, cuando se hace el lanzamiento de alguna versión del sistema, puede que algunos requerimientos ya no sean necesarios o hayan cambiado por alguna disposición del cliente u otra razón [1].

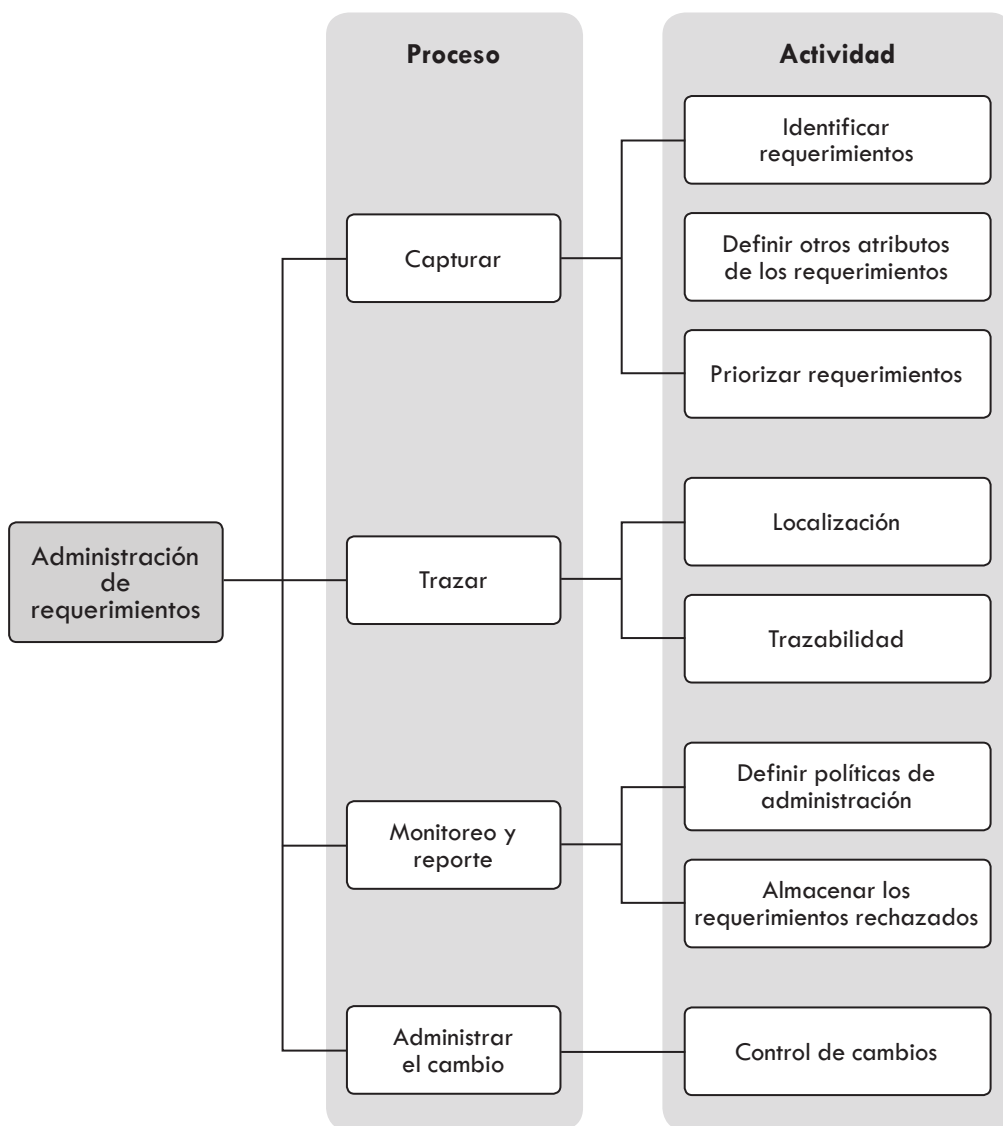
Además del control de cambios, la administración de requerimientos debe encargarse del control de versiones, hacer seguimiento del estado de los requerimientos y realizar la trazabilidad que permita ubicar el origen y lugar donde se encuentra el requerimiento en cualquier artefacto producto del proceso de desarrollo. Estas responsabilidades del proceso permiten asegurar la precisión, la integridad y que los requerimientos se encuentren siempre actualizados [2].

Una vez realizadas la recolección, el análisis y la especificación de los requerimientos, es necesario empezar a realizar las tareas de administración de requerimientos para, así, aumentar las posibilidades de éxito, ya que dentro de las causas por las cuales fracasan los proyectos de software, según el estudio CHAOS [3], realizado por el Standish Group, se encuentran, entre otras, la incapacidad de manejar el cambio, la inexactitud e incomprensión de los requerimientos y la planeación inapropiada, las cuales están estrechamente relacionadas con la administración de los requerimientos.

4.2. Proceso de administración de requerimientos

Para alcanzar una gestión de requerimientos exitosa se deben realizar las actividades registradas en la figura 4.1.

FIGURA 4.1. PROCESO DE ADMINISTRACIÓN DE REQUERIMIENTOS



Fuente: adaptado de [2], [4].

4.2.1. Capturar

4.2.1.1. Definir los atributos de requerimientos

“Los requerimientos son objetos de ingeniería y deben ser organizados y rastreados como tal” [5], es decir, los requerimientos son parte fundamental del proyecto y son herramientas poderosas, las cuales permiten definir cuál debe ser el comportamiento del sistema. Por lo tanto, además de la descripción textual de los requerimientos, cada uno de estos debería contar con ciertos atributos que permiten la contextualización del requerimiento dentro del sistema, pues los requerimientos, al igual que otros artefactos, cambian su estado y es necesario saber qué cambia, cómo cambia, por qué cambia, quién lo cambia, entre otras cuestiones.

Los atributos de los requerimientos, como bien lo dicen Alexander y Stevens [5], sirven para rastrear el estado de los requerimientos en cualquier etapa del proyecto; sin embargo, no todos los requerimientos tienen la misma necesidad de ser monitoreados todo el tiempo, por su simplicidad, bajo riesgo respecto al cambio o baja prioridad; pero existen atributos que aplican a todos los requerimientos en general. Por lo tanto, debe de haber un conjunto de atributos para utilizar de manera general en los proyectos y para realizar un proceso de administración que cumpla con las características mínimas [6].

Con este propósito, se presenta una clasificación de los atributos de los requerimientos (tabla 4.1). En primera instancia, están los atributos que describen la información del requerimiento. En segunda, se hallan los atributos que se desarrollan durante el proyecto, ya sea porque dependen de otros requerimientos para obtener su valor o porque existen otros artefactos que pertenecen a otras fases que influyen en este (por ejemplo, localización o trazabilidad). Por último, se encuentran los atributos opcionales, que pueden albergar una gama amplia dependiendo de lo que el proyecto requiera [6].

TABLA 4.1. DISTRIBUCIÓN DE LOS ATRIBUTOS DE LOS REQUERIMIENTOS

Atributo	Descripción
Atributos propios del requerimiento	
Fecha	Día, mes y año en el cual fue creado el requerimiento.
Descripción	Describe el requerimiento identificado en la fase de recolección.
Versión	Versión actual del requerimiento.
Autor	Nombre de la persona que creó el requerimiento.
Responsable	Persona que debe asegurar que el requerimiento se cumple.
Dueño	Persona a la que pertenece el requerimiento o una lista de <i>stakeholders</i> .
Estado	Fase en que se encuentra el requerimiento, es decir, el requerimiento puede estar propuesto, rechazado, implementado o verificado, dependiendo de los criterios que cada grupo tenga.
Versión línea base	Número de la versión del requerimiento, el cual pertenece a la línea base actual.
Origen	Describe las fuentes del requerimiento (véase sección “2.1. Recolección”, punto d).
Razón	Se utiliza para saber el motivo de la última revisión.
Cambios realizados	Lista de los cambios realizados al requerimiento.
Responsable del cambio	Indica el responsable de los cambios realizados en los requerimientos.
Fecha del último cambio	Fecha en la cual fue modificado por última vez el requerimiento.
Atributos para el proyecto	
Prioridad	Importancia del requerimiento.
Requerimientos asociados	Requerimientos que se relacionan, ya sea porque pertenecen al mismo módulo.
Subsistemas	Lista de subsistemas en los que se encuentra el requerimiento.
Casos de uso asociados	Casos de uso que cumplen con el requerimiento. Este atributo está relacionado con los que tratan sobre la trazabilidad vertical.
Tipo de requerimiento	Según la clasificación que se maneje en el grupo, a qué grupo de requerimientos pertenece.

Atributo	Descripción
Volatilidad	Este campo indica qué tan volátil es el requerimiento (véase sección “3.2.1.1.1. Volatilidad en los requerimientos”).
Localización	Especifica el componente o subsistema donde se encuentra asignado el requerimiento.
Riesgo	Indica el nivel de riesgo que puede tener un requerimiento, debido a diferentes factores como la limitación de la herramienta de software que se utilice, la falta de conocimiento de la herramienta, la dificultad de la interfaz, el conjunto de habilidades en el equipo de desarrollo y la criticidad con el incumplimiento de una obligación se propagará el riesgo.
Dificultad	Indica la dificultad para los desarrolladores para implementar el requerimiento.
Beneficio	Indica el nivel de beneficio para el proyecto en general, puede ser distribuido según diferentes criterios. Para mayor ampliación del tema véase la sección “3.2.1.2. Priorización”.
Trazabilidad	Muestra los artefactos que permiten seguir la vida del requerimiento (véase sección “3.2.2.2. Trazabilidad”).
Atributos adicionales	
Casos de prueba asignados	Indica el caso de prueba asignado para validar el requerimiento en cada una de las fases de desarrollo.
Responsables en cada una de las fases	No siempre el responsable en la fase de requerimientos es el mismo en la fase de diseño, implementación o pruebas; por lo tanto, es necesario saber quién está a cargo del requerimiento en cada una de las fases.
Método de calificación	El método que se utilizará en pruebas de aceptación para demostrar la conformidad del producto con el requisito.
Métricas asociadas	Resultados obtenidos de las métricas asociadas al método de calificación, si lo van a aplicar.
Aprobación (según estándar que se maneje)	Porcentaje del resultado que indica si cumple con los estándares de calidad definidos en el proyecto, según las listas de chequeo aplicadas.

