



**Universidad Nacional de La Matanza
Escuela de Posgrado
Maestría en Informática**

Mitigación de amenazas a requisitos en el desarrollo global de software usando LEL y Escenarios

Autor: Ing. Juan Pablo Mighetti

Directora: Dra. Graciela D. S. Hadad

Mayo 2019

Tabla de Contenido

Tabla de Contenido	2
Resumen	4
1. Introducción	5
1.1 Motivación	5
1.1.1 <i>Los requisitos dentro del proceso de desarrollo</i>	5
1.1.2 <i>Orígenes del desarrollo global de software</i>	6
1.1.3 <i>Consideraciones sobre el desarrollo global de software y los requisitos</i>	6
1.1.4 <i>Ámbito de desarrollo de la tesis</i>	8
1.2 Problemática analizada	11
1.3 Antecedentes	12
1.4 Justificación del estudio	12
1.5 Alcance del trabajo	13
1.6 Objetivo	13
1.7 Hipótesis	15
2. Marco Teórico	16
2.1 Introducción a los modelos del proceso de ingeniería de requisitos	16
2.2 Léxico Extendido del Lenguaje	18
2.3 Características del LEL	19
2.4 Construcción del LEL	21
2.5 Escenarios	24
2.6 Características de los escenarios	24
2.7 Escenarios actuales y escenarios futuros	27
2.8 Utilidad de los escenarios	28
2.9 Amenazas, vulnerabilidad y riesgos	29
2.10 El desarrollo global de software	31
2.11 Amenazas del desarrollo global de software	32
3. Análisis de un proyecto de desarrollo global sin mitigación de amenazas	45
3.1 Características del proyecto Venezuela	45
3.2 Métricas definidas para los proyectos bajo estudio	52
3.3 Características de las especificaciones del proyecto Venezuela	56
3.4 Resultados de métricas obtenidas del Proyecto Venezuela	60
3.5 Tiempo de creación de especificaciones de requisitos para el proyecto Venezuela	61

3.6	Aprobaciones y rechazos de especificaciones de requisitos para el proyecto Venezuela.....	63
3.7	Transición de especificaciones de requisitos a la fábrica de software para el proyecto Venezuela.....	66
3.8	No conformidades en la certificación de usuarios para el proyecto Venezuela 72	
3.9	Análisis de resultados del Proyecto Venezuela.....	75
3.10	Conclusiones del análisis del Proyecto Venezuela.....	77
<hr/>		
4.	Propuesta de Mitigación de Amenazas a Requisitos aplicada en otro Proyecto.....	83
4.1	Características del proyecto Brasil.....	83
4.2	Acciones de mitigación de las amenazas mediante LEL y Escenarios.....	84
4.3	Introducción de mejoras al proceso de software en el proyecto Brasil.....	94
4.4	Resultados obtenidos al implementar los modelos LEL y Escenarios en el proyecto Brasil.....	98
4.5	Tiempo de creación de especificaciones de requisitos para el proyecto Brasil 99	
4.6	Aprobaciones y rechazos de especificaciones de requisitos para el proyecto Brasil.....	101
4.7	Transición de especificaciones de requisitos a la fábrica de software para el proyecto Brasil.....	103
4.8	No conformidades en la certificación de usuarios para el proyecto Brasil.....	107
4.9	Análisis comparativo de métricas entre ambos proyectos.....	109
4.10	Comparación de tiempos de creación de especificaciones de requisitos.....	110
4.11	Comparación de aprobaciones y rechazos de especificaciones de requisitos 112	
4.12	Comparación en la transición de especificaciones de requisitos a la fábrica de software.....	114
4.13	Comparación de no conformidades en la certificación de usuarios.....	116
4.14	Resumen comparativo de métricas.....	117
<hr/>		
5.	Conclusiones.....	122
6.	Referencias.....	130
6.1	Referencias Bibliográficas.....	130
6.2	Índice de Figuras.....	139
6.3	Índice de Tablas.....	140
<hr/>		
Anexo A – Proceso de software en el proyecto Venezuela.....		142
Anexo B – Proceso de software en el proyecto Brasil.....		148

Resumen

La especificación de requisitos es la entrada para las siguientes etapas del proceso de desarrollo de software, por lo que la calidad de la misma es vital para el éxito del proyecto, entendiendo que el éxito del mismo consiste en satisfacer acabadamente los requisitos. Los proyectos de desarrollo global de software, se ven afectados por diversas amenazas, tales como distancia geográfica y temporal, diferentes culturas e idiomas de los involucrados, entre otros factores, las que afectan e impactan en forma directa e indirecta en los requisitos.

Es importante conocer cuáles son las amenazas a las que se está expuesto cuando se trabaja bajo el modelo distribuido y cuál es el nivel de exposición que se tiene; este último estará dado, entre otros aspectos, por el conocimiento del equipo de trabajo en el dominio del problema, las diferencias idiomáticas y culturales. Conocer tardíamente estas amenazas y/o subestimarlas, podrá traer consecuencias nefastas para los requisitos, en primer lugar, y para todo el proyecto en general, tanto en calidad como en tiempo y costos. Mientras que conocerlas con anticipación y tomar las medidas adecuadas para su mitigación puede ayudar a conseguir los resultados esperados.

En tal sentido, se ha elaborado una propuesta utilizando los modelos Léxico Extendido del Lenguaje y Escenarios, con el fin de mitigar las amenazas a los requisitos en proyectos de desarrollo global. La propuesta fue puesta a prueba en un proyecto real de gran envergadura, obteniéndose resultados suficientemente promisorios al compararlos contra los de otro proyecto de similares características donde no se había tomado en consideración las amenazas a requisitos.

1. Introducción

1.1 Motivación

A continuación, se describen los motivos que llevaron a plantear y desarrollar el presente trabajo de tesis, en función de las complejidades que presentan los diversos procesos de ingeniería de requisitos, a los que se le suman las particularidades inherentes al desarrollo global de software.

1.1.1 Los requisitos dentro del proceso de desarrollo

La industria en sí tiene como propósito cubrir una necesidad existente en un determinado ámbito, y la industria del software no escapa a esta regla. Coincidentemente es necesario entender el problema para poder elaborar las mejores soluciones potenciales para ello y generar las especificaciones necesarias para la construcción del producto software. Por lo tanto, el entendimiento del problema y la elaboración de las especificaciones de requisitos son pasos claves dentro del desarrollo de software (Brooks, 1987).

Los requisitos son elementos de entrada para las subsecuentes etapas del proceso de desarrollo y, por lo tanto, una buena construcción de requisitos es vital para la salud del proyecto.

Dicha construcción incluye la práctica de obtener los requerimientos de los usuarios, clientes o sponsors, lo cual implica que en la construcción de requisitos está involucrado un componente humano importante. En la mayoría de los casos, el conocimiento está disponible en las personas, lo cual sumado a otros factores como problemas para identificar el alcance del sistema, insuficiente información, falta de experiencia de las partes intervinientes, disponibilidad limitada e intereses contrapuestos por parte de los usuarios o patrocinadores, ambigüedad e inconsistencia en la información capturada, bajo compromiso en la organización cliente, entre otros, generan inconvenientes diversos que atentan contra los plazos pactados, los costos del proyecto y la calidad de un buen producto software.

Es lógico decir que los requisitos, al igual que otros artefactos del proceso de software,

pueden estar amenazados por factores negativos, pero a diferencia de muchos otros, estos tienen un impacto mucho más nocivo cuanto más tarde son identificados en el proceso de desarrollo de software, y se agrava con el software en operación (Boehm, 1981).

1.1.2 Orígenes del desarrollo global de software

Trasladar las actividades de desarrollo fuera de los límites del país de origen es cada vez más común y sigue tomando cada vez más fuerza (Herbsleb, 2007). Actualmente continúa como una alternativa vigente que proporciona buenos resultados. Principalmente consiste en trasladar las tareas de desarrollo de software a otras locaciones, usualmente con diferentes culturas, idiomas, husos horarios, donde los equipos de trabajo locales y remotos colaboran con un mismo objetivo, comúnmente interactuando en forma remota (Ul Haq, y otros, 2011).

El crecimiento de la conectividad, particularmente fibra óptica intercontinental, y el esparcimiento de internet hacen que los costos asociadas a este ramo se reduzcan drásticamente (Fryer, y Gothe 2008). Principalmente, estos dos factores han dado mayor impulso al desarrollo global del software.

Décadas atrás algunas organizaciones comenzaron a experimentar con empresas de desarrollo de software localizado en forma remota, lo que les permitía acceso a mano de obra calificada a menores costos (Prikladnicki, y otros, 2003). Como dato anecdótico se puede mencionar que dentro de las 50 ubicaciones más competitivas para trabajar en forma offshore se encuentran 5 países latinoamericanos: Argentina, Brasil, Chile, Costa Rica y México (Kearney, 2009), (Gereffi, y otros, 2009).

El incremento de la competencia, generado por la globalización, obliga cada vez más a las compañías a buscar nuevas alternativas para ser competitivas, las cuales son evaluadas por las diferentes empresas multinacionales y consultoras de informática.

1.1.3 Consideraciones sobre el desarrollo global de software y los requisitos

El mundo tecnológico está en constante crecimiento y pone a disposición de las empresas

elementos que permiten que el trabajo des-localizado sea cada vez más atractivo, porque permiten entre otras cosas (Fryer, y Gothe 2008):

Posible reducción de costos.

Incrementar la flexibilidad y competencias profesionales de los equipos.

Lograr una rápida adaptabilidad por parte de la empresa, a los cambios de mercado para el logro de objetivos de corto plazo.

Competitividad mediante la reducción de costos en la creación de soluciones.

Flexibilidad en los modelos de contrataciones.

Mitigación de riesgos de carácter laboral e impositivo.

Posibilidad de expansión a nuevos mercados.

En general, el beneficio de un proyecto de desarrollo global de software, también denominado proyecto *multi-site*, radica en obtener una ventaja competitiva al aumentar la velocidad en las entregas y reducir los costos mediante el aprovechamiento de recursos calificados pero de menor costo, la tercerización con subcontratistas experimentados y la superposición de zonas horarias. A medida que más empresas adoptan esta forma de trabajo, no hacerlo se convierte en una desventaja competitiva (Fryer, y Gothe 2008).

Por otro lado, el avance en las telecomunicaciones, internet y herramientas colaborativas, como video conferencias, tablets o pizarras compartidas, permiten en tiempo real simular estar en un mismo lugar físico.

Si bien este modo de trabajo está cada vez más profundizado, no deja de ser una tarea sumamente compleja, en donde existen diversas amenazas que impactan en forma negativa en las diferentes etapas y componentes del proceso de software. Algunas de estas amenazas suelen ser idioma, cultura, confianza, comunicación, huso horario, distancia, entre otros (Aranda, y otros, 2008), (Richardson, y otros, 2012), (Damian, y otros, 2003).

Estas amenazas pueden dar lugar a ambigüedades, errores, contradicciones y omisiones en los requisitos y afectar a todo el proceso de software en general. Dado que los requisitos son las capacidades que un sistema debe satisfacer y su correcto cumplimiento determina el

éxito o el fracaso del proyecto (Oberg, y otros, 2000), entonces estas amenazas deberían ser identificadas, analizadas en profundidad y mitigadas.

Para entender la dimensión de los requisitos dentro del proceso de software, se puede citar que los problemas asociados al proceso de ingeniería de requisitos son la principal causa de fracaso de los proyectos (Oberg, y otros, 2000). Cerca del 70% de los requisitos son difíciles de identificar y un 54% no están claros ni bien organizados (Oberg, y otros, 2000).

1.1.4 Ámbito de desarrollo de la tesis

El trabajo que se presenta en esta tesis surge como consecuencia de haber identificado, en etapas tardías de un proyecto software real, un impacto significativo sobre la calidad del software, los plazos estimados y el presupuesto.

Dicho proyecto, es parte de un programa de transformación regional ya en marcha y tiene por objetivo la creación de un software para la industria de seguros que permita la unificación de las plataformas regionales de Latinoamérica. El programa tiene una duración prevista de 5 años, dispone de un presupuesto total de 60 millones de dólares: 20 millones para el primer año y 10 millones para los subsiguientes y al momento de la escritura de esta tesis se encuentra en su tercer año de ejecución. El programa está caracterizado por poseer varios de los equipos de trabajo en diversas zonas geográficas; no solo en Latinoamérica sino también en Europa y Asia, en total 7 locaciones en 6 países. El mismo también contempla la incorporación paulatina de los diferentes países, comenzando en principio por Venezuela, continuando secuencialmente, y con algún solapamiento, con Brasil, ambos con su operación diaria en marcha, y siguiendo con el resto de los países. Estos dos países identifican a los proyectos donde se dispone de los destinatarios del software y, por ende, de los usuarios, con quienes se realizan las actividades de elicitación. A su vez, el proyecto planea la generación de releases, donde cada release contemple funcionalidades de las principales áreas de la compañía.

La distribución geográfica de los involucrados en este programa regional de software se presenta en la Figura 1, donde están incluidos también los proveedores de tecnología.



Figura 1 - Distribución geográfica de involucrados

El proyecto inicial de Venezuela, el cual ya tiene su primera fase cerrada, y se está en proceso con la segunda fase, se trató principalmente de una migración y unificación tecnológica con aproximadamente un 20% de incremento de alcance que poseen las plataformas locales de cada país, por lo que se optó por un ciclo de vida en cascada (Royce, 1970). En la Figura 2 se muestra el ciclo de vida para el proyecto Venezuela, que fue también utilizado para el proyecto Brasil.

