



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“Deuda Técnica: Método de Gestión de la Deuda Técnica en los
Requerimientos”**

TESIS

QUE PARA OPTAR POR EL GRADO DE:

**MAESTRO EN CIENCIA E INGENIERÍA
DE LA COMPUTACIÓN**

PRESENTA:

Fernando Antonio Sánchez Montoya

DIRECTOR DE TESIS:

M. C. María Guadalupe Elena Ibargüengoitia González Facultad de Ciencias

Ciudad Universitaria, CDMX a Enero, 2021



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Tesis para obtener el grado de Maestro en Ciencia e Ingeniería en Computación
Primera Edición, 12 de enero de 2021

Agradecimientos

Agradezco a toda mi familia, en especial a mis abuelos que siempre que los miro me devuelven una sonrisa y cada vez que hablamos me apoyan y motivan en todo. ¡Los quiero mucho!

Abuelo Fernando Sánchez Rivera, descansa en paz.

A mis tíos que siempre han sido un gran ejemplo y siempre su apoyo ha estado presente.

A mis padres que son lo más importante en mi vida y me han dado muchas enseñanzas. Los quiero mucho.

A Martha que ha sido mi acompañante de viajes a la escuela y con ella he pasado muchos momentos especiales durante este periodo de maestría.

A Mariel que es mi confidente, me ha animado mucho, enseñado muchas cosas interesantes y que fue, es y será muy especial siempre para mí. Te amo.

A mi China y Banana por dejarse apapachar y siempre hacerme sonreír.

A mis amigos Kevin, David, Robert, Iván, Antonio, Nestor, Mónica y Kevin a pesar de los años son como son, siempre nos hemos apoyado mutuamente y son un ejemplo a seguir.

A la UNAM que siempre me ha dado grandes oportunidades y me ha abierto los ojos al conocimiento y a la realidad. ¡Gracias! Mi segunda

casa.

A Lupita por apoyarme siempre y confiar en mí. ¡Gracias, maestra!

Gracias a Lulú, Cecilia y Amalia que siempre están para nosotros con la mejor disposición. Un abrazo con cariño.

Al CONACYT por el apoyo y oportunidad de hacer una maestría.

Índice general

1. Introducción	1
1.1. Objetivo	4
1.1.1. Objetivo principal	4
1.1.2. Objetivos específicos	4
1.2. Problema u Oportunidad	4
1.3. Pregunta de Investigación	5
1.4. Contribución y relevancia	5
2. Marco teórico	7
2.1. Deuda Técnica	7
2.1.1. Propiedades de la Deuda Técnica.	8
2.1.2. Clasificación de la Deuda Técnica	9
2.1.2.1. Taxonomía de McConnell	9
2.1.2.2. Clasificación de Fowler	10
2.1.2.3. Panorama de Kruchten.	11
2.1.2.4. Clasificación de los tipos de Deuda Técnica.	12
2.1.3. Manejo de la Deuda Técnica	18
2.2. Requerimientos del Software	18
2.2.1. Ingeniería de Requerimientos	19
2.2.2. Los Requerimientos	19
2.2.3. SWEBOK	21
2.2.3.1. Proceso de los requerimientos.	21
2.2.3.2. Captura de los requerimientos.	21
2.2.3.3. Análisis de requerimientos.	22
2.2.3.4. Especificación de los requerimientos.	23
2.2.3.5. Validación de requerimientos.	23
2.2.3.6. Consideraciones Prácticas.	23

2.2.4.	ISO/IEC 29148-2011.	24
2.2.4.1.	Conceptos	24
2.2.4.2.	Procesos. 29	
2.2.4.3.	Olor en los requerimientos. 35	
2.3.	ISO/IEC 29110. Perfil Básico.	36
2.3.1.	La ISO/IEC 29110 y sus partes.	36
2.3.1.1.	Descripción y funciones de la ISO/IEC 29110.	36
2.3.2.	ISO/IEC 29110-5-1-2.	38
2.3.2.1.	Procesos de la ISO/IEC 29110-5-1-2.	39
2.4.	KUALI-BEH	46
2.4.1.	Conceptos.	46
3.	Estado del Arte	49
3.1.	Deuda Técnica en Requerimientos del Software	50
3.1.1.	Definiciones y clasificaciones de la Deuda Técnica en los Re- querimientos	50
3.1.2.	Causas de la Deuda Técnica en Requerimientos del Software .	51
4.	Método propuesto.	53
4.1.	Método de Gestión de la Deuda Técnica en los Requerimientos.	53
4.2.	Las prácticas del método.	56
4.2.1.	Práctica de Prevención de la Deuda Técnica en los Reque- rimientos.	57
4.2.1.1.	Actividades de la práctica y sus tareas asociadas. . .	59
4.2.1.2.	Conocimientos generales y sugerencias de he- rramientas para la implementación de la prác- tica.	59
4.2.2.	Práctica de Detección de la Deuda Técnica en los Requerimien- tos.	63
4.2.2.1.	Actividades de la práctica y sus tareas asociadas. . .	63
4.2.2.2.	Conocimientos generales y sugerencias de he- rramientas para la implementación de la prác- tica.	63
4.2.3.	Práctica de Manejo de la Deuda Técnica en los Requerimientos. 65	
4.2.3.1.	Actividades de la práctica y sus tareas asociadas. 65	
4.2.4.	Práctica de retrospectiva.	67

4.2.4.1. Actividades de la práctica y sus tareas asociadas.	68
5. Evaluación del método propuesto.	71
5.1. Evaluación del Método	71
5.1.1. Evaluación de la P1. Práctica de Prevención de la Deuda Técnica en los Requerimientos.	71
5.1.2. Evaluación de la P2. Practica de Detección de la Deuda Técnica en los Requerimientos.	72
5.1.3. Evaluación de la P3. Practica de Manejo de la Deuda Técnica en los Requerimientos.	72
5.1.4. Evaluación de la P4. Practica de Retrospectiva.	73
5.2. Observaciones de los expertos.	73
5.3. Cambios al método.	73
6. Conclusiones y trabajo futuro	75
6.1. Conclusiones	75
6.2. Trabajo futuro	76
A. Propuesta de validación para expertos del método de gestión de la deuda técnica en los requerimientos	77

Índice de figuras

2.1. Cuadrante de la Deuda Técnica [1]	11
2.2. Panorama de la Deuda Técnica [2].	12
2.3. 13 Tipos de Deuda Técnica y sus indicadores. Traducida de [3].	15
2.4. 10 tipos de Deuda Técnica según el estudio sistemático de Li [4]. Traducida.	17
2.5. Partes de la ISO/IEC 29110 [5]	36
2.6. Serie ISO/IEC 29110 [5]	38
2.7. Procesos del Perfil Básico [5]	39
2.8. Diagrama de los procesos de gestión del proyecto [5]	42
2.9. Diagrama del procesos de implementación del software. [5]	45
2.10. Plantilla para método de KUALI-BEH.[6]	47
2.11. Plantilla de práctica de KUALI-BEH.[6]	48
4.1. Método de Gestión de la Deuda Técnica en Requerimientos (MGDTR).	55
4.2. Tabla del Método de Gestión de la Deuda Técnica en Requerimientos (MGDTR).	56
4.3. Definición de la Práctica 1 (P1) del MGDTR.	58
4.4. Tabla de revisión de las características y atributos de los requerimientos.	62
4.5. Definición de la Práctica 2 (P2) del MGDTR.	64
4.6. Gáfica burn-down para la detección de la Deuda Técnica en los Requerimientos.	65
4.7. Definición de la Práctica 3 (P3) del MGDTR.	66
4.8. Tabla para las actividades a realizar.	67
4.9. Definición de la Práctica 4 (P4) del MGDTR.	68
4.10. Tabla de las actividades realizadas.	69
4.11. Tabla de las actividades realizadas que no dieron resultados esperados.	70

Introducción

El desarrollo de software en pequeñas, medianas y grandes empresas dedicadas a la producción de este, la mayoría de las veces está condicionado a un factor imparable como lo es el tiempo. En estas empresas se tiene que cumplir con cronogramas al haber fechas de inicio, entregas parciales y fin de iteraciones o cierre; en las cuales el equipo de desarrollo se puede ver presionado por los límites de éstas, lo cual conlleva omitir actividades, tomar atajos, incumplir buenas prácticas o desviarse de alguna metodología, por mencionar algunos ejemplos. Incumplir estas actividades pueden funcionar perfectamente a corto plazo para realizar una entrega o liberar un producto a tiempo, pero a mediano o a largo plazo las omisiones, falta de buenas prácticas o incumplimiento de la metodología pueden conducir al equipo de desarrollo a incurrir en una Deuda Técnica.

En el capítulo uno se explica la Deuda Técnica. El termino fue acuñado por Ward Cunningham, haciendo referencia a una deuda financiera.

Podemos preguntarnos si: ¿Adquirir una deuda financiera es “malo” o “bueno”?

Para la mayoría de las personas, adquirir una deuda no es “tan malo” como parece o en otras palabras, es más común de lo que parece, pues si es necesario salir rápidamente de un apuro o aprovechar una promoción a crédito, es sencillo decidir adquirir una deuda, pensando en que la misma tiene que ser saldada en cierto tiempo para no generar más intereses posteriores. De acuerdo con lo anterior, podríamos afirmar que para adquirir una deuda se debería contemplar un plan para pagarla en un periodo a futuro, por lo tanto, incurrir en una deuda tendría que ser una decisión prudente

1. INTRODUCCIÓN

y consciente, tomada por el que se convertirá en deudor, en el mejor de los casos. Además, no se consideraría una mala acción, sino todo lo contrario.

Análogamente, una Deuda Técnica, es una obligación de una organización de desarrollo de software que ha incurrido en una deuda a corto plazo con la finalidad de salir de apuros y realizar funcionalidades o entregar productos en el tiempo debido, suponiendo o pensando pagar esa deuda lo más pronto posible para no acumular más, o tener que asumir más intereses con el paso del tiempo.

La siguiente pregunta expande las nociones y repercusiones de la Deuda Técnica:

¿Hasta qué punto Deuda Técnica “buena” se vuelve una Deuda Técnica “mala”?

Se habla de incurrir en una deuda técnica a favor o en beneficio del proyecto y se debe tener un plan de cómo “pagarla” con el paso del tiempo, esto pasa cuando el equipo de desarrollo de software es consciente de la situación en la que se encuentran y llevan a cabo un plan de gestión de la Deuda Técnica.

¿Qué pasa cuando no es así? Cuando un equipo empieza a ignorar la Deuda Técnica y hace caso omiso de esta situación de la que se tiene el conocimiento, es justo dónde la Deuda Técnica se puede transformar en una “mala deuda”. Otra manera en que la Deuda Técnica podría ser “mala”, es cuando ha pasado desapercibida o es completamente ignorada y al momento de saber de su existencia, el proyecto de desarrollo de software puede estar en grave peligro.

Se sabe bien que en la mayoría de las situaciones en las que se incide en una deuda financiera esta lleva intereses implícitos por el simple hecho de contraerla, en el momento en que esta deuda se deja de saldar, los intereses crecen y crecen.

En términos de Deuda Técnica, los intereses se traducen en: horas humano extras, mantenimiento, refactorización, esfuerzo o dinero; así las consecuencias de tener una Deuda Técnica pueden estar presentes en la calidad, en la poca estabilidad frente a los cambios, productividad ineficiente o incumplimiento en el cronograma del proyecto.

Ya que es un tema bastante amplio y cada vez toma mayor importancia en el terreno de la ingeniería de software y a su vez, se va relacionando con otros procesos dentro del ciclo de vida de un proyecto, es por eso que en el presente trabajo, se

desarrollará un método para la **gestión de la Deuda Técnica en los Requerimientos del Software**, tomando como referencia los procesos de la ISO/IEC 29110 (Perfil Básico) y en la ISO/IEC 29148.

1. INTRODUCCIÓN

1.1 Objetivo

1.1.1 Objetivo principal

Diseñar un método para la gestión de *la Deuda Técnica en los Requerimientos de un proyecto de desarrollo de software*.

1.1.2 Objetivos específicos

- Revisar las ISO/IEC que servirán para el desarrollo del método.
- Diseñar el conjunto de prácticas que formarán parte del método para la gestión de la Deuda Técnica en los requerimientos.

1.2 Problema u Oportunidad

Cuando Cunningham [7] introdujo, lo que después se acuñara como el término de Deuda Técnica, lo hizo con una metáfora de una deuda financiera. Está metáfora la desarrolló aludiendo a “la primera entrega de código”, por lo tanto, la Deuda Técnica se asoció por un tiempo sólo a la práctica de codificación. Aún conforme se fue desarrollando el término y con el paso del tiempo, la mayoría de las veces el concepto de Deuda Técnica se continuó relacionando con el desarrollo o la arquitectura de código del proyecto.

Dado que el uso del término de Deuda Técnica se ha hecho cada vez más popular en la Ingeniería de Software, se han ido abarcando situaciones más allá de la primera definición establecida.

Así, la Deuda Técnica se asocia también a otras actividades como la documentación, requerimientos, arquitectura, diseño y pruebas de un proyecto [4]. Dicho esto, se toma la oportunidad de trabajar con la Deuda Técnica en los Requerimientos del software.

1.3 Pregunta de Investigación

La pregunta de investigación del presente trabajo es la siguiente:
¿Un método podría ayudar a gestionar la Deuda Técnica en los Requerimientos del software de un proyecto?

1.4 Contribución y relevancia

A pesar de que la Deuda Técnica es un concepto “nuevo”, recientemente es que ha tomado mayor relevancia. En México en la actualidad hay pocos procesos o metodologías que contribuyan a su detección, manejo y seguimiento.

La creación de herramientas que han contribuido a la identificación, manejo y seguimiento se han especificado en la Deuda Técnica en el Código o Arquitectura de un proyecto de software, tales herramientas como: SonarQube [8] o PMD[9] se han implementado en muchas organizaciones, pero hay muy pocas herramientas o métodos dirigidos hacia algún otro tipo de Deuda Técnica y como lo hemos mencionado, la Deuda Técnica en otras actividades como en los Requerimientos de Software, probablemente se le ha dirigido menos atención.

En los tipos de deuda en requerimientos, pruebas, diseño o documentación la atención que han obtenido, en su mayoría ha sido para su definición e identificación de factores que generan la deuda.

Este trabajo y el método que se propone es relevante ya que, busca dar un amplio panorama de qué es la Deuda Técnica, qué es la Deuda Técnica en los Requerimientos y brindar una serie de diversas prácticas para la identificación para procurar que se mantenga estable y llevar su seguimiento, de la Deuda Técnica en los Requerimientos, para que el (o los) proyecto(s) de desarrollo de software que se realizan, satisfagan criterios de calidad establecidos y restricciones iniciales; evitando que pueda ponerse en peligro el ciclo de vida de los proyectos. Además, busca aportar una nueva opción a la investigación de este tipo de Deuda Técnica y llamar la atención para que se sumen otras investigaciones sobre la Deuda Técnica en los Requerimientos .

Este trabajo con base en los procesos que ofrecen algunas normas ISO/IEC, busca construir un método para la gestión de la Deuda Técnica en los Requerimientos del Software, que por el tiempo y la extensión del tema se limitó el alcance solo a la investigación de los requerimientos. En este trabajo se considera que todas las fases y procesos que conforman la creación de un proyecto de software, son relevantes y

1. INTRODUCCIÓN

añadiendo que la “parte práctica” como lo es la arquitectura del código y el código ya han sido explotadas, al menos más que otras se decidió realizar el presente trabajo sobre la Deuda Técnica en los Requerimientos.

En el segundo capítulo se mostrarán los conocimientos teóricos en los que se desarrolla este trabajo de tesis. En este capítulo, se pretende dar una introducción al tema más importante de este trabajo: La Deuda Técnica.

El tercer capítulo abordará el estado actual de la investigación sobre la Deuda Técnica en Requerimientos, donde se definirá este tipo de deuda.

En el cuarto capítulo se desarrollará el método propuesto de gestión de la Deuda Técnica en los Requerimientos, el cual tiene cuatro prácticas: de prevención, detección, manejo y retrospectiva de la deuda.

En el quinto capítulo se mostrarán los comentarios de la evaluación de expertos que revisaron la propuesta del método para la gestión de la Deuda Técnica en los Requerimientos.

Marco teórico

Este capítulo muestra la teoría sobre la cual será desarrollada el método que se propone en el objetivo de este trabajo. Los conceptos presentados son: Deuda Técnica, un breve repaso desde el surgimiento, la evolución del concepto hasta los diferentes tipos de Deuda Técnica; la norma ISO/IEC 29110 en el perfil básico y también, la ISO/IEC 29148.

2.1 Deuda Técnica

El término de Deuda Técnica es un concepto presente en la ingeniería de software, Ward Cunningham, lo utilizó por primera vez como una metáfora de una deuda financiera:

“Cuando se libera código por primera vez lleva implícita una deuda. Un poco de deuda puede acelerar el desarrollo siempre y cuando se pague de inmediato con la refactorización. El peligro ocurre cuando la deuda no se paga. Cada minuto dedicado al código no del todo correcto cuenta como interés en esa deuda” [7].

Desde entonces, el concepto de Deuda Técnica ha evolucionado para abarcar diferentes aspectos del desarrollo de software que van, desde problemas de arquitectura y diseño; hasta problemas de documentación, requerimientos, personas y despliegue [2].

Las siguientes definiciones son algunas que podemos encontrar en la literatura sobre Deuda Técnica :

2. MARCO TEÓRICO

“Está constituida por un código no óptimo u otros artefactos relacionados con el desarrollo de software que brindan beneficios a corto plazo, pero causan un costo adicional a largo plazo durante el ciclo de vida del software” [10].

“Es el costo de refactorizar artefactos de software que no satisfacen los requerimientos de calidad deseados, más el interés incurrido para mantener esos artefactos” [11].

“Es la deuda de trabajo que se adquiere al producir código pobre, incumpliendo prácticas aconsejadas para el desarrollo de software” [12].

“Es la diferencia entre la entrega de un producto planificado o informado y la entrega real de este” [13].

En un seminario, en Dagstuhl [14], un grupo de expertos intentó refinar y acordar una definición de Deuda Técnica :

“En los sistemas intensivos en software, la deuda técnica consiste en construcciones de diseño o implementación que son convenientes a corto plazo, pero establecen un contexto técnico que puede hacer que un cambio futuro sea más costoso o imposible. La deuda técnica es un pasivo contingente cuyo impacto se limita a las cualidades del sistema interno, principalmente la capacidad de mantenimiento y la capacidad de evolución” [14].

2.1.1 Propiedades de la Deuda Técnica.

Brown [15] menciona que para definir el concepto y la caracterización de los tipos de Deuda Técnica, es útil describir una serie de propiedades de la deuda técnica.

Estas propiedades son las siguientes:

- **Visibilidad.**

Cuando la deuda no es visible puede causar problemas de gran importancia en el ciclo de vida del proyecto, es por eso que uno de los propósitos de la investigación sobre la Deuda Técnica se centra en encontrar la manera de lograr que la deuda sea visibilidad y que permita ser considerada en la toma de decisiones.

- **Valor.**
El valor es la diferencia económica entre el sistema tal como está y el sistema en un estado ideal.
- **Valor presente.**
El valor presente es el valor de los costos incurridos como resultado de la deuda. Además, el tiempo y la incertidumbre hasta el impacto.
- **Acrecentamiento.**
El nivel en que la deuda va acrecentando al punto de que esta que pueda llevar a un sistema a peligrar su ciclo de vida.
- **Ambiente.**
La deuda es relativa a un entorno dado o supuesto.
- **Origen.**
Es importante distinguir claramente entre la deuda estratégica, asumida para obtener alguna ventaja, y la deuda no intencional, que se adquiere a través de malas prácticas “más adelante se explican las maneras de contraer una deuda técnica”.
- **Impacto.**
Los elementos donde se encuentra la deuda técnica deben de estar localizados.

2.1.2 Clasificación de la Deuda Técnica

2.1.2.1 Taxonomía de McConnell

Steve McConnell, a partir de la definición de Cunningham, propone una clasificación [16] basada en el modo de incurrir en una Deuda Técnica de dos maneras (Tipo I y Tipo II).