Fuente: [2], [6], [7].

Sin embargo, como menciona Heumann en su artículo “The Five Levels of Requirement Management Maturity” [7], no todos los requerimientos son iguales; existen requerimientos más estables que otros, existen requerimientos que son más importantes que otros, razón por la cual se les presta mayor atención. Por lo tanto, no es necesario aplicar dentro de la especificación de los requerimientos (véase sección “3.1. Especificación de requerimientos”) todos los atributos descritos en la tabla 4.2, simplemente se pueden tomar algunos como base para modificar y agregar nuevos atributos que sean necesarios, a fin de que se pueda considerar que un requerimiento es completo y apropiado para el proyecto.

Los requerimientos se pueden considerar volátiles por dos razones: la primera es que puede que uno o más clientes no estén de acuerdo con la funcionalidad o característica que describe el requerimiento y la segunda razón es que existen ciertos tipos de eventos externos al proyecto que no pueden ser controlados y pueden generar algunos cambios sobre los requerimientos [8]. Un ejemplo clave de requerimientos volátiles son las interfaces externas.

Los requerimientos volátiles pueden ser de diferentes tipos, por ejemplo:

- Requerimientos cambiantes: son los que cambian debido al ambiente donde se encuentra o se encontrará el sistema.
- Requerimientos emergentes: estos surgen cuando el sistema ya está en fases de diseño y de implementación, ya que no es común verlos cuando el sistema se especifica.
- Requerimientos “consecuentes”: surgen de las especulaciones hechas en el momento de realizar la especificación, sobre la manera en la que va a ser usado por los clientes del sistema.
- Requerimientos de compatibilidad: son requerimientos sobre la compatibilidad que debe tener el sistema en caso de que surjan nuevas actualizaciones sobre el ambiente en el que va a funcionar el sistema.

Se debe considerar realizar el análisis de la volatilidad del requerimiento dentro del proceso, ya que uno de los beneficios es la facilidad para los desarrolladores en el momento hacer un cambio en los componentes del sistema que no se encuentran fuertemente acoplados. Por esto, teniendo la lista de requerimientos volátiles, estos se pueden ubicar en los componentes más desacoplados, para así facilitar la modificación del sistema.

4.2.1.2. Priorización

La priorización es un proceso necesario dentro de la ingeniería de requerimientos, ya que un proyecto depende de diferentes recursos como el tiempo, el personal y el dinero. Por esto, es preciso entregar un producto que contenga lo esencial (funcionalidades importantes) [9].

Para saber lo “esencial”, es preciso organizar los requerimientos de tal forma que se tenga un conjunto de requerimientos indispensables dentro del producto [10]. Según Berander y Andrews, el proceso de priorización de requerimientos apoya diferentes actividades como [10]:

- Negociar el alcance del proyecto, pues a veces se generan conflictos con algunas restricciones, como programa, presupuesto, recursos, tiempo en el mercado y calidad.
- Que los *stakeholders* puedan decidir cuáles son los requerimientos básicos del sistema.
- Hacer frente a exigencias contradictorias, es decir, para enfocarse en el proceso de negociación y resolver los desacuerdos entre los *stakeholders*.
- Equilibrar las repercusiones de los requerimientos en la arquitectura de software y la evolución futura del producto y sus costos asociados.
- Seleccionar solo un subconjunto de los requerimientos y aún producir un sistema que satisfacer al cliente(s).
- Estimar las expectativas del cliente.
- Conseguir una ventaja técnica y optimizar las oportunidades de mercado.
- Minimizar la repetición de trabajos y atrasos en el calendario (estabilidad del plan).
- Establecer la importancia relativa de cada requerimiento con el objetivo proporcionar el mayor valor al menor costo.

4.2.1.2.1. Criterios

Existen diferentes factores que pueden intervenir en la priorización, ya que se debe tomar en cuenta la opinión de los *stakeholders* del proyecto; por lo tanto, los factores que existen son muchos dependiendo de los objetivos y prioridades de cada *stakeholder*. Para esto, diferentes autores como Berander y Andrews [10] o Botta y Bahill [11] han agrupado diversos criterios útiles para establecer prioridades que pueden tomarse en cuenta para realizar el proceso de priorización de requerimientos. En la tabla 4.2 se presenta un conjunto de los criterios que pueden ser usados.

TABLA 4.2. CRITERIOS DE PRIORIZACIÓN DE REQUERIMIENTOS

	Criterio	Justificación
Importancia Los <i>stakeholders</i> deben decidir qué requerimiento es más importante para el sistema. Dado que cada uno va a seguir su propio criterio, este ítem puede llegar a ser multifacético. Por lo tanto, es necesario especificar el área de importancia que se quiere manejar para asegurar, por lo menos, que se está calificando con las mismas reglas.	Satisfacción de cliente	“Características que pueden incrementar la satisfacción del cliente deben ser de prioridad alta”, pues el producto es desarrollado a petición del cliente. Su opinión es una de las más importantes en esta etapa.
	Compromiso	Si se ha llegado a tener un acuerdo de lo que TIENE o DEBERÍA hacer el sistema, esto se debe cumplir, dado que ha existido un previo trato con el cliente.
	Prioridad de escenarios	Es importante dar prioridad a las características que participan en los escenarios (casos de uso), ya que pueden ser importantes para los objetivos del negocio.
	Beneficio	Este beneficio se refiere a características relacionadas con requerimientos no funcionales del sistema, como desempeño, tiempo entre fallas, entre otros.
	Seguridad (<i>safety</i>)	Las características críticas de seguridad deben ser de alta prioridad, ya que se está involucrando la seguridad de los seres humanos.
	Cuando es necesario	En algunas ocasiones, existen características que deben ser planeadas de manera temprana, debido al deseo o a restricciones por parte del cliente; por lo tanto, el tiempo es algo que también se debe tomar en cuenta.
Impacto negativo (penalización) Es el valor que tiene el requerimiento para el <i>stakeholder</i> , pero se aplica al contrario, es decir, qué tanto se perjudica el sistema si se realiza o no un requerimiento. Al igual que el anterior grupo de criterios, se necesita especificar el área que se va a evaluar.	Arquitectura	Dar alta prioridad a las características que tendrán un gran impacto en la arquitectura del sistema. Esto se debe a que los cambios que se realicen sobre ellas posteriormente pueden afectar diferentes entidades y causar así más cambios en diferentes artefactos, lo cual llega a ser costoso para el proyecto.
	Dependencias	Este tipo de relaciones entre los requerimientos pueden indicar el grado de impacto que puede causar si existe cambio sobre algún requerimiento.
	Frecuencia de uso	Objetos que pueden ser usados muchas veces durante el desarrollo del proyecto deben ser de alta prioridad. Por ejemplo, un fragmento de código que realiza una funcionalidad específica repetidas veces, debe ser elaborado y optimizado.

	Criterio	Justificación
Costo Como su nombre lo indica, es el esfuerzo en el que incurre el <i>stakeholder</i> para llevar a cabo el requerimiento. Según el <i>stakeholder</i> , el recurso a tener en cuenta difiere.	Costo	El costo es el valor numérico que se le asigna a cada requerimiento, cuantificando los recursos invertidos (tiempo, personas, viajes). Depende de los <i>stakeholders</i> analizar su costo y si el retorno de la inversión es satisfactorio para el cliente, el cual se puede obtener revisando el beneficio del requerimiento. Si el beneficio que trae desarrollar el requerimiento es alto para el cliente, entonces se puede considerar incluirlo.
	Proporción costo-beneficio	Como su nombre lo indica es la combinación del costo-beneficio, obteniendo así características que ofrecen mayor beneficio con menor costo.
	Complejidad	Las características que son complejas de desarrollar deben ser de alta prioridad, pues se debe asignar a los mejores miembros para llevarlas a cabo.
	Potencial de reúso	Si es un ítem altamente reusable, debe tener una prioridad alta, pues puede influir en el tiempo de desarrollo.
	Dificultad de implementación	Los requerimientos que cuentan con un grado de dificultad alto en el momento de poner en práctica deberían tener alta prioridad; por ejemplo, los atributos que pueden hacer difícil la aplicación incluyen gran tamaño, la incertidumbre, la novedad, el número de personas involucradas y las restricciones organizacionales. La complejidad y la dificultad de implementación son diferentes, pues la segunda se refiere más a la dificultad en aplicar la solución, mientras que la primera se refiere al costo que puede incurrir para el sistema.
	Tiempo de implementación	Como su nombre lo indica, es el tiempo que puede necesitar un requerimiento para ser aceptado; por lo tanto, los que necesiten más tiempo deben ser planeados con anterioridad y tener prioridad, para no crear posibles caminos críticos en el proyecto.

	Criterio	Justificación
Riesgo Es la probabilidad de fallo que tiene el requerimiento de no ser implementado o aplicado dentro del sistema.	Riesgo	“Trabajar sobre los requerimientos de alto riesgo en primer lugar para reducir el riesgo y, además porque son más propensos a cambiar, produciendo así cambios en otras características”. Con esto se evita el gasto de dinero en la aparición de cambios no planeados.
	Volatilidad	“La volatilidad de los requerimientos se considera un factor de riesgo y en ocasiones se manejan como parte del aspecto de riesgo. Otros piensan que la volatilidad se debe analizar por separado”. Existen diferentes razones, por las cuales la volatilidad puede variar como cambios en el mercado, en los requerimientos de negocio, cambios legislativos, etc. Sin embargo, independiente de la razón, estos cambios causan atrasos en el cronograma lo que lleva al gasto de dinero adicional en el proyecto. Por lo tanto, hay que definir estos requerimientos con alta prioridad y así evitar problemas en el desarrollo del proyecto. Véase sección “3.2.1.1.1. Volatilidad en los requerimientos”.
	Estabilidad	“Desarrollar requerimientos estables en primer lugar” ayuda a mantener una base casi intacta sobre la que se puede trabajar el resto de los requerimientos. Además, permite visualizar los cambios de otros requerimientos con antelación [52] y evitar así el gasto de dinero en cambios que pudieron ser evitados.