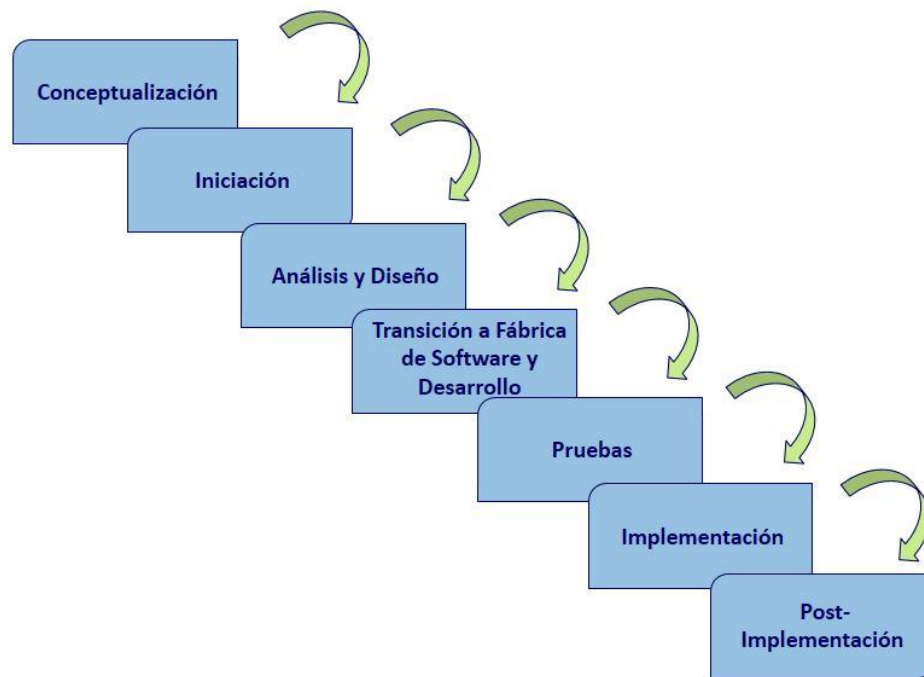


Figura 2 - Ciclo de vida de los proyectos Venezuela y Brasil

Durante la certificación de usuarios (subproceso *Pruebas*), en el proyecto Venezuela, se identificaron numerosos rechazos por parte de los usuarios locales, que encendieron algunas alarmas para avanzar con el siguiente proyecto, cuyos usuarios eran de Brasil, y que se encontraba a punto de comenzar. Estos rechazos fueron debidamente documentados a fin de calcular las métricas definidas dentro del proyecto, procesarlas y tomar acciones que permitieran mejorar el proceso. En base a la experiencia en el proyecto Venezuela, se analizaron las causas que provocaron rechazos y se estableció una propuesta para reducir los mismos, la que se aplicó en el proyecto Brasil.

Inicialmente el proyecto Brasil estaba planteado para iniciarse con 8 meses de diferencia al proyecto Venezuela, pero finalmente se inició con 13 meses de desfasaje debido a la necesidad de implementar las acciones de mitigación adecuadas. Al ser un proyecto de similares características, en cuanto a los servicios de software a proveer, se optó por el mismo ciclo de vida presentado en la Figura 2.

1.2 Problemática analizada

La globalización es cada vez más usual para muchas industrias (Fryer, y Gothe 2008). Las comunicaciones de alta velocidad como la fibra óptica, la accesibilidad de viajes, oportunidades globales y otras ventajas tecnológicas y sociales, han hecho que las compañías se expandan globalmente (Fryer, y Gothe 2008). Tradicionalmente las actividades de proyectos software se llevaban a cabo en una misma locación, posteriormente se evaluaron alternativas como fragmentar o des-localizar los procesos de desarrollo en diferentes ubicaciones o países, lo que se conoció inicialmente como offshoring (Gereffi, y otros, 2009).

Si bien la mayoría de las empresas conoce las amenazas relacionadas con la producción de los requisitos (Goguen, y otros, 1993) (Nuseibeh y Easterbrook, 2000), muchas veces no tienen en cuenta aquellas nuevas que aparecen o las subestiman, tales como las amenazas potenciadas por la modalidad de trabajo multi-site (Nonyelum Ogwueleka, 2012).

Si se analiza en profundidad la difícil tarea de construir los requisitos, donde se enfrentan inconvenientes, tales como interpretar correctamente a los usuarios en su necesidad (Nuseibeh y Easterbrook, 2000), comprender la complejidad del contexto en donde se encuentran inmersos esos requisitos (Jackson, 2009), identificar potenciales omisiones (Basili, y otros, 1981) (Westland, 2002), reducir problemas de comunicación (Leite, y Franco 1993), obtener la disponibilidad y el interés necesarios de los involucrados, identificar e implementar las herramientas adecuadas (Kaindl, y otros, 2002), minimizar la comunicación asincrónica, entre otras dificultades, entonces se pone en evidencia que el proceso de ingeniería de requisitos es por sí solo sumamente desafiante (Nuseibeh y Easterbrook, 2000).

Es necesario entender que la actividad de elicitación involucra una comunicación donde generalmente los puntos de vista respecto a la necesidad y a la solución no suelen ser los mismos para los usuarios y para los analistas (Leite, y Freeman 1991). La comunicación se advierte aún más amenazada cuando se está bajo un modelo distribuido (Boland, y otros, 2004), (Richardson, y otros, 2012), (Fryer, y Gothe 2008), (Damian, y otros, 2003), provocando una amenaza mayor sobre la correcta interpretación y construcción de los

requisitos.

Por lo tanto, es necesario identificar las amenazas, ya existentes, nuevas o potenciadas, sobre los requisitos y trabajar sobre la mitigación de las mismas con el propósito de evitar los impactos nocivos de estas.

1.3 Antecedentes

Existen numerosos trabajos en la literatura que ponen en evidencia la compleja tarea de construir los requisitos funcionales y no funcionales de un sistema de software (Van Lamsweerde, y otros, 1998), (Cockburn, 2000), (Leffingwell, y otros, 2003), como así también existe evidencia sobre las dificultades de trabajar en un modelo distribuido de desarrollo de software (Kruchten, 2004), (Damian, y otros, 2003), (Ebert, y otros, 2001), (Richardson, y otros, 2012), (Fryer, y Gothe 2008).

Por otro lado, dado que el presente trabajo de tesis contempla la utilización del modelo de escenarios como parte de la estrategia para la mitigación de las amenazas identificadas, se ha realizado una revisión de los trabajos de los principales exponentes sobre este modelo, con el fin de entender las posibilidades que brinda como así también sus debilidades (Potts, 1995), (Sutcliffe, y otros, 1998), (Kaindl, 2000), (Leite, y otros, 2004), (Hadad, 2008), (Alrajeh, y otros, 2007), (Seyff, y otros, 2009) (Zowghi, 2007).

Toda la bibliografía recolectada fue profundamente revisada, ordenada en forma cronológica y catalogada, estableciéndose puntos en común y diferencias, que permitieron identificar omisiones o carencias en algunos estudios que podrían ser complementados con otros.

Adicionalmente, se tuvieron en cuenta las características de cada uno de los estudios para contrastarlas con las características de este trabajo de tesis.

1.4 Justificación del estudio

La globalización del software empuja cada día más a buscar alternativas rentables fuera de los límites de los países, ya sea mediante la contratación de proveedores, adquiriendo compañías, trabajando colaborativamente con otras filiales, o mediante asociaciones con

otras empresas. Estas decisiones están motivadas por varias causas que poseen beneficios como los presentados en (Fryer, y Gothe 2008), (Prikladnicki, y otros, 2003), (Aranda, y otros, 2008) y enumerados en la sub-sección 1.1.3.

Este conjunto de ventajas va acompañado de amenazas de diversa índole que deben ser monitoreadas para evitar efectos negativos que afecten a los requisitos; efectos que confabulan contra los tiempos, costos y calidad de los proyectos, minimizando las posibilidades de éxito. Como se menciona en (Davis, y otros, 2006), los problemas en la identificación de los requisitos se asocian con ser el principal factor de fracaso del 90 % de los grandes proyectos de software.

El desconocimiento de estas amenazas y la no mitigación de las mismas puede afectar los requisitos y llevar a la pérdida de las ventajas competitivas, la desmotivación del equipo de trabajo y la obtención de un software de escasa calidad (Richardson, y otros, 2012).

Por lo cual, este estudio persigue principalmente la identificación de las principales amenazas a los requisitos bajo el modelo de trabajo distribuido, sus causas e impacto en los requisitos, junto con la presentación de las técnicas y las ventajas que aportan el uso de los modelos Léxico Extendido del Lenguaje (LEL) (Leite, y Franco 1993) y Escenarios (Leite, y otros, 2000) en la mitigación de las amenazas.

1.5 Alcance del trabajo

El alcance de este trabajo está enmarcado en las amenazas identificadas en un proyecto real de desarrollo regional bajo la modalidad multi-site, de grandes dimensiones con equipos distribuidos en 7 locaciones diferentes.

Específicamente, se centra en la problemática de los requisitos de software a especificar en el marco de desarrollos globales de software. La propuesta de solución para mitigar estas amenazas se basa en el uso de modelos escritos en lenguaje natural como parte de un proceso de ingeniería de requisitos bien definido y difundido (Leite, y otros, 2004).

1.6 Objetivo

El objetivo general del presente trabajo es:

Establecer las mejoras que pueden lograrse en la mitigación de las amenazas a los requisitos en proyectos de desarrollo global, mediante el uso del LEL y de los Escenarios.

Como sub-objetivos del trabajo se han planteado:

Identificar las virtudes que tienen los modelos LEL y Escenarios.

Identificar las principales amenazas sobre los requisitos de software en el desarrollo global de software, y las sub-amenazas derivadas de las principales que afectan a los requisitos en el desarrollo global de software.

Obtener una medida cuantitativa del impacto de las amenazas identificadas en el modelo distribuido sobre los requisitos.

Identificar la mitigación que permiten el LEL y los Escenarios sobre las amenazas respecto a los requisitos en el desarrollo global de software.

Cuantificar cómo el LEL y los Escenarios pueden contribuir en la mitigación del impacto de las amenazas a los requisitos, en el desarrollo global de software.

Las siguientes preguntas son las que guiaron el presente trabajo:

- 1. ¿Qué papel desempeñan el LEL y los Escenarios en la construcción de requisitos?*
- 2. ¿Cuáles son las principales características del desarrollo global de software?*
- 3. ¿Qué beneficios brindan al desarrollo global de software el LEL y los Escenarios?*
- 4. ¿Cuáles son las principales amenazas que presenta el desarrollo global de software?*
- 5. ¿Cuáles son las sub-amenazas derivadas de las principales en el desarrollo global de software?*
- 6. ¿Cuál es el impacto que estas amenazas pueden tener sobre la calidad de los requisitos?*
- 7. ¿Cómo puede ayudar el uso del LEL y los Escenarios a la mitigación de estas*

amenazas?

1.7 Hipótesis

Es bien conocido que los requisitos son un componente esencial para la generación de un producto software, y su correcta elaboración es vital dentro del proceso de software (Nahar & Student, 2013).

Existen numerosas amenazas que ponen en riesgo la calidad de las especificaciones de requisitos de software y aún cuando estas amenazas son de una gran importancia, se asume que son controlables, es decir, su impacto puede ser cuantificado mediante el uso de métricas específicas y moderado con la incorporación de heurísticas, también específicas.

Se conoce que tanto la utilización del LEL como de los Escenarios, además de ser técnicas de modelado, colaboran con una apropiada elicitación y potencian la comunicación entre los involucrados cuando el proceso de requisitos ocurre en una única locación (Hadad, 2008). Se presupone que el uso de estos dos modelos también influenciarían positivamente en un proceso de requisitos llevado a cabo en un ambiente multi-site como lo es el desarrollo global de software.

2. Marco Teórico

2.1 Introducción a los modelos del proceso de ingeniería de requisitos

En la mayoría de las ingenierías se obtienen productos tangibles, que son previamente caracterizados en maquetas, planos, bosquejos, diagramas u otros modos de representación; posteriormente, mediante estos elementos se planifican los procesos de producción procurando garantizar las expectativas iniciales (Hadad, 2008). El software es en sí mismo una representación del mundo real, un producto intangible, que al igual que en todo proceso de ingeniería, necesita ser representado mediante alguno de los elementos antes mencionados. En este sentido, es necesario que esta representación sea lo más completa, correcta y precisa posible, evitando omisiones, ambigüedades, errores y contradicciones, las que podrían impactar sobre otras etapas posteriores. Está ampliamente probado el incremento exponencial del costo relativo de reparación de requisitos cuanto más tarde en el ciclo de vida sean detectados estos defectos (Boehm, 1981), y confirmado posteriormente por (Westland, 2002). La relación entre el costo de reparación de un defecto en los requisitos y el tiempo transcurrido hasta su detección puede verse en la siguiente Figura 3, según datos de (Boehm, 1981).