- **Tipo I. Deuda no intencional.**
McConnell, menciona que esta deuda se adquiere involuntariamente y en algunas ocasiones, se puede incurrir sin darse cuenta. “Esta deuda técnica es el resultado no estratégico de hacer un mal trabajo”[16].
- **Tipo II. Deuda intencional.**
Esta deuda se adquiere a propósito, al tomar una decisión previamente ana-

2. MARCO TEÓRICO

lizada por los miembros del equipo y que puede ser momentáneamente, una buena solución a la situación actual de un producto.

McConnell, hace hincapié en el Tipo II de la deuda técnica, cuando esta se piensa saldar en dos periodos de tiempo: a corto y a largo plazo.

- **Tipo II.A. Deuda a corto plazo.**

Esta deuda se contrae de manera táctica, por lo general se adquiere entre brechas cortas de tiempo, como entre la entrega de una iteración y el inicio de una nueva. Siempre se tiene a consideración que debe ser pagada constantemente o de forma rápida.

- **Tipo II.B. Deuda a largo plazo.**

La deuda adquirida a largo plazo se puede mantener por años, pues como se adquirió intencional y estratégicamente, el equipo puede pagarla en este tiempo, priorizando otros asuntos.

2.1.2.2 Clasificación de Fowler

Martin Fowler, clasifica la adquisición de Deuda Técnica en un cuadrante [1], en él explica que la Deuda Técnica puede ser prudente o imprudentemente y se incurre en ella inadvertida o deliberadamente.

- **Deliberada -Imprudente.**

Esta clase de deuda, es la que se adquiere al no seguir buenas prácticas de desarrollo o no tener la debida precaución al escribir código, por el poco tiempo o poca planeación.

- **Inadvertida-Imprudente.**

La clase de Deuda Técnica más costosa causada por el poco conocimiento de buenas prácticas o la falta de experiencia del equipo de desarrollo, lleva a adquirir una deuda de la cual no se tiene idea de su existencia y de que tan inmensa puede llegar a ser.

- **Deliberada-Prudente.**

Es la Deuda Técnica que se adquiere sabiendo que se va a incurrir en ella y vale la pena asumir el riesgo, ya que, el beneficio de endeudarse puede ser benéfico a corto plazo.

Además, el equipo es consciente de que en algún momento tiene que ser saldada.

- **Inadvertida-Prudente.**

Se incurre en una Deuda Técnica de manera inadvertida y prudente, cuando se ha comprendido la manera de resolver un proyecto y se está de acuerdo con la solución.

Al final, se obtienen buenos resultados, pero el equipo de desarrollo se da cuenta que había una manera más adecuada para solucionar el proyecto.

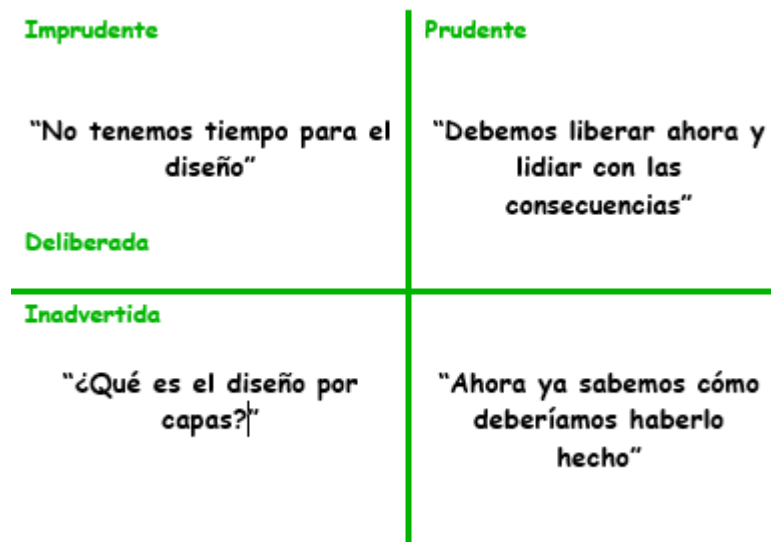


Figura 2.1: Cuadrante de la Deuda Técnica [1]

2.1.2.3 Panorama de Kruchten.

Philippe Kruchten, [2], propone limitar el concepto de Deuda Técnica a elementos invisibles que en su mayoría, es con lo que lidia el equipo de desarrollo internamente o, lo que el cliente no ve (código, documentación, arquitectura, calidad interna, etc.). Cuando la deuda es asociada con temas de calidad y de mantenimiento, estas se vuelven nuevas probabilidades para incurrir en una Deuda Técnica. En la figura 2.2 se muestra "el panorama de la deuda técnica", la organización propuesta por Kruchten, consta de:

- **Elementos visibles.**

2. MARCO TEÓRICO

- Defectos para corregir (Temas de calidad externa).
 - Nuevas funcionalidades para desarrollar (Temas de evolución).
- **Elementos invisibles.** Son los aspectos que puede ser causantes de la deuda, incluyendo aspectos invisibles de la calidad y de evolución interna.

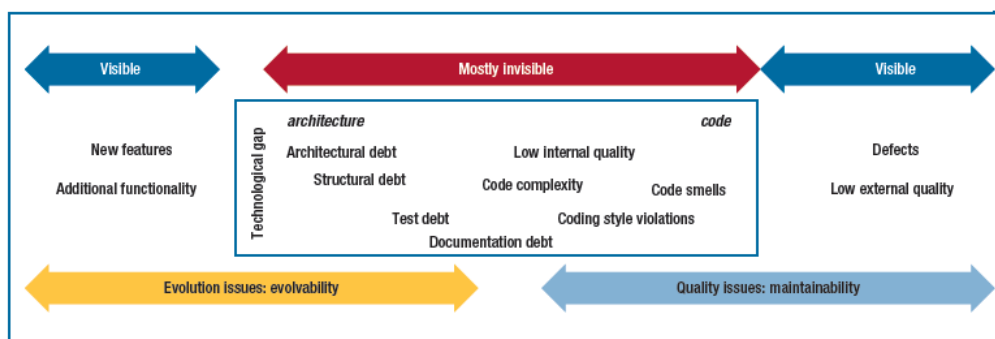


Figura 2.2: Panorama de la Deuda Técnica [2].

2.1.2.4 Clasificación de los tipos de Deuda Técnica.

Conforme se fue desarrollando el término de Deuda Técnica a lo largo de la historia, muchas de las definiciones no clasificaban la deuda por su naturaleza específica. Un grupo de académicos decidió clasificar la deuda técnica por su tipo e indicadores claves para cada tipo.

Como se había mencionado anteriormente, la deuda se asocia a actividades que se realizan durante los procesos de gestión e implementación del proyecto.

En el resultado de Alves [3] se mencionan 13 tipos de Deuda Técnica que se identificaron en el resultado de esta investigación.

Alves, menciona que el trabajo desarrollado, “contribuye al panorama de la deuda técnica mediante la organización de los diferentes tipos de deuda técnica que se han considerado en la literatura técnica. Los tipos identificados se organizaron utilizando una ontología, lo que permitirá compartir un vocabulario común para la comunidad de investigadores en deuda técnica”[3].

El estudio realizado consideraba que los tipos e indicadores no estaban organizados y además, resultaba difícil establecer términos comunes que indicaran la dirección

o el área a tomar para tratar la Deuda Técnica existente, en la que se había incurrido durante el ciclo de vida del proyecto.

1. **Deuda de Arquitectura.** Se refiere a los problemas encontrados en la arquitectura del proyecto. La falta de modularidad es un ejemplo, ya que podría estar afectando a los requerimientos arquitectónicos, implicando una mayor inversión de esfuerzo y tiempo al tener que implementar desarrollos más complejos para su intervención.
2. **Deuda de Compilación.** Se refiere a los problemas relacionados con la compilación, problemas que pueden ser innecesarios, pero que dificultan las tareas a realizar y por lo tanto, se consume mayor tiempo en su compilación.
3. **Deuda del Código.** Se refiere a los problemas que se pueden encontrar en el código fuente o por las malas prácticas de codificación que podrían afectar negativamente, haciendo que se vuelva complicado la examinación de este.
4. **Deuda de Defectos.** Se refiere a los defectos conocidos o desconocidos en el código fuente del proyecto de software, que son identificados por medio de las actividades de pruebas, pero debido a la priorización y recursos limitados se dejan para otro momento.
5. **Deuda en el Diseño.** Es la deuda que se encuentra al momento de analizar el código fuente y se identifican prácticas que violan los principios del buen diseño.
6. **Deuda de la Documentación.** Se refiere a la deuda en la documentación del proyecto de software y puede observarse en la falta de documentación ya sea incompleta o inadecuada, es decir, que no cumpla los criterios de calidad establecidos.
7. **Deuda en la Infraestructura.** Es la deuda que se encuentra en la estructura de una organización de software que podría retrasar u obstaculizar algunas actividades de desarrollo.
8. **Deuda de las Personas.** Es la deuda que se adquiere con las personas y que pueden obstaculizar o retrasar actividades dentro del desarrollo del software.

2. MARCO TEÓRICO

9. **Deuda en los Procesos.** Se refiere a la deuda que se encuentra en los procesos ineficientes o inadecuados.

10. **Deuda en los Requerimientos.** La deuda de requerimientos se refiere a las compensaciones hechas con respecto a los requerimientos que el equipo de desarrollo necesita implementar o cómo implementarlo, pueden ser requerimientos incompletos o mal implementados.

11. **Deuda del Servicio.** Es la deuda que se puede dar cuando existe la necesidad de sustituir el servicio web, ya que este debe ser gestionado, compensado y transformado de ser una obligación a ser un valor agregado.

12. **Deuda en la automatización de pruebas.** Se refiere a la deuda del trabajo involucrado en la automatización de pruebas de funcionalidades previamente desarrolladas para soportar una integración continua y ciclos de desarrollo más rápidos.

13. **Deuda en las Pruebas.** Se refiere a los problemas en las actividades de prueba que pueden afectar la calidad de la misma.

La ontología propuesta, muestra una lista de indicadores que permiten identificar el tipo de deuda en la que se incurre, considerando la naturaleza de estos tipos como criterio de clasificación.

En la figura a continuación, se muestran los 13 tipos de Deuda Técnica con sus indicadores.

2.1 Deuda Técnica

Tipo de Deuda Técnica	Indicador
Deuda de Arquitectura	Problemas en la arquitectura del software. Análisis estructural Dependencias estructurales Violación de la modularidad
Deuda de Compilación	"Banderas muertas" "Objetivos Zombi" Dependencia Visibilidad
Deuda del Código	Métricas de código Código fuera de los estándares Código duplicado Corrección multiproceso (ASA) Algoritmos lentos
Deuda de Defectos	Defectos conocidos no corregidos
Deuda en el Diseño	Método del cerebro Métricas de código Olor en el código Clase de datos Grupos de datos Acoplamiento disperso Código duplicado Clase de Dios (o clase grande) Basura Acoplamiento intensivo Problemas en el diseño del software Análisis estructural
Deuda de la Documentación	Documentación inexistente Especificaciones del diseño incompletas Documentación incompleta Insuficientes comentarios en el código
Deuda en la Infraestructura	-
Deuda de las Personas	-
Deuda en los Procesos	-
Deuda en los Requerimientos	Backlog de requerimientos
Deuda del Servicio	Selección/Reemplazo de servicios web
Deuda en la automatización de pruebas	-
Deuda en las Pruebas	Pruebas incompletas Baja cobertura

Figura 2.3: 13 Tipos de Deuda Técnica y sus indicadores. Traducida de [3].

Por otro lado, Li [4], en su estudio sistemático, a través del análisis de una colección de estudios se logró clasificar la deuda técnica en diez tipos. Esta clasificación está determinada de acuerdo con las fases del ciclo de vida de un proyecto en la que se puede incurrir en la Deuda Técnica.

2. MARCO TEÓRICO

1. **Deuda Técnica en los Requerimientos.** Se refiere a la distancia entre la solución óptima de la especificación de los requerimientos y la solución que se implementó hasta el momento. Esta definición es tomada de Ernst [17].
2. **Deuda Técnica de Arquitectura.** Es la deuda causada por las decisiones en la arquitectura que comprometen algunos aspectos de calidad interna, como la capacidad del mantenimiento.
3. **Deuda Técnica en el Diseño.** Es la deuda que se refiere a los atajos técnicos que se toman en un diseño detallado.
4. **Deuda Técnica del Código.** Es la deuda que se adquiere por el código pobremente escrito que viola las buenas prácticas o reglas.
5. **Deuda Técnica en las Pruebas.** Se refiere a los atajos tomados en la fase de pruebas.
6. **Deuda Técnica de Compilación.** Son las fallas en un sistema de software, en su sistema de compilación o en su proceso, que hacen que la compilación sea demasiado compleja y difícil.
7. **Deuda Técnica de la Documentación.** Se refiere a la insuficiente, incompleta o tardía documentación en cualquier aspecto del desarrollo de software.
8. **Deuda Técnica en la Infraestructura.** Se refiere a una configuración subóptima de procesos relacionados con el desarrollo, tecnologías, herramientas de soporte, que afecta negativamente la capacidad del equipo para realizar un producto de calidad.
9. **Deuda Técnica de Versionamiento.** Se refiere a los problemas de versionamiento en el código fuente.
10. **Deuda Técnica de Defectos.** Es la deuda que se refiere a los defectos, bugs o fallas encontradas en los sistemas de software.

Estos tipos de deuda son los más explícitos y mencionados en los estudios que ayudaron a presentar la clasificación, aunado a esta clasificación Li, muestra la cla-

2.1 Deuda Técnica

sificación de subtipos considerados como causa de cada tipo de deuda.

En la figura 2.4, se muestran los diez tipos de Deudas con sus respectivos subtipos.

		Tipo de Deuda Técnica	Subtipo de Deuda Técnica
D E U D A	T É C N I C A	Requerimientos	Sobre ingeniería.
		Arquitectura	Olor en la arquitectura Anti-patrones arquitectónicos Comportamiento complejo de dependencias arquitectónicas Violación de buenas prácticas de arquitectura Problemas de cumplimiento arquitectónico
		Diseño	Olor en el código Clases o métodos complejos Basura Especificación incompleta del diseño
		Código	Mala calidad del código Código duplicado Código complejo Violaciones en el Código
		Pruebas	Baja cobertura Aplazamiento de las pruebas Falta de automatización de pruebas Defectos no encontrados en las pruebas Pruebas caras Errores de estimación en plan de esfuerzo de pruebas
		Compilación	Malas dependencias Proceso de compilación manual
		Documentación	Documentación atrasada Documentación incompleta Documentación insuficiente Falta de comentarios en el código
		Infraestructura	Uso de tecnología vieja/fuera de uso Uso de herramientas de soporte viejas/fuera de uso Falta de integración continua Planeación pobre de lanzamientos
		Versionamiento	Bifurcaciones innecesarias de código Soporte de multi-versión
		Defecto	Defectos/bugs

Figura 2.4: 10 tipos de Deuda Técnica según el estudio sistemático de Li [4]. Traducida.

2. MARCO TEÓRICO

2.1.3 Manejo de la Deuda Técnica

El manejo de la Deuda Técnica se comprende a las actividades que se realizan para evitar que “se salga de control” manteniéndola en un nivel razonable. En esta sección, se muestran algunas actividades que se implementan para el manejo de la Deuda Técnica.

En [4], se menciona un conjunto de las actividades que buscan o prevén que un equipo incurra en una deuda o buscan lidiar con la deuda ya existente y manejarla para mantenerla en términos razonables.

Estas actividades son las siguientes:

- **Identificación.** Identificar la deuda causada por decisiones técnicas intencionales o no intencionales en un proyecto.
- **Medición.** El propósito de esta actividad es cuantificar el beneficio y el costo de la deuda existente en un proyecto por medio de diferentes técnicas de estimación.
- **Priorización.** Clasificar la deuda señalando qué se necesita "pagar primero" para que la deuda no aumente y qué se debe “pagar después” en tanto, no afecte y se pueda tolerar hasta versiones posteriores.
- **Prevención.** Tiene como objetivo evitar que la deuda siga creciendo potencialmente.
- **Monitoreo.** Es la actividad encargada de observar los cambios en el costo y beneficio de las actividades extraordinarias de la deuda.
- **Reembolso.** Resolver o mitigar la deuda en un proyecto mediante técnicas como reingeniería o refactorización.
- **Representación/Documentación.** Proporcionar la manera de representar y codificar la deuda de manera uniforme abordando las preocupaciones de las partes interesadas.
- **Comunicación.** El objetivo de esta actividad es hacer la deuda visible para el equipo y las partes interesadas y que se puedan tomar decisiones más conscientes.

En esta guía se identifican diez áreas de conocimiento en las cuales los procesos son categorizados.

2.2 Requerimientos del Software

Cuando se desea realizar un desarrollo de software se deberían establecer claramente las necesidades que se buscan solucionar para así, encontrar una o más ma-

neras de proponer un resultado satisfactorio en la realización de este, es por lo que los requerimientos de software son una parte fundamental en el proceso de desarrollo.

2.2.1 Ingeniería de Requerimientos

Antes de definir y describir los requerimientos es importante conocer que la ingeniería de software, es la encargada de comprender las necesidades reales y satisfacerlas en un producto de software, es decir, se encarga de las tareas asociadas a la determinación de estas necesidades.

La ingeniería de requerimientos se define como: *“Una función interdisciplinaria media entre los dominios del adquirente y el proveedor para establecer y mantener los requerimientos que debe cumplir el sistema”*. [18]

Zave [19] define la ingeniería de requerimientos como “una rama de la ingeniería de software que se ocupa de los objetivos del mundo real, las funciones y las limitaciones de los sistemas de software. También se preocupa por la relación de estos factores con las especificaciones precisas del comportamiento del software y con su evolución a lo largo del tiempo y entre familias de software”

A su vez, Leite [20] propone que la ingeniería de requerimientos es “el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo dque genera un documento de requerimientos.”

Es interesante que las definiciones incluyen a actores que están involucrados, objetivos, recopilar, sistema y requerimientos. A través de la recopilación de los objetivos, los involucrados pueden obtener los requerimientos para modelar un sistema que satisfaga las necesidades que se plantean. Por lo tanto, los requerimientos son fundamentales en un proyecto.

2.2.2 Los Requerimientos

La definición tomada del SWEBOK [21] dice lo siguiente:

2. MARCO TEÓRICO

“Los requerimientos del software expresan las necesidades y los apremios colocados en un producto de software que contribuye a la solución de un cierto problema del mundo real” [21].

También, la IEEE en [22] define a los requerimientos de la siguiente manera:

“como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado”.

Entre los requerimientos existe una importante distinción al momento de establecerse, pues es cuando se establecen los requerimientos del producto de software y cuando se establecen las restricciones sobre el desarrollo.

Así, los requerimientos se catalogan en dos tipos (Requerimientos Funcionales y Requerimientos No Funcionales) y su definición es la siguiente:

Requerimientos Funcionales (RF): “Es la declaración y descripción de los servicios o funciones que el software va a ejecutar.”[21, 23]

En [24] los RF se definen como: “Un subconjunto de los requerimientos del usuario. Requisitos que describen lo que debe hacer el software, en términos de tareas y servicios” .

En cambio, Somerville menciona que los requerimientos del software “ son declaraciones de servicios que el sistema debería proporcionar, cómo debería reaccionar el sistema a entradas particulares y cómo debería comportarse el sistema en situaciones particulares” [25].

Y la ISO/IEC 29148 [18] menciona que “los requerimientos funcionales describen el sistema, las funciones y tareas de los elementos del sistema a realizar.”

Requerimientos No Funcionales (RNF): “Son las restricciones de los servicios o funciones ofrecidas por el software que actúan para obligar la solución. También, se reconocen como los requerimientos de calidad.”[21, 23]

Los RF y RNF son diferentes en esencia, pero a su vez complementarios. Dada una descripción del problema, por ejemplo:

- el tipo de usuarios a los que va dirigido,

2.2 Requerimientos del Software

- las funcionalidades que se van a implementar o
- el tipo de software a desarrollar

Este tipo de necesidades definen los Requerimientos Funcionales y No Funcionales del sistema y la ingeniería de requerimientos es la encargada de recopilarlos.