Fuente: [10], [11].

4.2.1.2.2. Proceso

Existe un proceso para priorizar requerimientos que recomienda el autor Firesmith para integrarlo a la administración de requerimientos [12]:

- Convencer a los *stakeholders*: es necesario dar a conocer la importancia de la priorización de requerimientos a los *stakeholders*; de esta forma se pueden obtener mejores resultados.
- Entrenar a los *stakeholders*: el entrenamiento a los *stakeholders* de requerimientos es necesario, ya que así se asegura que todos los involucrados en el proceso de

priorización de requerimientos están tomando el mismo concepto y criterio a la hora de evaluar.

- Escoger técnica: se debe escoger una técnica de priorización para saber los criterios a tener en cuenta para evaluar la importancia de cada uno de los requerimientos. La selección de la técnica depende de la organización, del cliente, de la complejidad del proyecto y de sus *stakeholders* [10].
- Priorizar los requerimientos actuales: “Durante el análisis de requerimientos, el equipo debe trabajar con el representante de los stakeholders para priorizar los requerimientos actuales” [12]. Dentro de esta fase se incluyen actividades como la negociación con los *stakeholders* sobre los requerimientos que se van a desarrollar y su validación.
- Publicar las prioridades: durante la especificación de requerimientos, el equipo debe publicar las prioridades resultantes después del análisis, con el fin de que todas las partes interesadas estén informadas y puedan utilizar este informe de prioridades.
- Estimación el esfuerzo: con el equipo de arquitectura y el de desarrollo se registran estimaciones realistas sobre los esfuerzos a incurrir en los requerimientos seleccionados.
- Desarrollar un cronograma: con los resultados obtenidos en la fase anterior se desarrolla un cronograma para las etapas posteriores del desarrollo del producto. Este cronograma no es permanente, ya que puede cambiar dependiendo de las posibles modificaciones de los requerimientos en el futuro.
- Mantenga las prioridades: “Durante la administración de requerimientos, el equipo debe trabajar con los stakeholders para mantener los parámetros de exigencia a medida que cambian” [12]. Generalmente, este campo suele ser como un atributo del requerimiento al final de todo el análisis.

4.2.1.2.3. Técnicas

En el momento de priorizar los requerimientos, los *stakeholders* asignan un valor de importancia al requerimiento según el criterio; sin embargo, en proyectos donde la complejidad es alta, el asignar un valor no soluciona el problema, pues el resultado no logrará satisfacer los objetivos propuestos en principio, es decir, el conjunto de requerimientos estará vagamente delimitado y no se tendrá la certeza de que esos requerimientos merecen estar dentro de ese conjunto. Además, el hecho de que influyan diferentes *stakeholders* lo hace aún más complicado, ya que podría encontrarse con resultados contradictorios, donde no se tiene un criterio de decisión más que escoger los de mayor puntuación. Para evitar este tipo de problemas se han

presentado diferentes técnicas de priorización de requerimientos desde hace varios años. A continuación, se presentarán las más comunes.

Para empezar, se debe mencionar que existen diferentes escalas para evaluar un requerimiento, ya que dependiendo de la granularidad que se esté manejando, esta puede variar [9]. Las más comunes son:

- Bajo, medio y alto
- Esencial, condicional y opcional
- Bueno tenerlo, objetivo, altamente deseada y se debe lograr
- Numérica (por ejemplo, 1-5 o 1-10)

Dependiendo de la técnica, se escoge la que mejor se adecue, pues según la escala se puede elegir qué tan detallado resulta el análisis.

4.2.1.2.4. Asignación numérica

La asignación numérica es una de las técnicas más sencillas y comunes [10]. Consiste en agrupar los requerimientos en diferentes niveles (véase sección “4.2.1.2.3. Técnicas”), donde cada uno de los *stakeholders* tiene que agruparlos según su criterio. Para evitar inconvenientes, como el considerar todos los requerimientos importantes, se debe limitar el número de requerimientos por grupo. De esta forma se tiene un conjunto de requerimientos semiordenados.

4.2.1.2.5. Priorización basada en valor, costo y riesgo

Esta técnica la propone Wiegers en su artículo “First Things First: Prioritizing Requirements” [9], el cual menciona que se deben tener en cuenta las diferentes perspectivas de los *stakeholders* para estimar las prioridades relativas de un conjunto de requerimientos. Esta consiste en evaluar cuatro criterios:

- Beneficio
- Impacto negativo (*penalty*)
- Costo
- Riesgo

Los *stakeholders* asociados a esta técnica suelen ser el director del proyecto, para mediar entre las partes implicadas; los representantes del área de desarrollo, para estimar los riesgos y costos que se van a afrontar, y los representantes de los clientes “clave”, para que indiquen qué beneficios e impactos negativos podrían tener si el requerimiento está o no dentro del producto.

Luego de evaluar cada requerimiento se deben asignar unos pesos a cada criterio, es decir, un valor de importancia a cada ítem; por ejemplo, si se considera que el proyecto es de alto riesgo (variedad de caminos críticos, presupuesto o tiempo), se le proporciona más valor al criterio *riesgo*. Por último, se les aplica una fórmula a los valores asignados sobre los requerimientos respecto a cada criterio [2]:

$$\text{Prioridad} = \frac{\% \text{valor}}{(\% \text{costo} * \text{peso del costo} + \% \text{riesgo} * \text{peso del riesgo})}$$

Así se logra un valor específico de cada requerimiento, se organiza la tabla de mayor a menor y se obtiene un conjunto priorizado de requerimientos.

4.2.1.2.6. Analytical Hierarchy Process

Es un método para la toma de decisiones adaptado a la priorización de requerimientos de software [10], [13]. Se utiliza “por pares”, ya que se usa una matriz de comparación para calcular el valor relativo y los costos de los requerimientos entre sí. Mediante el uso del *Analytical Hierarchy Process* (AHP), el analista de requerimientos puede confirmar la coherencia del resultado, pues los errores pueden identificarse durante las comparaciones redundantes que se realizan. El AHP posee diferentes ventajas, como el evitar errores subjetivos, además de aumentar la probabilidad de que los resultados sean confiables. El método de esta técnica se distribuye en cuatro puntos generales [13]:

1. Revisar los requerimientos asociados al requerimiento candidato.
2. Aplicar el método de comparación por pares para evaluar el valor relativo de cada uno de los requerimientos candidatos.
3. Aplicar el método de comparación por pares para evaluar el costo relativo de los requerimientos candidatos.
4. Calcular el valor relativo de cada requerimiento candidato y costo de implementación, para luego realizar un diagrama de costo-valor.

Utilice el diagrama de costo-valor como un mapa para analizar los requerimientos candidatos.

4.2.1.2.7. Valor acumulativo: la prueba de 100 dólares

La prueba de 100 dólares es una técnica muy sencilla de priorización en la cual a los interesados se les da 100 unidades imaginarias para distribuir entre los requerimientos [13]. El resultado de la priorización se presenta en una escala de razón [10].

Un problema con esta técnica se presenta cuando hay demasiados requerimientos; sin embargo, un estudio al respecto mostró resultados positivos en cuanto al uso de otros valores, aparte de 100 (por ejemplo, 1000, 10.000 o 1.000.000) [10].

4.2.1.2.8. Árbol de búsqueda binario

El árbol binario de búsqueda es un algoritmo que suele utilizarse en la búsqueda de información y puede emplearse en la priorización de los requerimientos [13]. La clasificación que usa esta técnica se basa en una escala ordinal, donde el requerimiento más importante ocupa la posición número 1, y el menos importante, la posición número n (de n requerimientos) [10]. El proceso que se utiliza es:

1. Ubicar los requerimientos en una lista (no importa el orden).
2. Tomar el primer requerimiento y posicionarlo como raíz.
3. Tomar otro requerimiento y realizar la comparación con el nodo raíz.
4. Si el requerimiento es menos importante que el nodo raíz, debe compararlo con el nodo hijo izquierdo. Si el requerimiento es más importante que el nodo raíz, debe compararlo con el nodo hijo derecho. En el caso de que el nodo no tenga nodos secundarios, introducir el nuevo requerimiento como el nuevo nodo secundario hacia la derecha o izquierda, dependiendo de si el requerimiento es más o menos importante.
5. El grupo que realiza el análisis debe repetir los pasos 3 y 4 hasta que todos los requerimientos se hayan comparado e insertado en el árbol de búsqueda binario (BST).

Cuando se requiera ver una presentación más común, se debe recorrer todo el BST en orden. Con esto se obtienen los requerimientos en una lista ordenada de mayor a menor importancia, para llegar así a lo que Berander y Andrews llaman *técnica de priorización ranking* [10].

4.2.1.2.9. Ejemplo de priorización

Para tener una mejor comprensión de este tema, se expone un ejemplo utilizando una de las técnicas descritas en las secciones anteriores. Para este caso se utilizará la técnica de costo-beneficio, propuesta por Wiegers [9].

Se tiene un conjunto de 12 requerimientos, el cual debe priorizarse. Se tienen recursos limitados para la primera entrega del proyecto y el objetivo es cumplir con un alcance definido por el cliente. Para ello se ha propuesto a algunos *stakeholders* que evalúen los requerimientos dependiendo de 4 variables, como las que expone Wiegers: costo, beneficio, riesgo y penalidad. El *costo* es el valor numérico de los recursos invertidos (tiempo, personas, materiales o viajes) que se necesitan para

realizar un requerimiento. El *beneficio*, como su nombre lo indica, es qué tan bueno para el cliente o para el proyecto sería realizar este requerimiento en este momento. La *penalidad* es todo lo contrario: se evalúa qué tan malo sería para el cliente o para el proyecto no realizar este requerimiento. Por último, está el *riesgo*, que indica la probabilidad de fallo o grandes cambios durante el desarrollo del requerimiento, lo cual es considerado un riesgo para el proyecto. En la tabla 4.3 se muestran los datos recolectados y totalizados para cada requerimiento.