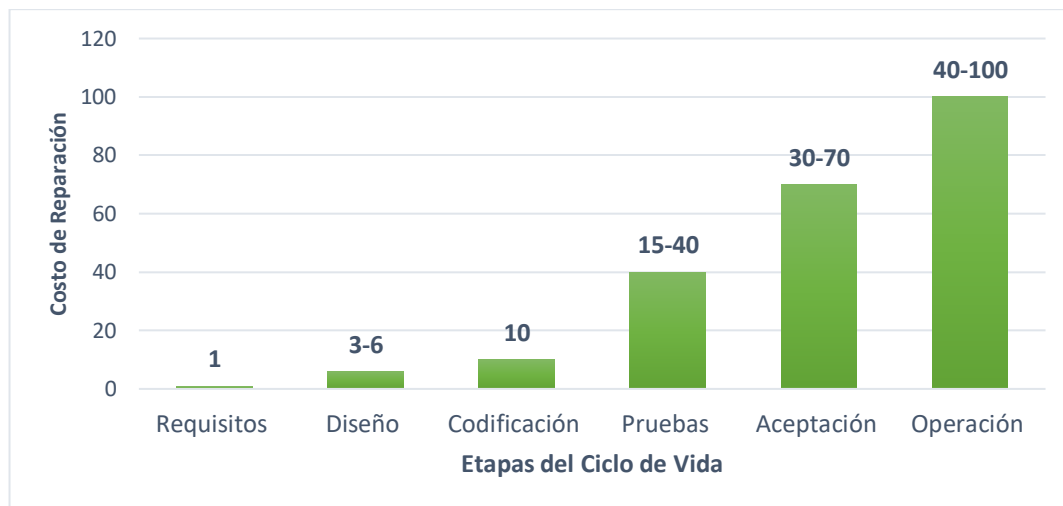


Figura 3 - Costo relativo de reparación de defectos de requisitos a lo largo del tiempo

Aquí es donde la ingeniería de requisitos juega un papel fundamental, pues su intención es mejorar el entendimiento de las necesidades y colaborar con una correcta definición de las

prestaciones y características que debe satisfacer el sistema de software. En otras palabras, el propósito de la ingeniería de requisitos como disciplina es la de estructurar y sistematizar el proceso de producción de requisitos, el cual abarca las actividades de descubrir, analizar, comprender, modelar, validar y mantener el conjunto de requisitos del sistema de software (Hadad, 2008).

El enfoque apropiado frente a la creciente complejidad de los sistemas de software es comprender en profundidad el contexto de aplicación y la necesidad real antes de trabajar sobre la solución a esa necesidad, y la ingeniería de requisitos colabora ampliamente en ese sentido.

Es evidente que la ingeniería de requisitos tiene un ingrediente social de carácter colectivo, el cual incluye a usuarios, analistas, expertos y otros involucrados en el proyecto, donde cada uno de ellos aporta un componente importante al proceso de software.

Los involucrados en este proceso de requisitos, dado su rol dentro de él, poseen diferentes habilidades, conocimiento y vocabularios, que hacen que el entendimiento, fundamental para una correcta representación del sistema y sus necesidades, sea una tarea ardua. Para tener un proceso de ingeniería de requisitos exitoso es vital contar una buena comunicación entre las partes y comprender profundamente el contexto de aplicación (Hadad, 2008).

Este es el punto donde la ingeniería de requisitos aporta métodos, técnicas y herramientas que permitan una colaboración entre todos los involucrados en el proceso de elicitación, especificación, verificación y validación de las condiciones que debe satisfacer el sistema. Los modelos Léxico Extendido del Lenguaje (LEL) y Escenarios están basados en el uso del lenguaje natural, lo cual proporciona ventajas no solo para el proceso de construcción de los requisitos, sino a lo largo de todo el ciclo de vida de software, como un medio de comunicación de fácil entendimiento entre las partes (Hadad, y otros, 1999). El LEL es un glosario que define los términos utilizados en el contexto de aplicación (Leite, y Franco 1993). Respecto a los escenarios, se distinguen los escenarios actuales (Leite, y otros, 2000), que describen las situaciones observadas en el contexto de aplicación, y los escenarios futuros (Hadad, 2008), que describen las situaciones que se proyecta ocurrirán en el contexto de aplicación con la introducción del sistema de software a desarrollar.

La estrategia basada en los modelos LEL y Escenarios se desarrolla en cuatro etapas (Hadad, 2008):

- 8. Comprender el vocabulario del contexto, que implica crear, verificar y validar el LEL;*
- 9. Comprender el contexto actual, que implica crear, verificar y validar escenarios actuales;*
- 10. Definir el contexto del software, que implica crear, verificar y validar escenarios futuros; y*
- 11. Explicitar los requisitos del software, que implica crear, verificar y validar las especificaciones de requisitos.*

2.2 Léxico Extendido del Lenguaje

La construcción de los requisitos es un proceso complejo donde la comunicación juega un papel fundamental en las relaciones humanas, lo que influye notoriamente en la calidad de los requisitos (Hadad, y otros, 1999). En general, existen jergas o lenguajes propios del dominio del problema bajo estudio, con palabras específicas y que tienen un significado particular en ese contexto. Por lo que no entender este lenguaje puede conllevar a defectos en las especificaciones, que derivan, entre otras cosas, en re-trabajo al tener que ser corregidas, y nuevamente verificadas y validadas.

El LEL es un modelo diseñado para colaborar en la elicitación y la representación del lenguaje de la aplicación, y persigue como objetivo el entendimiento del lenguaje utilizado en el contexto sin preocuparse por entender el problema mismo (Leite, y otros, 2000), (aunque facilita colateralmente la comprensión general del contexto de aplicación).

Cuando se tienen diferentes léxicos - el de los usuarios, el de los ingenieros de software y el de otros involucrados - la comunicación tiene una mayor posibilidad de fracasar (Hadad, y otros, 1999). Por el contrario, si los involucrados en un proyecto comparten el mismo lenguaje entonces la comunicación entre ellos mejora (Leite, 1989).

El LEL es una representación del lenguaje natural de aquellas palabras o frases que resultan

peculiares para el grupo de analistas y en definitiva para el proyecto. Adicionalmente, proporciona otras ventajas como ayudar a la relación con los usuarios, mejorando la comunicación con estos, facilita la validación de modelos escritos en lenguaje natural y minimiza las ambigüedades en los modelos y documentos que se generan, entre otros beneficios (Leite, y otros, 2004).

En la Tabla 1 se resumen los objetivos que atiende el Léxico Extendido del Lenguaje, junto con el impacto de dichos objetivos en el proceso de software.

Tabla 1 - Objetivos y consecuencias del LEL (Hadad, y otros, 1999)

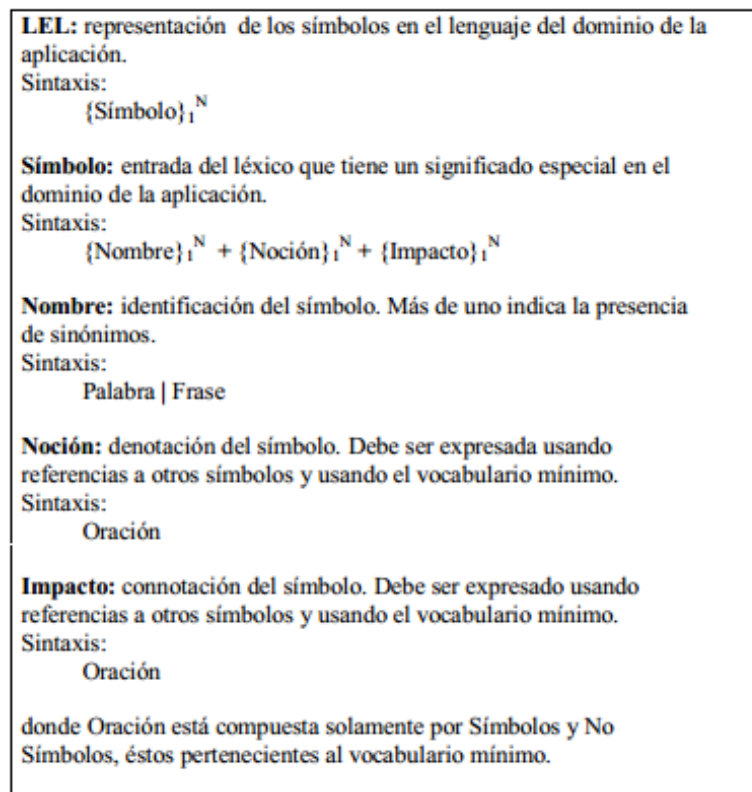
Objetivo	Consecuencia
<i>Conocer el vocabulario del contexto</i>	<ul style="list-style-type: none"> * <i>Asegurar la comunicación</i> * <i>Facilitar la validación de los requisitos con los usuarios</i> * <i>Mantener el mismo vocabulario durante todo el proceso de desarrollo</i>
<i>Contar con un instrumento simple de rastreabilidad</i>	<ul style="list-style-type: none"> * <i>Documentar consistentemente</i> * <i>Capacitar a nuevos miembros del equipo de trabajo en la terminología empleada</i> * <i>Generar versiones del LEL a medida que evoluciona el proceso de desarrollo</i>

Un enfoque interesante es el planteado por Leite (Leite, 1989), quien propone la construcción del LEL como paso inicial del proceso de definición de los requisitos, de esta forma pospone para una fase posterior el entendimiento del problema, y se focaliza en conocer el vocabulario del usuario y su contexto a fin de evitar los problemas básicos relacionados con la comunicación.

2.3 Características del LEL

Durante la lectura de documentación o la interacción con los usuarios se intenta recopilar la mayor cantidad de palabras o frases específicas o peculiares al contexto, cada uno de estos términos serán entradas en el modelo LEL, las que se denominan símbolos.

Cada símbolo está identificado por un nombre o más de uno (en el caso de los sinónimos) y posee dos descripciones la primera llamada “Noción” que describe simplemente qué es el símbolo y la segunda, llamada “Impacto” que describe como el símbolo actúa dentro del contexto de aplicación, es decir, qué connotación tiene dentro del contexto (Leite, y otros, 2000). La Figura 4 muestra el modelo del LEL.



+ significa composición, {x} significa cero o más ocurrencias de x, | representa or

Figura 4 - Modelo del Léxico Extendido del Lenguaje, extraído de (Kaplan, y otros, 2000)

Los conjuntos de símbolos forman una red, que permite representar al LEL en un hipertexto (Leite, y Franco, 1990) y navegar en él para conocer todo el vocabulario del contexto de aplicación.

Al momento de la construcción del LEL tiene que tenerse en cuenta dos principios básicos, llamados “Principio de circularidad” y “Principio de vocabulario mínimo” (Leite, y Franco 1993). El primero propone la utilización de símbolos existentes en el LEL en la descripción de otros símbolos y el segundo propone que se minimice el uso de términos externos al

contexto de la aplicación en la descripción de los símbolos.

2.4 Construcción del LEL

En la Figura 5 se grafica el proceso de construcción del LEL, que involucra actividades básicas de elicitación, modelado, verificación y validación.

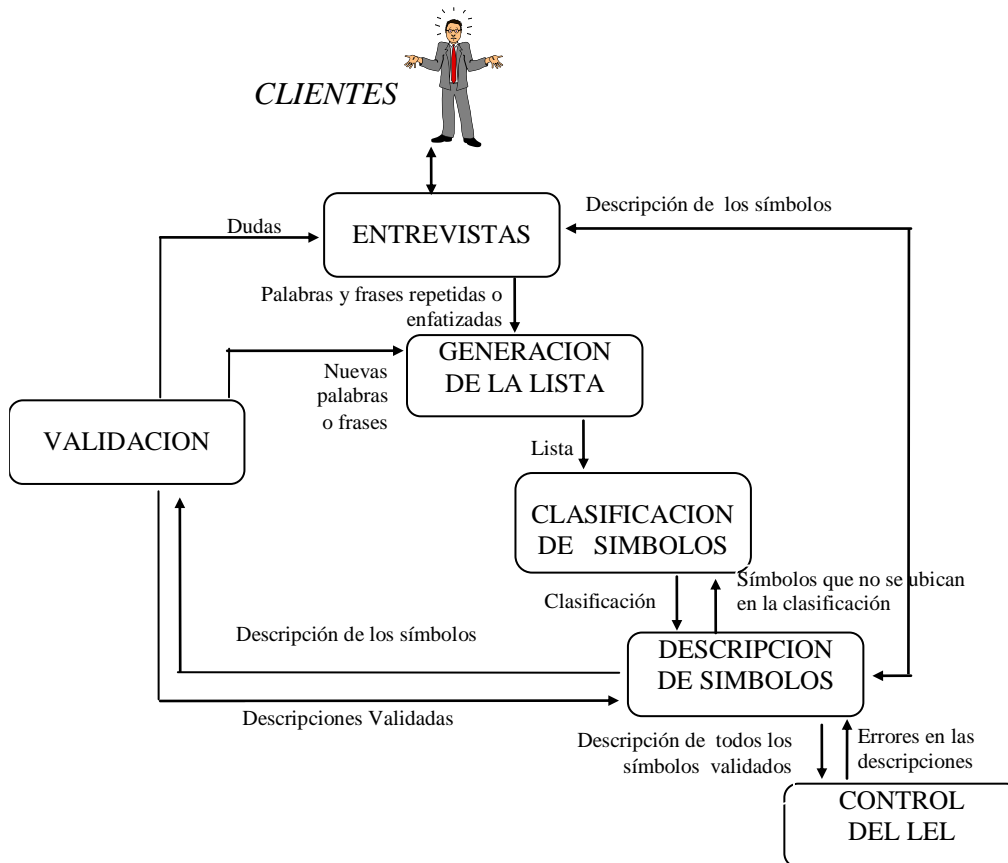


Figura 5 - Etapas para la construcción del LEL, extraída de (Hadad, y otros, 1999)

A continuación se describen las actividades involucradas al construir el LEL, resumidas de (Hadad, y otros, 1999).

a) Entrevistas

Las entrevistas permiten identificar y conocer el vocabulario que los usuarios emplean en su ámbito de trabajo. La cantidad de entrevistas necesarias para la construcción del LEL dependerá de varios factores (Hadad, y otros, 1999) como ser, la complejidad de la

aplicación, la experiencia que el analista tenga en la construcción del LEL y el conocimiento de este respecto al tema en estudio.

Los analistas seleccionados para llevar adelante estas entrevistas deberían ser aquellas que tengan mayor conocimiento sobre el contexto de aplicación, permitiendo de esta forma recopilar la mayor cantidad posible de símbolos y evitar introducir defectos que incrementen luego los tiempos de validación.

Un punto de suma importancia a tener en claro al momento de iniciar estas entrevistas es recordar que esta fase no tiene por finalidad la identificación de funcionalidades del sistema, como puede ser un proceso abocado a la elicitación de conocimiento, sino que el objetivo es familiarizarse con la jerga o el lenguaje del contexto de la aplicación.

b) Generación de la lista de símbolos

Durante las entrevistas se obtienen los símbolos que conforman la lista de símbolos del LEL. Inicialmente se genera una lista de símbolos candidatos que son aquellas frases o palabras que son recurrentes en el discurso del entrevistado.