Los requerimientos deberían ser descritos detalladamente, con la finalidad de ser lo más explícitos posibles para su desarrollo y a manera de prevención para que sean lo menos ambiguos o imprecisos.

2.2.3 SWEBOK

La definición del área del conocimiento de los requerimientos del software según el SWEBOK es la siguiente:

“Se refiere al análisis, a la especificación, y a la validación de los requerimientos del software” [23].

El área del conocimiento los requerimientos del software se detealla en siete fases que se describirán en el siguiente bloque.

2.2.3.1 Proceso de los requerimientos.

Este proceso establece la configuración de las actividades sobre cómo analizar, especificar y validar, los diferentes tipos de proyectos.

Entre las actividades importantes del proceso se establecen los agentes involucrados y su función dentro del mismo, como lo son los usuarios del software, los clientes, analistas de mercado, reguladores e ingenieros de software. Además, se introducen recursos de la gerencia requeridos y consumidos durante el proceso, también se procesan los requerimientos para establecer costos y puntualidad del producto, en busca de mejorar el proceso y la calidad de este.[21, 23]

2.2.3.2 Captura de los requerimientos.

Parte esencial del área del conocimiento de los requerimientos del software, es la recolección de estos; cómo se presentan, identifican y extraen los requerimientos de

2. MARCO TEÓRICO

un proyecto.

Previo a la identificación de las fuentes de los requerimientos, los ingenieros de software se dedican a la intensiva tarea de identificación y extracción de los requerimientos, una tarea de cuidado y de entendimiento; basada en una buena comunicación e ingeniería de software. La habilidad para extraer la información de las fuentes de los requerimientos, a veces resulta ser clave, pues muchas veces es complejo para la fuente (interesados o clientes) describir o expresar las tareas que se convertirán en los requerimientos y puede haber mucha información vaga [21, 23].

Para esto, existen diversas maneras de recolectar los requerimientos del software, llamadas técnicas de levantamiento o captura. Algunas de estas técnicas son las siguientes:

- Entrevistas.
- Prototipos.
- Escenarios.
- Reuniones.
- Observación.

2.2.3.3 Análisis de requerimientos.

La descripción exacta de los requerimientos debe permitir que estos puedan ser validados, implementados y que su costo se pueda estimar.

Analizar los requerimientos lleva consigo los objetivos de identificar y resolver los conflictos, con esto también descubrir los límites del software, podríamos decir que, analizar los requerimientos es tomarlos uno a uno para darles una clasificación e identificar si son requerimientos funcionales o no funcionales; si un requerimiento abarca varios requerimientos o si alguno es derivado de otro de más alto nivel y también, la prioridad y el alcance del requerimientos.

El entendimiento del problema como parte fundamental antes de iniciar con la propuesta de la solución, es influido por la naturaleza del problema, sujeto a la maestría del ingeniero de software y a las limitaciones del cliente, pero pudiera ser vital crear el contexto conceptual del software para establecer una conexión entre el software previsto y su ambiente externo [21].

2.2.3.4 Especificación de los requerimientos.

La especificación de requerimientos del software se refiere a establecer en un documento físico o electrónico, las exigencias de este.

En este caso se despliega una serie de tres documentos:

- Documento de la definición del sistema: Enumera los requerimientos del sistema junto con información de fondo con los objetivos del sistema, misión y requerimientos no funcionales.
- Especificación de requerimientos del sistema: Especifica la visión, requerimientos del sistema y los requerimientos del software derivados de los requerimientos del sistema.
- Especificación de requerimientos del software: Establece la base para el acuerdo entre el cliente y el proveedor. Se establece la base también, de lo que hay que hacer en el software, siendo realista y que permita estimar costos y riesgos.

2.2.3.5 Validación de requerimientos.

La validación de los requerimientos se puede realizar para verificar que estos son comprensibles, constantes y finitos, indicando los procesos de examinación de los documentos de requerimientos para constatar que estos definen al software como se espera, estos procesos de validación pueden ser de inspección, revisión o de prototipado, en los cuales se asigna a un grupo para identificar fallas, problemas o poca claridad; en busca de soluciones, nuevas propuestas e incluso, nuevos requerimientos. Por esto es importante realizar estas pruebas de aceptación de los requerimientos.

2.2.3.6 Consideraciones Prácticas.

“Está extensamente reconocido dentro de la industria del software que los proyectos de la ingeniería de software son críticamente vulnerables cuando estas actividades (análisis, especificación y validación de los requerimientos de software) se realizan mal”[21].

Es muy mencionado que una cultura en la documentación de los requerimientos, cuando la organización de desarrollo esta en crecimiento es la llave del éxito de cualquier proceso.

El entendimiento de los requerimientos es un constante refinamiento de estos, durante el ciclo de vida del proyecto, esto nos lleva a incurrir en cambios en los

2. MARCO TEÓRICO

requerimientos y esta es una manera de entenderlos y reconocer que, el cambio es inevitable por cualquier razón que pueda impactarlos.

El proceso de los requerimientos no es al inicio, o una actividad previa del desarrollo, sino es una actividad constante e iterativa en el ciclo de vida del software.

2.2.4 ISO/IEC 29148-2011.

La ISO/IEC 29148 [18] es un compendio de normas con el objetivo de brindar una guía para el manejo de procesos y productos pertenecientes a la ingeniería de requerimientos durante el ciclo de vida de un producto de software.

Se dice que es un compendio de varias normas ya que hace una “armonización” de varias fuentes, en su mayoría normas ISO:

- ISO/IEC 12207:2008 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes.
- ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes.
- ISO/IEC/IEEE 15289:2011, Systems and software engineering — Content of life-cycle information products (documentation).
- ISO/IEC TR 19759, Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK).
- IEEE Std 830, IEEE Recommended Practice for Software Requirements Specifications .
- IEEE Std 1233, IEEE Guide for Developing System Requirements Specifications.
- IEEE Std 1362, IEEE Guide for Information Technology — System Definition — Concept of Operations (ConOps) Document.
- ISO/IEC TR 24748-1, Systems and software engineering — Life cycle management — Part 1: Guide for life cycle management.
- ISO/IEC/IEEE 24765, Systems and software engineering — Vocabulary.

2.2.4.1 Conceptos

En esta sección de la ISO/IEC 29148-2011 se presentan los conceptos que se aplican a los requerimientos y a sus propiedades y también, a los procesos de obtención,

análisis, asignación, documentación, gestión y sus elementos de información generados durante estos.

Fundamentos de los requerimientos

■ **General.**

Es la ingeniería de requerimientos la encargada de que estos de las partes interesadas se puedan descubrir, obtener, desarrollar, analizar, verificar, validar, comunicar y gestionar, de manera que el resultado sea una jerarquía de requerimientos que se puedan implementar, proporcionando una base para verificar diseños y aceptar soluciones, esta puede ser representada en una o más especificaciones de requerimientos.

■ **Interesados.**

Son las partes interesadas y varían según el proyecto, el conjunto mínimo de interesados se constituye de usuarios y compradores, y los requerimientos del proyecto pueden requerir de la inclusión de dos o más grupos de interesados y la organización que desarrolla mantiene y opera el sistema y las autoridades reguladoras.

■ **Transformación de las necesidades a requerimientos.**

En esta sección se habla de las intenciones de las partes interesadas en forma de necesidades, metas u objetivos, que aún no son requerimientos pues no se han definido ni analizado su consistencia o viabilidad. Para ayudar con la comprensión de estas intenciones se hace uso del concepto de operaciones y del concepto operativo del sistema, así la ingeniería lleva estas intenciones a declaraciones formales y mejor estructuradas que se transforman en requerimientos del sistema.

■ **Construcción de requerimientos.**

Un requerimiento bien formado es una declaración que:

- puede ser verificada,
- debe ser implementado por un sistema,
- debe resolver un problema específico,
- debe ser calificada por condiciones medibles y

2. MARCO TEÓRICO

- debe ser limitado por restricciones.

En este estándar un requerimiento es *“una declaración que traduce o expresa una necesidad y sus restricciones y condiciones asociadas”*. [18]

Esta declaración puede tomar forma de lenguaje natural, por lo tanto debe contener un sujeto, un verbo y un complemento y a esto podría sumarse una condición.

Una condición es *“un atributo cuantitativo o cualitativo medible que se estipula para un requerimiento. Además, califica un requerimiento que se necesita y proporciona atributos que permite que un requerimiento se formule y establezca de una manera que puede validarse y verificarse”*. [18]

■ Características de los requerimientos individuales.

Cada requerimiento por sí mismo deberá de poseer las siguientes características:

1. **Necesario.** El requerimiento define una capacidad esencial, característica, restricción y/o factor de calidad, si se llega a eliminar no podría ser satisfecha por alguna otra capacidad del producto.
2. **Implementación libre.** El requerimiento debe establecer qué es lo que se requiere y no cómo se debe cumplir.
3. **No ambiguo.** El requerimiento se establece de forma simple y fácil de entender y que se pueda implementar de una sola manera.
4. **Consistente.** El requerimiento está libre de conflictos con otros requerimientos.
5. **Completo.** El requerimiento establecido no necesita más ampliación.
6. **Singular.** La declaración de requerimientos incluye un solo requerimiento, sin uso de conjunciones.
7. **Factible.** El requerimiento es alcanzable.
8. **Trazable.** El requerimiento puede ser rastreado hacia arriba (declaraciones de necesidades documentadas) o hacia abajo (requerimientos especí-

ficos).

9. **Verificable.** El requerimiento tiene los medios para demostrar que el sistema cumple el requerimiento especificado.

■ Características de un conjunto de requerimientos.

Existen ciertas características que deberían cumplirse para garantizar que el conjunto de requerimientos cumplirá con una solución factible, por lo tanto, cada conjunto de requerimientos deberá tener las siguientes características:

1. **Completo.** El conjunto de requerimientos no necesita alguna ampliación para cumplir con una tarea específica.
2. **Consistente.** En el conjunto no existen contradicciones entre los requerimientos individuales.
3. **Asequible.** El conjunto de requerimientos puede satisfacerse con una solución que sea factible dentro de las limitaciones del ciclo de vida del proyecto.
4. **Acotado.** El conjunto de requerimientos mantiene el alcance identificado y no va más allá de lo necesario para cumplir con las necesidades del usuario.

■ Criterios de lenguaje de los requerimientos.

Los requerimientos deben expresar necesidades para que el sistema de interés sea la solución a éstas. Por lo tanto, los requerimientos deben de ser el “qué”, se deben evitar los términos vagos y generales, es decir se prefiere evitar requerimientos que sean difíciles o imposibles de verificar o que puedan tener más de una interpretación. He aquí una lista de términos que suelen ser ambiguos o generales:

1. **Superlativos.** Ej. mejor, más.
2. **Lenguaje subjetivo.** Ej. fácil de usar, rentable.

2. MARCO TEÓRICO

3. **Pronombres vagos.** Ej. eso, esto.
4. **Adverbios o adjetivos ambiguos.** Ej. casi siempre, minimal.
5. **Términos abiertos o no verificables.** Ej. como mínimo, proporcionar soporte.
6. **Frases comparativas.** Ej. mejor que, mayor calidad.
7. **Lagunas o vacíos.** Ej. sí es posible, según corresponda.
8. **Referencias incompletas.** Ej. no especificar fechas o versiones.
9. **Declaraciones negativas.** Ej. que el sistema no haga.

■ **Atributos de los requerimientos.**

Para que los requerimientos estén “bien formados” deben de tener atributos definidos que ayuden a su comprensión y gestión. Algunos ejemplos de atributos que podrían tener los requerimientos son los siguientes:

1. **Identificación.** Los requerimientos deben ser identificados únicamente, es decir, darles un número, una etiqueta o un nombre. Al darles un identificador único se fija una relación o vínculo con la finalidad de poder rastrearlos o poder acceder fácilmente al requerimiento
2. **Prioridad del interesados.** Esta propiedad se puede establecer mediante un consenso con los interesados. Es preciso establecer una escala o esquema para asignar a cada requerimiento una prioridad.
3. **Dependencia.** Definir o establecer explícitamente la dependencia entre requerimientos, con la finalidad de identificar cuales son esenciales para el sistema y en caso de cambio o eliminación de un requerimiento primario, poder eliminar o cambiar los que dependen de este.
4. **Riesgo.** Las técnicas de análisis de riesgos se pueden usar para determinar una calificación para los requerimientos, en término de sus consecuencias o grado para evitar riesgos.

2.2.4.2 Procesos.

Proceso de requerimientos.

En el presente apartado se proporciona al usuario orientación adicional sobre la planificación e implementación de procesos relacionados con los requerimientos. Además, en esta sección se esclarece la notación que se utilizará y también, se hace hincapié en la utilización de la ISO/IEC 15288:2008 (IEEE Std 15288-2008) [26]. En esta sección se retoman en su totalidad dos de sus procesos: el proceso de la definición de los requerimientos de los interesados y el proceso de análisis de requerimientos. En conjunto, estos dos procesos brindan un grupo básico de requerimientos que tienen como destino, ser la entrada al proceso del diseño de la arquitectura, pues los requerimientos se asignan, descomponen y rastrean los elementos del sistema.

Proceso de la definición de los requerimientos de los interesados.

El propósito del proceso de la definición de los requerimientos de los interesados es definir los requerimientos para un sistema que puede proporcionar servicios necesarios para usuarios y otros interesados en un entorno definido.

Los resultados que se pretenden obtener, son los siguientes:

- Especificación de las características requeridas del sistema y el contexto de uso de las funciones y servicios del producto.
- Definir restricciones en una solución al sistema.
- Lograr la trazabilidad de los requerimientos de los interesados.
- Definir los requerimientos de los interesados.
- Se identifican los requerimientos de las partes interesadas para su validación.

Durante el proyecto se implementarán las siguientes tareas y actividades de acuerdo con las políticas y procedimientos aplicables de la organización con respecto al proceso de definición de requerimientos de los interesados.

- **Obtener requerimientos de los interesados.**

2. MARCO TEÓRICO

1. **Identificar a las partes interesadas individuales o a las clases de partes interesadas que tienen un interés legítimo en el sistema a lo largo del ciclo de vida.** Se menciona que es preciso identificar todas las clases de partes interesadas para identificar después a las partes interesadas que tienen una fuerte influencia sobre los objetivos, estrategias, operaciones y el sistema. La información que se obtenga de cualquiera de las partes interesadas dependerá del rol, responsabilidad y su posición, por lo tanto, es importante identificar a las partes interesadas. Para obtener una visión real del problema a resolver, sería interesante recopilar información de todas las clases o niveles de partes interesadas.
2. **Obtener los requerimientos de las partes interesadas directamente de estas.** Como se menciona anteriormente es importante identificar todas las fuentes potenciales de requerimientos y recibir información directamente de estas fuentes.
Al momento de recibir información de estas fuentes se deben abordar puntos como: objetivos, escenarios, perfil de la misión, entornos y contextos, despliegue operativo, rendimiento, efectividad y el ciclo de vida operacional.

Definir los requerimientos de los interesados.

1. **Definir las restricciones en una solución del sistema que son consecuencias inevitables de los acuerdos existentes, las decisiones de gestión y las decisiones técnicas.** Las restricciones son un tipo de requerimientos y estas pueden ser impuestas por interesados externos, otros sistemas, actividades técnicas (transición o mantenimiento) o medidas de efectividad.
2. **Definir un conjunto representativo de secuencias de actividades para identificar todos los servicios que corresponden a escenarios y entornos operativos y de soporte anticipado.** Plantear o proponer escenarios puede servir para identificar los requerimientos que de otro modo no podrían ser fáciles de identificar.
3. **Identificar la interacción usuario-sistema.** Determinar los requerimientos de usabilidad estableciendo el desempeño humano y la interacción humano-sistema, considerando las capacidades humanas y las limitaciones de habilidades, definiendo: capacidades (mentales o aprendidas), entorno, condiciones

normales, inusuales y de emergencia y reclutamiento, capacitación y cultura.

4. Analizar y mantener los requerimientos de los interesados.

- **Analizar el conjunto completo de los requerimientos.** Analizar los requerimientos basado en las cláusulas: características de los requerimientos individuales y características de un conjunto de requerimientos. Además, identificar y priorizar los que sea conflictivos, faltantes, incompletos, ambiguos, inconsistentes, incongruentes o inverificables.
- **Resolución de problemas de los requerimientos.** Los conflictos en los requerimientos ocurren constantemente, por esa razón es importante identificarlos para establecer negociaciones con las partes interesadas, para proponer una resolución a estos.
- **Retroalimentación de los requerimientos analizados a los interesados para garantizar que las necesidades y expectativas se hayan capturado y expresado correctamente.** Explicar y establecer un acuerdo sobre las propuestas para resolver los requerimientos de los interesados, en los que se presentan conflictos.
- **Establecer con los interesados que los requerimientos estén expresados correctamente.** Confirmar que los requerimientos de los interesados son comprensibles para los creadores y en el caso de haber tenido conflictos, la solución o acuerdo propuestos no comprometen las intenciones iniciales.
- **Registrar los requerimientos de los interesados de manera adecuada para su gestión durante el ciclo de vida y más allá.** Los registros serán la base para la trazabilidad de los requerimientos del sistema y pueden ser una fuente confiable de conocimiento para los requerimientos de sistemas posteriores.
- **Mantener la trazabilidad de los requerimientos de los interesados a las fuentes de necesidades de los interesados.** La trazabilidad se debe mantener para documentar cómo los requerimientos están destinados a cumplir con los objetivos de los interesados.

2. MARCO TEÓRICO

Proceso de análisis de requerimientos

El propósito del proceso de análisis de requerimientos es la transformación de la visión de los servicios deseados, basada en los requerimientos de los interesados a una vista técnica de un producto requerido que ofrezca estos servicios.

Los resultados esperados son los siguientes:

- Especificación de las características, atributos, requerimientos funcionales y de rendimiento para la solución del producto.
- Especificación de las restricciones que afectan el diseño arquitectónico del sistema y los medios para realizarlo.
- Integridad y trazabilidad de los requerimientos del sistema a los requerimientos de los interesados.
- Definición de una base para verificar que se cumplan los requerimientos del sistema.

Durante el proyecto se implementarán las siguientes tareas y actividades [26].

- **Definir los requerimientos del sistema**
 1. **Definir el límite funcional del sistema en términos del comportamiento y propiedades dadas.** Se deben establecer condiciones de alcance o límite del sistema con los interesados antes de la definición de los requerimientos.
 2. **Definir todas las funciones del sistema que son fundamentales.**
 3. **Definir las restricciones de implementación necesarias que se introducen por los requerimientos de los interesados.** Es preciso validar las restricciones con los interesados y garantizar que sean correctamente expresadas y entendidas, antes del desarrollo de los requerimientos y el diseño arquitectónico.
 4. **Definir medidas técnicas y de calidad que permitan evaluar logros técnicos.** Establecer parámetros de desempeño asociados a las me-

didas de efectividad identificadas en los requerimientos de los interesados.

5. **Especificar los requerimientos y funciones del sistema, según la identificación de riesgos o criticidad del sistema.**

- **Analizar y mantener los requerimientos del sistema.**

1. **Analizar la integridad de los requerimientos del sistema para asegurar que cada requerimiento o cada conjunto de requerimientos tengan la integridad general.** Asegurarse que los requerimientos cumplen con las cláusulas de características de los requerimientos individuales y características de un conjunto de requerimientos.
2. **Realizar una retroalimentación de los requerimientos analizados a los interesados para garantizar que los requerimientos del sistema reflejen adecuadamente las necesidades y expectativas de los interesados.** Esta retroalimentación asegurará que los requerimientos de los interesados han evolucionado correctamente a requerimientos del sistema.
3. **Demstrar la trazabilidad entre los requerimientos del sistema y los requerimientos de los interesados.** Mantener la trazabilidad entre requerimientos de los interesados y los requerimientos del sistema para asegurar que, los requerimientos de los interesados cumplen uno o más requerimientos del sistema.
4. **Mantener durante el ciclo de vida del sistema el conjunto de requerimientos del sistema con su correspondiente justificación, decisiones y supuestos asociados.** Tener presente ante cualquier situación el conjunto de requerimientos que facilitará la solución a cualquier conflicto que involucre a los requerimientos del sistema.

Actividades de ingeniería de requerimientos en otros procesos técnicos:

- **Requerimientos en el diseño de la arquitectura.** En esta sección se explica la correlación entre los requerimientos y la arquitectura, ya que la solución al diseño de la arquitectura está dada en términos de los requerimientos.