TABLA 4.3. EJEMPLO DE DATOS PARA LA PRIORIZACIÓN DEL REQUERIMIENTO

Pesos	40 %	20 %	10 %	30 %	Total
ID REQ	Beneficio	Penalidad	Riesgo	Costo	
R1	6,4	4,8	1,6	6,3	1,7
R2	4,1	7,0	2,8	7,3	1,2
R3	6,8	7,8	3,1	2,8	3,7
R4	6,1	5,8	2,5	6,1	1,7
R5	5,3	6,0	3,6	2,1	3,3
R6	5,9	6,3	2,8	5,5	1,8
R7	3,1	3,6	1,6	3,8	1,5
R8	6,6	7,0	3,3	5,6	2,0
R9	7,0	3,6	2,5	4,1	2,3
R10	7,6	7,8	3,8	3,6	3,1
R11	7,5	7,5	3,8	3,5	3,1
R12	5,8	6,5	3,3	5,8	1,7

Fuente: elaboración propia.

En la tabla 4.3 se muestran los cuatro criterios de evaluación que se tuvieron en cuenta para el proyecto. Para este se acordó utilizar una escala de 1 a 10, donde es posible utilizar cifras decimales.

Como lo indica Wiegers [9], es recomendable poner un peso a cada uno de los criterios, si es necesario en el proyecto. Esto con el fin de priorizar también los criterios en la especificación —por ejemplo, en este proyecto se define, por el grupo de proyecto, que el riesgo que se presenta en cada requerimiento no es significativo;

por lo tanto, el porcentaje es el menor de todos los criterios (10 %)—. Esta opción también puede aplicarse a los *stakeholders*, donde el porcentaje de decisión por persona es diferente y los datos no son promediados. Esto es permitido mientras todo el grupo de proyecto esté de acuerdo [10].

Siguiendo con el ejemplo, se puede observar que el criterio de mayor peso es el *beneficio* (40 %); por lo tanto, se deduce que la apreciación del cliente para esta primera entrega es mayor, así el costo que implique realizarlo no sea significativo. Por ejemplo, el requerimiento R2 tiene un costo alto (implica tiempo de desarrollo, análisis y diseño, personal); sin embargo, para esta entrega no queda dentro de los requerimientos seleccionados para implementar (resaltados en verde), dado que para el cliente el beneficio que alcanza a percibir no es tan alto como el de otros requerimientos, en este tipo de situaciones es donde se valora una priorización a conciencia, porque genera resultados bien argumentados.

También se puede ver el caso contrario: que son los requerimientos con bajo costo, pero con un alto impacto para el cliente, en este caso el requerimiento R3, que tiene el valor de prioridad más alto de los 12 requerimientos evaluados. Si en esta evaluación solo se hubiera tenido en cuenta la opinión del grupo de desarrollo, muy seguramente este requerimiento no se implementaría para la primera entrega, dado su bajo costo para el desarrollador; sin embargo, para el cliente ver esta funcionalidad en ejecución sí es importante, lo cual demuestra que en el proceso de priorización es recomendable tener *stakeholders* con diferentes perspectivas (en este caso, el cliente).

A partir del ejemplo anterior, se concluye que la priorización es importante dentro del desarrollo del proyecto, pues permite planear de mejor manera el desarrollo de requerimientos y obtener una mejor apreciación del cliente, que al final de todo el proceso es uno de los objetivos.

4.2.2. Trazar

4.2.2.1. Localización

La localización es un procedimiento que consiste en asignar requerimientos de alto nivel (véase sección “1.5.3. Requerimientos de sistema”) a los subsistemas específicos. Es necesario que durante el desarrollo se incluyan tanto componentes de hardware y software como productos complejos que contienen múltiples subsistemas de software [2]. La asignación se realiza después de que los requerimientos de sistema se especifican y la arquitectura del sistema se ha definido.

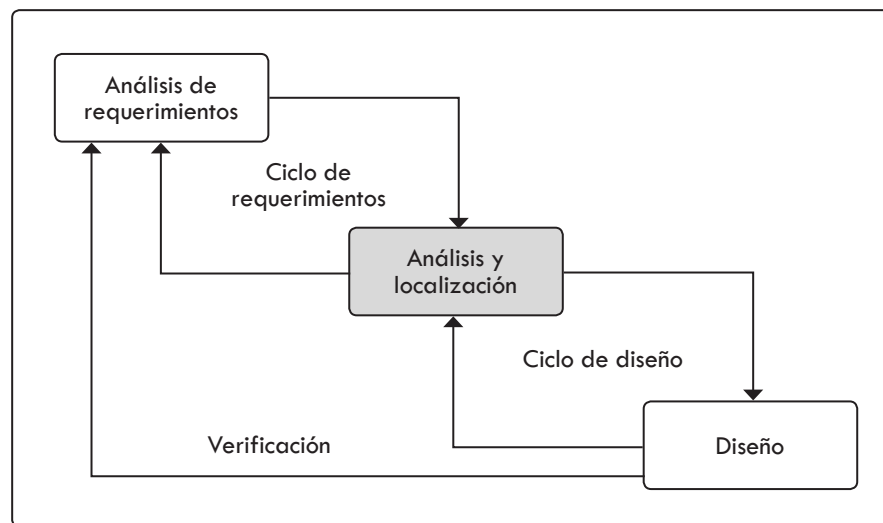
La localización es necesaria para asegurar que todos los requerimientos del sistema son cumplidos por un subsistema o componente o por un conjunto de ellos colaborando juntos, pues no es seguro que un requerimiento se cumpla con solo un subsistema o componente. Daneva y Herrmann [14] presentan una forma de distribuir los diferentes grupos donde pueden ser asignados los requerimientos:

- Subsistemas: grupo de funcionalidades lógicas.
- Componentes del sistema: hardware, software, entrenamiento y documentación.

La fase de diseño de la arquitectura es especialmente crítica para los sistemas que incluyen componentes de software y hardware [2]. Una etapa importante para el análisis es asignar requerimientos de sistema a los distintos subsistemas y componentes. Los requerimientos de sistema se deben descomponer en requerimientos funcionales y no funcionales para los diferentes subsistemas y componentes [9]. Adicional a lo anterior, la información sobre la trazabilidad permite al equipo de desarrollo abordar cada requerimiento en el diseño.

El proceso de localización ayuda a retroalimentar el proceso de análisis de requerimientos, de igual forma que el proceso de localización es retroalimentado por el proceso de diseño [15], tal y como se muestra en la figura 4.2.

FIGURA 4.2. LOCALIZACIÓN DENTRO DEL PROCESO DE INGENIERÍA



Fuente: tomado de [16].

“La localización es importante para permitir el análisis detallado de las necesidades” [15], ya que una vez un conjunto de requerimientos es asignado a un componente, las necesidades individuales pueden surgir y ser analizadas para descubrir más necesidades que pueden ser cubiertas por el mismo u otros componentes. Como se muestra en la figura 4.2, en grandes proyectos la asignación estimula una nueva ronda de análisis de cada subsistema [15].

Para llevar a cabo este proceso, Wiegers y Beatty [2] recomiendan empezar siempre desde la parte general del sistema, ya que esto permite el surgimiento de nuevos requerimientos durante el proceso [1], [15], es decir, desde los *requerimientos de negocio* hasta los *requerimientos del sistema*, los cuales se encargan de describir la arquitectura y la funcionalidad del sistema distribuida en diferentes componentes.

Como se muestra en la figura 4.2, para obtener un buen diseño, se debe iterar con ayuda del análisis de requerimientos, porque una de las ventajas es que se pueden encontrar puntos de ambigüedad y confusión [2]. Si se encuentran muchos de esos problemas, los requerimientos no eran lo suficientemente claros o no estaban detallados de modo adecuado.

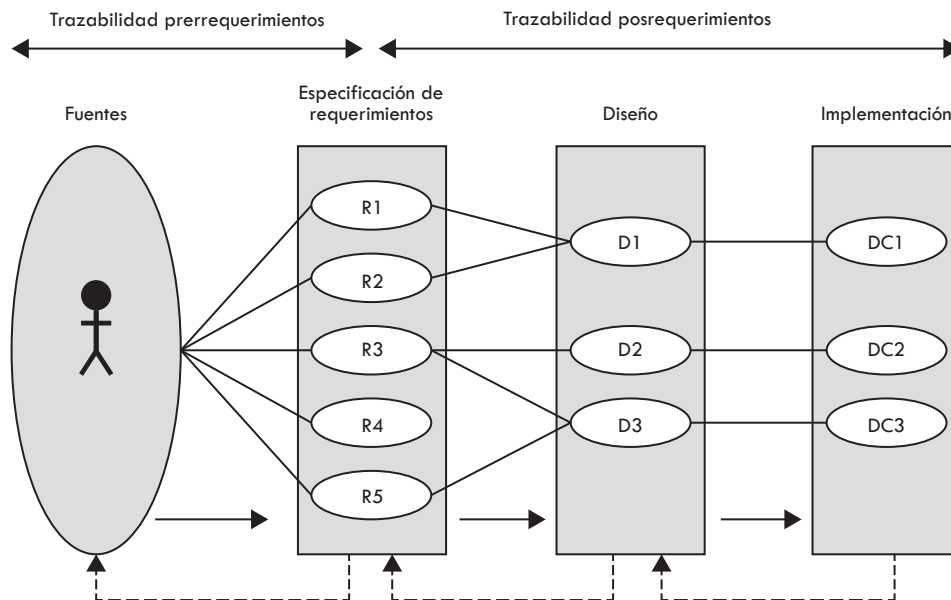
Adicional a lo mencionado, otra de las ventajas de realizar el proceso de localización es asegurar que el proceso de verificación (véase sección “3.2. [Verificación](#)”) pueda realizarse de la mejor manera, a fin de asegurar que los requerimientos, en su mayoría, han sido comprendidos, aceptados por el cliente y se encuentran dentro del sistema.