Esta lista sufre frecuentes modificaciones al avanzar con las distintas entrevistas, en donde el entrevistado proporciona más símbolos y sinónimos. Asimismo, se pueden eliminar símbolos para los cuales se advierte un significado común a cualquier contexto y se clarifican aquellos ambiguos o confusos, hasta obtener la lista definitiva de la cual se pretende que posteriormente sufra solo cambios mínimos.

Es también posible detectar símbolos con el mismo nombre y distintos significados, es decir, puede aparecer el uso de homónimos en un mismo contexto de aplicación.

c) Clasificación de los símbolos

Para asegurar la integridad y la homogeneidad de las definiciones se recurre a la clasificación de los símbolos (Hadad, y otros, 1999). Dicha clasificación estará alineada con la definida en (Leite, y Franco, 1990), donde los símbolos se agrupan en los siguientes tipos: Sujeto, Verbo, Objeto y Estado. Esta tarea es especialmente importante cuando se tiene un grupo de analistas realizando descripciones de distintos símbolos del

LEL, dado que para cada tipo de símbolo se dispone de una plantilla que indica qué información colocar en la noción y qué en el impacto.

d) Descripción de los símbolos

Esta actividad implica describir la noción y el impacto de cada símbolo. Estas descripciones dependen del tipo asignado al símbolo en la actividad previa. Puede ocurrir que un símbolo no pueda ser descripto según el tipo indicado, entonces este debe ser reclasificado.

e) Validación con los usuarios

Las validaciones iniciales posibilitan corroborar la lista candidata de símbolos; las validaciones posteriores complementan las iniciales, de forma tal que se pueda confirmar que el conocimiento obtenido sea el correcto y lo más completo posible; de no ser así, es necesario modificar o ampliar el LEL.

f) Control del LEL

El objetivo de esta actividad es verificar que el LEL construido sea consistente y homogéneo, algo vital cuando participa más de un analista. Este paso es recurrente y sumamente necesario para obtener un LEL de calidad. Una de las técnicas de mayor efectividad que puede aplicarse es la inspección (Kaplan, y otros, 2000), o pueden realizarse revisiones menos formales utilizando, por ejemplo, checklists.

Como conclusión de lo expuesto anteriormente, se observa que la ambigüedad de los modelos y documentos puede ser reducida mediante la utilización del LEL, empleando la terminología contenida en este para la escritura de dichos documentos. A su vez el LEL contribuye a alcanzar una buena comunicación, mejorando el conocimiento del contexto y fortaleciendo el vínculo entre los involucrados. A lo largo de este trabajo de tesis, se presentará cómo este modelo impacta positivamente en la reducción de las amenazas a las que están sometidos los requisitos inmersos en el desarrollo global de software.

2.5 Escenarios

Los escenarios son flujos o descripciones de funcionalidades requeridas del sistema en un momento particular (Hadad, y otros, 1999). Comienzan describiendo situaciones de un macro sistema y evolucionan a través del proceso de construcción de software describiendo como esas situaciones cambian en el macro sistema (Leite, y otros, 2004). Esta afirmación es consistente con la recomendación de (Oberg, y otros, 2000) de desarrollar inicialmente una visión simplista del sistema con el fin de entender sus principales objetivos (Oberg, y otros, 2000).

Por otro lado, los escenarios permiten conocer el problema a resolver o una necesidad insatisfecha, unificar criterios sobre cómo se interactuará con el sistema, incrementar el compromiso de los cliente y usuarios con el proceso mismo de desarrollo, organizar los detalles involucrados en la comprensión del contexto actual y facilitar el entrenamiento a nuevos integrantes del equipo de trabajo (Leite, y otros, 2000).

2.6 Características de los escenarios

Si bien los escenarios no son ni requisitos ni especificaciones funcionales, pueden ser la fuente de conocimiento para nuevos requisitos y la base para las especificaciones funcionales. Asimismo, permiten describir propiedades no funcionales asociándolas a aspectos funcionales.

Por otro lado, al no ser formales pueden estar representados de varias formas: narrativa textual, storyboards, video mock-ups, prototipos escritos o situaciones físicas (Hadad, y otros, 1999), (Rolland, y otros, 1998). Aunque habitualmente, suelen usarse narrativas escritas en lenguaje natural con una cierta estructura. En un principio, son usados para entender el contexto de aplicación, pero posteriormente también son usados para comprender la funcionalidad requerida del problema a resolver, describiendo en ambos casos una situación específica de cómo es o cómo deberá ser el comportamiento en ese contexto (Leite, y otros, 2004).

El modelo de escenario a utilizar en esta tesis está compuesto por título, objetivo, contexto, recursos, actores, episodios y excepciones, como puede observarse en la Figura 6 (Hadad,

2008). Asimismo, cabe destacar que los escenarios se escriben maximizando el uso de términos definidos en el LEL, a fin de reducir la ambigüedad en sus descripciones.

Escenario: descripción de una situación que ocurre en el contexto del problema.
 Sintaxis: Título + Objetivo + Contexto + {Recursos}₁^N + {Actores}₁^N + {Episodios}₂^N + {Excepciones}

Título: identificación del Escenario. En el caso de un sub-Escenario, el título es el mismo que la sentencia episodio, sin las restricciones.
 Sintaxis: Frase | ([Actor | Recurso] + Verbo + Predicado)

Objetivo: finalidad a ser alcanzada. El Escenario describe la forma de lograr el objetivo.
 Sintaxis: [Actor | Recurso] + Verbo + Predicado

Contexto: compuesto por al menos uno de los siguientes ítems:
Ubicación Geográfica: lugar físico donde se produce el Escenario.
Ubicación Temporal: especificación de tiempo para el desarrollo del Escenario.
Precondición: estado inicial del Escenario.
 Sintaxis: {Ubicación Geográfica} + {Ubicación Temporal} + {Precondición}
 donde Ubicación Geográfica es: Frase | Nombre
 donde Ubicación Temporal es: Frase | Nombre
 donde Precondición es: [Sujeto | Actor | Recurso] + Verbo + Predicado

Recursos: elementos físicos o información relevantes que deben estar disponibles en el Escenario.
 Sintaxis: Nombre

Actores: personas, dispositivos o estructuras organizacionales que tienen un rol en el Escenario.
 Sintaxis: Nombre

Episodios: conjunto de acciones que detallan al Escenario y proveen su comportamiento. Un episodio también puede ser descripto como un Escenario.
 Sintaxis (usando BNF parcial):
 <episodios> ::= <serie grupo> | <serie episodios>
 <serie grupo> ::= <grupo> <grupo> | < grupo no secuencial> | <serie grupo> <grupo>
 <grupo> ::= <grupo secuencial> | < grupo no secuencial>
 <grupo secuencial> ::= <sentencia básica> | <grupo secuencial > <sentencia básica>
 <grupo no secuencial> ::= # <serie episodios> #
 <serie episodios> ::= <sentencia básica> <sentencia básica> | <serie episodios> <sentencia básica>
 <sentencia básica> ::= <sentencia simple> | <sentencia condicional> | <sentencia optativa>
 <sentencia simple> ::= <sentencia episodio> CR
 <sentencia condicional> ::= **Si** <condición> **entonces** <sentencia episodio> CR
 <sentencia optativa> ::= [<sentencia episodio>] CR
 donde <sentencia episodio> se describe:
 (([Actor] + Verbo + Predicado) | ([Actor] + [Verbo] + Título)) + {Restricción}

donde **Restricción:** es un requisito de calidad o alcance referido a una dada entidad, y se describe:
 ([Sujeto | Actor | Recurso] + [**No**] **Debe** + Verbo + Predicado) | Frase

Excepciones: generalmente reflejan la falta o mal funcionamiento de un recurso. Una excepción impide el cumplimiento del objetivo del Escenario. Se puede indicar el tratamiento de la excepción a través de un Escenario o de una acción simple.
 Sintaxis: Causa (Solución)
 donde Causa es:
 (Frase | ([Sujeto | Actor | Recurso] + Verbo + Predicado)) + {Nro. Episodio}
 donde Solución es:
 Título | ([Sujeto | Actor | Recurso] + Verbo + Predicado)

+ significa composición, {x} significa cero o más ocurrencias de x, () para agrupar, | para or, [x] significa que x es opcional.

Figura 6 - Modelo de Escenario, extraído de (Hadad, 2008)

En este modelo, los actores no son necesariamente personas sino que representan un rol dentro del contexto de aplicación: ellos son las entidades que hacen uso del sistema para satisfacer las necesidades (Gil, 2002) o que participan en la situación que ocurre en el

contexto de aplicación (Leite, y otros, 2004). A su vez estas necesidades pueden trasladarse a objetivos que representan las condiciones a ser alcanzadas. Los objetivos de los escenarios se alcanzan cumplimentando sus episodios, los cuales representan a su vez un grupo de actividades o tareas asociadas a actores específicos. Dichos episodios están formados gramaticalmente por un grupo de oraciones en lenguaje natural que permite la especificación del comportamiento esperado en el contexto de aplicación. Estos episodios pueden ser simples, condicionales u opcionales. Los episodios simples son aquellos necesarios para concluir satisfactoriamente el escenario; los condicionales son aquellos cuya ocurrencia depende de alguna condición específica; y los opcionales son aquellos que pueden o no ocurrir sin poderse expresar claramente una condición de ocurrencia.

Mientras se llevan a cabo los episodios pueden aparecer excepciones, representando un obstáculo para el cumplimiento del objetivo. Estas excepciones se especifican indicando la causa de la excepción y el tratamiento que se da a la misma. El tratamiento puede describirse mediante una simple oración o la inclusión del título de otro escenario, el que será el encargado de tratar esta excepción.

El contexto representa el entorno en el cual está inmerso el escenario pudiendo detallar su localización geográfica o temporal o alguna precondition necesaria.

El nivel de detalle requerido de cada escenario dependerá de dos factores (Hadad, y otros, 1999):

- ✓ *El grado de importancia que el usuario le otorgue a los hechos específicos del problema.*
- ✓ *La fase en la que se encuentra el proceso de desarrollo.*

Al igual que los requisitos, los escenarios están relacionados unos con otros y asociados a otros entregables dentro del proceso de desarrollo (Oberg, y otros, 2000), es decir, los escenarios no son independientes entre sí, aunque representen una descripción particular de un comportamiento del sistema. Cada escenario tiene una relación semántica con los otros escenarios (Booch, 1994).

2.7 Escenarios actuales y escenarios futuros

Los escenarios actuales son aquellos que describen circunstancias actuales que tienen lugar en el contexto de aplicación (Hadad, 2008). Los escenarios futuros modelan el comportamiento esperado del sistema e interactúan fuertemente con los objetivos del sistema de software (Hadad, 2008).

Durante la etapa inicial de la creación de los escenarios actuales, el LEL aporta información útil que sirve para la primera versión de los escenarios actuales (Hadad, 2008). Adicionalmente, en las organizaciones y a disposición de los proyectos existe diversa información que puede ser usada para ampliar y mejorar los escenarios, como por ejemplo; documentación de procesos, flujos, reportes, imágenes, bosquejos, sistemas de software en uso, entre otros.

Posteriormente, estos escenarios ampliados y mejorados deben ser organizados a efectos de obtener un conjunto consistente de escenarios que representen el contexto actual (Hadad, 2008). Estos escenarios son un punto de partida para definir los escenarios futuros (Leite, y otros, 2004). La relación entre los escenarios actuales y futuros es estrecha, si bien ambos comparten el mismo contexto del problema y tienen una estructura similar, son diferenciados por el contenido que presentan (Leite, y otros, 2004).

El entendimiento del contexto actual es sumamente importante para comprender la problemática actual y posteriormente elaborar propuestas que satisfagan las necesidades expresadas por los clientes y/o usuarios. En este sentido, los escenarios actuales son los que ayudan a comprender el contexto y las limitaciones del futuro sistema.

El LEL y los escenarios actuales son el punto de partida para la rastreabilidad necesaria entre los artefactos construidos (modelos, documentos y componentes de software) y también hacia las fuentes de información; sin una rastreabilidad apropiada la gestión de los requisitos se transforma en una tarea casi imposible de llevar a cabo (Leite, y otros, 2004). La rastreabilidad incluye posteriormente a los escenarios futuros y hacia las especificaciones de requisitos. A partir de estas especificaciones podrá tener lugar la post-rastreabilidad de los requisitos (Hadad, 2008)

Se puede decir entonces que los escenarios futuros están más ligados a los requisitos y las

necesidades que debe satisfacer el sistema. Hadad (2008) menciona tres estrategias para la creación de escenarios futuros:

- ✓ *Derivando los escenarios futuros directamente de los escenarios actuales existentes.*
- ✓ *Derivando los escenarios futuros a través de los objetivos establecidos para el software a construir.*
- ✓ *Derivando los escenarios futuros mediante una combinación de las estrategias precedentes.*

El proceso de construcción de escenarios actuales no se describe en esta tesis, el mismo se encuentra detallado extensamente en “Defining System Context using Scenarios” (Leite, y otros, 2004). Una descripción más detallada del proceso de construcción de escenarios futuros puede estudiarse en “Uso de Escenarios en la Derivación de Software” (Hadad, 2008).