2. MARCO TEÓRICO

- **Requerimientos en verificación.** Se confirma que los requerimientos del diseño especificado del sistema se hayan cumplido. Para su comprobación se recomienda definir y documentar un plan de verificación basado en los requerimientos del sistema, el método de verificación bien definido aborda algunas o todas las consideraciones del cómo, dónde, cuándo y quién, para producir información objetiva y poder demostrar la satisfacción del requerimiento y con cuatro métodos estándares de verificación para obtener evidencia objetiva de que los requerimientos se han cumplido:
 1. Inspección. Una revisión a contra documento para confirmar el cumplimiento del requerimiento. Generalmente usado para verificar propiedades observables.
 2. Análisis. Se incluye un uso de datos analíticos como simulaciones definidas para mostrar el cumplimiento de lo especificado.
 3. Demostración. Exhibición generalmente cualitativa realizada sin instrumentación o equipo de prueba mínimo, se hacen observaciones y se comparan respuestas predeterminadas.
 4. Prueba. Acción en la cual la operabilidad, capacidad de soporte o rendimiento de un elemento se verifica cuantitativamente sometido a condiciones reales o simuladas.

- **Requerimientos en validación.** Se proporciona evidencia objetiva de que los servicios que presta un sistema en uso, cumple con los requerimientos de las partes interesadas. Esta validación está dada generalmente por una autoridad fuerte dentro del proyecto y las partes interesadas esperan la confirmación de que los requerimientos reflejan adecuadamente las necesidades de estas.

Gestión de requerimientos

La gestión de requerimientos abarca aquellas tareas que registran y mantienen los requerimientos en evolución y el contexto asociado, así como la información histórica de las actividades de la ingeniería de requerimientos. Además, establece procesos para definir, controlar y publicar los requerimientos para su disponibilidad en todos los niveles del sistema de interés.

Se menciona que raramente los requerimientos son estáticos y a su vez, muchos de ellos pueden evolucionar por razones diversas como su contexto o conforme avanza el ciclo de vida, por eso es importante identificar y comunicar estos requerimientos ya sea con las partes interesadas como con la comunidad técnica, puesto que, la comprensión de que los requerimientos continúa evolucionando es una parte crucial en todo el ciclo de vida del proyecto.

En un proyecto típico la ingeniería en requerimientos está presente desde la obtención hasta la gestión del cambio de los requerimientos.

2.2.4.3 Olor en los requerimientos.

En [27] se define el *Olor en los Requerimientos* (“*requirements smell*”) como “*un indicador de una violación a la calidad, que nos puede llevar a un defecto, con una ubicación concreta y un mecanismo de detección concreto*”.

Este término introducido en [27] , deriva de la aplicación en los requerimientos, del término de “olores en el código” siendo estos una indicación imprecisa de la mala calidad que puede tener el código; también deriva de la ISO/IEC 29148 de la que previamente hemos hablado, ya que proporciona al usuario una lista de criterios de lenguaje de los requerimientos donde se busca la adecuada elección del lenguaje para los artefactos de los requerimientos.

Para definir los olores en los requerimientos el autor toma la sección de la ISO/IEC 29148 [18] Criterios de lenguaje de los requerimientos.

Los olores se definen de la siguiente manera:

Nombre del olor: Lenguaje subjetivo.

1. **Entidad:** Palabra.
2. **Explicación:** El lenguaje subjetivo se refiere a palabras cuya semántica no está definida objetivamente.
3. **Ejemplo:** La arquitectura y la programación deben garantizar una mantenibilidad **simple y eficiente**.

2. MARCO TEÓRICO

2.3 ISO/IEC 29110. Perfil Básico.

La ISO/IEC 29110 [5] es un conjunto de normas que tiene como propósito proporcionar una guía para que las pequeñas organizaciones puedan ofrecer productos y/o servicios de mejor calidad, así como mejorar el desempeño de sus procesos.

Una Pequeña Organización (PO) es una entidad conformada por hasta 25 personas. Además, según la Organización para la Cooperación Económica y Desarrollo, las pequeñas y medianas empresas constituyen el tipo de organizaciones de negocio dominante en el mundo, por lo tanto, el reconocimiento de las PO como proveedores de software de alta calidad es requerido de manera frecuente.

2.3.1 La ISO/IEC 29110 y sus partes.

La ISO/IEC 29110 está formada de cinco partes las cuales van dirigidas a una audiencia objetivo y además, cada parte cumple diferentes funciones.

ISO/IEC 29110	Título	Audiencia Objetivo
Parte 1	Visión general	PO, evaluadores, productores de estándares, vendedores de herramientas y vendedores de metodologías.
Parte 2	Marco de trabajo y taxonomía	Productores de estándares, vendedores de herramientas y vendedores de metodologías. No dirigidas a PO.
Parte 3	Guía de evaluación	Evaluadores y PO
Parte 4	Especificaciones del perfil	Productores de estándares, vendedores de herramientas y vendedores de metodologías. No dirigidas a PO
Parte 5	Guía de gestión e Ingeniería	PO

Figura 2.5: Partes de la ISO/IEC 29110 [5]

2.3.1.1 Descripción y funciones de la ISO/IEC 29110.

2.3.1.1.1 La ISO/IEC 29110-1.

- Definir los términos de negocio comunes al Conjunto de Documentos del Perfil de la PO.
- Introducir los conceptos de procesos, ciclo de vida y normalización, y la serie ISO/IEC 29110.
- Presentar las características y requerimientos de una PO.

2.3 ISO/IEC 29110. Perfil Básico.

- Aclarar los fundamentos para los perfiles, documentos, estándares y guías de una PO específica.

2.3.1.1.2 La ISO/IEC 29110-2.

- Introducir los conceptos para el perfil normalizado de ingeniería de software para las PO.
- Definir los términos comunes para el Conjunto de Documentos del Perfil de las PO.
- Establecer la lógica detrás de la definición y aplicación de perfiles normalizados.
- Especificar los elementos comunes para todos los perfiles normalizados (estructura, conformidad, evaluación).
- Introducir la taxonomía de perfiles de la ISO/IEC 29110.

2.3.1.1.3 La ISO/IEC 29110-3.

- Definir los lineamientos y requerimientos de conformidad de la evaluación de proceso, necesarios para alcanzar el propósito de los perfiles de la PO definidos.

2.3.1.1.4 La ISO/IEC 29110-4-m.

- Proveer la especificación para todos los perfiles del Grupo de Perfiles Genérico.

2.3.1.1.5 La ISO/IEC 29110-5-m-n.

- Proveer una guía de implementación sobre gestión e ingeniería para el Perfil Básico del Grupo del Perfil Genérico especificado en la ISO/IEC 29110 Parte 4-1.

2. MARCO TEÓRICO

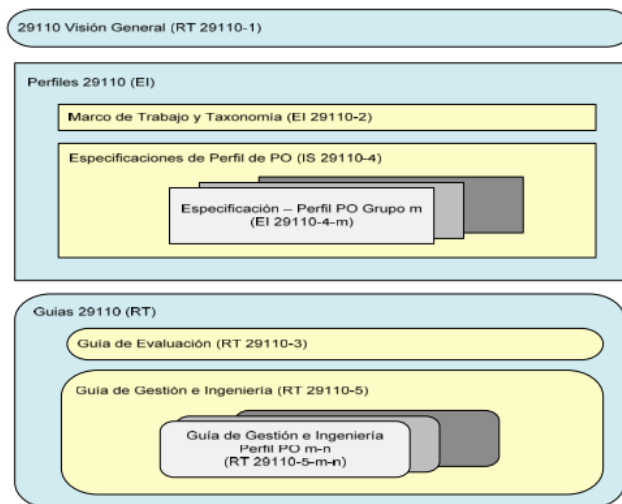


Figura 2.6: Serie ISO/IEC 29110 [5]

2.3.2 ISO/IEC 29110-5-1-2.

En la tabla 2.1, las partes de las ISO/IEC 29110, la parte cuatro es la encargada de brindar la especificación para los perfiles del grupo de perfiles genérico, es importante mencionar que el grupo consta de cuatro perfiles: entrada, básico, intermedio y avanzado. En la parte cinco se menciona que esta será la parte encargada de implementar lo que provee la parte cuatro, por lo tanto, la ISO/IEC 29110-5-1-2 es la guía de implementación de ingeniería del perfil básico.

Al usar esta guía, la PO puede obtener los siguientes beneficios:

1. Un conjunto acordado de requerimientos del proyecto y productos esperados es entregado al cliente.
2. Un proceso de gestión disciplinado que proporciona visibilidad y acciones correctivas sobre los problemas y desviaciones del proyecto es realizado.
3. Un proceso sistemático de implementación de Software que satisfaga las necesidades del cliente y asegure la calidad de los productos.

Para implementar la guía y poder obtener estos beneficios, la pequeña organización deberá cumplir con las siguientes condiciones de entrada:

1. El enunciado de trabajo del proyecto está documentado.

2. La viabilidad del proyecto fue realizada antes de su inicio.
3. El equipo del proyecto, incluyendo el gestor del proyecto, está asignado y entrenado; y los bienes, servicios e infraestructura para iniciar el proyecto están disponibles.

2.3.2.1 Procesos de la ISO/IEC 29110-5-1-2.

El perfil básico de la ISO/IEC consta de dos procesos: Gestión del Proyecto e Implementación del Software. Véase en la figura 2.7.

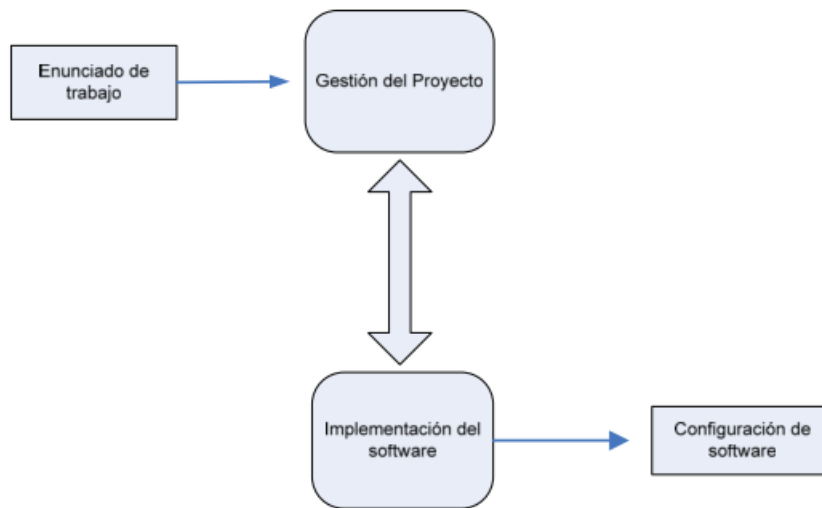


Figura 2.7: Procesos del Perfil Básico [5]

2.3.2.1.1 Gestión del Proyecto.

“El propósito de este proceso es establecer y llevar de manera sistemática las tareas del proyecto de implementación del software, permitiendo así, satisfacer los objetivos de este en cuestiones de calidad, tiempo y dinero deseados”[5]

Los objetivos particulares del proceso de Gestión del Proyecto son los siguientes:

- GP.O1. El Plan del Proyecto para la ejecución del proyecto es desarrollado de acuerdo al Enunciado de Trabajo y revisado y aceptado por el cliente. Las

2. MARCO TEÓRICO

tareas y los recursos necesarios para completar el trabajo son dimensionados y estimados.

- GP.O2. El avance del proyecto es monitoreado contra el Plan del Proyecto y registrados en el registro de estado del avance. Las correcciones para resolver los problemas y desviaciones respecto del plan son realizadas cuando los objetivos del proyecto no son logrados. El cierre del proyecto es ejecutado para conseguir la aceptación documentada del cliente en el Documento de Aceptación.
- GP.O3. Las Solicitudes de Cambio son atendidas mediante su recepción y análisis. Los cambios a los requerimientos del software son evaluados por su impacto técnico, en costo y en el cronograma.
- GP.O4. Reuniones de revisión con el equipo de trabajo y el cliente son realizadas. Los acuerdos que surgen de estas reuniones son documentados y se les hace seguimiento.
- GP.O5. Los riesgos son identificados en el desarrollo y durante la realización del proyecto.
- GP.O6. Una Estrategia de Control de Versiones de Software es desarrollada. Los elementos de configuración del software son identificados, definidos e incorporados a la línea base. Las modificaciones y releases de los elementos son controlados y puestos a disposición del cliente y del equipo de trabajo. El almacenamiento, la manipulación y la entrega de los elementos son controlados.
- GP.O7. El Aseguramiento de Calidad del Software es realizado para proporcionar garantía de que los productos y procesos de trabajo cumplen con el Plan del Proyecto y Especificación de Requerimientos.

Actividades de los procesos de gestión del proyecto son las siguientes:
Véase figura 2.8.

- GP.1 Planificación del Proyecto
- GP.2 Ejecución del Plan del Proyecto
- GP.3 Evaluación y Control del Proyecto
- GP.4 Cierre del Proyecto

En el proceso de Gestión del Proyecto antes del inicio, durante el proceso y al terminarlo; se tienen diversos productos de entrada, internos y de salida, con la finalidad de llevar y cumplir ordenadamente cada actividad y objetivo de este.

Se detallan los productos y su origen o destino, según correspondan.

Productos de entrada de la GP:

1. Enunciado del trabajo por parte del cliente.
2. Configuración del software por parte de la implementación del software.

2.3 ISO/IEC 29110. Perfil Básico.

3. Solicitud de cambio por parte del cliente y de la implementación del software.

Productos internos de la GP:

1. Solicitud de cambio.
2. Acciones correctivas.
3. Acta de Reunión.
4. Resultados de Verificación.
5. Reporte de avance.
6. Respaldo del repositorio del proyecto

Productos de salida de la GP:

1. Plan del proyecto hacia la implementación de software.
2. Acta de aceptación hacia la alta dirección.
3. Repositorio del proyecto para la implementación de Software.
4. Acta de reunión para el cliente.
5. Configuración del software para el cliente.

2. MARCO TEÓRICO

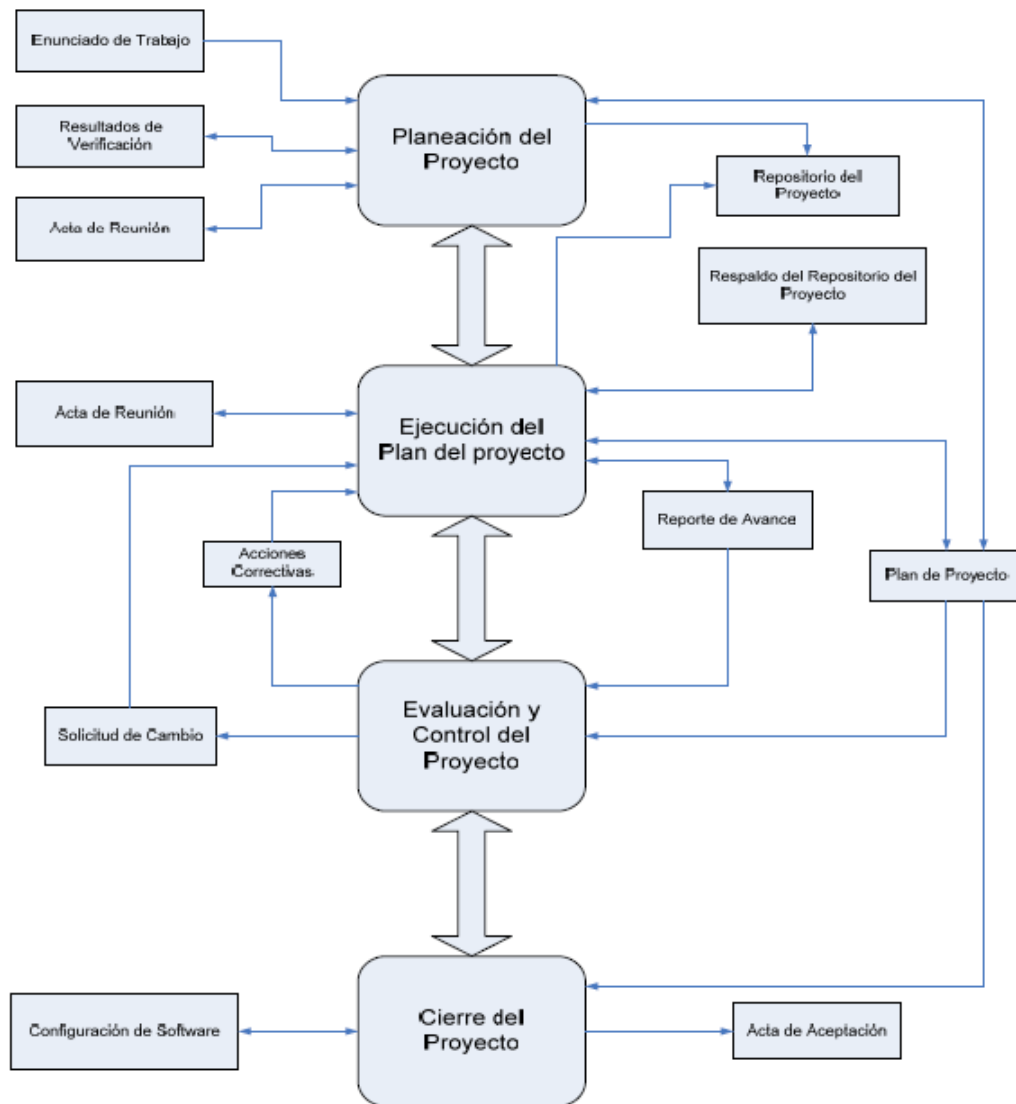


Figura 2.8: Diagrama de los procesos de gestión del proyecto [5]

2.3.2.1.2 Implementación del Software

El propósito de este proceso es la realización de manera sistemática las actividades de análisis, diseño, construcción, integración y pruebas del o los productos de software, nuevos o modificados, de acuerdo con lo establecido en el proceso de gestión del proyecto. [5]

Los objetivos particulares del proceso de Implementación de Software son los siguientes:

- IS.O1. Las Tareas de las actividades son realizadas a través del cumplimiento del Plan del Proyecto actual.
- IS.O2. Los requerimientos del Software son definidos, analizados para su correctitud y testeabilidad, aprobados por el Cliente, incorporados a la línea base y comunicados.
- IS.O3. La arquitectura y diseño detallado del Software son desarrollados e incorporados a la línea base. Aquí se describen los Componentes de Software y sus interfaces internas y externas. La consistencia y trazabilidad de los requerimientos de Software son establecidos.
- IS.O4. Los Componente de Software definidos por el diseño son producidos. Las pruebas unitarias son definidas y ejecutadas para verificar la consistencia de los requerimientos y el diseño. La trazabilidad de los requerimientos y el diseño son establecidas.
- IS.O5. El Software es producido ejecutando la integración de los Componente de Software y es verificado usando los Casos de Prueba y Procedimientos de Prueba. Los resultados son registrados en el Reporte de Pruebas. Los defectos son corregidos y la consistencia y trazabilidad hacia el Diseño de Software son establecidos.
- IS.O6. La Configuración de Software, que cumpla con la Especificación de Requerimientos según lo acordado con el Cliente, que incluye la documentación de usuario, operación y mantenimiento es integrada, incorporada a la línea base y almacenada en el Repositorio del Proyecto. Las necesidades de cambios para la Configuración de Software son detectadas y las solicitudes de cambio relacionadas son iniciadas.
- IS.O7. Las Tareas de verificación y validación de todos los productos de trabajo requeridos son realizados utilizando los criterios definidos para lograr la coherencia entre los productos de entrada y salida en cada actividad. Los defectos son identificados y corregidos; los registros son almacenados en los Resultados de Verificación/Validación.

2. MARCO TEÓRICO

Actividades del proceso de implementación del software son las siguientes:
Véase figura 2.9.

- IS.1 Inicio de la Implementación de Software.
- IS.2 Análisis de requerimientos del Software.
- IS.3 Arquitectura y Diseño Detallado del Software.
- IS.4 Construcción del Software.
- IS.5 Integración y Pruebas del Software.
- IS.6 Entrega del Producto.