Por último, la localización también está relacionada con el proceso de pruebas, pues como mencionan Wiegers y Beatty [2], los requerimientos proveen la base para el análisis de pruebas del sistema, pues estas deben ser verificadas contra lo que se especifica en el *Software Requirements Specification* (SRS); por lo tanto, los requerimientos deben ser trazados y localizados (véase sección “4.2.2.2. Trazabilidad”). Esto se debe a que sería posible que el producto muestre correctamente todas las conductas descritas en los casos de prueba basados en el código, pero eso no quiere decir que implemente correctamente las exigencias funcionales [2].

4.2.2.2. Trazabilidad

La trazabilidad de requerimientos es la habilidad de definir, capturar y seguir [17] la vida del requerimiento hacia atrás y hacia adelante [18], es decir, es el grado como se puede establecer una relación entre el requerimiento y su origen y el requerimiento y los productos que se generan al implementarse [19]. Esto hace referencia al código, a las pruebas y a la documentación [16]. Existen dos tipos de trazabilidad:

- Trazabilidad prerrequerimientos: se refiere a la habilidad de describir y seguir la vida del requerimiento antes de que este sea incluido en la especificación de requerimientos [16], es decir, seguir el rastro del requerimiento desde que este es recolectado (véase sección “2.1. Recolección”) hasta el proceso de análisis de requerimientos (véase sección “2.2. Análisis”).
- Trazabilidad posrequerimientos: hace referencia a la habilidad de describir y seguir la vida del requerimiento después de que este ha sido incluido en la especificación de requerimientos [16]. Es decir, se debe seguir el rastro del requerimiento durante la especificación (véase sección “3.1. Especificación de requerimientos”) hasta los elementos de la implementación, por ejemplo, un método que haya sido incluido dentro del código para satisfacer el requerimiento. La figura 4.3 muestra a qué se refiere esta clasificación de la trazabilidad de los requerimientos.

FIGURA 4.3. TRAZABILIDAD PRERREQUERIMIENTOS Y TRAZABILIDAD POSREQUERIMIENTOS

Fuente: adaptado de [16].

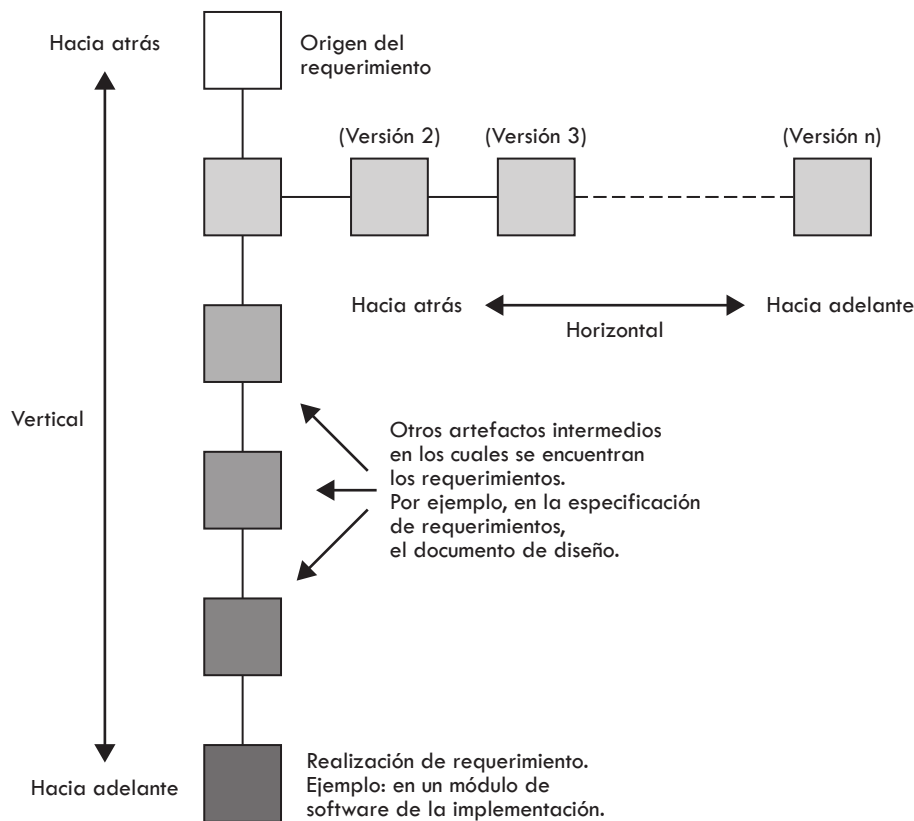
Además de los dos tipos anteriores, también existen los siguientes:

- Trazabilidad hacia atrás: puede ser vertical u horizontal. Si es vertical, se refiere a la habilidad de describir las relaciones del requerimiento desde las fuentes (véase sección “2.1. Recolección”) hasta su especificación [18]. Si es horizontal,

se refiere a la relación existente entre las versiones que se desarrollan a lo largo de la vida del requerimiento [17].

- Trazabilidad hacia adelante: al igual que la trazabilidad hacia atrás, esta puede ser vertical u horizontal. Si es vertical, describe la relación entre los requerimientos hasta las fases posteriores del ciclo de vida del desarrollo del sistema, es decir, los requerimientos se pueden ubicar dentro de la implementación del sistema, en documentos como manuales de usuario, en secciones del código. Si es horizontal, esta trazabilidad hace referencia a las versiones posteriores del requerimiento. En la figura 4.4 se muestra la diferencia entre estos cuatro tipos de trazabilidad, para aclarar posibles confusiones.

FIGURA 4.4. TIPOS DE TRAZABILIDAD HACIA ADELANTE Y HACIA ATRÁS



Fuente: tomado y adaptado de [16].

4.2.2.2.1. Beneficios de realizar la trazabilidad

Dentro de los beneficios identificados por Wiegers y Beatty [2] se encuentran:

- Utilizar la trazabilidad para verificar, validar (véase sección “3.2. Verificación”) y demostrar que se han implementado todos los requerimientos.
- Realizar cambios de manera correcta durante el mantenimiento, ya que teniendo una matriz de trazabilidad (tabla 4.4), esta permite saber cuáles son los cambios que se deben llevar a cabo en caso de que, por ejemplo, cambie un requerimiento de negocio.
- Afirmar con seguridad cuál es el estado de implementación del sistema y cuáles son los requerimientos que no se han implementado.
- Facilitar la reutilización de componentes del sistema, ya que la trazabilidad identifica los paquetes u otras unidades de código que están relacionadas tanto con los requerimientos como con el diseño, las pruebas y otros paquetes.

Cuando se presentan un resultado inesperado después de ejecutar alguna de las pruebas, se puede ir a la matriz de trazabilidad (tabla 4.4) para identificar cuáles son los componentes o unidades de código que deben revisarse.

TABLA 4.4. TABLA DE TRAZABILIDAD

Fuentes del requerimiento	Requerimiento	Elementos de diseño	Unidad de código	Casos de prueba	Manuales

Fuente: adaptado de Westfall [18].

4.2.2.2.2. Información necesaria para realizar trazabilidad

Para realizar cualquier tipo de trazabilidad se necesita información que permita establecer las relaciones entre los requerimientos y las demás etapas del proceso de ingeniería de software. Algunos ítems para tener en cuenta son [16], [19]:

- Fuente del requerimiento: consiste en identificar cuál es el origen del requerimiento. Las fuentes se ven reflejadas en la recolección de los requerimientos en procesos como las entrevistas con los *stakeholders*. Por esto es necesario tener en cuenta los documentos que surgen de la especificación de requerimientos.
- Razón de ser del requerimiento: muestra la descripción del porqué el requerimiento fue especificado.

- Personas involucradas con el requerimiento: relaciona el requerimiento con su autor y con las personas encargadas.
- Requerimientos asociados: permite relacionar el requerimiento con otros requerimientos que de alguna manera dependen de él.
- Diseño del sistema: relaciona por medio de la localización los componentes del diseño del sistema con los requerimientos, así asegura que cada uno de estos es implementado en el sistema.
- Arquitectura del sistema: relaciona el requerimiento con el módulo en el que se encuentra implementado; pero no solo el módulo, también se debe relacionar el requerimiento con la sección de código (ya sea un paquete, un objeto, una clase o un método) que realmente implemente dicho requerimiento [4].
- Pruebas: permite relacionar los requerimientos con las pruebas que son diseñadas para comprobarlo. Los requerimientos de usuario, por ejemplo, se relacionan con pruebas de aceptación; mientras que los requerimientos de sistema se relacionan con las pruebas de sistema e integración.

Antes de detallar cuáles son las técnicas que se usan para realizar la trazabilidad, cabe notar que es necesario que tanto los requerimientos como los demás productos que surgen durante el desarrollo del sistema, cuenten con un identificador único (véase sección “4.2.1.1. Definir los atributos de los requerimientos”).

4.2.2.3. Técnicas utilizadas para realizar la trazabilidad

Matriz de trazabilidad: es una matriz de referencia cruzada [16], es decir, la matriz muestra las relaciones entre los elementos ubicados en las filas con los elementos ubicados en las columnas [4]. Esta matriz se utiliza con el fin de relacionar los requerimientos con los productos que se generan para satisfacerlos [20].

La anterior tabla 4.4 muestra un ejemplo de una tabla de trazabilidad, en la cual se puede tener los dos tipos de trazabilidad vertical, tanto hacia adelante como hacia atrás, ya que se describe desde la fuente del requerimiento, pasando por los elementos de diseño y las unidades de código, hasta los casos de prueba y los manuales.