2.8 Utilidad de los escenarios

Los escenarios tienen distintos propósitos dependiendo de la fase en la que sean usados. Estos se detallan en la Tabla 2

Tabla 2 - Objetivos y consecuencias del uso de Escenarios, extraído de (Hadad, y otros, 1997)

Objetivo	Consecuencia
<i>Capturar requisitos</i>	<ul style="list-style-type: none"> * Evitar abstracciones orientadas a una solución técnica. * Brindar una visión más amplia del comportamiento del macro sistema.
<i>Proveer un medio de comunicación</i>	<p>(Derivan de la utilización del lenguaje del contexto de aplicación)</p> <ul style="list-style-type: none"> • Asegurar la comunicación entre los usuarios y los ingenieros de software • Garantizar la comunicación entre los miembros del equipo • Facilitar la validación de los requisitos con los usuarios • Comprometer a los usuarios en el desarrollo • Validar el LEL
<i>Contar con un instrumento de rastreabilidad</i>	<ul style="list-style-type: none"> • Documentar los requisitos • Capacitar a nuevos miembros del equipo para comprender la aplicación • Promover la evolución de los escenarios a medida que avanza el proceso de desarrollo • Generar casos de prueba

2.9 Amenazas, vulnerabilidad y riesgos

Las amenazas se encuentran presentes en todos los ámbitos y, por lo tanto, también lo estarán cuando se trata la producción de requisitos. Las amenazas también pueden ser vistas como factores que evitan que una organización pueda capitalizar las oportunidades asociadas al desarrollo global de software (Richardson, y otros, 2012).

Si se considera que des-localizar los equipos de trabajo es una decisión estratégica y que aunque persigue varias ventajas, entre ellas la reducción de costos, conlleva una alta inversión y está afectada por fuertes desafíos (Fryer, y Gothe 2008), por lo que el impacto frente a un fracaso no será menor.

Teniendo en cuenta algunas definiciones aportadas por Cardona Arboleda (2001), se puede llegar a las siguientes definiciones en relación a los proyectos de software:

✓ *Amenaza, peligro o peligrosidad (A) es la probabilidad de ocurrencia de un suceso*

potencialmente desastroso durante un cierto período de tiempo.

- ✓ *Vulnerabilidad (**V**) es el grado de pérdida de un elemento o grupo de elementos bajo riesgo, resultado de la probable ocurrencia de un suceso desastroso, expresada en una escala de 0 (sin daño) a 1 (pérdida total).*
- ✓ *Riesgo específico (**Rs**) es el grado de pérdida esperada debido a la ocurrencia de un suceso particular, como una función de la amenaza y la vulnerabilidad.*
- ✓ *Elementos en riesgo (**E**) son los documentos, personas, recursos, procesos, herramientas y actividades, expuestos en un proyecto determinado.*
- ✓ *Riesgo total (**Rt**) es definido como el grado de pérdida esperada para un elemento o conjunto de elementos expuestos a un determinado riesgo específico. Puede ser obtenido mediante el producto del riesgo específico **Rs**, y los elementos en riesgo **E**.*
- ✓ *De esta manera, la evaluación del riesgo puede llevarse a cabo mediante la siguiente formulación general (Cardona Arboleda, 2001):*

$$Rt = E * Rs = E * A * V \quad (1)$$

Cardona Arboleda (2001) propuso también eliminar la variable exposición **E**, por considerarla implícita en la vulnerabilidad **V**. En otras palabras, expresa que no se “es vulnerable” si no se “está expuesto”.

Reformulando lo mencionado anteriormente, se puede expresar que, una vez conocida la amenaza o peligro **Ait**, entendida como la probabilidad de que se presente un suceso con una intensidad mayor o igual a **i** durante un período de exposición **t**, y conocida la vulnerabilidad **Ve**, entendida como la predisposición intrínseca de un elemento expuesto **e** a ser afectado o de ser susceptible a sufrir un daño ante la ocurrencia del anteriormente mencionado suceso con una hipotética intensidad **i**, entonces, el riesgo **Rie** se expresa como la probabilidad de que se presente una pérdida sobre el elemento **e**, como resultado de la ocurrencia de un suceso con una intensidad mayor o igual a **i** (Cardona Arboleda, 2001).

Por lo que el Riesgo en un determinado período de tiempo se puede expresar mediante la siguiente fórmula:

$$Rie|t = (Ai, Ve)|t \quad (2)$$

El propósito de esta tesis está basado sobre las amenazas, variable **A**, puntualmente sobre los requisitos, el cual es un elemento en riesgo **e**, cuando se implementa un modelo distribuido, y cómo es posible su mitigación mediante la utilización de los modelos LEL y Escenarios.

La vulnerabilidad de los elementos de riesgo expuestos debería ser analizada y planeada estratégicamente en el momento que la organización decide optar por trabajar bajo la modalidad multi-site a efectos de robustecerlos y de esta forma estar mejor preparados para trabajar de este modo.

Lo que también se puede mencionar es que no importa cuánto esfuerzo una organización haga por estar preparada de antemano para este cambio de paradigma en su modo de trabajar, siempre podrá ser mejorado y, por lo tanto, podrá reducir su vulnerabilidad.

Este es un proceso de mejora continua que debería ser considerado dentro de la organización.

2.10 El desarrollo global de software

La globalización se hizo presente a lo largo de múltiples industrias, y no es de sorprender que la industria del software y la tecnología se hayan introducido rápidamente en esta tendencia (Fryer, y Gothe 2008).

Ya en los 90' varias organizaciones experimentaron con el trabajo remoto, persiguiendo objetivos como la reducción de costos o el rápido acceso a recursos capacitados (Prikladnicki, y otros, 2004). Hoy ya se ha transformado en una práctica más común, celebrándose la primera conferencia internacional sobre desarrollo global en el año 2006 [ICGSE \(IEEE International Conference on Global Software Engineering\)](#).

Básicamente se puede describir este modelo como una técnica de tercerización en la cual

todo o parte del equipo de trabajo, proveniente de diversas locaciones, participan del proyecto y proveen sus servicios mediante diversas herramientas y plataformas de trabajo como internet (Ul Haq, y otros, 2011).

Las dos formas más comunes actualmente en uso incluyen la llamada “Offshore Outsourcing”, la cual implica la contratación de servicios a una organización externa situada en otra locación diferente, y la denominada “Offshore Insourcing”, la cual consta de la contratación de los servicios de una subsidiaria localizada en otro/s lugar/es (Prikladnicki, y otros, 2004).

Si bien la primera opción es la más utilizada, la dificultad de construir una relación con un extraño en una locación remota hace que las organizaciones muchas veces opten por crear, o adquirir sus propios centros de desarrollo remotos en países como India, Rusia o Argentina, entre otros varios, intentando mitigar la diferencia de intereses que pueden existir.

El desarrollo global de software posee además ciertas características distintivas, que entre otras cosas, complejizan el ciclo de vida del proyecto, comprometiendo significativamente las tareas de ingeniería de requisitos (Zowghi, 2007).

Si se tiene en cuenta que las tareas relacionadas a la ingeniería de requisitos son consideradas las más influyentes dentro del proceso de desarrollo, dado que en esta etapa se identifican, analizan y definen las funcionalidades, las limitaciones y el propósito del sistema (Zowghi, 2007), es razonable pensar que las organizaciones que opten por esta modalidad de trabajo deben contar con procesos especialmente definidos para esta práctica, herramientas adecuadas, medios y formas de comunicación apropiadas, políticas claras, entre otros, para hacer frente a las diversas amenazas que rodean a esta forma de trabajo.

2.11 Amenazas del desarrollo global de software

Como se mencionó anteriormente la fase de construcción de requisitos es la más desafiante en todo el ciclo de vida del proyecto, y actúa como cimiento para el desarrollo global del sistema (Nahar & Student, 2013), (Nonyelum Ogwueleka, 2012), (Richardson, y otros, 2012). Si ocurre un problema en esta etapa, impactará en cascada en las demás fases del

proyecto deteriorando el producto en su totalidad (Nahar & Student, 2013).

Si a ello se suma la complejidad de trabajar bajo un modelo distribuido en donde se tienen amenazas como ser: diferencias culturales, idiomáticas, husos horarios distintos, etc., entonces la complejidad y las amenazas se ven potenciadas (Richardson, y otros, 2012), (Nonyelum Ogwueleka, 2012), (Zowghi, 2007), (Damian, y otros, 2003), (Hanisch, y otros, 2007).

Varios son los trabajos que enumeran las amenazas más comunes dentro del modelo distribuido (Prikladnicki, y otros, 2003), (Prikladnicki, y otros, 2004), (Richardson, y otros, 2012), (Fryer, y Gothe 2008), (Ul Haq, y otros, 2011), (Aranda, y otros, 2008) (Nonyelum Ogwueleka, 2012), (Damian, y otros, 2003), las que se enumeran a continuación:

- I. Comunicación inadecuada*
- II. Diferencias culturales y/o idiomáticas*
- III. Distancia geográfica y temporal*
- IV. Gestión del conocimiento*

Por otro lado, se tienen otras amenazas importantes que surgen como consecuencia de algunas de las anteriores y que requieren una correcta mitigación con el objetivo de evitar impactos en los requisitos. Ellas son:

- V. Falta de confianza e involucramiento*
- VI. Carencia del dominio del problema*
- VII. Ambigüedad, contradicciones y/o falta de claridad en las especificaciones*
- VIII. Problemas técnicos o de herramientas*
- IX. Rotación del equipo*
- X. Objetivos individuales*

En la Figura 7 se visualizan las amenazas principales, las amenazas derivadas y la relación

de consecuencia entre ellas.

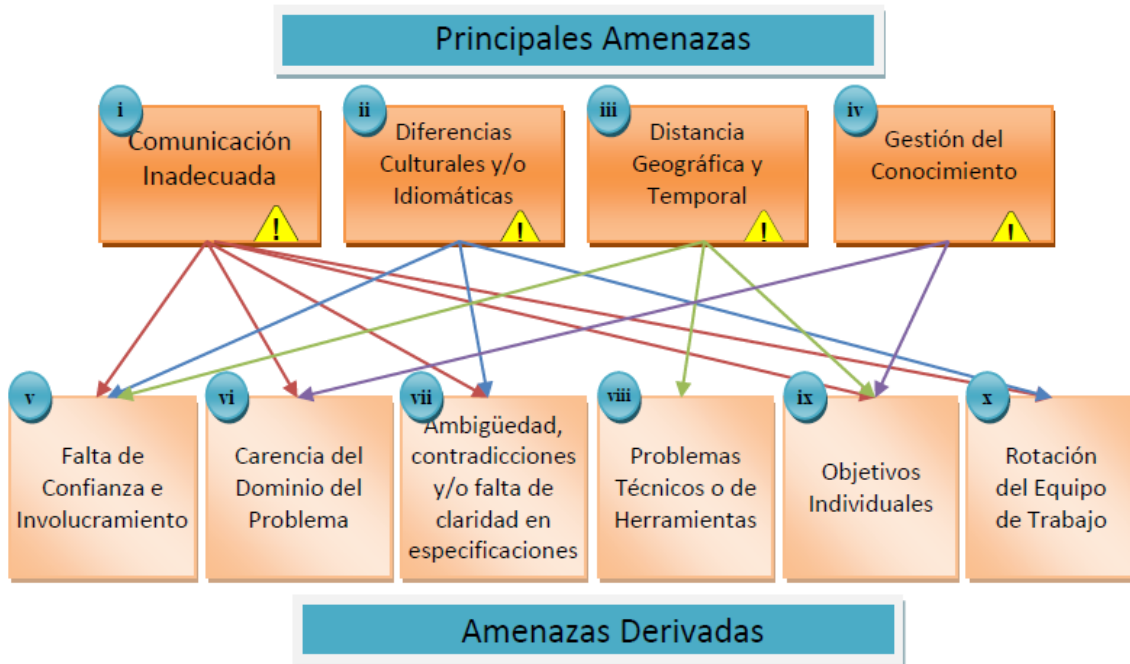


Figura 7 - Principales Amenazas y sus derivadas en el desarrollo global de software

A continuación se abordan y analizan estas amenazas de forma de establecer sus orígenes e impacto.

I. Comunicación inadecuada:

La comunicación es un elemento indispensable que todos los actores tienen que manejar a diario durante todo el proceso de desarrollo de software (Hadad, y otros, 1999), y una buena comunicación dentro de dicho proceso es una condición indispensable para obtener productos de calidad, que cumplan las necesidades de los usuarios en tiempo y dentro de los costos esperados (Cherry, y otros, 2004).

En particular, la etapa de construcción de requisitos solo podrá ser exitosa si existe una efectiva comunicación entre el cliente, los usuarios y los analistas, de modo que no existan discrepancias en el entendimiento de las necesidades (Nahar & Student, 2013).

La fase de construcción de requisitos es la etapa más demandante de comunicación de todo el proceso de desarrollo (Hanisch, y otros, 2007), dado que es donde más interacción se

requiere entre quienes poseen la necesidad y quienes deben interpretarla para proveer soluciones (Zowghi, 2007).

Como instrumentos que colaboran con una buena comunicación se puede incluir a la comunicación informal y cara a cara, las cuales son una herramienta sumamente utilizada en proyectos localizados en el mismo lugar.

Usualmente la comunicación entre miembros de un equipo bajo el modelo distribuido es mediante medios electrónicos y de forma asincrónica, básicamente limitada a correos electrónicos. El abuso de este tipo de comunicación es también perjudicial para los proyectos, generando entre otras cosas, desentendimientos, re-trabajo, ambigüedades, e impactando inevitablemente en los costos y la calidad de los productos (Richardson, y otros, 2012), (Damian, y otros, 2003). Fryer y otros (2008) mencionan que el abuso de este tipo de comunicación puede hacer que la productividad en los proyectos de desarrollo global decaiga hasta en un 50% y que sea necesario un re-trabajo de entre 2 y 5 veces mayor que en un proyecto localizado en el mismo lugar.