En el proceso de Implementación del Software antes del inicio, durante el proceso y al terminarlo; se tienen diversos productos de entrada, internos y de salida, con la finalidad de llevar y cumplir ordenadamente cada actividad y objetivo de este.

Se detallan los productos y su origen o destino, según correspondan.

Productos de entrada de la IS:

1. Plan del proyecto por parte de la gestión del proyecto.
2. Repositorio del proyecto por parte de la gestión del proyecto.

Productos internos de la IS:

1. Resultado de validación.
2. Resultados de verificación.

Productos de salida de la IS:

1. Solicitud de cambios para la gestión del proyecto.
2. Configuración del software para la gestión del proyecto.

2.3 ISO/IEC 29110. Perfil Básico.

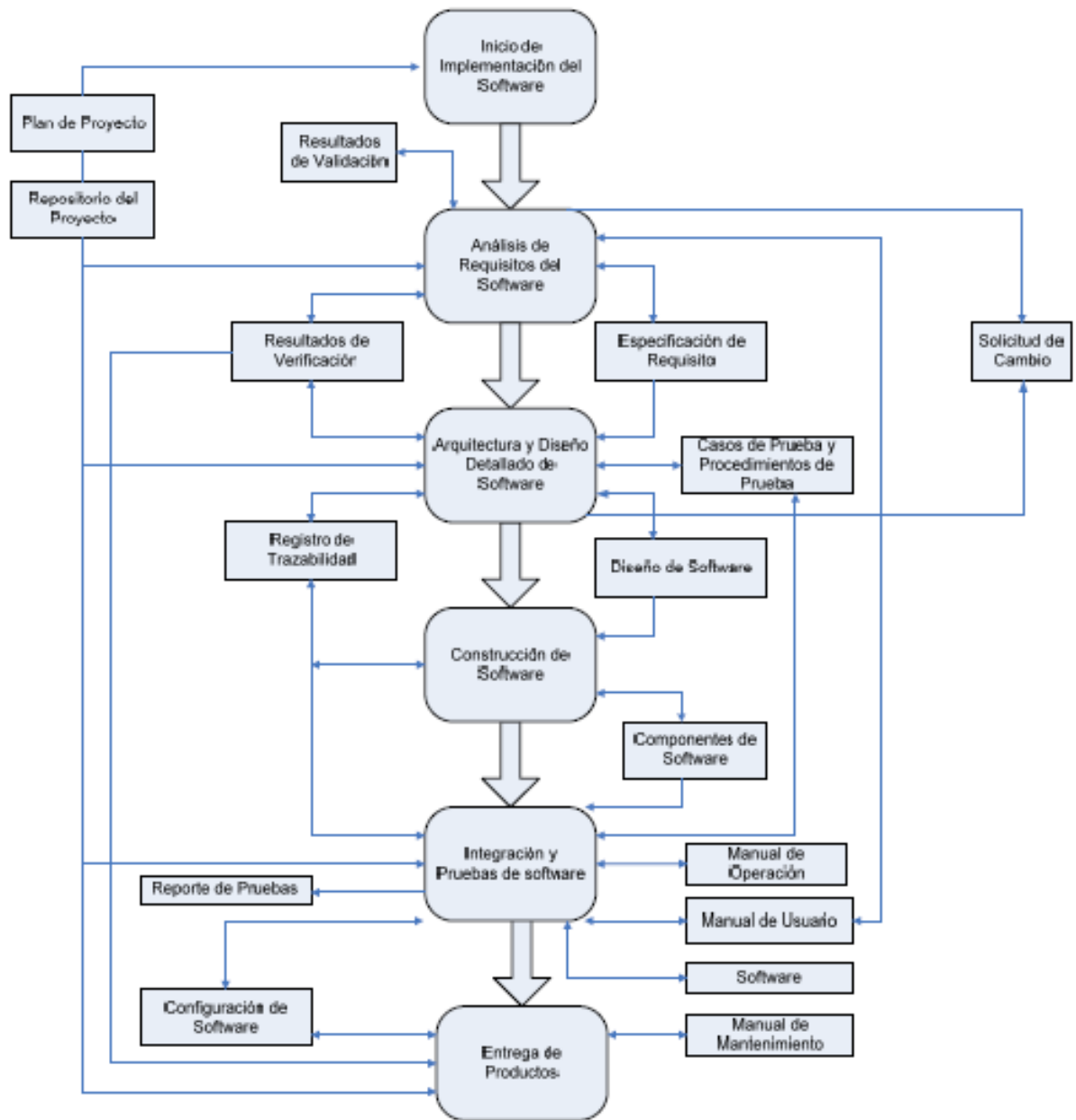


Figura 2.9: Diagrama del procesos de implementación del software. [5]

2. MARCO TEÓRICO

2.4 KUALI-BEH

KUALI-BEH [28], es una descripción de conceptos comunes y básicos, relacionados a la ingeniería de software que brinda una base para la construcción de métodos y prácticas para un equipo de desarrollo de software.

Una de las dos vistas que compone a KUALI-BEH, es la vista estática en la cual se escriben los conceptos comunes que se involucran en los proyectos de software como: *productos de software, equipo de trabajo, practicantes, interesados, **prácticas y métodos***, entre otros. Para fines de este trabajo utilizaremos los señalados en letras en negritas para la construcción del método que se describe en la sección "Método propuesto".

2.4.1 Conceptos.

En esta sección se describen los conceptos que se utilizarán para la creación de nuestro método y sus prácticas.

*“Un **método** es una articulación de un conjunto de prácticas coherente, consistente y completo, con un propósito específico que satisface las necesidades de los interesados bajo condiciones específicas ”*[28].

La definición de un método contiene los siguientes elementos comunes:

- **Propósito:** Es el objetivo que se debe alcanzar al aplicar el método en su totalidad.
- **Entrada:** Son las características esperadas de un producto de trabajo y/o condiciones para dar inicio con la ejecución de la práctica.
- **Resultado:** Son las características esperadas de un producto de trabajo y/o condiciones requeridas como resultado después de la ejecución de una práctica.
- **Prácticas:** Conjunto de prácticas que constituyen al método y tienen como fin común cumplir el propósito de este.

Usaremos la siguiente plantilla para la definición del método:

[identificador]	Método	
[nombre]		
Propósito		
[propósito]		
Entrada	Resultado	
[necesidades de los involucrados, condiciones del proyecto,...]	[características del producto de software,...]	
Prácticas		
[Practica de Requerimientos, ..., Practica de Entrega, ...]		

Figura 2.10: Plantilla para método de KUALI-BEH.[6]

“Una **práctica** es una guía del trabajo, con un objetivo en específico que proporciona cómo producir un resultado a partir de una entrada. La guía brinda un conjunto de actividades sistemáticas y repetibles enfocadas en alcanzar el objetivo y el resultado de la práctica. Se requieren conocimientos y habilidades particulares para seguir la guía de la práctica, que se puede realizar opcionalmente con herramientas” [28].

La definición de una práctica contiene los siguientes elementos comunes:

- **Entrada:** Son las características esperadas de un producto de trabajo y/o condiciones para dar inicio con la ejecución de la práctica.
- **Objetivo:** Es el propósito que se espera lograr después de la ejecución de la práctica.

2. MARCO TEÓRICO

- **Resultado:** Son las características esperadas de un producto de trabajo y/o condiciones requeridas como resultado después de ejecutar una práctica.
- **Actividad:** Es el conjunto de tareas que contribuyen a alcanzar el objetivo de la práctica.
- **Conocimientos y habilidades:** Conjunto de habilidades, competencias o logros adquiridos por el practicante y necesarios para realizar la práctica.
- **Herramientas:** Dispositivo para realizar una función en particular.

Usaremos la siguiente plantilla para la definición de una práctica:

<i>[identificador]</i>	Práctica	
<i>[nombre]</i>		
Objetivo		
<i>[objetivo]</i>		
Entrada	Resultado	
<i>[características esperadas,...]</i>	<i>[características esperadas,...]</i>	
Actividades		
<i>[lista de actividades]</i>		
Herramientas (opcional)		
<i>[lista de herramientas propuestas]</i>		
Conocimientos y habilidades		
<i>[habilidades, competencias, talentos,...]</i>		

Figura 2.11: Plantilla de práctica de KUALI-BEH.[6]

Estado del Arte

Este capítulo abordará el estado actual de la investigación, respecto a la Deuda Técnica en Requerimientos.

El término de Deuda Técnica es relativamente nuevo y desde que se mencionó por primera vez, se asoció a la actividad de programar. Conforme se ha ido popularizando y ampliando la definición, la Deuda Técnica ha sido relacionada a las actividades realizadas en un proyecto de desarrollo o mantenimiento de software.

En el estudio que realiza Lenarduzzi [29], recopila artículos acerca investigaciones sobre la Dauda Técnica (el estudio contempla artículos hasta 2018) y encuentra que la deuda más estudiada es la relacionada con el código (Deuda en el Código) con un 38 por ciento de los artículos; en segundo lugar, se encuentra la Deuda en la Arquitectura con 24 por ciento y con un diez por ciento se encuentra la Deuda en el Diseño. Sin embargo, se menciona que un 24 por ciento no especifica el tipo de deuda que investiga, pero desarrolla en general el tema de Deuda Técnica.

Es por eso que la Deuda Técnica en los Requerimientos ha sido realmente poco desarrollada o llevada a otros niveles en comparación de la deuda en otras actividades como la arquitectura o en el código.

3. ESTADO DEL ARTE

3.1 Deuda Técnica en Requerimientos del Software

3.1.1 Definiciones y clasificaciones de la Deuda Técnica en los Requerimientos

Neil Ernst, define la deuda de los requerimientos como la “distancia entre la solución óptima a los requerimientos y la solución que se implementó hasta el momento”[17]. Ernst, sugiere que los requerimientos deben ser tratados como objetos de primera clase durante el diseño y durante todo el ciclo de vida.

En la ontología de Alves [3], la deuda en los requerimientos recae en la implementación de estos o en la forma de implementarlos, de manera que los requerimientos funcionales no satisfacen del todo a los requerimientos no funcionales, es decir que los requerimientos funcionales no satisfacen reglas de seguridad, calidad, comportamiento, etc.

La definición, según la naturaleza de la Deuda Técnica en los Requerimientos es la siguiente:

“La deuda de requerimientos se refiere a las compensaciones hechas con respecto a los requerimientos que el equipo de desarrollo necesita implementar o cómo implementarlos”.

Lenarduzzi [29], al querer definir la deuda técnica en los requerimientos, crea un marco de referencia, con tres clasificaciones de Deuda en los Requerimientos, en el que cada clasificación nos va dando indicios o guías; de tal forma, que podríamos detectar, medir y pagar esta deuda. Es de importancia mencionar que Lenarduzzi plantea esta definición y “guías” en un entorno de desarrollo de aplicaciones móviles.

Lenarduzzi así clasifica sus tres tipos de Deuda Técnica en los Requerimientos:

- **Deuda en los requerimientos del tipo 0:**

Necesidades incompletas del cliente. Es aquella deuda en la que se desatienden las necesidades de los usuarios o interesados ignorando los canales de comentarios donde estos expresan sus necesidades, esta se puede identificar clasificando automáticamente comentarios de usuarios y partes interesadas. Se paga después de haber identificado las necesidades que se habían descuidado y se incluyen en el documento de especificación de los requerimientos del software.

3.1 Deuda Técnica en Requerimientos del Software

- **Deuda en los requerimientos del tipo 1:**

Olor en los requerimientos. Es aquella deuda en la que se incurre cuando los encargados de los requerimientos formalizan los requerimientos con la documentación pertinente y se podrían identificar violaciones a la norma de la calidad de los requerimientos (ISO29148), provocando que se pueda ocasionar una implementación incorrecta de estos. Se identifica en la documentación de los requerimientos y podría ser cuantificada con el costo resultante de eliminar los olores en los requerimientos, es decir mitigar el impacto negativo de estos “olores”.

- **Deuda en los requerimientos del tipo 2:**

Implementación no coincidente. Deuda en la que la implementación real del sistema no es la misma que la que está establecida por la parte interesada, se puede detectar con la trazabilidad de los requerimientos comparada y la solución implementada en código. Se paga a manera de empatar o implementar la solución para que coincida con las necesidades reales.

Lenarduzzi, retoma la definición de Ernst (la primera que se define en este capítulo) y la extiende, definiendo sus tres tipos de ReD" (Requeriments Debt).

En el tipo 0, incluyen decisiones con respecto a la obtención de las necesidades de los usuarios en una etapa temprana de la gestión de los requerimientos del software. El tipo 1, considera que la formalización de los requerimientos es necesaria para la identificación de problemas en estos, ya que, al ser capturados en lenguaje natural puede existir ambigüedad . (ver. Olor en los requerimientos).

Y se extiende la definición con el tipo 2, al plantear que, dada la diferencia entre la implementación real de los requerimientos y la implementación deseada esta no coincide, por lo tanto es prudente tratar de igualar o minimizar la diferencia encontrada.

3.1.2 Causas de la Deuda Técnica en Requerimientos del Software

Neil Ernst [17], dice que cuando se ha incurrido en la deuda en los requerimientos, es porque se le ha dado una priorización mayor a requerimientos que no son importantes y brindan poco valor al cliente. Todo lo anterior debido a una pobre o inadecuado análisis o elicitación de los requerimientos; a esto podemos agregar la mala administración de estos, así como su completitud y formulación.

3. ESTADO DEL ARTE

Alves [3], en la definición de este tipo de deuda, menciona que puede ser causada por los requerimientos que están parcialmente implementados o incompletos y también, requerimientos implementados, pero que no cubren todos los casos o excepciones a reglas.

Método propuesto.

Este capítulo trata sobre el método propuesto para la gestión de la Deuda Técnica en los Requerimientos. El método consistirá en un conjunto de prácticas de aplicación que sirvan de apoyo para la gestión de este tipo de deuda técnica y así evitar riesgos que puedan ser consecuencia de incurrir en ella. Se recomienda que este método se implemente como complemento a los procesos, los métodos o las prácticas de Gestión de los Requerimientos (captura, análisis, especificación y validación), según la organización.

4.1 Método de Gestión de la Deuda Técnica en los Requerimientos.

El método y las prácticas de las que se compone fueron construidos y desarrollados en su mayoría con base en la ISO/IEC 29110 y la ISO/IEC 29148, como se mencionó anteriormente y para su definición se utilizaron las plantillas que proporciona Quali-Beh. En el perfil básico de la ISO/IEC 29110 hay dos procesos con sus respectivas actividades:

- **Gestión de proyectos.**
 1. Planificación del proyecto.
 2. Ejecución del plan del proyecto.

4. MÉTODO PROPUESTO.

3. Evaluación y control del proyecto.
4. Cierre del proyecto.

▪ **Implementación del software.**

1. Inicio de la implementación del software.
2. Análisis de los requerimientos del software.
3. Arquitectura y diseño detallado del software.
4. Construcción del software.
5. Integración y pruebas del software.
6. Entrega del producto.

Ya que el método propuesto está apegado a la ISO/IEC 29110, las prácticas que forman parte de este, están relacionadas directamente con las actividades y procesos de esta norma. Por lo tanto, explicaremos la relación existente entre las prácticas con la norma:

La práctica número uno del método, **Práctica de Prevención de la Deuda Técnica en los Requerimientos**, está relacionada con la actividad dos del **Proceso de Implementación del Software**, que es la actividad de **Análisis de los requerimientos del Software**. Cabe mencionar que estas actividades se llevan a cabo en las actividades **Inicio, Planificación y Ejecución del Proyecto**, que pertenecen al proceso de **Gestión del Proyecto**.

La segunda práctica del método propuesto es la **Práctica de Detección de la Deuda Técnica en los Requerimientos**, que se basa en una comparación de lo que se ha construido del producto de software, por lo tanto, se relaciona con las actividades de **Construcción e Integración y Pruebas del Software**, que pertenecen al **Proceso de Implementación del Software**, las cuales se llevan a cabo en las actividades de **Ejecución y Evaluación, y Control del Proyecto del proceso de Gestión del Proyecto**.

4.1 Método de Gestión de la Deuda Técnica en los Requerimientos.

La relación de la tercera práctica del método con la ISO/IEC 29110 es que la **Práctica de Manejo de la Deuda Técnica en los Requerimientos**, en la que se lleva el seguimiento de la deuda, es en las actividades de **Construcción e Integración y Pruebas del Software**, que pertenecen al **Proceso de Implementación del Software**, ya que la práctica se plantea como una actualización del trabajo que se ha ido realizando durante las iteraciones, al igual que la práctica dos del método. A su vez estas actividades también se llevan a cabo en las actividades de **Ejecución y Evaluación, y Control del Proyecto** del proceso de **Gestión del Proyecto**.

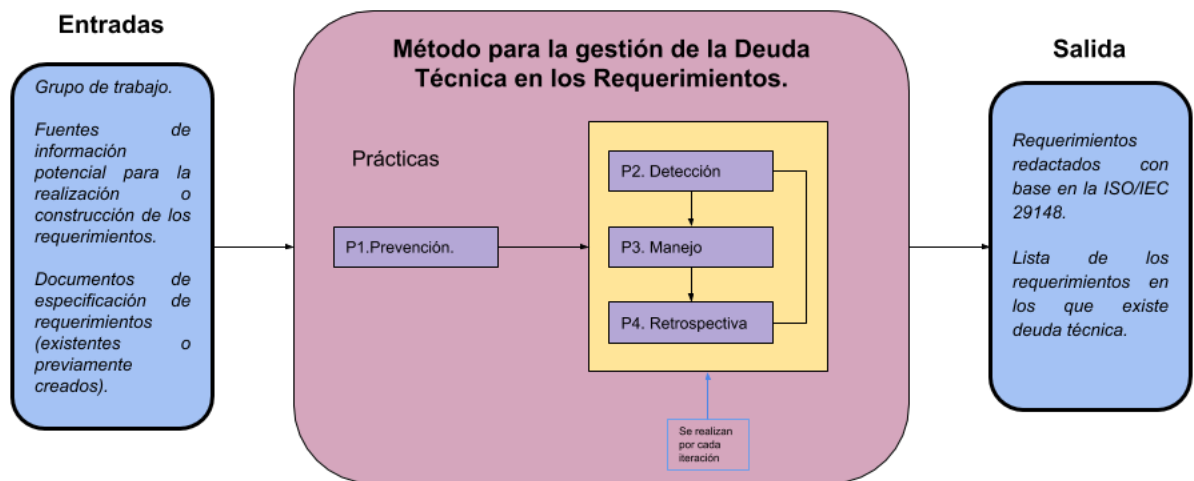


Figura 4.1: Método de Gestión de la Deuda Técnica en Requerimientos (MGDTR).

4. MÉTODO PROPUESTO.

En la cuarta y última práctica del método, **Práctica de Retrospectiva**, se realiza para retroalimentación del grupo de trabajo y generalmente al final de una iteración o al final del proyecto. Esta se relaciona con la **Ejecución, Evaluación y Control**, y **Cierre del Proyecto** del proceso de **Gestión del Proyecto**.

Así se define el método de gestión de la deuda técnica en los requerimientos tomando la plantilla de definición del método de KUALI-BEH.

M1	Método
<i>Método de Gestión de la Deuda Técnica en los Requerimientos.</i>	
Propósito	
<i>El propósito del método de gestión de la deuda técnica en los requerimientos es brindar una guía para que en un proyecto de desarrollo se pueda prevenir, detectar y manejar la deuda técnica en los requerimientos, para esto se proporciona una serie de prácticas que sirven de apoyo durante el proceso de los requerimientos.</i>	
Entrada	Resultado
<i>Grupo de trabajo. Fuentes de información potencial para la realización o construcción de los requerimientos.</i>	<i>Requerimientos redactados con base en la ISO/IEC 29148 [5]. Lista de los requerimientos en los que existe deuda.</i>
Prácticas	
<ol style="list-style-type: none"> 1. Práctica de Prevención de la Deuda Técnica en los Requerimientos (P1). 2. Práctica de Detección de la Deuda Técnica en los Requerimientos (P2). 3. Práctica de Manejo de la Deuda Técnica en los Requerimientos (P3). 4. Práctica de Retrospectiva (P4). 	

Figura 4.2: Tabla del Método de Gestión de la Deuda Técnica en Requerimientos (MGDTR).