Pero no solo se pueden realizar matrices de trazabilidad como la que se muestra en la tabla 4.4, también se es posible hacer matrices que muestren la relación solo entre dos elementos, por ejemplo, requerimientos y casos de uso, como la tabla 4.5.

TABLA 4.5. MATRIZ DE TRAZABILIDAD

Requerimiento funcional	UC-1	UC-2	UC-3	UC-4
RF-1	X			
RF-2	X			
RF-3			X	
RF-4			X	
RF-5		X		X

Fuente: tomado de [16].

Lista de trazabilidad: es una técnica más simple y más compacta que la matriz, donde cada requerimiento tiene una lista que muestra la información de trazabilidad. Estas listas, en general, como se muestra en la tabla 4.6, consisten en ubicar en la primera columna los requerimientos y en la segunda poner los requerimientos que están asociados o de los cuales depende el requerimiento de la primera columna [16], lo cual más adelante permite crear una visualización gráfica más clara de los requerimientos, como son los grafos de dependencias, que se utilizan para identificar rutas críticas y ayudar con la priorización de requerimientos (véase sección “4.2.1.2. Priorización”). La desventaja de utilizar las listas a cambio de la matriz de trazabilidad es que no se puede evaluar de manera inversa la relación [16].

TABLA 4.6. LISTA DE TRAZABILIDAD

Requerimiento	Dependencia
RF-1	-
RF-2	RF-1
RF-3	RF-2, RF-1
RF-4	RF-1, RF-3
RF-5	RF-2

Fuente: adaptado de [4].

4.2.2.2.4. Ejemplo de localización y trazabilidad

El siguiente ejercicio tiene como objetivo principal aclarar las posibles dudas que surjan al estudiar los conceptos de localización y trazabilidad. Para esto se listan a continuación ciertos elementos que hacen parte de un proyecto realizado para la materia Ingeniería de Software de la Pontificia Universidad Javeriana, llamado *Demented Movie Game*, el cual pretendía simular el juego Super Triumph.¹ No se incluyen los componentes registrados en el documento de diseño, debido a que el diseño de la aplicación no puede ser resumido.

El ejercicio consiste en relacionar los requerimientos con su origen y los objetos derivados. Para esto, al final del ejercicio se encuentran las tablas en las cuales se debe registrar la trazabilidad tanto horizontal como vertical.

Reglas:

TABLA 4.7. REGLAS DEL EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD

ID regla	Descripción
REG1	Super Triumph permite jugar en dos diferentes modalidades.
REG2	El jugador debe elegir con cuál característica apuesta. Puede elegir si la apuesta a la mayor o a la menor. Eso quiere decir que dentro de las cartas que se ponen en juego gana el que tenga el mayor o el menor puntaje, dependiendo si está a la mayor o a la menor. Quien gana se lleva todas las cartas que están en la mesa.
REG3	El turno de un jugador termina cuando le pide a un contrincante una carta y el contrincante no la tiene.
REG4	Los cuartetos se forman con la letra del identificador de la carta.

Fuente: elaboración propia.

¹ Super Triumph es la versión latina del juego Ace Trump, véase: https://en.wikipedia.org/wiki/Ace_Trumps

Casos de uso:

TABLA 4.8. CASOS DE USO DEL EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD

ID caso de uso	Nombre	Descripción
IMCU-023	Pedir carta a otro usuario	El jugador puede pedir cartas a otro jugador.
IMCU-024	Escoger modalidad	Escoger la segunda modalidad que permite el juego.
IMCU-026	Repartir cartas	Repartición equitativa de las cartas a todos los usuarios presentes en la partida. En caso de que el número no permita una repartición equitativa de cartas, las cartas se repartirán hasta que se terminen, en el orden de inscripción de juego por parte de los usuarios.
IMCU-040	Apostar a la mayor	El jugador, después de escoger la modalidad que desea jugar, cuando tiene su turno de juego, toma la decisión de jugar a la mayor de alguna de las características que tiene la primera carta de su baraja. Consiste en apostar su carta a que los demás jugadores tienen menor puntaje que él en la característica que escoge.
IMCU-041	Apostar a la menor	El jugador, después de escoger la modalidad que desea jugar, cuando tiene su turno de juego, toma la decisión de jugar a la menor de alguna de las características que tiene la primera carta de su baraja. Consiste en apostar su carta a que los demás jugadores tienen mayor puntaje que él en la característica que escoge.

Fuente: elaboración propia.

Componentes definidos:

TABLA 4.9. COMPONENTES DEL EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD

Componente	Descripción
Jugar	Componente definido para reunir los requerimientos relacionados con la manera en la cual se desarrolla el juego.
Consultas	Componente definido para contener los requerimientos relacionados con la base de datos.
Administración	Son los requerimientos asociados a la gestión de la información de los participantes del juego.

Fuente: elaboración propia.

*Requerimientos:***TABLA 4.10. REQUERIMIENTOS DEL EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD**

ID requerimiento	Descripción	Versión
IMR002	El sistema debe validar que el usuario y la contraseña dados existan en la base de datos.	2.0
IMR002	El sistema debe validar los datos para acceder al sistema.	1.0
IMR016	El sistema debe permitir que el anfitrión escoja la modalidad con la cual se va a jugar la partida.	1.0
IMR027	El sistema debe permitir al jugador en turno apostar al mayor valor o al menor valor, en la segunda modalidad.	1.0
IMR039	El sistema debe poder terminar el turno de un jugador si sus posibilidades de juego son nulas en la primera modalidad.	1.0
IMR024	El sistema debe validar cuál es la carta que tiene la mejor característica elegida por el jugador en turno en la segunda modalidad.	2.0
IMR027	El sistema debe permitir al jugador en turno apostar al mayor valor en la segunda modalidad.	2.0
IMR028	El sistema debe permitir al jugador en turno apostar al menor valor en la segunda modalidad.	2.0
IMR039	El sistema debe validar cuál carta es la mejor para la segunda modalidad.	1.0

Fuente: elaboración propia.

Pruebas:

TABLA 4.11. PRUEBAS DEL EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD

ID prueba	Qué se prueba	Resultados
P1	Registro de usuario.	Aprobado
P2	La carta seleccionada es la que tiene la mejor característica.	Aprobado
P3	El sistema asignó el turno al siguiente jugador, debido a que el jugador actual no puede realizar ninguna jugada.	Aprobado
P4	El jugador puede seleccionar la modalidad con la cual quiere jugar una partida.	Aprobado
P5	El jugador apuesta "A la mayor".	Aprobar
P6	El jugador apuesta "A la menor".	Aprobado

Fuente: elaboración propia.

Solución: la tabla 4.12 se debe utilizar para definir la localización de los requerimientos. Para esto marque con una X la casilla correspondiente al componente en el cual usted cree que se encuentra el requerimiento.

TABLA 4.12. PLANTILLA DE REQUERIMIENTOS: EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD

Requerimiento	Jugar	Consultas	Administración
IMR002			
IMR016			
IMR027			
IMR039			
IMR024			

Fuente: elaboración propia.

En la tabla 4.13 registre las fuentes relacionadas con el requerimiento, a fin de definir la trazabilidad *vertical hacia atrás*. Para la trazabilidad *vertical hacia adelante*, ubique los casos de prueba relacionados con el requerimiento.

TABLA 4.13. PLANTILLA DE REQUERIMIENTOS 2: EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD

Fuente 1	Fuente 2	Requerimiento	Casos de prueba
		IMR002	
		IMR016	
		IMR027	
		IMR039	
		IMR024	

Fuente: elaboración propia.

En cuanto a la trazabilidad horizontal, utilice las tablas a continuación para ubicar las versiones de los requerimientos listados anteriormente (figura 4.5).

FIGURA 4.5. PLANTILLA DE REQUERIMIENTOS 3: EJERCICIO DE LOCALIZACIÓN Y TRAZABILIDAD

ID requerimiento	Versión	➡	ID requerimiento	Versión

ID requerimiento	Versión	➡	ID requerimiento	Versión

ID requerimiento	Versión	➡	ID requerimiento	Versión

ID requerimiento	Versión	➡	ID requerimiento	Versión

Fuente: elaboración propia.

4.2.3. Monitoreo y reporte

4.2.3.1. Definir políticas de administración

Definir las políticas de administración no es más que establecer el plan de administración de requerimientos, en el cual se describen los procedimientos y los estándares que deberían seguirse durante la administración de los requerimientos. Ya que se define como el plan de administración de requerimientos, debe incluir a las personas responsables de llevar a cabo las actividades del plan [21].

Para definir las políticas de administración de requerimientos, es preciso empezar por una lista general que contenga los siguientes lineamientos [4]:

- Cuáles son los estándares que se deben seguir durante el proceso.
- Cómo se deben recolectar los requerimientos y cuáles son las personas involucradas en este procedimiento.
- Cómo se van a identificar cada uno de los elementos contenidos en la especificación.
- Qué tipo de atributos se considerarán al momento de especificar cada requerimiento (véase sección “4.2.1.1. Definir los atributos de los requerimientos”).
- Qué tipo de atributos debe contener la especificación de requerimientos.
- Quiénes son los encargados de realizar la especificación de requerimientos.
- Teniendo en cuenta las políticas organizacionales, cómo se realizará el análisis de los requerimientos.
- Especificar las políticas de administración de cambio de requerimientos (véase sección “4.2.4. Administración del cambio”).
- Especificar quiénes serán los encargados, qué tipo y qué técnica de trazabilidad será utilizada en el desarrollo del sistema (véase sección “4.2.2.2. Trazabilidad”).
- Definir cuál es la forma correcta de almacenar los requerimientos rechazados [22].

Una vez definida la lista de políticas (o si ya se tiene una lista general), se deben seleccionar las políticas más adecuadas para el proceso de administración del proyecto específico.