Entonces al no disponer, o disponer en forma escasa, de esta comunicación informal, sincrónica y cara a cara en el modelo distribuido, se hace muy complejo el entendimiento del dominio del problema y de la jerga utilizada en él. Adicionalmente, la carencia de una correcta comunicación, además de reducir la calidad y la cantidad de información que es transmitida entre los diferentes equipos des-localizados, representa una barrera para la transferencia de conocimiento (Fryer, y Gothe 2008), (Richardson, y otros, 2012), impactando sensiblemente en la calidad obtenida en los artefactos producidos, la motivación de los equipos, la confianza entre los involucrados respecto al proyecto y generando una cultura de “ellos contra nosotros” (Richardson, y otros, 2012).

Las personas tienden a generar empatía con otras que dominan sus mismos conceptos o intereses. En este sentido, es fundamental que los equipos distribuidos manejen el mismo léxico, dado que una comunicación basada en diferentes léxicos tiene una mayor probabilidad de fracasar (Hadad, y otros, 1999). El manejo de un mismo vocabulario potencia la relación de los involucrados (Leite, 1989).

Por lo expuesto anteriormente, se desprende que es de vital importancia implementar

instrumentos que permitan mejorar aquellos aspectos de la comunicación que impactan en los proyectos de desarrollo global de software.

II. Diferencias culturales y/o idiomáticas:

La cultura y el idioma son puntos clave dentro de este modelo. La cultura puede diferir acorde a las siguientes áreas (Damian, y otros, 2003):

- ✓ *Organizacional*
- ✓ *Geográfica*
- ✓ *Cultura nacional*
- ✓ *Religión*
- ✓ *Distancia de poder*

Por ejemplo, algunas culturas pueden ser sensibles a reportar al sexo femenino, o los individuos de culturas que poseen una distancia de poder alta son muy deferentes con las figuras de autoridad y generalmente aceptan una distribución desigual del poder, sin cuestionar ninguna de las decisiones (Fryer, y Gothe 2008), mientras que los individuos de culturas con una distancia de poder baja cuestionan fácilmente la autoridad y esperan participar en las decisiones que les afectan (Richardson, y otros, 2012).

Dependiendo de la cultura involucrada y de las herramientas que utiliza el proyecto de software, puede ocurrir que la transferencia del conocimiento se transforme en una tarea altamente desafiante, dado que las dudas o contradicciones no se evacuarían o cuestionarían, impactando en la calidad de los requisitos.

El idioma es un factor crítico que impacta directamente en las actividades de elicitación de información y su validación (Damian, y otros, 2003). Esta situación puede darse entre los individuos de la organización o el área cliente y los analistas; entre los analistas y desarrolladores o testers, y otros. En cualquiera de las fases donde se presente, puede dar lugar a ambigüedades, omisiones, errores, y contradicciones.

El uso de idiomas que no son la lengua madre de los miembros del equipo representa barreras adicionales para una correcta comunicación (Richardson, y otros, 2012).

Adicionalmente, si los involucrados no tienen un nivel adecuado del idioma común a utilizar, esto hace que se encuentre cierta reticencia a la comunicación sincrónica, potenciando una interacción asincrónica con los efectos negativos antes mencionados. Aún cuando el idioma puede ser el mismo para los involucrados, dependiendo del país de origen el significado de ciertas palabras puede ser diferente.

Por otro lado, la formación, la posición y el contexto de trabajo de cada uno de los involucrados, hace que se manejen diferentes jergas, por lo que el entendimiento entre los diferentes miembros del equipo puede verse afectado. En este sentido, los instrumentos que colaboren con un correcto entendimiento del vocabulario del contexto de aplicación serán de suma utilidad.

III. Distancia geográfica y temporal:

Es indudable que bajo este modelo exista una distancia geográfica entre los involucrados, la cual limita o representa un desafío para una comunicación cara a cara o informal, (Cherry, y otros, 2004), potenciando uno de los principales problemas en la elicitación de información, como lo es la carencia de una correcta comunicación entre los involucrados (Damian, y otros, 2003).

La comunicación asincrónica limita la profundidad de la relación entre los miembros del equipo. En este sentido, la distancia temporal, la cual reduce en muchos casos la cantidad de horas laborales de solapamiento entre las diferentes locaciones, atenta contra la comunicación sincrónica, tales como uso de teléfono, video conferencias, chat, entre otros medios, la cual es una herramienta útil para la rápida resolución de discrepancias, falta de información y ambigüedades antes de que se transformen en problemas de mayor magnitud (Damian, y otros, 2003), y que afectan adicionalmente a las actividades de negociación y priorización de requisitos (Zowghi, 2007).

Esta carencia afecta no solo al vínculo de los miembros del equipo, sino también al correcto entendimiento del contexto de aplicación y, por consiguiente, de las necesidades del cliente y los usuarios (Damian, y otros, 2003). La distancia geográfica o temporal es un aspecto propio del modelo distribuido, y afecta a la coordinación y el control de los proyectos (Zowghi, 2007). La mitigación de esta amenaza puede y debe llevarse a cabo mediante la

utilización de métodos y/o herramientas que potencien una apropiada interacción entre los involucrados (Fryer, y Gothe 2008), con el fin de lograr una adecuada interpretación de las necesidades y cumplir con la tarea de construir una correcta especificación de requisitos.

IV. Gestión del conocimiento:

Como se comentó anteriormente, cuanto mayor sea el conocimiento que el equipo de trabajo tenga sobre el dominio del problema o sobre el problema mismo, mejor serán los beneficios para el proyecto.

El propósito de la gestión del conocimiento es la de absorber el capital intelectual de la organización con el fin de ser usado posteriormente, refiriéndose también a la recopilación de información de los proyectos pasados o en curso (Prikladnicki, y otros, 2003). En la fase de construcción de requisitos, suele haber una gran cantidad y variedad de información proveniente de diversas fuentes de información, que debiera estar disponible para todos los involucrados. Este factor debe ser considerado especialmente en un modelo distribuido donde los involucrados están geográficamente dispersos y, por ende, deben tener canales formales para acceder a la información a compartir, ayudando a evitar de alguna manera el conocimiento tácito y la transmisión informal de información (Zowghi, 2007). En este sentido, la creación de un repositorio consistente de información puede ser de ayuda en el proceso de la toma de decisiones, incrementando la calidad del proceso y, por consiguiente, la calidad del producto, reduciendo los costos y los tiempos del proyecto (Prikladnicki, y otros, 2003).

Es importante mencionar que no es solo recopilar material por el solo hecho de incrementar y/o proporcionar la información, sino que el objetivo a seguir es el de mejorar la calidad de esa información con el fin de que sea más comprensible y fácil de procesar, lo cual proveerá un soporte fundamental para la elaboración de la solución del problema a resolver.

La rastreabilidad se vuelve fundamental en el modelo distribuido donde diferentes equipos en diferentes lugares acceden a la misma información y es de vital importancia conocer el origen de dicha información, y las modificaciones por las que ha pasado un requisito como así también quien las ha solicitado y aprobado, y fundamentalmente los motivos que llevaron a cada cambio. La rastreabilidad no es solamente necesaria para una correcta

administración de los elementos de configuración, como son los requisitos, sino que proporciona, mediante la identificación de aquellos elementos relacionados, la facilidad de poder entender el contexto del problema y las necesidades (Fryer, y Gothe 2008).

V. Falta de confianza e involucramiento:

Para algunos autores, establecer y mantener la confianza entre los involucrados, es el punto más desafiante entre equipos distribuidos (Ul Haq, y otros, 2011).

La confianza es factible de construir cuando se trata de equipos de trabajo en un mismo lugar, donde la comunicación cara a cara, informal, las charlas en pasillos, máquinas de café, almuerzos en conjunto o similares, son más comunes. La confianza es difícil de establecer cuando se trata de equipos multi-site donde las relaciones interpersonales y la sociabilización se ven limitadas por la distancia geográfica y temporal (Fryer, y Gothe 2008), (Richardson, y otros, 2012), (Damian, y otros, 2003).

Si bien la creación de perfiles, las herramientas colaborativas, tal como video conferencias, y otras, pueden ayudar a mitigar esta amenaza (Fryer, y Gothe 2008), es necesario un correcto entendimiento entre los involucrados para lograr funcionar como un equipo detrás de un mismo objetivo.

Es claro, además, que sin una relación de confianza entre los analistas, los usuarios y clientes es imposible tener éxito en las fases de elicitación de conocimiento (Nahar & Student, 2013), y esto se fundamenta principalmente en las carencias para manejar una misma jerga y/o para hablar un mismo lenguaje (Richardson, y otros, 2012).

Mientras que los analistas hablan un lenguaje semi-técnico propio de su disciplina, los usuarios utilizan una jerga totalmente propia del dominio del problema, por lo que la comunicación y el entendimiento se ven seriamente deteriorados. La utilización de lenguaje natural en las herramientas de comunicación y el manejo del mismo léxico contribuyen con una adecuada comunicación y ayudan a la construcción de la relación de confianza entre los involucrados.

VI. Carencia del dominio del problema:

El conocimiento del contexto de aplicación es de vital importancia en un proyecto, de lo

contrario puede derivar en un grave obstáculo para el propio proyecto (Christel, y otros, 1992) y conducir a un ping-pong de preguntas y repuestas, o validaciones continuas y frustrantes, ocasionando retrasos significativos. La falta de conocimiento sobre el contexto del problema puede dar como resultado una incorrecta interpretación de las necesidades, una recepción de requerimientos excesivamente extensos o ambiguos, lo que derivará casi inevitablemente en una solución inadecuada.

Cuando los involucrados en el proyecto se encuentran distribuidos, el entendimiento del dominio se vuelve más complejo (Kumar, y otros, 2011). Y este ping-pong de preguntas y respuestas al estar inmerso en el modelo distribuido, donde existen las amenazas de comunicación anteriormente mencionadas o la distancia geográfica o temporal, hará que los tiempos del proyecto se vean seriamente afectados y por ende sus costos.

Cuando los miembros del proyecto no comparten un mismo idioma, esto se dificulta aún más (Kumar, y otros, 2011). Muchas de las definiciones que comparte el usuario a través de mail, notas, documentos y otros medios, pueden estar plagadas de palabras específicas del dominio del problema, las cuales al intentar ser traducidas mediante traductores automáticos, pueden perder totalmente su significado dado que carecen del contexto del problema. La interpretación de esta información por parte de alguien que no conoce el contexto puede ser muy difícil (Dharmadas, 2008), o puede llevar a que aquellos que conocen el dominio del problema se vean abrumados por preguntas o necesidades de recursos de otras locaciones respecto a explicaciones, generando cuellos de botella y ralentizando el proceso de desarrollo (Richardson, y otros, 2012).

VII. Ambigüedad, contradicciones y/o falta de claridad en las especificaciones:

La ambigüedad de los requisitos es uno de los puntos que el estándar IEEE 29148 menciona que debe ser evitado al especificar un requisito; explícitamente menciona que debe haber una sola interpretación para cada uno de ellos y que esta debe ser simple y fácil de comprender (IEEE_29148, 2011).

El hecho de que el proyecto esté inmerso en el modelo distribuido y se tengan diferencias culturales, idiomáticas, un conocimiento del dominio limitado y generalmente se abuse de la comunicación asincrónica, hace que las probabilidades de tener requisitos ambiguos se

vean potenciadas.

Asimismo, si el conocimiento proviene de diversas fuentes de información, la distancia entre los múltiples emisores de la información (por ejemplo, usuarios) y los receptores (analistas) puede dificultar el entendimiento de la información transmitida, conduciendo a la falta de claridad en las especificaciones. Esta misma situación hace propensa la aparición de contradicciones en la información, donde la comunicación a distancia no facilita la resolución de dichas contradicciones.

Los requisitos ambiguos, poco precisos y/o contradictorios impactarán en cascada en otras fases del proyecto (Nahar & Student, 2013), y darán lugar a una construcción errónea que probablemente no sea lo que el cliente necesite, especialmente cuando estos defectos no son detectados tempranamente.

El uso de ontologías como elemento para la reducción de las diferencias o interpretaciones del lenguaje es mencionado por varios autores (Nonyelum Ogwueleka, 2012), (Aranda, y otros, 2008) y se refiere a la descripción formal y completa del vocabulario relacionado a un dominio (Nonyelum Ogwueleka, 2012). Esto es importante durante la elicitación y negociación de los requisitos porque clarifica la estructura del conocimiento y reduce las ambigüedades del dominio del problema (Nonyelum Ogwueleka, 2012). Glosarios menos formales que las ontologías pueden brindar también un buen apoyo a este tipo de amenaza (Kovitz, 1998), (Regnell, y otros, 1999), (Hadad, Doorn & Kaplan, 2009).

VIII. Problemas técnicos o de herramientas:

Como en todo proceso, las actividades involucradas deben utilizar herramientas, las cuales facilitan su desempeño. Los problemas técnicos o la carencia de herramientas hacen más complejo el logro de los objetivos propuestos.

En este sentido, la no creación y/o inadecuada administración de los repositorios de información como así también la gestión deficiente de los usuarios, perfiles y accesos, a los elementos de configuración del proyecto, es una amenaza para la integridad de las especificaciones de requisitos, así como de todo otro artefacto que se construya durante el proceso. La falta de rastreabilidad entre los requisitos, o entre los requisitos y los demás elementos de configuración, potencia la incorporación de omisiones u otros defectos al

momento de introducir cambios en las especificaciones de requisitos.

Por otro lado, la incapacidad de una adecuada recolección de información que ayude en el cálculo de las métricas del proyecto llevará a la toma de decisiones incorrectas con un impacto negativo en el proyecto.