4.2 Las prácticas del método.

Las cuatro prácticas que constituyen al método de gestión de la deuda técnica están diseñadas de manera que mantengan una relación entre sí. La P1 Prevención, busca que los requerimientos se definan de una manera clara, simple y que sean explícitos; obteniendo este objetivo y conforme se desarrollan otras tareas o procesos de la gestión de los requerimientos. La P2 Detección, se centra en verificar que los

4.2 Las prácticas del método.

requerimientos están siendo completados en toda la extensión de la palabra, y si no es así, crear una lista de aquellos requerimientos que están incompletos o que no han sido implementados y ya deberían estarlo, está lista será la entrada para la P3 Manejo de los Requerimientos en los que se ha identificado alguna deuda. Todo esto será monitoreado a través de la P4 Retrospectiva, en donde se señalarán las actividades que están resultando benéficas y las que no, esta retroalimentación servirá para seguir trabajando de la misma manera o proponer mejoras a las actividades realizadas, respectivamente.

Las practicas desarrolladas del método de gestión de la deuda técnica en los requerimientos están expresadas con una platilla de práctica de Kuali-Beh.

- **Plantilla de práctica (Kuali-Beh).**

Siguiendo la plantilla de Kuali-Beh para la definición de prácticas, se define el ID, nombre, objetivo, entradas, resultados, actividades, herramientas y conocimientos y habilidades de la práctica.

- **Actividades de la práctica y sus tareas asociadas.**

Definida la platilla de práctica según Kuali-Beh, se establecen las actividades especificadas y se explicitan sus tareas en caso de tenerlas; para indicar los pasos a seguir explícitamente y cumplir con el objetivo de las tareas y a su vez las actividades.

- **Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.**

En esta sección se da el contexto, si es que se requiere de temas o de herramientas que ayudarán a entender y en consecuencia proponer el funcionamiento de las herramientas que se recomienda sean utilizadas para la realización de la práctica y las actividades dentro de esta.

4.2.1 Práctica de Prevención de la Deuda Técnica en los Requerimientos.

La primera práctica que compone al método es la de Prevención de la Deuda Técnica en los Requerimientos, durante esta práctica se realiza un análisis de las fuentes potenciales de información para la construcción de los requerimientos del proyecto.

4. MÉTODO PROPUESTO.

Esta práctica se nombra de prevención porque su principal motivación es que la Deuda Técnica no se haga presente desde el inicio de la identificación y construcción de los requerimientos.

<i>P1</i>	Práctica
<i>Práctica de Prevención de la Deuda Técnica en los Requerimientos.</i>	
Objetivo	
<i>El objetivo de esta práctica es esclarecer las necesidades, metas u objetivos de los stakeholders o partes interesadas, para ayudar/iniciar la construcción de los requerimientos por medio de sus características y atributos, basado en los "fundamentos de los requerimientos" de la ISO/IEC 291458 [cita29148], como medida de prevención deuda técnica en los requerimientos.</i>	
Entrada	Resultado
<i>Fuentes de información potencial sobre las necesidades de los stakeholders o partes interesadas:</i> <ul style="list-style-type: none"> • <i>Stakeholders o partes interesadas.</i> • <i>Planteamiento del problema o necesidades.</i> • <i>Documentos de especificación de requerimientos (existentes o previamente creados).</i> • <i>Textos, grabaciones, resumen o notas tomadas en entrevistas con stakeholders o partes interesadas.</i> 	<i>Requerimientos que cumplen con los "fundamentos de los requerimientos" de la de la ISO/IEC 291458.</i>
Actividades	
<ol style="list-style-type: none"> 1. <i>Redactar los requerimientos basado en los "criterios de lenguaje de los requerimientos".</i> 2. <i>Verificar "las características de los requerimientos individuales" y "los atributos de los requerimientos".</i> 	
Herramientas (opcional)	
<i>Control de versiones.</i> <i>"Repositorio físico o digital".</i> <i>Fuentes de información.</i> <i>Tablas y conocimientos específicos brindados en el contexto de la práctica.</i>	
Conocimientos y habilidades	
<i>Capacidad de análisis, redacción, comprensión de textos e ideas principales.</i>	

Figura 4.3: Definición de la Práctica 1 (P1) del MGDTR.

4.2.1.1 Actividades de la práctica y sus tareas asociadas.

1. **Redactar los requerimientos basado en “los criterios de lenguaje de los requerimientos”.** (Ir a sección Requerimientos del software » ISO/IEC29148 » Conceptos).
 - Leer, escuchar y/o tomar notas de las necesidades u objetivos de las fuentes potenciales de información con las que contamos para la realización de los requerimientos.
 - Escribir los requerimientos basados en la información que brindan “los criterios de lenguaje de los requerimientos”, siguiendo las recomendaciones que se hacen en la siguiente sección.
2. **Verificar “las características de los requerimientos individuales”.** (Ir a sección Requerimientos del software » ISO/IEC29148 » Conceptos).
 - Realizar la matriz para comparar cada requerimiento con las características de los requerimientos individuales.

4.2.1.2 Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

1. ¿Cómo identificar un requerimiento?

En muchos casos, los interesados o las partes interesadas nos cuentan las necesidades u objetivos de diversas maneras, pueden ser muy peculiares, o simplemente muestran un escrito formal. Independientemente de la forma en la que este dada esa fuente sabemos que un requerimiento se caracteriza por ser una declaración de modo imperativa, por lo tanto, en su mayoría está acompañada de la palabra “deberá”.

Las palabras “debería” y “puede” suelen ser utilizada por los interesados o partes interesadas para declarar un deseo o alguna declaración puesto que no tienen la certeza o seguridad de que sea necesaria o alcanzable, por esa razón es importante discutirla y llegar a un acuerdo para evitar algún

4. MÉTODO PROPUESTO.

tipo de dificultad (desde un problema de ambigüedad hasta un problema de incumplimiento).

2. **¿Cómo escribir o redactar un requerimiento?**

Un requerimiento debe establecer el sujeto, la acción que realizará y si existen condiciones o restricciones.

Es preferible utilizar declaraciones:

- Simples.
- Positivas, evitando las negativas como “no debería”.
- Usar voz activa en estas declaraciones, precisamente porque nos interesa el actor y qué es lo que hace.

Se hace la recomendación de usar la lista de palabras que se encuentra en la sección de “los criterios de lenguaje de los requerimientos” para evitar el uso de términos que podrían ocasionar ambigüedad, complicar su entendimiento o hasta qué punto llevarlos.

El uso del siguiente formato es recomendable para la redacción de los requerimientos o textos previos a la formalización de estos.

- [Condición]-[Sujeto]-[Acción]-[Objeto]-[Restricción].

Ejemplo:

- [Cuando el cliente dé clic en exportar información,] [el sistema] deberá [generar] [un PDF] [con solo el RFC, nombre, CURP y código QR del cliente.]

La redacción y el entendimiento puede facilitarse ya que esta manera de escribir los requerimientos hace que esta sea clara y simple, además cumpliría con las características antes mencionadas.

3. **Las características y atributos de los requerimientos individuales.**

Otra recomendación que se emite en esta práctica es **realizar la matriz siguiente, con el propósito de verificar que los requerimientos que se**

4.2 Las prácticas del método.

están analizando cumplen con las características deseadas de un requerimiento individual. (Ir a sección Requerimientos del software » ISO/IEC29148 » Conceptos » Características de los requerimientos individuales para saber cuáles son estas características.

4. MÉTODO PROPUESTO.

						ID	REQUERIMIENTOS
						Requerimiento	Características individuales de los requerimientos.
						Necesario	
						Imp. libre	
						No ambiguo	
						Consistente	
						Completo	
						Singular	
						Factible	
						Trazable	
						Verificable	
						Prioridad	

Figura 4.4: Tabla de revisión de las características y atributos de los requerimientos.

4.2.2 Práctica de Detección de la Deuda Técnica en los Requerimientos.

Para Neil Ernst, la deuda técnica en los requerimientos se define como la “distancia entre la solución óptima a los requerimientos y la solución que se implementó hasta el momento” [17], nosotros podríamos considerar esa distancia de lo que se ha implementado hasta el momento, con lo que debería estar implementado para satisfacer la solución a los requerimientos, como la existencia de la Deuda Técnica en los Requerimientos, pues detectar la deuda podría ser bastante útil para el proyecto haciéndonos conscientes de la brecha que nos separa del objetivo.

Así se define la práctica con la plantilla de KUALI-BEH:

4.2.2.1 Actividades de la práctica y sus tareas asociadas.

1. Revisar y comparación de las listas de requerimientos definidas.

- Describir y detallar las diferencias que se encuentran entre la lista de los requerimientos que se deberían tener implementados hasta la fecha de esta práctica, con la lista de los requerimientos implementados realmente hasta el momento.

2. Realizar de una gráfica “burn-down” con los datos que se obtienen de la comparación de ambas listas

- Realizar la gráfica “burn-down” con los datos encontrados en la actividad anterior para monitorear las diferencias encontradas en el punto uno y poder identificar si existe deuda técnica en los requerimientos.

4.2.2.2 Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

1. Diferencias importantes para identificar

Es importante definir las diferencias en un nivel de alta descripción o a detalle para no omitir actividades pendientes que forman parte de la implementación de un requerimiento y así poder identificar dónde puede existir la deuda técnica

4. MÉTODO PROPUESTO.

P2	Práctica
<i>Práctica de Detección de la Deuda Técnica en los Requerimientos.</i>	
Objetivo	
<i>El objetivo de esta práctica es realizar una inspección sobre los requerimientos implementados hasta el momento de aplicación de esta práctica, para realizar una comparación de los requerimientos que deberían estar implementados a la fecha según lo establecido, con lo que está implementado en realidad.</i>	
Entrada	Resultado
<i>Lista de requerimientos a implementar hasta el momento. Lista de requerimientos implementados hasta el momento. Medición o estimación previa de los requerimientos según la organización.</i>	<i>Lista de requerimientos o partes de requerimientos faltantes de implementación.</i>
Actividades	
<ol style="list-style-type: none"> 1. <i>Revisar la lista de requerimientos a implementar según lo acordado con los stakeholders o partes interesadas y las partes encargadas del realizar la implementación.</i> 2. <i>Revisar la lista de requerimientos implementados hasta la fecha.</i> 3. <i>Comparar ambas listas y definición de las actividades, tareas o subtarear incompletas o no implementadas de algún requerimiento (o requerimiento no implementado en su totalidad).</i> 4. <i>Realizar una gráfica burn-down con los datos que se obtienen de la comparación de ambas listas, usando la medición o estimación previa realizada.</i> 5. <i>Identificar la existencia de deuda técnica en los requerimientos.</i> 	
Herramientas (opcional)	
<i>Control de versiones. "Repositorio físico o digital". Burn-down chart.</i>	
Conocimientos y habilidades	
<i>Capacidad de análisis, comparación, comprensión de textos e ideas principales, redacción y facilidad de realizar gráficas.</i>	

Figura 4.5: Definición de la Práctica 2 (P2) del MGDTR.

en los requerimientos que nos puede ayudar en el seguimiento de esta.

2. Gráfica “burn-down”

Este tipo de gráficas nos sirven para monitorear el trabajo pendiente y el tiempo con el que contamos para completar el objetivo del proyecto. Así, en su mayoría los ejes de esta gráfica muestran las actividades (eje Y) en contra del tiempo (eje X) mostrando la diferencia entre el avance y lo planeado. Además, las líneas muestran el avance comparado con lo planeado haciendo explícitos los detalles que hacen falta o el buen avance que se tiene del proyecto.

4.2 Las prácticas del método.

La entrada de estas gráficas, generalmente son estimaciones o mediciones de cada requerimiento.

Se muestra un ejemplo de una gráfica “burn-down” utilizada en un proyecto de desarrollo.

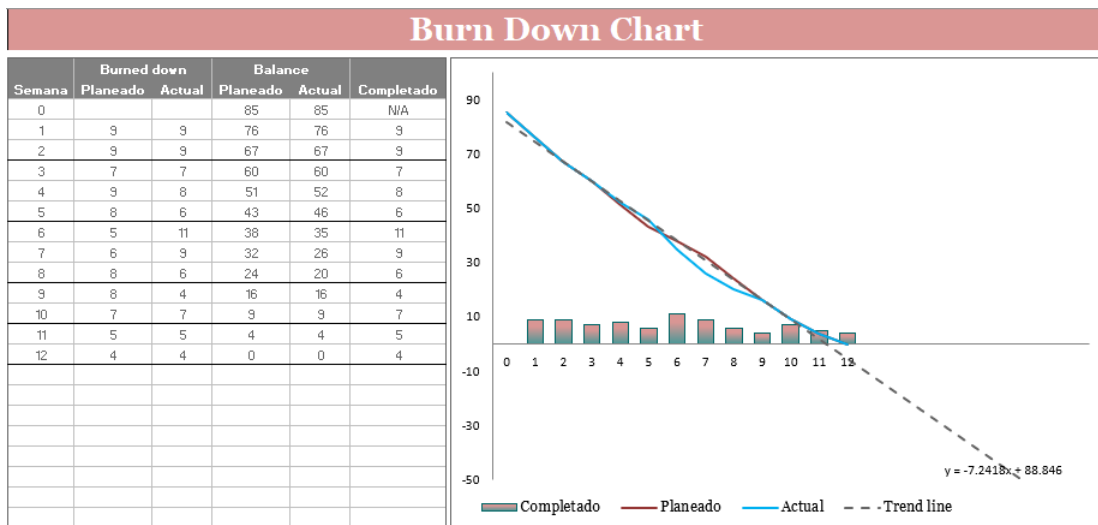


Figura 4.6: Gráfica burn-down para la detección de la Deuda Técnica en los Requerimientos.

4.2.3 Práctica de Manejo de la Deuda Técnica en los Requerimientos.

En esta práctica consiste en mantener presente la lista de requerimientos o actividades ligadas a los requerimientos faltantes o incompletos que se han detectado con deuda técnica. Se pretende que en esta lista se vayan eliminando los elementos que van siendo completados o incorporando elementos que van quedando pendientes o incompletos o actividades relacionadas a la gestión de los requerimientos.

4.2.3.1 Actividades de la práctica y sus tareas asociadas.

1. Proponer las actividades que se realizarán para completar los requerimientos o partes de los requerimientos faltantes o incompletos.

4. MÉTODO PROPUESTO.

P3	Práctica
<i>Práctica de Manejo de la Deuda Técnica en los Requerimientos.</i>	
Objetivo	
<i>El objetivo de esta práctica es llevar un seguimiento sobre las actividades a realizar para manejar la deuda técnica en los requerimientos, así como una lista de requerimientos en los que identificamos la existencia de deuda técnica en los requerimientos.</i>	
Entrada	Resultado
<i>Lista de requerimientos o partes de requerimientos faltantes de implementación (Resultado de la práctica 2).</i>	<i>Lista de requerimientos en los que se identifica que existe Deuda Técnica. Lista de actividades a realizar para el manejo de esta deuda.</i>
Actividades	
<ol style="list-style-type: none"> 1. <i>Proponer actividades a realizar para el manejo y evitar que la deuda técnica en los requerimientos aumente de acuerdo con la lista de requerimientos o partes de requerimientos faltantes de implementación.</i> 2. <i>Realizar la lista de las actividades del punto 1.</i> 3. <i>Colocar la lista en un lugar que este a la vista y que sea de fácil acceso o consulta de todo el equipo.</i> 4. <i>Establecer una revisión de los elementos cumplido o verificar las actividades y sus resultados conforme el paso del tiempo.</i> 	
Herramientas (opcional)	
<i>Control de versiones. "Repositorio físico o digital" Fuentes de información. Resultado de prácticas anteriores. Tablero kanban (físico o digital).</i>	
Conocimientos y habilidades	
<i>Capacidad de análisis, redacción, organizacional, comprensión de textos e ideas principales.</i>	

Figura 4.7: Definición de la Práctica 3 (P3) del MGDTR.

- Analizar el requerimiento o sus partes faltantes para asignar las actividades para completarlas.
 - Realizar la lista de las actividades del punto anterior para cumplir con los requerimientos incompletos.
2. Colocar la lista en un lugar que este a la vista y que sea de fácil acceso o consulta de todo el equipo.

4.2 Las prácticas del método.

- Asegurar que la lista la tenga cada miembro del equipo y que sea fácil de consultar. Puede ser a través de un tablero kanban físico o digital.
3. **Establecer una revisión de los elementos o verificar las actividades y sus resultados conforme el paso del tiempo.**

- Esta actividad sería mejor si se realizara semanas después de la identificación de la deuda existente o junto con la retrospectiva.
- Verificar las actividades y actualizar la lista de deuda técnica en los requerimientos.
- Quitar o tachar las actividades ya cumplidas y agregar los nuevos requerimientos identificados con deuda técnica.

Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

1. **Realizar la lista de las actividades que se realizarán para completar los requerimientos o partes de los requerimientos faltantes o incompletos.**

ID	Requerimiento incompleto	Actividad propuesta	Por hacer	Hecho

Figura 4.8: Tabla para las actividades a realizar.

4.2.4 Práctica de retrospectiva.

La Práctica de Retrospectiva puede ayudar a identificar si las tareas y actividades que se llevaron a cabo para la gestión de la deuda técnica en los requerimientos se realizaron de manera correcta y si se obtuvo un beneficio, así como brindar retroalimentación inmediata de los errores incurridos durante la aplicación de estas; para

4. MÉTODO PROPUESTO.

volver a plantear soluciones o mejoras sobre las actividades que presentaron dificultades durante el desarrollo del proyecto y en específico, en la gestión de la deuda técnica en los requerimientos.

P4		Práctica
<i>Práctica de Retrospectiva.</i>		
Objetivo		
<i>El objetivo de esta práctica es estar conscientes de las actividades que se están realizando bien o se realizaron bien para gestionar la deuda técnica en los requerimientos, enfatizar las actividades que no han producido resultados esperados y proponer mejoras para el trabajo a futuro.</i>		
Entrada		Resultado
<i>Resultados y fuentes internas de información (miembros del equipo) como opiniones, comentarios, etc.</i>		<i>Lista de actividades que se hicieron bien y actividades a mejorar a futuro.</i>
Actividades		
<ol style="list-style-type: none"> 1. <i>Realizar una lista de actividades que el equipo realizó o está realizando bien de acuerdo con la gestión de la deuda técnica en los requerimientos.</i> 2. <i>Identificar las actividades que no están generando resultados esperados o deseados.</i> 3. <i>Realizar lista de mejoras a implementar en las actividades del punto 2.</i> 		
Herramientas (opcional)		
<i>Repositorio físico o digital Fuentes internas de información.</i>		
Conocimientos y habilidades		
<i>Capacidad de análisis, redacción, comprensión de textos e ideas principales.</i>		

Figura 4.9: Definición de la Práctica 4 (P4) del MGDTR.

4.2.4.1 Actividades de la práctica y sus tareas asociadas.

1. Realizar e identificar la lista de actividades que se realizaron correctamente o se están realizando bien para la gestión de la deuda técnica y las actividades de las cuales no se están obteniendo buenos resultados.
 - Enlistar las actividades que se realizaron en el proceso de gestión de la deuda técnica en los requerimientos y si existe alguna otra actividad complementaria que se implementó, añadirla a la lista.

4.2 Las prácticas del método.

- Mencionar las actividades y recibir opiniones de los integrantes del equipo de acuerdo con su perspectiva, si la actividad implementada tuvo un efecto positivo o negativo en el proyecto.
 - De acuerdo con las opiniones emitidas marcar la actividad como realizada correctamente o con resultado negativo.
2. **Realizar la lista de propuestas de mejoras para las actividades, de la lista del punto 1 marcadas como negativas.**
- Enlistar nuevamente las actividades que no dieron los resultados esperados.
 - Escribir propuestas de mejoras a algunas actividades para obtener un resultado que pueda beneficiar al proyecto.

Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

1. **Enlistar las actividades que se realizaron en el proceso de gestión de la deuda técnica en los requerimientos.**

ID	Actividad	Positiva	Negativa	Observaciones

Figura 4.10: Tabla de las actividades realizadas.