El plan de administración de requerimientos debe incluir secciones que se encarguen de definir:

- Los objetivos que se van a cumplir en el proceso de administración de requerimientos. Se debe especificar todo, por ejemplo, los objetivos que se quieren cumplir en determinado lapso, el alcance que tendrá el documento [21] y las reglas aplicables en cualquier situación (ya sea una entrevista con el cliente o el proceso que se realizará para solicitar algún cambio). Además, caben mencionar

las herramientas y capacitaciones sobre ellas para tener un mejor desempeño en el proceso de administración [22].

- El proceso de recolección de requerimientos que se va a seguir [23] (véase sección “2.1. Recolección”).
- Los estándares que se deben seguir para hacer la descripción de los requerimientos y el SRS, por ejemplo el estándar IEEE 29148 [24].
- Cómo realizar el control de cambios [23] (véase sección “4.2.4. Administración del cambio”). Dada la situación general de casi todos los proyectos, donde el cambio en los requerimientos es más que inevitable, es necesario definir estrategias que permitan administrar el cambio de los requerimientos, por ejemplo, mantener un historial de cambios o establecer un canal único para el cambio [21]; también se pueden usar herramientas que complementen esta actividad. Una que menciona Ralph Young es *Change Control Boards* (CCB) [22].
- Cómo realizar las revisiones y la validación de los requerimientos: las revisiones no requieren mucho trabajo y se obtiene un beneficio [22], que es el aseguramiento de la calidad de los documentos e informes de la especificación.
- Cuál es la técnica (ya sea la matriz o la lista) y qué información [25] se utilizará para llevar a cabo la trazabilidad (véase sección “4.2.2.2. Trazabilidad”).
- Las métricas de la administración de requerimientos: esto con el fin de conocer en qué medida está avanzando el proyecto, por ejemplo, el estado de cada requerimiento, el número de cambios acumulado [21], etc.
- Qué excepciones en cuanto al plan se pueden realizar, en qué casos y con la autorización de quién.

Los beneficios que trae realizar un plan de administración de requerimientos son:

- El tener las políticas por escrito permite que las personas involucradas en el desarrollo del proyecto tengan el conocimiento de cómo se debe hacer el proceso de administración de requerimientos, lo cual permite que se ahorre tiempo durante este proceso.
- De alguna manera, permite que el grupo de desarrollo sea proactivo a cualquier eventualidad, lo cual es una gran ventaja, pues todos saben qué se debe hacer en caso de que se presente un problema o retraso en el proceso.

En cuanto a los problemas que se pueden presentar, está el hecho de que se debe consultar a las personas que están involucradas en el proceso de ingeniería de requerimientos para modificar, proponer y revisar las políticas, con el fin de generar un entrenamiento que asegure que las personas involucradas en el proceso conozcan y apliquen dichas políticas, lo cual puede tomar mucho tiempo.

4.2.3.2. Almacenar requerimientos rechazados

Ya que los requerimientos rechazados pueden llegar a surgir durante alguna de las etapas posteriores a la especificación de requerimientos (véase sección “2.3. Especificación de requerimientos”), es indispensable contar con un registro que los almacene y las razones por las cuales se decidió que estos no deberían ser implementados, con el fin de evitar un proceso de análisis de requerimientos (véase sección “2.2. Análisis”), a efectos de verificar por qué no se tuvieron en cuenta en el análisis anterior [8]. Para almacenar estos requerimientos Sommerville y Sawyer [4] recomiendan utilizar una base de datos.

4.2.4. Administración del cambio

4.2.4.1. Control de cambios

Varios autores, como Lam o Wiedemann, indican que “Los cambios siempre ocurren en un proyecto y en cualquier fase” [26]. Para esto se es preciso contar con un proceso de control de cambios que le permita al equipo de trabajo manejar el cambio en los requerimientos de la mejor manera, a fin de evitar cualquier incoherencia o atraso dentro del proceso [27]. A estos efectos, se deben definir los lineamientos o las políticas que se van a seguir cuando se va a efectuar un cambio en los requerimientos.

4.2.4.1.1. Proceso de control de cambios

El objetivo de definir un proceso de control de cambios es establecer cuáles deben ser los pasos formales cuando se desea hacer un cambio en algún requerimiento. Tener en cuenta los siguientes pasos permite ese control de cambios:

1. Cuál debe ser el proceso para realizar la solicitud del cambio en los requerimientos: el definir un proceso de control del cambio permite asegurar que los cambios propuestos son evaluados o aplicados de manera consistente [4]. Para una solicitud de cambio se realizan las siguientes actividades:
 - *Llenar la plantilla de solicitud de cambios.* Esta debe contener atributos como: el número de la solicitud, la fecha en la cual se hace, el nombre de la persona que solicita el cambio, la descripción del problema, la descripción del cambio, el costo del cambio, y campos en los cuales posteriormente se aceptará o se registrarán las razones por las cuales no se acepta el cambio [28]. En la tabla 4.14 se muestra un ejemplo de cuáles son los atributos en la solicitud del cambio.

TABLA 4.14. POSIBLES ATRIBUTOS PARA EL CONTROL DE CAMBIOS

Solicitud de cambio	
Dueño	Identificador único
Resultados del análisis	Fecha de registro
Referencia	Prioridad
Fecha de decisión	Persona que propone
Estado	Descripción
Justificación	Explicación

Fuente: adaptado de [26].

- *Analizar la solicitud de cambio* [28] y *cálculo de costos* [26]: después de que es realizada la solicitud del cambio, esta se analiza para verificar si es válido. Es obligatorio evaluar si es o no factible, cuál es el efecto que tendrá y cuáles son los costos en los que se incurrirá en caso de hacer la modificación en el sistema. Existen tres factores principales que pueden influir o no en el cambio de un requerimiento [26]:
 - * Los resultados del análisis (en particular la incidencia en los costos y el calendario).
 - * La estructura técnica del sistema que se va a desarrollar.
 - * Las estructuras organizativas del cliente y el proveedor.
 - *Implementación del cambio*: una vez aprobada la solicitud de cambio, puede aplicarse al sistema, es decir, se debe modificar el SRS, y si es necesario también el diseño, la implementación, las pruebas y los manuales [29]. También es preciso que se registre el cambio realizado, para informar a todo el grupo de trabajo cuál fue el cambio de la especificación y qué debe modificarse en adelante [26].
 - *Mantener el historial de cambios*: cuando se efectúa el cambio es importante almacenar toda la información pertinente a este.
2. Establecer un comité de control de cambios [4]: esto con el fin de asegurar que todas las solicitudes de cambio serán evaluadas. Para establecer este comité se siguen los siguientes pasos:
- *Seleccionar a los miembros*: el objetivo de este paso es seleccionar a las personas correctas que se encarguen de evaluar todos los cambios. Todos los *stakeholders* pueden ser parte del comité [28].

- *Reuniones para evaluar los cambios*: los miembros del comité deben reunirse de manera constante para asegurarse de que las solicitudes de cambios se evalúan y que son contestadas en un tiempo determinado [28].
3. Definir cómo notificar el cambio a los *stakeholders* afectados [4]: se debe informar a los *stakeholders* afectados los cambios que fueron aceptados e implementados.

La ventaja de definir un proceso de control de cambios es que los cambios se llevarán a cabo de manera controlada; entonces, cada vez que se realice uno, se analizará el impacto que generará en el proyecto, y al finalizar, los *stakeholders* podrán conocer cuáles fueron las modificaciones hechas.

4.2.4.1.2. Usar una base de datos de requerimientos

Para mantener un mejor control sobre los cambios del sistema, se recomienda contar con una base de datos de requerimientos que almacene todos los atributos, los estados y los cambios realizados. Esta práctica trae beneficios como:

- Realizar búsquedas de grupos de requerimientos y mantener los enlaces de cada requerimiento hacia otros requerimientos.
- Tener una base de datos que funciona como un repositorio en el cual se puede trabajar de manera concurrente sin la posibilidad de que se presenten conflictos [4].
- Publicar al grupo el estado de la solicitud del cambio si fue aceptado, rechazado o está pendiente para ser evaluado [30].

4.2.4.1.3. Ejercicio control de cambios

De acuerdo con el proceso mencionado para el control de cambios, defina qué atributos debe contener la plantilla de solicitud de un cambio, a fin de que se acomode al proceso de administración de requerimientos dentro su proyecto. Para esto utilice la tabla 4.15.

TABLA 4.15. EJERCICIO DE CONTROL DE CAMBIOS

Atributo	Descripción

Fuente: elaboración propia.

Después de definir los atributos mínimos que se diligencian para solicitar un cambio en los requerimientos, establezca el proceso para informar a los *stakeholders* del proyecto que se ven afectados por los cambios realizados.

4.3. Beneficios de la administración de requerimientos

Dentro de los beneficios de hacer una buena administración de requerimientos se encuentran:

- Permite verificar que el sistema que se está construyendo es el mismo durante toda la fase de construcción [31].
- Reusabilidad, ya que la información que se registra durante la especificación del proyecto puede ser reutilizada en la construcción de otros [26].
- Seguridad legal: ya que los requerimientos son prácticamente el único medio escrito mediante el cual el cliente y el gerente del proyecto pueden comprobar que el sistema que se entrega es el mismo sistema que se solicitó.
- Da seguridad para todos los *stakeholders* del proyecto, ya que se puede tener una imagen clara de qué se está desarrollando, cómo y con qué [26].
- La administración de requerimientos permite tener una especificación de requerimientos clara que no se preste para malos entendidos ni confusiones [32].
- Como la administración de requerimientos tiene que agregar los atributos descritos en la sección “4.2.1.1. Definir los atributos de los requerimientos”, es muy fácil generar un reporte de avance del proyecto de acuerdo con la información registrada en el campo de estado del requerimiento, ya que indica el estado de implementación de cada uno [26].
- Permite mantener los acuerdos sobre el alcance del proyecto entre los desarrolladores, usuarios y los demás *stakeholders* del proyecto [32], esto además conlleva que se genere un sistema de alta calidad [33].
- Se establecen los criterios de aceptación del sistema que será entregado [32].
- Permite generar las pruebas adecuadas que verifiquen que el proyecto tiene las funcionalidades que el cliente solicitó [26].
- Se reducen las posibilidades de realizar la entrega del proyecto por fuera de la fecha límite y se reducen los costos, debido a que la administración de requerimientos disminuye el número de errores [33].
- Además de los beneficios ya mencionados, también se pueden obtener los beneficios descritos en la sección “4.2.2.2. Trazabilidad”.