La conectividad juega un papel sumamente importante cuando se trata del desarrollo global de software. La no previsión de herramientas de comunicación apropiadas, como líneas telefónicas, video conferencias, herramientas de chats, entre otras, para suplir la pérdida de la comunicación cara a cara o informal, representa una seria amenaza para los objetivos del proyecto. Asimismo, la seguridad interna y externa tiene un papel fundamental en este tipo de proyectos. La selección inadecuada de herramientas o su incorrecta administración dará lugar a problemas que afectarán los tiempos y la calidad de los entregables del proyecto. Por el contrario, una selección adecuada de las herramientas, enfoques y técnicas para los proyectos de desarrollo global de software colaborará con la correcta ingeniería de requisitos aumentando las posibilidades de éxito del proyecto (Hanisch, y otros, 2007). Dentro de los factores críticos de éxito para el desarrollo global de software se pueden encontrar: crear y administrar adecuadamente los repositorios de información, seleccionar y proporcionar herramientas para una adecuada comunicación, y desarrollar una infraestructura clara y flexible (Fryer, y Gothe 2008).

IX. Rotación del equipo de trabajo:

La pérdida de recursos experimentados durante el avance de un proyecto tiene un impacto directamente proporcional en la productividad, en la transferencia de conocimiento, y en la motivación del equipo, sobrecargando al equipo remanente para recuperar el retraso generado (Richardson, y otros, 2012).

Este impacto no se limita a lo expuesto anteriormente, sino que también representa dedicar aún mayor tiempo del equipo remanente para la capacitación de los nuevos recursos a incorporarse. Esta capacitación es una tarea desmoralizadora, que muchas veces se lleva adelante en forma apresurada o en forma inadecuada para suplir la falta de tiempo, lo cual irremediablemente impacta en la calidad del proceso de capacitación y, por consiguiente, en las tareas que esos recursos llevarán a cabo posteriormente.

X. Objetivos individuales:

La carencia de logros u objetivos compartidos puede dar lugar a la desmotivación de los equipos, pobre productividad y potenciar la cultura de “ellos contra nosotros” (Richardson, y otros, 2012), afectando significativamente el resultado del proyecto.

Las metas pueden no ser las mismas entre los diferentes equipos distribuidos, al no reportar a la misma estructura gerencial o incluso a diferentes compañías. Lo mencionado anteriormente hace complejo que los equipos entiendan y se comprometan con el mismo objetivo (Fryer, y Gothe 2008).

De cualquier forma, el perseguir un mismo propósito sería el primer paso para que los involucrados lleguen a construir una cultura de equipo.

En resumen, el desconocimiento o sub-estimación de alguno de estos temas puede impactar en la calidad de los requisitos considerablemente y, lo que es peor aún, el identificar este impacto negativo en etapas muy posteriores incrementará significativamente los costos de resolución (Boehm, 1981), (Westland, 2002). La Tabla 3 resume las amenazas descriptas y qué autores las identifican en sus trabajos, esto se ha marcado con una “X” en cada una de las celdas correspondientes.

Tabla 3 - Amenazas identificadas por los distintos autores en el tema

<i>Autores</i>	(Richardson, y otros, 2012)	(Prikladnicki, y otros, 2003)	(Fryer, y Gothe 2008)	(Ul Haq, y otros, 2011)	(Aranda, y otros, 2008)	(Nonyelum Ogwueleka, 2012)	(Nahar & Student, 2013)	(Zowghi, 2007)	(Damian, y otros, 2003)
<i>Amenaza</i>									
<i>Comunicación inadecuada</i>	X	X	X	X	X	X	X	X	X
<i>Diferencias culturales y/o idiomáticas</i>	X	X	X	X	X	X		X	X
<i>Distancia geográfica y temporal</i>	X	X	X	X	X	X		X	X
<i>Gestión del conocimiento</i>	X	X	X		X	X		X	X
<i>Falta de confianza e involucramiento</i>	X	X	X	X			X	X	X
<i>Carencia del dominio del problema</i>	X		X		X	X	X		
<i>Ambigüedad, contradicciones y/o falta de claridad en las especificaciones</i>					X		X		X
<i>Problemas técnicos o de herramientas</i>		X	X	X	X	X		X	
<i>Rotación del equipo</i>	X								
<i>Objetivos individuales</i>	X		X				X		

3. Análisis de un proyecto de desarrollo global sin mitigación de amenazas

3.1 Características del proyecto Venezuela

Como ya se ha mencionado, evitar o al menos minimizar el impacto de las amenazas del desarrollo global sobre los requisitos es de vital importancia, dado que de no ocurrir repercutirá sobre otras etapas posteriores.

Esto queda claramente evidenciado con el análisis post-mortem realizado del proyecto Venezuela, donde se presentará a continuación cómo las principales amenazas y sus derivadas se materializaron e impactaron en forma negativa en él.

Este impacto se cuantificó mediante el uso de métricas específicas, cuyos resultados fueron posteriormente evaluados a fin de elaborar una estrategia proactiva de mitigación para el proyecto Brasil.

El programa regional de una compañía multinacional de seguros en estudio consistió inicialmente en la implementación de una plataforma regional común para Venezuela; este fue el primero de los países incorporados, lo que permitió luego de su primera entrega identificar la necesidad de introducir mejoras en el proceso dado que los resultados obtenidos no fueron los esperados. Estas mejoras se enfocaron básicamente en el proceso de requisitos.

En el proyecto Venezuela se dispuso de un equipo de 13 analistas funcionales en Argentina (Buenos Aires), 38 desarrolladores en India (Pune), 7 Testers en India (Chennai) y alrededor de 20 usuarios especializados en Venezuela (Caracas).

Se debe observar que, de los 13 analistas funcionales, solamente 4 contaban con experiencia en el negocio de seguros, el resto contaba con competencias de análisis funcional pero no conocimientos profundos de la industria. Respecto al proveedor seleccionado para el desarrollo, de los 38 desarrolladores, solo 13 contaban con experiencia en proyectos relacionados a seguros. En la Tabla 4 se muestra la distribución de parte del equipo según su conocimiento o no del dominio del problema, donde se observa que el 69% de los analistas no tenían conocimiento en el dominio de seguros, y el 66% de los

desarrolladores tampoco. Esto se debió a la falta de recursos disponibles en los mercados laborales de Buenos Aires y Pune; esta misma circunstancia se presentó también en el proyecto Brasil.

Tabla 4 - Distribución de competencias del equipo para el Proyecto Venezuela

Proyecto Venezuela		
Roles	Analistas Funcionales	Desarrolladores
Con Conocimiento del Dominio	4	13
Sin Conocimiento del Dominio	9	25
Cantidad total	13	38

Un resumen del ciclo de vida cascada adoptado con sus 7 subprocesos se describe a continuación (ver Figura 2 en el capítulo 1):

- I. Conceptualización, que involucraba actividades de análisis de factibilidad, de preparación de herramientas e infraestructura, de configuración, definición de alcance y requerimientos del negocio, y definición preliminar de la arquitectura;*
- II. Iniciación, que involucraba actividades de gestión de riesgo y planificación, reunión de lanzamiento del proyecto, y entrenamiento del equipo;*
- III. Análisis y Diseño, que involucraba actividades de gestión para actualizar la planificación del proyecto y definición del plan de pruebas, actividad de análisis del sistema, actividad de diseño de la arquitectura, del modelo de datos, de las clases, y de los casos de prueba;*
- IV. Transición a Fábrica y Desarrollo, que involucraba las actividades de transferencia de las especificaciones de requisitos a la fábrica de software, creación de código, pruebas unitarias, desarrollo de manuales de usuario, de instalación y operativos;*
- V. Pruebas, que involucraba las actividades de ejecución de Pruebas de*

Integración y de Pruebas de Certificación de usuarios;

- VI. Implementación, que involucraba actividades de gestión del proyecto, formalizar la aceptación de entregables y reuniones de cierre, actividades de entrenamiento a usuarios, plan de soporte, implementación en los ambientes de producción, y actividades de gestión del conocimiento creando documentación de transferencia de conocimiento del equipo del proyecto al equipo de soporte; y*
- VII. Post-Implementación, que involucraba reuniones de lecciones aprendidas, liberación de recursos y revisión post-implementación.*

En el Anexo A, se detalla el proceso de software llevado a cabo para el proyecto Venezuela, identificándose sus subprocesos y actividades.

El proceso de requisitos en el proyecto Venezuela se desarrolló a lo largo del primer y tercer subproceso, y sus actividades se detallan a continuación:

En el subproceso de Conceptualización, la actividad de definición de alcance y estrategia de elicitación de requerimientos involucraba:

- ✓ *Elicitación y análisis de los requerimientos de alto nivel generados por el negocio.*
- ✓ *Creación del registro de requerimientos del negocio.*
- ✓ *Creación de la matriz de rastreabilidad.*
- ✓ *Validación mediante sesiones de los requerimientos del negocio.*
- ✓ *Aprobación formal de los requerimientos del negocio.*

En el subproceso de Análisis y Diseño la actividad de análisis del sistema involucraba:

- ✓ *Revisión general y validación detallada de los requerimientos del negocio.*
- ✓ *Priorización de requerimientos del negocio.*
- ✓ *Creación de las especificaciones de requisitos.*

- ✓ *Validación de las especificaciones de requisitos mediante sesiones.*
- ✓ *Aprobación formal de las especificaciones de requisitos.*

Después de la aprobación formal de las especificaciones de requisitos durante los subsecuentes procesos, se realizaba una actividad de evolución de las especificaciones, a pesar de practicarse un modelo en cascada.

Según este proceso, los analistas se movilizaban desde Argentina a Venezuela, donde tomaban contacto con los responsables de cada área de la compañía, quienes ya habían participado de la reunión de lanzamiento del proyecto y estaban interiorizados de la metodología y estrategia del proyecto. Paso siguiente se identificaban las fuentes de información idóneas para proveer más detalle sobre los requerimientos del negocio, y se comenzaba con las tareas de elicitación. Del conjunto de técnicas de elicitación tradicionales más usadas, como ser, encuestas, entrevistas, cuestionario y análisis de documentos (Antonelli, y otros, 2002), se optó por las entrevistas y los cuestionarios dada la experiencia del equipo en estas técnicas y la historia de la compañía en el uso de las mismas. Esto no es inusual si se considera que son las más utilizadas por los equipos de profesionales de Argentina (Antonelli, y otros, 2002). Y es natural que una compañía que históricamente utilizó estas técnicas para la elicitación co-localizada utilice esas mismas en ambientes distribuidos de desarrollo de software (Zapata, 2013).

Entrevistas

Los analistas que se movilizaban a Venezuela asumían el rol de entrevistadores, llevando a cabo las entrevistas. La duración de las entrevistas fue entre 60 y 90 minutos aproximadamente. Posteriormente, se elaboraba un acta de la entrevista a validar por el entrevistado.

Cuestionarios

En el proyecto Venezuela resultaron sumamente útiles para evacuar dudas puntuales sobre comportamientos actuales y futuros, de una forma asincrónica, permitiendo que sean respondidos en los momentos libres o en varios lapsos de tiempo. Dado que requieren menos tiempo por parte de los informantes que las entrevistas, fue aceptado por el área

operativa la distribución de estos cuestionarios entre varias personas claves a fin de elicitación y validar conceptos importantes de los requerimientos del negocio para el sistema.

Con la información recopilada en las entrevistas y las respuestas de los cuestionarios, se procedía a la construcción de las especificaciones de requisitos. Los formatos eran diversos dependiendo del tipo de requisito, siendo el más usado el documento dividido en secciones, escrito en prosa y complementados con diagramas de secuencia, diagramas de actividades, imágenes y tablas, entre otros artefactos.

En la Figura 8 se muestra la estructura de un documento de especificación de requisito de complejidad baja. Este documento presenta una introducción, un resumen del proceso de negocio involucrado y sus reglas de negocio, una descripción funcional de la solución para el proceso de negocio y una sección con los requisitos no funcionales involucrados. Esta estructura es la utilizada por la compañía de seguros en todo el proyecto de software. Se debe observar que la documentación generada fue escrita en inglés dado las diferencias idiomáticas entre las locaciones del proyecto.

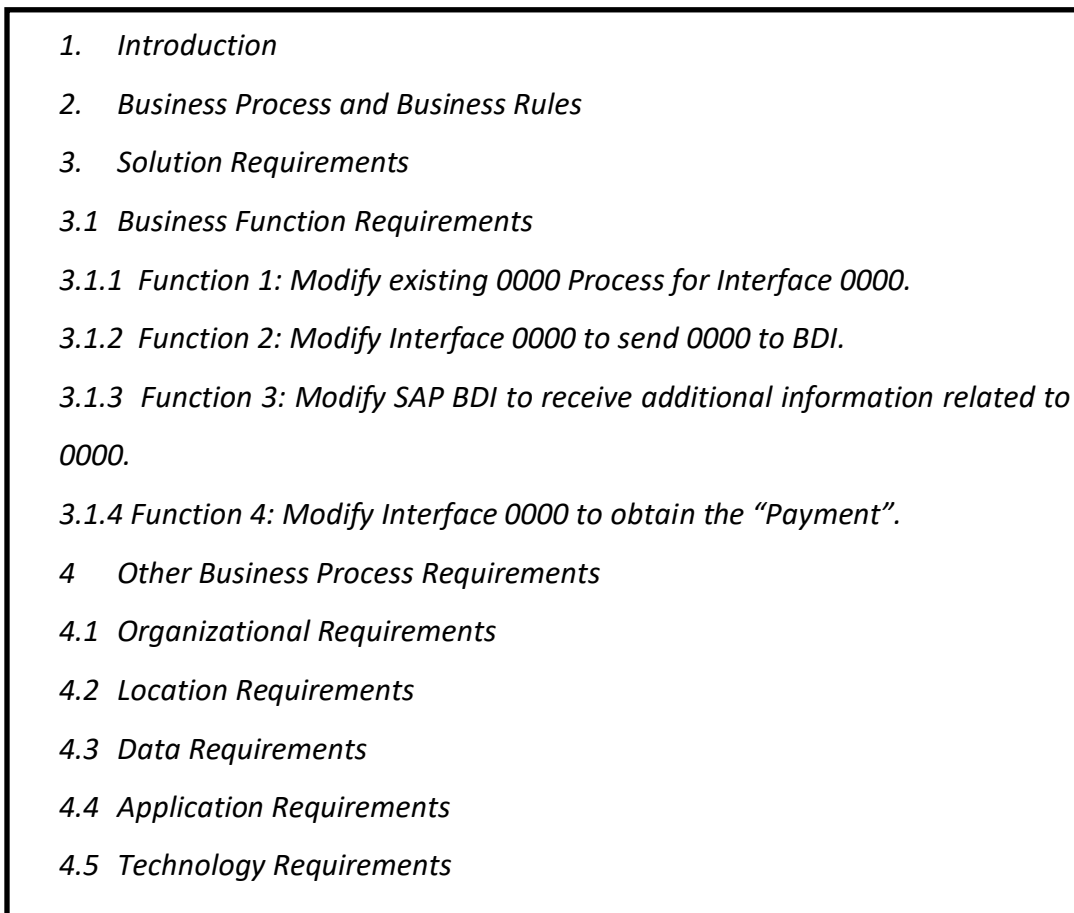


Figura 8 - Ejemplo de Estructura de una Especificación de Requisito

Finalmente, el documento procedía a ser validado por el área de negocios quien podía rechazarlo solicitando ajustes o modificaciones, las que el analista llevaba a cabo y nuevamente solicitaba la aprobación. Una vez aprobado el documento, los analistas funcionales regresaban a Buenos Aires para hacer la transición de las especificaciones de requisitos a la fábrica de software en India, que habían sido planificadas al comienzo de esta fase.