2. **Enlistar las actividades en las que no se obtuvieron los resultados esperados y se proponen mejoras que puedan ayudar a obtener buenos resultados en la gestión de la deuda técnica en los requerimientos.**

4. MÉTODO PROPUESTO.

ID	Actividad	Propuesta de mejora	Observaciones

Figura 4.11: Tabla de las actividades realizadas que no dieron resultados esperados.

Evaluación del método propuesto.

En este capítulo se exponen los comentarios de la evaluación de los expertos a los que se les hizo llegar la propuesta del método para la gestión de la Deuda Técnica en los Requerimientos.

Para la evaluación de este método de gestión, se les hizo llegar a cuatro expertos, el método para que ellos emitieran su opinión para saber si el método se podía utilizar o si este brindaría un valor benéfico al desarrollo del proyecto en una organización. Por el tiempo de preparación y por los motivos de la pandemia este método no se pudo sujetar a más revisiones que se habían contemplado; por lo que solamente se tomaron en cuenta los comentarios y opiniones de los expertos anteriormente mencionados y a los cuales se les agradece su participación y tiempo.

5.1 Evaluación del Método

5.1.1 Evaluación de la P1. Práctica de Prevención de la Deuda Técnica en los Requerimientos.

Los evaluadores consideran que la P1 de prevención, del presente método, es una “herramienta excelente”, teniendo en cuenta que esta práctica se debe realizar como complemento del proceso de gestión de los requerimientos. Los evaluadores hacen énfasis en que los resultados dependen del equipo que implemente la práctica o el método. Si el equipo es lo suficientemente maduro y además, se siguen las buenas

5. EVALUACIÓN DEL MÉTODO PROPUESTO.

prácticas, el método podría ser una herramienta primordial, o de lo contrario sería inútil.

Consideran también, que la P1 evita confusión, ya que, ayuda a analizar, descomponer y catalogar los requerimientos en términos concisos.

Las actividades presentadas sientan una base, tanto en el lenguaje como en la forma de comunicación dentro del equipo u organización, beneficiando en el cumplimiento de las necesidades de los interesados.

5.1.2 Evaluación de la P2. Practica de Detección de la Deuda Técnica en los Requerimientos.

La P2 de detección, se considera una práctica indispensable porque detectar una deuda técnica tardía podría poner en peligro el proyecto.

Esta práctica podría ayudar a corregir el rumbo del proyecto, en caso de que no se este cumpliendo con el backlog del proyecto, además, permitiría al equipo a dar un seguimiento constante al avance del proyecto, lo cual, sería benéfico para el ciclo de vida.

5.1.3 Evaluación de la P3. Practica de Manejo de la Deuda Técnica en los Requerimientos.

Los requerimientos cobran una importancia de cabo a rabo en el ciclo de vida de cualquier proyecto, por eso se considera que al realizar la P3 de seguimiento de la deuda técnica en los requerimientos, crearía una concientización en las personas que son responsables de la toma de decisiones (administradores, gerentes e incluso el cliente) y también, en el equipo de desarrollo.

Consideran que al realizar una lista de actividades (con tareas o requerimientos faltantes o pendientes) con deuda técnica sería un control para reducirla, posiblemente desde una etapa temprana evitando que crezca exponencialmente y ponga en riesgo el resultado del proyecto.

5.1.4 Evaluación de la P4. Practica de Retrospectiva.

La P4 de retrospectiva, es considerada una buena herramienta para identificar actividades que tuvieron un resultado negativo en el proyecto, con la finalidad de que estas conductas, comportamientos, problemas o situaciones sean previstas y sirvan como “lecciones aprendidas” en futuros proyectos.

Esta información le aportaría un valor agregado a los proyectos para la organización o el equipo, pues el equipo es el principal encargado de registrar observaciones y las valoraciones de las acciones tomadas para no volver a incurrir en ellas, siempre y cuando el equipo evite omitir u obviar situaciones acontecidas.

5.2 Observaciones de los expertos.

En una sección del apartado que se les presentó a los expertos (ver Apéndice A), se solicitaba hacer observaciones que pudieran complementar al método y que los expertos consideraban necesarias. Las observaciones fueron:

- Realizar un diagrama de secuencia para la representación del método con sus prácticas.
- El apartado uno se enfocaría en considerar el ambiente tecnológico (cambios de versiones o de software/hardware).
- Observaron que faltan definir acciones respecto al cambio de bibliotecas de software, vulnerabilidades y cambios de software.
- Consideraron que en los elementos de entrada a las prácticas se deberían incluir diagramas empresariales, así como la documentación existente del software, ya que no siempre se contemplan las restricciones que los documentos organizacionales definen.

5.3 Cambios al método.

Se considera realizar algunos cambios en el método tomando en cuenta las observaciones que realizaron los expertos.

- Realizar un diagrama de secuencia para la representación del método con sus prácticas.

5. EVALUACIÓN DEL MÉTODO PROPUESTO.

Se tomó en cuenta esta observación para representar de manera gráfica el método con sus prácticas. (ver figura 4.2)

- Las observaciones: **Observaron que faltan definir acciones respecto al cambio de bibliotecas de software, vulnerabilidades y cambios de software. y El apartado uno se enfocaría en considerar el ambiente tecnológico (cambios de versiones o de software/hardware).**

Se considera que, el ambiente tecnológico en caso de haber cambios de versiones o de software/hardware, sería con respecto a otro tipo de deuda técnica como: Deuda del código, deuda en el diseño, deuda de los procesos o en la infraestructura.

- **Consideraron que en los elementos de entrada a las prácticas se deberían incluir diagramas empresariales, así como la documentación existente del software, ya que no siempre se contemplan las restricciones que los documentos organizacionales definen.**

Esta observación se tomó en cuenta para explicitar más las entradas del método y la práctica 1, se busca hacer énfasis en que el equipo debe recurrir a la documentación de los requerimientos existente o que se haya realizado hasta el momento.

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones del presente trabajo de investigación y el trabajo futuro que conlleva.

6.1 Conclusiones

El presente trabajo de tesis aborda y da un panorama sobre el concepto de Deuda Técnica, ya que es un concepto que ha tomado una gran relevancia en la ingeniería de software, hasta el punto de que el concepto se ha ido expandiendo cada vez más y la connotación con la que la introdujo Ward Cunningham[7] por primera vez, ha cambiado totalmente y tomado presencia en las demás áreas de la ingeniería de software, es decir, ya no se asocia solamente a la actividad de escribir código.

Según los estudios que se retoman [4] y [3], la Deuda Técnica se catáloga en varios tipos, uno de ellos es la deuda que existe en los requerimientos del software.

La aportación principal de este trabajo consiste en una propuesta que da una visión general y un método que brinde ayuda para la gestión sobre la Deuda Técnica en los Requerimientos.

El método que se propuso tiene cuatro prácticas con las cuales se busca hacer un seguimiento de la Deuda Técnica en los Requerimientos, con la finalidad de que no pueda presentar un peligro para el proyecto.

6. CONCLUSIONES Y TRABAJO FUTURO

Algo que propone el método presente es que la gestión o manejo de la Deuda Técnica brinde actividades que permitan prevenir, detectar y manejar la Deuda Técnica (es por eso que se nombran así las prácticas del método).

El propósito es hacer notar las propiedades mencionadas en el capítulo de Deuda Técnica: que la deuda sea visible, conocer el valor presente de esta, evitar que acrecente, poder distinguir su origen y saber dónde se encuentra la Deuda Técnica en los Requerimientos.

Estas prácticas del método se desarrollaron con base en estándares y buenas prácticas como las ISO/IEC 29148 [18], 15288 [26] y la 29110 [5]. Además, para la definición del método y las prácticas se utilizaron las plantillas de KUALI-BEH [28]. Después de realizar las prácticas del método, fue enviado a algunos expertos para evaluaran su practicidad y si el método aportaba un valor al proyecto, considerando que este ayudaría a gestionar la deuda en los requerimientos.

6.2 Trabajo futuro

El presente trabajo tiene un vasto campo de investigación, por su gran amplitud y latencia que ha tenido a través del tiempo. Por falta de recursos y tiempo, sería preciso a futuro considerar las siguientes acciones:

- Creación de una herramienta que pueda efficientar y automatizar el Método de Gestión de la Deuda Técnica con conceptos de Inteligencia Artificial.
- Creación completa de un método de gestión para cada una de las diferentes clasificaciones de la deuda técnica.
- Unificar los diferentes métodos de gestión de los tipos de deuda técnica para la creación de un método general de gestión de la Deuda Técnica.

Propuesta de validación para expertos del método de gestión de la deuda técnica en los requerimientos

Documento presentado a los expertos.
Inicia documento.

**A. PROPUESTA DE VALIDACIÓN PARA EXPERTOS DEL
MÉTODO DE GESTIÓN DE LA DEUDA TÉCNICA EN LOS
REQUERIMIENTOS**

Método de gestión de la deuda técnica en los requerimientos.

Fernando Antonio Sánchez Montoya
feroasmon@gmail.com

Dado que el uso del término Deuda Técnica se ha hecho cada vez más popular en la Ingeniería de Software, y abarca situaciones más allá de la primera definición establecida como la documentación, requerimientos, arquitectura, diseño y pruebas de un proyecto [7]. En este trabajo se toma la oportunidad de trabajar con la Deuda Técnica en los Requerimientos del software.

La **hipótesis** del presente trabajo de tesis es la siguiente:

¿Se puede proponer un método para ayudar a gestionar la Deuda Técnica en los Requerimientos del software de un proyecto?

Por lo tanto, este trabajo busca definir un método basado en los procesos que ofrecen algunas normas ISO/IEC, para la gestión de la Deuda Técnica en los Requerimientos del software, para evitar que sea un obstáculo durante la ejecución del proyecto.

Por lo que esta tesis brindará un método constituido de una serie de cuatro prácticas para la identificación de la Deuda Técnica en los Requerimientos, procurar que se mantenga estable y llevar su seguimiento para que el proyecto de desarrollo de software, satisfagan criterios de calidad establecidos.

En la siguiente sección, se presenta el método que se propone para satisfacer la hipótesis de este trabajo y es por medio del presente, que le hago la petición de evaluar el método respondiendo las preguntas que se encuentran después de cada práctica y al final del método, con la finalidad de poder recibir retroalimentación, información importante, sugerencias u opiniones de expertos que, puedan complementar este trabajo y sobre todo, que pueda ser funcional para la comunidad de la ingeniería de software. (La sección de preguntas de cada práctica y del método están subrayadas en amarillo).

Método de gestión de la Deuda Técnica en los Requerimientos.

El método y las prácticas de las que se compone fueron construidos y desarrollados en su mayoría con base en la ISO/IEC 29110 [5] y la ISO/IEC 29148 [18], como se mencionó anteriormente y para su definición se utilizaron las plantillas que proporciona Quali-Beh [6][21].

Las cuatro prácticas que constituyen al método de gestión de la deuda técnica están diseñadas de manera que mantengan una relación entre sí, la P1 Prevención busca que los requerimientos se definan de una manera clara, simple y que sean explícitos; obteniendo este objetivo y conforme se desarrollan otras tareas o procesos de la gestión de los requerimientos, la P2 Detección, se centra en verificar que los requerimientos están siendo completados en toda la extensión de la palabra, y si no es así, crear una lista de aquellos requerimientos que están incompletos o que no han sido implementados y ya deberían estarlo, esta lista será la entrada para la P3 Manejo de los

Requerimientos en los que se ha identificado alguna deuda. Todo esto será monitoreado a través de la P4 Retrospectiva en donde se señalarán las actividades que están resultando benéficas y las que no. Esta retroalimentación servirá para seguir trabajando de la misma manera o proponer mejoras a las actividades realizadas, respectivamente.

Así se define el método de gestión de la deuda técnica en los requerimientos tomando la plantilla de definición del método de KUALI-BEH.

M1	Método
<i>Método de Gestión de la Deuda Técnica en los Requerimientos.</i>	
Propósito	
<i>El propósito del método de gestión de la deuda técnica en los requerimientos es brindar una guía para que en un proyecto de desarrollo se pueda prevenir, detectar y manejar la deuda técnica en los requerimientos, para esto se proporciona una serie de prácticas que sirven de apoyo durante el proceso de los requerimientos.</i>	
Entrada	Resultado
<i>Grupo de trabajo. Fuentes de información potencial para la realización o construcción de los requerimientos.</i>	<i>Requerimientos redactados con base en la ISO/IEC 29148 [5]. Lista de los requerimientos en los que existe deuda.</i>
Prácticas	
<ol style="list-style-type: none"> 1. Práctica de Prevención de la Deuda Técnica en los Requerimientos (P1). 2. Práctica de Detección de la Deuda Técnica en los Requerimientos (P2). 3. Práctica de Manejo de la Deuda Técnica en los Requerimientos (P3). 4. Práctica de Retrospectiva (P4). 	

Las Prácticas del método.

Las cuatro prácticas que se definen más adelante contienen tres secciones:

- **Plantilla de práctica (Kuali-Beh).**
Siguiendo la plantilla de Kuali-Beh para la definición de prácticas se define el ID, nombre, objetivo, entradas, resultados, actividades, herramientas y conocimientos y habilidades de la práctica.
- **Actividades de la práctica y sus tareas asociadas.**
Definida la plantilla de práctica según Kuali-Beh, se establecen las actividades especificadas y se explicitan sus tareas, en caso de tenerlas, para indicar los pasos detallados a seguir para cumplir con el objetivo de las tareas y a su vez las actividades.
- **Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.**
En esta sección se da el contexto, si es que se requiere de temas o de herramientas que ayudarán a entender y en consecuencia proponer el funcionamiento de las herramientas que se recomienda sean utilizadas para la realización de la práctica y las actividades dentro de esta.

P1. Práctica de Prevención de la Deuda Técnica en los Requerimientos.

La primera práctica que compone al método es la de *Prevención de la Deuda Técnica en los Requerimientos*, durante esta práctica se lleva un análisis de las fuentes potenciales de información para la construcción de los requerimientos del proyecto. Esta práctica se nombra de prevención porque su principal motivación es que la Deuda Técnica no se haga presente desde el principio de la identificación y construcción de los requerimientos.

P1	Práctica
<i>Práctica de Prevención de la Deuda Técnica en los Requerimientos.</i>	
Objetivo	
<i>El objetivo de esta práctica es esclarecer las necesidades, metas u objetivos de los stakeholders o partes interesadas, para ayudar/iniciar la construcción de los requerimientos por medio de sus características y atributos, basado en los “fundamentos de los requerimientos” de la ISO/IEC 291458 [18], como medida de prevención de deuda técnica en los requerimientos.</i>	
Entrada	Resultado
<i>Fuentes de información potencial sobre las necesidades de los stakeholders o partes interesadas:</i> <ul style="list-style-type: none"> • <i>Stakeholders o partes interesadas.</i> • <i>Planteamiento del problema o necesidades.</i> • <i>Textos, grabaciones, resumen o notas tomadas en entrevistas con stakeholders o partes interesadas.</i> 	<i>Requerimientos que cumplen con los “fundamentos de los requerimientos” de la ISO/IEC 29148.</i>
Actividades	
<ol style="list-style-type: none"> 1. <i>Redactar los requerimientos basado en los “criterios de lenguaje de los requerimientos”.</i> 2. <i>Verificar “las características de los requerimientos individuales” y “los atributos de los requerimientos”.</i> 	
Herramientas (opcional)	
<i>Control de versiones. “Repositorio físico o digital”. Fuentes de información. Tablas y conocimientos específicos brindados en el contexto de la práctica.</i>	
Conocimientos y habilidades	
<i>Capacidad de análisis, redacción, comprensión de textos e ideas principales.</i>	

Actividades de la práctica y sus tareas asociadas.

1. **Redactar de los requerimientos basada en “los criterios de lenguaje de los requerimientos”** (Ir a sección Requerimientos del software >> ISO/IEC29148 >> Conceptos).
 - Leer, escuchar y/o tomar notas de las necesidades u objetivos de las fuentes potenciales de información con las que contamos para la realización de los requerimientos.
 - Escribir los requerimientos basado en la información que brindan “los criterios de lenguaje de los requerimientos” y siguiendo las recomendaciones que se hacen en la siguiente sección.
2. **Verificar “las características de los requerimientos individuales”.** (Ir a sección Requerimientos del software >> ISO/IEC29148 >> Conceptos).
 - Realizar la matriz que compare cada requerimiento con las características de los requerimientos individuales.

Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

1. ¿Cómo identificar un requerimiento?

En muchos casos, los stakeholders o partes interesadas nos cuentan las necesidades u objetivos de diversas maneras, pueden ser muy peculiares, o simplemente muestran un escrito formal. Independientemente de la forma en la que este dada esa fuente sabemos que un requerimiento se caracteriza por ser una declaración de modo imperativa, por lo tanto, en su mayoría está acompañada de la palabra “deberá”.

Las palabras “debería” y “puede” suelen ser utilizada por los stakeholders o partes interesadas para declarar un deseo o alguna declaración puesto que no tienen la certeza o seguridad de que sea necesaria o alcanzable, por esa razón es importante discutirla y llegar a un acuerdo para evitar algún tipo de dificultad (desde un problema de ambigüedad hasta un problema de incumplimiento).

2. ¿Cómo escribir o redactar un requerimiento?

Un requerimiento debe establecer el sujeto, la acción que realizará y si existen condiciones o restricciones.

Es preferible utilizar declaraciones:

- Simples
- Positivas, evitando las negativas como “no debería”

- Usar voz activa en estas declaraciones, precisamente porque nos interesa el actor y qué es lo que hace.

Se hace la recomendación de usar la lista de palabras que se encuentra en la sección de “los criterios de lenguaje de los requerimientos” para evitar el uso de términos que podrían ocasionar ambigüedad, complicar su entendimiento o hasta qué punto llevarlos.

El uso del siguiente formato para la redacción de los requerimientos o textos previos a la formalización de estos.

- [Condición]-[Sujeto]-[Acción]-[Objeto]-[Restricción].

Ejemplo:

- Cuando el cliente dé clic en exportar información, el sistema deberá generar un PDF con solo el RFC, nombre, CURP y código QR del cliente.

La redacción y el entendimiento puede facilitarse ya que esta manera de escribir los requerimientos hace que esta sea clara y simple, además cumpliría con las características antes mencionadas.

3. Las características y atributos de los requerimientos individuales.

Otra recomendación que se emite en esta práctica es **realizar la matriz siguiente, con el propósito de verificar que los requerimientos que se están analizando cumplen con las características deseadas** de un requerimiento individual. (Ir a sección Requerimientos del software >> ISO/IEC29148 >> Conceptos >> Características de los requerimientos individuales para saber cuáles son estas características.

						ID	REQUERIMIENTOS
						Requerimiento	Características individuales de los requerimientos.
						Necesario	
						Imp. libre	
						No ambiguo	
						Consistente	
						Completo	
						Singular	
						Factible	
						Trazable	
						Verificable	
						Prioridad	

Preguntas de validación de la primera práctica.

1. ¿Usted cree que la redacción de los requerimientos influye en el manejo de estos (validación, implementación, etc.)?
2. Escribir o reescribir los requerimientos en la forma que se recomienda en esta práctica ¿ayudaría a entender o comprender de una mejor manera los requerimientos?
3. Para usted son importantes las observaciones o recomendaciones que se hacen en esta práctica para la identificación de los requerimientos, por ejemplo:

“Las palabras ‘debería’ y ‘puede’ suelen ser utilizadas por los stakeholders o partes interesadas para declarar un deseo o alguna declaración...”.

O, “realizar la matriz siguiente, con el propósito de verificar que los requerimientos que se están analizando cumplen con las características deseadas...”.

4. ¿Para usted sería útil realizar la verificación de **“las características de los requerimientos individuales”** y de **“los atributos de los requerimientos”**, en esta práctica?
5. ¿Considera usted, que estas actividades se podrían integrar con facilidad al proceso de manejo de los requerimientos de un equipo de trabajo?

Práctica de Detección de la Deuda Técnica en los Requerimientos.

Para Neil Ernst la deuda técnica en los requerimientos se define como la “distancia entre la solución óptima a los requerimientos y la solución que se implementó hasta el momento” [14], nosotros podríamos considerar esa distancia o diferencia de lo implementado hasta el momento con lo que debería estar implementado o satisfacer la solución a los requerimientos para detectar la existencia de deuda técnica en los requerimientos, pues detectar la deuda podría ser bastante útil para el proyecto haciéndonos conscientes de la brecha que nos separa del objetivo de este.