4.4. Beneficios de usar una herramienta de administración de requerimientos

Gracias al avance de la tecnología, se han desarrollado diferentes opciones para apoyar los procesos de construcción de productos de todo tipo, y la construcción de software no se queda atrás. Por esto existen en el mercado herramientas hechas con el propósito de apoyar los procesos de administración de requerimientos dentro del desarrollo de software. Cuando se utilizan este tipo de herramientas, son muchos los beneficios con los que se pueden encontrar los grupos de desarrollo, dentro de los cuales se encuentran los beneficios de negocio y los beneficios de proyecto.

4.4.1. Beneficios de negocio

Los beneficios que se pueden obtener son los siguientes:

- Ahorro de tiempo, ya que se automatizan muchos de los procesos de la administración de requerimientos [34].
- Disminución en el estrés del equipo de desarrollo, debido a la automatización de los procesos de trazabilidad y control de cambios [34].
- Algunas herramientas proporcionan permisos de acceso para los *stakeholders*, lo cual asegura que los requerimientos del proyecto solo están disponibles para los *stakeholders* involucrados [2].

4.4.2. Beneficios de proyecto

Los beneficios que se pueden obtener son los siguientes:

- Permiten que el proceso de control de cambios se realice de manera sencilla [35], ya que estas herramientas pueden mantener un historial de cambios para cada requerimiento. Ello posibilita la comunicación del cambio a los *stakeholders* y, además, si se quiere restaurar algún estado anterior de algún requerimiento, se puede hacer con facilidad [2].
- Es más eficiente el análisis de impacto de un cambio (véase sección “4.2.4. Administración del cambio”), debido a que la información que se utiliza para llevar a cabo este análisis se encuentra reunida en el mismo lugar [35].
- Se pueden guardar con mayor eficiencia los atributos del requerimiento, debido a que los campos de los atributos de cada requerimiento ya están listos para recibir la información [35].

- Algunas herramientas les permiten a los equipos estructurar sus requerimientos, es decir, el equipo de desarrollo decide cuáles son los atributos con lo que se va a especificar el requerimiento [34].
- Facilitan la implementación del proceso de trazabilidad [2]. En algunos casos se pueden implementar la trazabilidad hacia adelante y hacia atrás (véase sección “4.2.2.2. Trazabilidad”), ya que muchas de estas herramientas permiten la definición de casos de uso y componentes de diseño [35].
- Muchas de estas herramientas permiten saber casi de manera instantánea cuál es el estado del proyecto en cuanto a implementación de requerimientos [2].
- Permiten mejorar la colaboración entre los *stakeholders* del proyecto [34].
- Aseguran que el producto que se está realizando es el producto que el cliente está esperando [35].

Referencias

- [1] K. Wiegers, *More About Software Requirements: Thorny Issues and Practical Advice*, 1st ed. Redmond, WA: Microsoft Press, 2005.
- [2] K. Wiegers y J. Beatty, *Software Requirements*, 3rd ed. Redmond, WA: Microsoft Press, 2013.
- [3] “The Standish Group Report: CHAOS”. 2014. [Online]. Disponible: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
- [4] I. Sommerville y P. Sawyer, *Requirements Engineering: A Good Practice Guide*. London: John Wiley & Sons, 1997.
- [5] I. Alexander y R. Stevens, *Writing Better Requirements*, 1st ed. London: Addison-Wesley Professional, 2002.
- [6] The Land Software Engineering Centre, “Requirements Attributes”, 2010. [Online]. Disponible: http://www.lsec.dnd.ca/qsd_current_version/sw_eng/di/reqpro_attributes.htm
- [7] J. Heumann, “The Five levels of requirements management maturity”, The Rational Edge, 2003. [Online]. Disponible: http://vincentvanrooijen.com/container%5CRequirements%20Management%5CThe%205%20levels%20of%20ManagementMaturity_TheRationalEdge_Feb2003.pdf
- [8] I. F. Hooks y K. A. Farry, *Customer Centered Products: Creating Successful Products Through Smart Requirements Management*, 1st ed. New York: Amacom/American Management Association, 2000.
- [9] K. Wiegers, “First things first,” *Softw. Dev.*, vol. 7, no. 9, pp. 48-53, Sep. 1999.

- [10] P. Berander y A. Andrews, "Requirements prioritization", en *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Eds. Springer Berlin: Heidelberg, 2005, pp. 69-94.
- [11] R. Botta y A. T. Bahill, "A prioritization process", *Eng. Manag. J.*, vol. 19, no. 4, pp. 20-27, Dec. 2007.
- [12] D. Firesmith, "Prioritizing requirements", *J. Object Technol.*, vol. 3, no. 8, pp. 35-48, 2004.
- [13] N. Mead, "Requirements prioritization introduction", United States Computer Emergency Readiness Team (US-CERT), Sep. 26, 2006. [Online]. Disponible: <https://www.us-cert.gov/bsi/articles/best-practices/requirements-engineering/requirements-prioritization-introduction>. [Citado: 29-mar-2017].
- [14] M. Daneva y A. Herrmann, "Requirements prioritization based on benefit and cost prediction: a method classification framework", en *2008 34th Euromicro Conference Software Engineering and Advanced Applications*, 2008, pp. 240-247.
- [15] M. Eriksson, K. Borg, y J. Börstler, "7.2.1 The FAR Approach-Functional Analysis/Allocation and Requirements Flowdown Using Use Case Realizations", *INCOSE Int. Symp.*, vol. 16, no. 1, pp. 950-964, Jul. 2006.
- [16] M. Hokkanen, "Requirements traceability", Tesis de Maestría, Lappeenranta University of Technology, Finlandia, 2001. [Online]. Disponible: <https://core.ac.uk/download/pdf/39921616.pdf>
- [17] F. A. C. Pinheiro, "Chapter 5", en *Requirements Traceability*, s. d.
- [18] L. Westfall, "Bidirectional requirements traceability", The WestFallTeam, 2006. [Online]. Disponible: <http://www.compaid.com/caiinternet/ezone/westfall-bidirectional.pdf>
- [19] L. Antonelli y A. Oliveros, "Traceability en la etapa de elicitación de requerimientos," en *Anais do WER01-Workshop em Engenharia de Requisitos, Buenos Aires, Argentina, Novembro 22-23, 2001*, pp. 1-19.
- [20] Ludwig Consulting Services, "Interviewing User-Tips". 2009. [Online]. Disponible: <http://www.jiludwig.com/Interviews.html>. [Consultado: 29-mar-2017].
- [21] Ludwig Consulting Services, "Requirements management", en *Introduction to Requirements Management*. [Online]. Disponible: http://www.jiludwig.com/Requirements_Management.html. [Consultado: 29-mar-2017].
- [22] R. R. Young, "Twelve requirements basics for project success", *CrossTalk. The Journal of Defense Software Engineering*, Dec. 2006. [Online]. Disponible: http://www.compensationanalytics.com/_resources/RequirementsBasics.pdf

- [23] Construction Management Guide, “Create requirements management plan”, 2009. [Online]. Disponible: <http://www.cmguide.org/archives/1420>
- [24] International Organization for Standardization, “ISO/IEC/IEEE 26531:2015. Systems and software engineering”. [Online], Dec. 2011. Disponible: <https://www.iso.org/standard/43090.html>
- [25] “Guidelines: Requirements Management Plan”. [Online]. Disponible: http://sce.uhcl.edu/helm/rationalunifiedprocess/process/modguide/md_rmp.htm. [Consultado: 29-mar-2017].
- [26] C. Hood, S. Wiedemann, S. Fichtinger, y U. Pautz, *Requirements Management: The Interface Between Requirements*. Berlin: Springer, 2008.
- [27] A. Kobayashi y M. Maekawa, “Need-based requirements change management”, en *Proceedings. Eighth Annual IEEE International Conference and Workshop On the Engineering of Computer-Based Systems-ECBS 2001*, 2001, pp. 171-178.
- [28] “Activity: Establish Change Control Process”. [Online]. Disponible: http://sce.uhcl.edu/helm/rationalunifiedprocess/process/activity/ac_epcmp.htm. [Consultado: 29-mar-2017].
- [29] I. Sommerville, *Software Engineering*, 9 ed. Boston: Pearson, 2010.
- [30] M. Hoffmann, N. Kuhn, M. Weber, y M. Bittner, “Requirements for requirements management tools”, en *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International*, 2004, pp. 301-308.
- [31] E. Hull, K. Jackson, y J. Dick, *Requirements Engineering*, 3rd ed. London: Springer, 2010.
- [32] Oakleigh Consulting, “When is Projects go Bad”. [Online]. Disponible: <http://www.oakleigh.co.uk/page/1388/White-Papers/Whitepaper-Articles/When-IS-Projects-Go-Bad>. [Consultado: 29-mar-2017].
- [33] A. Davis y D. Leffingwell, “Using requirements management to speed delivery of higher quality applications”, 1995. [Online]. Disponible: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.194.7124&rep=rep1&type=pdf>
- [34] “Requirements Management Tools - Overview”. [Online]. Disponible: <http://pmblog.accompa.com/2009/07/30/requirements-management-tools-overview/>. [Consultado: 29-mar-2017].
- [35] “Benefits of Requirements Management Tool - Product Management School”. [Online]. Disponible: http://www.productmanagementschool.com/w1/Benefits_of_Requirements_Management_Tool. [Consultado: 29-mar-2017].