P2	Práctica
<i>Práctica de Detección de la Deuda Técnica en los Requerimientos.</i>	
Objetivo	
<i>El objetivo de esta práctica es realizar una inspección sobre los requerimientos implementados hasta el momento de aplicación de esta práctica, para realizar una comparación de los requerimientos que deberían estar implementadas a la fecha según lo establecido, con lo que está implementado en realidad.</i>	
Entrada	Resultado
<i>Lista de requerimientos a implementar hasta el momento. Lista de requerimientos implementados hasta el momento.</i>	<i>Lista de requerimientos o partes de requerimientos faltantes de implementación.</i>
Actividades	
<ol style="list-style-type: none"> 1. Revisar la lista de requerimientos a implementar según lo acordado con los stakeholders o partes interesadas y las partes encargadas del realizar la implementación. 2. Revisar la lista de requerimientos implementados hasta la fecha. 3. Comparar ambas listas y definición de las actividades, tareas o subtareas incompletas o no implementadas de algún requerimiento (o requerimiento no implementado en su totalidad). 4. Realizar una gráfica burn-down con los datos que se obtienen de la comparación de ambas listas. 5. Identificar la existencia de deuda técnica en los requerimientos. 	
Herramientas (opcional)	
<i>Control de versiones. “Repositorio físico o digital”. Burn-down chart.</i>	
Conocimientos y habilidades	
<i>Capacidad de análisis, comparación, comprensión de textos e ideas principales, redacción y facilidad de realizar gráficas.</i>	

Actividades de la práctica y sus tareas asociadas.

1. Revisar y comparación de las listas de requerimientos definidas.

- Describir y detallar las diferencias que se encuentran entre la lista de los requerimientos que se debían tener implementados hasta la fecha de esta práctica con la lista de los requerimientos implementados realmente hasta el momento.

2. Realizar de una gráfica burn-down con los datos que se obtienen de la comparación de ambas listas.

- Realizar la gráfica burn-down con los datos encontrados en la actividad anterior para monitorear las diferencias encontradas en el punto uno y poder identificar si existe deuda técnica en los requerimientos.

Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

1. Diferencias importantes para identificar.

Sería importante definir las diferencias en un nivel de alta descripción o a detalle para no omitir actividades pendientes que forman parte de la implementación de un requerimiento, así poder identificar dónde puede existir la deuda técnica en los requerimientos y que nos puede ayudar en el seguimiento de esta.

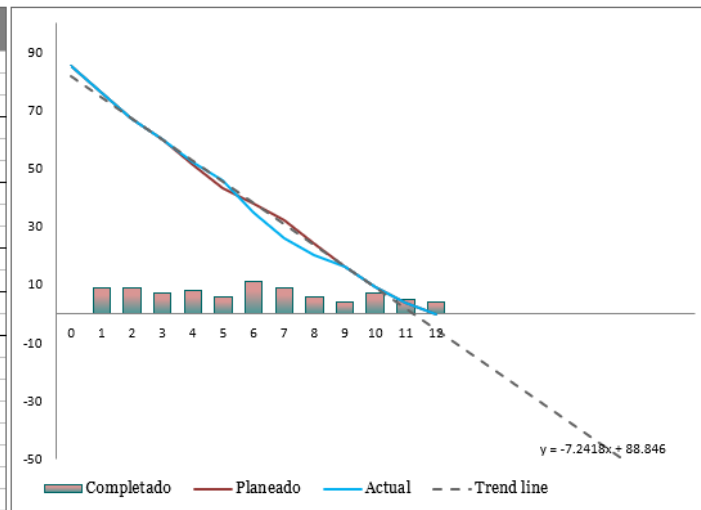
2. Gráfica burn-down.

Este tipo de gráficas nos sirven para monitorear el trabajo pendiente y el tiempo con el que contamos para completar el objetivo del proyecto. Así, en su mayoría los ejes de esta gráfica muestran las actividades (eje Y) en contra del tiempo (eje X) mostrando la diferencia entre el avance y lo planeado. Además, las líneas muestran el avance comparado con lo planeado haciendo explícitos los detalles que hacen falta o el buen avance que se tiene del proyecto.

Se muestra un ejemplo de una gráfica burn-down utilizada en un proyecto de desarrollo.

Burn Down Chart

Semana	Burned down		Balance		Completado
	Planeado	Actual	Planeado	Actual	
0			85	85	N/A
1	9	9	76	76	9
2	9	9	67	67	9
3	7	7	60	60	7
4	9	8	51	52	8
5	8	6	43	46	6
6	5	11	38	35	11
7	6	9	32	26	9
8	8	6	24	20	6
9	8	4	16	16	4
10	7	7	9	9	7
11	5	5	4	4	5
12	4	4	0	0	4



Preguntas de validación de la segunda práctica.

1. ¿Cree usted, que detectar la existencia de la Deuda Técnica en los requerimientos en su proyecto o proyectos, le ayudaría a manejar o darle seguimiento a su proyecto?
2. Llevar una lista de los requerimientos que se han realizado y una lista de los requerimientos que deberían estar hechos hasta la fecha, ¿sería de ayuda para realizar una comparación y tener presente qué es lo faltante en la realización de los requerimientos?
3. ¿Sería de ayuda realizar la gráfica burn-down para representar la diferencia, planteada en la pregunta anterior, de manera gráfica el avance del equipo?
4. Realizar y dar seguimiento a una lista específica y detallada de requerimientos o actividades que faltan por culminar, ¿sería de utilidad para usted o el equipo de desarrollo?

Práctica de Manejo de la Deuda Técnica en los Requerimientos.

En esta consiste en mantener presente la lista de requerimientos o actividades ligadas a los requerimientos faltantes o incompletos que se han detectado con deuda técnica en los requerimientos. Se pretende que en esta lista se vayan eliminando los elementos que van siendo completados o incorporando elementos que van quedando pendientes o incompletos o actividades relacionadas a la gestión de los requerimientos.

<i>P3</i>	Práctica
<i>Práctica de Manejo de la Deuda Técnica en los Requerimientos.</i>	
Objetivo	
<i>El objetivo de esta práctica es llevar un seguimiento sobre las actividades a realizar para manejar la deuda técnica en los requerimientos, así como una lista de requerimientos en los que identificamos la existencia de deuda técnica en los requerimientos.</i>	
Entrada	Resultado
<i>Lista de requerimientos o partes de requerimientos faltantes de implementación (Resultado de la práctica 2).</i>	<i>Lista de requerimientos en los que se identifica que existe Deuda Técnica. Lista de actividades a realizar para el manejo de esta deuda.</i>
Actividades	
<ol style="list-style-type: none"> <i>1. Proponer actividades a realizar para el manejo y evitar que la deuda técnica en los requerimientos aumente de acuerdo con la lista de requerimientos o partes de requerimientos faltantes de implementación.</i> <i>2. Realizar la lista de las actividades del punto 1.</i> <i>3. Colocar la lista en un lugar que este a la vista y que sea de fácil acceso o consulta de todo el equipo.</i> <i>4. Establecer una revisión de los elementos cumplido o verificar las actividades y sus resultados conforme el paso del tiempo.</i> 	
Herramientas (opcional)	
<i>Control de versiones. "Repositorio físico o digital" Fuentes de información. Resultado de prácticas anteriores. Tablero kanban (físico o digital).</i>	
Conocimientos y habilidades	
<i>Capacidad de análisis, redacción, organizacional, comprensión de textos e ideas principales.</i>	

Actividades de la práctica y sus tareas asociadas.

- 1. Proponer las actividades que se realizarán para completar los requerimientos o partes de los requerimientos faltantes o incompletos.**
 - Analizar el requerimiento o sus partes faltantes para asignar las actividades para completarlas.
 - Realizar la lista de las actividades del punto anterior para cumplir con los requerimientos incompletos.
- 2. Colocar la lista en un lugar que este a la vista y que sea de fácil acceso o consulta de todo el equipo.**
 - Asegurar que la lista la tenga cada miembro del equipo y que sea fácil de consultar. Puede ser a través de un tablero kanban físico o digital.
- 3. Establecer una revisión de los elementos o verificar las actividades y sus resultados conforme el paso del tiempo.**
 - Está actividad sería mejor si se realizara semanas después de la identificación de la deuda existente o junto con la retrospectiva.
 - Verificar las actividades y actualizar la lista de deuda técnica en los requerimientos.
 - Quitar o tachar las actividades ya cumplidas y agregar los nuevos requerimientos identificados con deuda técnica.

Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

- 1. Realizar la lista de las actividades que se realizarán para completar los requerimientos o partes de los requerimientos faltantes o incompletos.**

ID	Requerimiento incompleto	Actividad propuesta	Por hacer	Hecho

Preguntas de validación de la tercera práctica.

1. ¿Usted realizaría en conjunto con su equipo, una lista de actividades, como se indica en la práctica, que ayudarían a complementar las tareas o requerimientos faltantes y llevarla a cabo?
2. Realizar y administrar una “lista de deuda técnica en los requerimientos” ¿aportaría alguna mejora o motivación en la forma de trabajar de usted o su equipo?
3. ¿En qué beneficiaría a su proyecto un **manejo consciente** de la deuda Técnica en los requerimientos?

Práctica de Retrospectiva.

La *Práctica de Retrospectiva* puede ayudar identificar si las tareas y actividades que se llevaron a cabo para la gestión de la deuda técnica en los requerimientos se realizaron de manera correcta y se obtuvo un beneficio, así como brindar retroalimentación inmediata de los errores incurridos durante la aplicación de estas para volver a plantear soluciones o mejoras sobre las actividades que presentaron dificultades durante el desarrollo del proyecto y en específico, en la gestión de la deuda técnica en los requerimientos.

P4	Práctica
<i>Práctica de Retrospectiva.</i>	
Objetivo	
<i>El objetivo de esta práctica es estar conscientes de las actividades que se están realizando bien o se realizaron bien para gestionar la deuda técnica en los requerimientos, enfatizar las actividades que no han producido resultados esperados y proponer mejoras para el trabajo a futuro.</i>	
Entrada	Resultado
<i>Resultados y fuentes internas de información (miembros del equipo) como opiniones, comentarios, etc.</i>	<i>Lista de actividades que se hicieron bien y actividades a mejorar a futuro.</i>
Actividades	
<ol style="list-style-type: none"><i>1. Realizar una lista de actividades que el equipo realizó o está realizando bien de acuerdo con la gestión de la deuda técnica en los requerimientos.</i><i>2. Identificar las actividades que no están generando resultados esperados o deseados.</i><i>3. Realizar lista de mejoras a implementar en las actividades del punto 2.</i>	
Herramientas (opcional)	
<i>Repositorio físico o digital Fuentes internas de información.</i>	
Conocimientos y habilidades	
<i>Capacidad de análisis, redacción, comprensión de textos e ideas principales.</i>	

Actividades de la práctica y sus tareas asociadas.

1. Realizar e identificar la lista de actividades que se realizaron bien o se están realizando bien para la gestión de la deuda técnica y las actividades de las cuales no se están obteniendo buenos resultados.
 - Enlistar las actividades que se realizaron en el proceso de gestión de la deuda técnica en los requerimientos y si existe alguna otra actividad complementaria que se implementó, añadirla a la lista.
 - Mencionar las actividades y recibir opiniones de los integrantes del equipo de acuerdo con su perspectiva, si la actividad implementada tuvo un efecto positivo o negativo en el proyecto.
 - De acuerdo con las opiniones emitidas marcar la actividad como realizada bien o negativa.
2. Realizar la lista de propuestas de mejoras para las actividades, de la lista del punto 1 marcadas como negativas.
 - Enlistar nuevamente las actividades que no dieron los resultados esperados.
 - Escribir propuestas de mejorar algunas actividades para obtener un resultado que pueda beneficiar al proyecto.

Conocimientos generales y sugerencias de herramientas para la implementación de la práctica.

1. Enlistar las actividades que se realizaron en el proceso de gestión de la deuda técnica en los requerimientos.

ID	Actividad	Positiva	Negativa	Observaciones

2. Enlistar las actividades en las que no se obtuvieron los resultados esperados y se proponen mejoras que puedan ayudar a obtener buenos resultados en la gestión de la deuda técnica en los requerimientos.

ID	Actividad	Propuesta de mejora	Observaciones

Preguntas de validación de la cuarta práctica.

1. Realizar una retrospectiva del trabajo realizado ¿le brindaría información importante sobre las actividades realizadas y, si estas actividades dieron un resultado esperado y cuáles no?
2. Considera que en la actividad 1 con la tabla, brinda suficiente información al enlistar las actividades que se propusieron y detectar cuales tiene un efecto positivo o negativo en el avance del proyecto, después de haberlas implementado.
3. En la tabla de la actividad 2, en su opinión ¿considera relevante obtener información de las actividades que no dieron resultados positivos y proponer mejoras a estas actividades?

Cuestionario general del método de aplicación.

1. ¿Considera que el método de gestión de la deuda técnica en los requerimientos es aplicable?
2. ¿El método es entendible?
3. En su opinión, ¿el método de gestión está completo?
4. Considera usted que se podría agregar alguna otra opción faltante al método o a alguna práctica, que pueda complementarlo(a).

Bibliografía

[5] Norma Técnica Peruana, “Perfiles Del Ciclo De Vida Para Las Pequeñas Organizaciones Guía De Gestión De Ingeniería,” Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual, no. Lima41, p. 11, 2006. [Online]. Available: <http://bvirtual.indecopi.gob.pe/normas/29110-5-1-2.pdf>

[6] G. Ibargüengoitia, “Conceptos básicos de la ESENCIA y Kuali-Beh Contenido.”

[7] Z. Li, P. Avgeriou, and P. Liang, “A systematic mapping study on technical debt and its management,” *The Journal of Systems & Software*, vol. 101, pp.193–220, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.jss.2014.12.027>

[18] I. ISO and IEEE, “ISO/IEC/IEEE 29148:2011,” 2011. [Online]. Available: <https://standards.ieee.org/findstds/standard/29148-2011.html>

[21] H. Oktaba, M. Morales, and M. Dávila, “KUALI-BEH: Software Project Common Concepts,” no. August, 2012. [Online]. Available: http://scholar.google.com.mx/scholar?hl=en{\&}q=Kuali-Beh{\&}btnG={\&}as{_}sdt=1{\%}2C5{\&}as{_}sdt={\#}0

A. PROPUESTA DE VALIDACIÓN PARA EXPERTOS DEL MÉTODO DE GESTIÓN DE LA DEUDA TÉCNICA EN LOS REQUERIMIENTOS

Acaba documento.

Bibliografía

- [1] M. Fowler, “Technical Debt Quadrant.” [Online]. Available: <https://martinfowler.com/bliki/TechnicalDebtQuadrant.html>
- [2] P. Kruchten, R. L. Nord, and I. Ozkaya, “Technical Debt : From Metaphor,” *IEEE Software*, pp. 18–22, 2012.
- [3] N. S. Alves, L. F. Ribeiro, V. Caires, T. S. Mendes, and R. O. Spínola, “Towards an ontology of terms on technical debt,” *Proceedings - 2014 6th IEEE International Workshop on Managing Technical Debt, MTD 2014*, pp. 1–7, 2014.
- [4] Z. Li, P. Avgeriou, and P. Liang, “A systematic mapping study on technical debt and its management,” *The Journal of Systems & Software*, vol. 101, pp. 193–220, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.jss.2014.12.027>
- [5] Norma Técnica Peruana, “Perfiles Del Ciclo De Vida Para Las Pequeñas Organizaciones Guia De Gestion De Ingenieria,” *Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual*, no. Lima 41, p. 11, 2006. [Online]. Available: <http://bvirtual.indecopi.gob.pe/normas/29110-5-1-2.pdf>
- [6] G. Ibarg, “Conceptos básicos de la ESENCIA y Kualí-Beh Contenido.”
- [7] W. Cunningham, “The WyCash portfolio management system.” in *Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum)(OOPSLA '92)*, 1992, pp. 29–30. [Online]. Available: <https://doi.org/10.1145/157709.157715>
- [8] SonarQube SA, “SonarQube,” 2008. [Online]. Available: <https://www.sonarqube.org/>

BIBLIOGRAFÍA

- [9] PMD, “PMD,” 2020. [Online]. Available: <https://pmd.github.io/>
- [10] S. McConnell and C. S. B. Inc., “Managing technical debt,” *Construx, Software Development Best Practices*, vol. 10, no. 3.
- [11] J. C. Carver, J. Cabot, R. Capilla, and H. Muccini, “GitHub, Technical Debt, Code Formatting, and More,” *IEEE Software*, vol. 34, no. 2, pp. 105–107, 2017.
- [12] Scrummanager.net, “Deuda técnica,” 2014. [Online]. Available: https://www.scrummanager.net/bok/index.php?title=Deuda_\}t\{'e\}cnica
- [13] R. E. Fairley and M. J. Willshire, “Better Now Than Later: Managing Technical Debt in Systems Development,” *Computer*, vol. 50, no. 5, pp. 80–87, 2017.
- [14] Philippe Kruchten, “Refining the definition of technical debt.” 2016. [Online]. Available: <https://philippe.kruchten.com/2016/04/22/refining-the-definition-of-technical-debt/>
- [15] N. Brown, Y. Cai, Y. Guo, R. Kazman, M. Kim, P. Kruchten, E. Lim, A. MacCormack, R. Nord, I. Ozkaya, R. Sangwan, C. Seaman, K. Sullivan, and N. Zazworka, “Managing technical debt in software-reliant systems,” *Proceedings of the FSE/SDP Workshop on the Future of Software Engineering Research, FoSER 2010*, pp. 47–51, 2010.
- [16] S. McConnell, “Managing Technical Debt (White Paper),” *Workshop on Managing Technical Debt (part of ICSE 2013)*, no. June, pp. 1–14, 2013. [Online]. Available: <http://2013.icse-conferences.org/documents/publicity/MTD-WS-McConnell-slides.pdf>
- [17] N. A. Ernst, “On the Role of Requirements in Understanding and Managing Technical Debt,” *2012 Third International Workshop on Managing Technical Debt (MTD)*, pp. 61–64, 2012.
- [18] I. ISO and IEEE, “ISO/IEC/IEEE 29148:2011,” 2011. [Online]. Available: <https://standards.ieee.org/findstds/standard/29148-2011.html>
- [19] P. Zave, “Classification of research efforts in requirements engineering,” *Proceedings of the IEEE International Conference on Requirements Engineering*, no. May, pp. 214–216, 1995.
- [20] J Leite, “Ingeniería de Requisitos,” *Notas de Cátedra*, 1997.

- [21] IEEE COMPUTER SOCIETY, *Swebok - Guía al cuerpo de conocimiento de la Ingeniería de Software*, 2004.
- [22] M. G. L. F. S. J. A. C. V. J. C. B. Piattini, *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión.*, 1st ed., Spain, 1996.
- [23] I. Kotonya, G; Sommerville, *Requirements Engineering: Processes and Techniques.* by Wiley, 1998.
- [24] D. Baklizky, “The COSMIC Functional Size Measurement Method Versions 4.0.1/4.0.2 Guideline for Sizing Business Application Software,” no. May, pp. 1–76, 2017.
- [25] Ian Sommerville, *Software Engineering 8*, 7th ed., 2007.
- [26] I. ISO and IEEE, “ISO/IEC 15288:2008 (IEEE Std 15288-2008) Systems and software engineering — System life cycle processes,” 2008.
- [27] H. Femmer, D. Méndez Fernández, S. Wagner, and S. Eder, “Rapid quality assurance with Requirements Smells,” *Journal of Systems and Software*, vol. 123, pp. 190–213, 2017.
- [28] H. Oktaba, M. Morales, and M. Dávila, “KUALI-BEH: Software Project Common Concepts,” no. August, 2012. [Online]. Available: http://scholar.google.com.mx/scholar?hl=en&q=Kuali-Beh&btnG=&as{_}sdt=1{\\%}2C5{\\&}as{_}sdt={\\#}0
- [29] V. Lenarduzzi, T. Besker, D. Taibi, A. Martini, and F. A. Fontana, “Technical Debt Prioritization: State of the Art. A Systematic Literature Review,” 2019. [Online]. Available: <http://arxiv.org/abs/1904.12538>