Adicionalmente, los miembros del equipo, gestionados por el gerente del proyecto, llevaban adelante una validación de la planificación, y posteriormente comenzaban las sesiones de transición de las especificaciones a la fábrica de software. Estas sesiones consistían en enviar el documento con la especificación de requisito a la fábrica de software, en donde el gerente se encargaba de asignarla al desarrollador, en función de la disponibilidad de recursos y los conocimientos requeridos.

Posterior a una revisión por parte del recurso que llevaría adelante el desarrollo, se procedía a las sesiones de transición, donde el analista funcional explicaba el objetivo del requisito y profundizaba sobre las especificaciones, y el o los desarrolladores asignados planteaban dudas que el analista respondía. Si la revisión no era concluida en esa sesión de transición, dados los tiempos de solapamiento de 2 horas entre Buenos Aires y Pune, los desarrolladores analizaban el documento por separado hasta el día siguiente, donde exponían sus dudas y los analistas las respondían; este ciclo de preguntas-respuestas se repetía hasta que la comprensión por parte de los desarrolladores era lograda, y entre ambas partes aceptaban la transición como concluida.

Al finalizar la transición, los desarrolladores y testers llevaban adelante las tareas propias de construcción, pruebas unitarias y pruebas de integración, teniendo además la posibilidad de evacuar dudas con los analistas funcionales durante todo el proceso desde la transición hasta la entrega para las pruebas de certificación por parte de los usuarios.

Una vez concluido el desarrollo y las etapas de pruebas de integración, la fábrica de software en India suministraba los entregables al equipo de Latinoamérica, que eran instalados en los ambientes correspondientes por el equipo de promociones a fin de proceder con la etapa de certificación de usuarios.

Fue durante esta etapa de certificación donde se identificaron numerosos rechazos por parte de los usuarios, argumentando que la aplicación no proveía lo que ellos habían solicitado y, por lo tanto, no cumplía con las necesidades operativas.

En ese momento, si bien se contaba con información recopilada de las métricas definidas al comienzo del proyecto, no se había hecho un análisis profundo de las mismas, por lo cual tampoco se habían tomado las medidas correctivas adecuadas.

Se decidió en ese momento hacer un análisis profundo de esta información con el objetivo de identificar los principales problemas que derivaron en los rechazos de los usuarios e impactaron en la calidad del producto entregado y, por ende, en los tiempos y los costos del proyecto. Este análisis serviría adicionalmente para identificar las amenazas que el proyecto Brasil enfrentaría al momento de comenzar y tomar acciones adecuadas para su mitigación desde el inicio.

Las características del proyecto para Venezuela son las que se expusieron anteriormente. La información recopilada surge de las métricas obligatorias, definidas según el framework de procesos de la compañía, las que se detallan en la siguiente sección.

3.2 Métricas definidas para los proyectos bajo estudio

A continuación se detallan solo aquellas métricas relacionadas con los requisitos del software, las que permitieron identificar las causas de los impactos negativos en el proyecto Venezuela respecto a requisitos (ver Tabla 5 a 11). Estas mismas métricas fueron también las utilizadas para el proyecto Brasil.

Tabla 5 - Métrica DFE01: Entrega de especificación de requisitos

Indicador	Entrega de especificación de requisitos
Código	DFE01
Definición	Tiempo promedio necesario para la creación de una especificación de requisito determinada.
Subproceso	Análisis y diseño
Fórmula de cálculo	A/B
Definición de los parámetros	A: Tiempo total de entrega de todas las especificaciones de requisitos B: Número total de especificaciones
Unidad de medida	Días
Periodicidad	Mensual
Observaciones	Tiempo medio necesario para concluir una especificación, desde el momento en que se asigna hasta el momento que se aprueba. Se agruparán reportes según la cantidad de líneas de negocio existentes.

Tabla 6 - Métrica DFA01: Aprobación de especificación de requisitos

Indicador	Aprobación de especificación de requisitos
Código	DFA01
Definición	Tiempo medio necesario para la validación de una especificación de requisito.
Subproceso	Análisis y diseño
Fórmula de cálculo	A/B
Definición de los parámetros	A: Tiempo total de aprobación de todas las especificaciones de requisitos B: Número total de especificaciones
Unidad de medida	Días
Periodicidad	Semanal
Observaciones	Tiempo medio necesario para aprobar por parte del usuario/cliente las especificaciones de requisitos

Tabla 7 - Métrica DFR01: Rechazo de especificación de requisitos

Indicador	Rechazo de especificación de requisitos
Código	DFR01
Definición	Promedio de la cantidad de rechazos de las especificaciones por parte de los usuarios/clientes durante la aprobación de los mismos.
Subproceso	Análisis y diseño
Fórmula de cálculo	A/B
Definición de los parámetros	A: Número total de rechazos de especificaciones por el usuario o cliente B: Número total de especificaciones entregadas, realizadas en el período de medición
Unidad de medida	Unidades numéricas
Periodicidad	Semanal
Observaciones	N/A

Tabla 8 - Métrica TRATPT01: Tiempo promedio de transición a fábrica

Indicador	Tiempo promedio de transición a fábrica
Código	TRATPT01
Definición	Tiempo promedio de transición de las especificaciones de requisitos a la fábrica de software.
Subproceso	Transición a Fábrica y Desarrollo
Fórmula de cálculo	N/T
Definición de los parámetros	N: Total de horas de transición para todas las especificaciones de requisitos T: Total de especificaciones de requisitos
Unidad de medida	Horas
Periodicidad	Semanal
Observaciones	N/A

Tabla 9 - Métrica TRACPDF01: Cantidad de preguntas por especificación de requisitos

Indicador	Cantidad de preguntas por especificación de requisitos
Código	TRACPDF01
Definición	Cantidad promedio de preguntas por especificación de requisitos, durante la transición y desarrollo por parte de la fábrica de software.
Subproceso	Transición a Fábrica y Desarrollo
Fórmula de cálculo	N/T
Definición de los parámetros	N: Total de preguntas referidas a las especificaciones de requisitos T: Total de especificaciones de requisitos
Unidad de medida	Unidades numéricas
Periodicidad	Semanal
Observaciones	N/A

Tabla 10 - Métrica CERTNCFF01: Cantidad de no conformidades por errores, cambios o ambigüedades en las especificaciones de requisitos

Indicador	Cantidad de no conformidades por errores, cambios o ambigüedades en las especificaciones de requisitos
Código	CERTNCFF01
Definición	Porcentaje de no conformidades en las especificaciones de requisitos respecto al total de no conformidades en la etapa de certificación.
Subproceso	Pruebas
Fórmula de cálculo	$N/T * 100$
Definición de los parámetros	N: Cantidad de no conformidades en especificaciones de requisitos T: Total de defectos
Unidad de medida	Porcentaje
Periodicidad	Semanal
Observaciones	N/A

Tabla 11 - Métrica CERTINCPER01: Índice de no conformidades por errores, cambios o ambigüedades por especificaciones de requisitos

Indicador	Índice de no conformidades por errores, cambios o ambigüedades por especificaciones de requisitos
Código	CERTINCPER01
Definición	Índice de no conformidades, debido a errores, cambios o ambigüedades en requisitos, por cada especificación de requisitos, en la etapa de certificación.
Subproceso	Pruebas
Fórmula de cálculo	N/T
Definición de los parámetros	N: Cantidad de no conformidades debido a errores, cambios o ambigüedades en especificaciones de requisitos T: Total de especificaciones de requisitos
Unidad de medida	Unidades numéricas
Periodicidad	Semanal
Observaciones	N/A

3.3 Características de las especificaciones del proyecto Venezuela

La cantidad de especificaciones de requisitos creadas para el proyecto Venezuela fue de 104, las cuales se categorizaron según su complejidad en alta, media o baja, como puede observarse en la Figura 9.

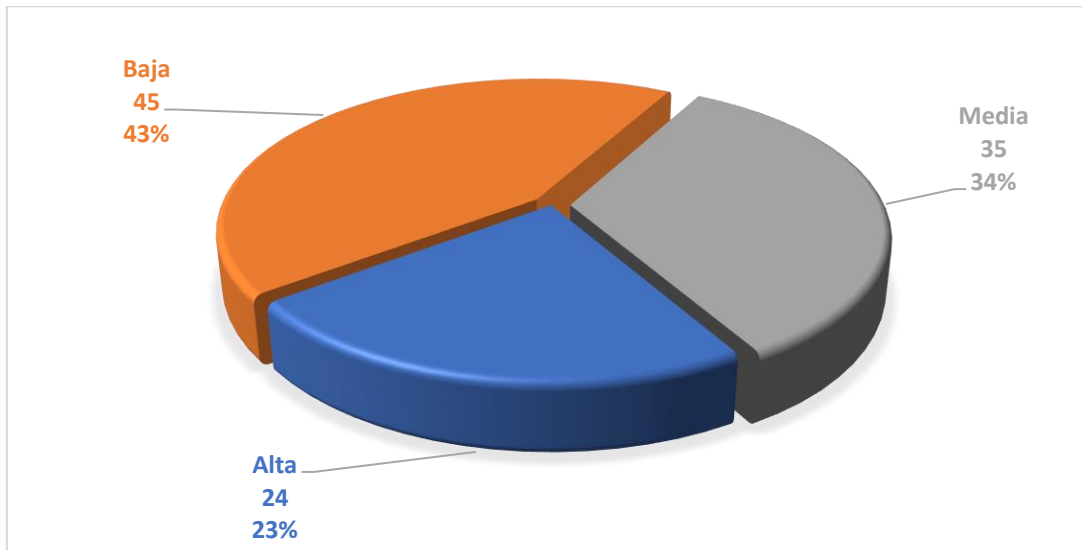


Figura 9 - Cantidad de Especificaciones de requisitos según su complejidad para el proyecto Venezuela

La cantidad promedio de páginas de las especificaciones es de 49, siendo 82 para las de alta complejidad, 48 para las de mediana complejidad y 18 para las de baja complejidad. La clasificación se llevó a cabo mediante un estimador común entre las diferentes locaciones y proveedores, a fin de tener un consenso sobre la complejidad de las especificaciones. Su complejidad depende de la cantidad de transacciones y elementos involucrados en las estimaciones. Este estimador está basado en puntos de función, que proporcionan una medición apropiada de la funcionalidad que entra a un sistema (Pressman, 2014). Si bien se sabe que las estimaciones de costo y esfuerzo no son una ciencia exacta, esta estimación proporciona una valoración razonablemente justa, con un componente de riesgo/incertidumbre aceptable. Los puntos de función se derivan usando una relación empírica basada en medidas contables (directas) del dominio de información del software y en valoraciones cualitativas de la complejidad del software (Pressman, 2014).

Dado que los formularios en las Figura 10 a Figura 13 están en inglés, se presentan los

ítems de valoración con sus siglas en inglés, de manera de facilitar la interpretación de los mismos.

Los valores de dominio de información del software, tienen la forma siguiente:

Número de entradas externas (EI):

Usualmente estas entradas tienen su fuente en los usuarios que interactúan con el sistema, o es información transmitida desde otras aplicaciones. En el caso de compañías aseguradoras, pueden ser por ejemplo aplicaciones de documentación de siniestros sobre algún bien asegurado, información procedente de algún banco o de la superintendencia de seguros. Esta información, que usualmente también actualiza archivos lógicos internos, puede traer datos de control o datos útiles para la aplicación.

Número de salidas externas (EO):

Esas salidas básicamente presentan información del sistema al usuario como pueden ser reportes, mensajes, etc.

Número de consultas externas (EQ):

Son esencialmente peticiones al sistema, el cual reacciona en forma inmediata proporcionando una respuesta, que en la mayoría de los casos es obtenida de un archivo lógico interno.

Número de archivos lógicos internos (ILF):

Cada uno de estos archivos lógicos son un conjunto de datos relacionados que se encuentran dentro de la aplicación y que se nutren de las entradas externas.

Número de archivos de interfaz externos (EIF):

Los archivos de interfaz externos son datos agrupados lógicamente que no pertenecen al sistema, sino que se hallan fuera del mismo, pero que aportan información útil que es usada por la aplicación.

En función de la recopilación del dominio de información ante mencionado y su evaluación según ciertos criterios que se hayan definido previamente, es entonces posible determinar la complejidad de cada una de las funcionalidades descritas en las especificaciones de

requisitos.

La utilización de este modelo permitió que todos los equipos involucrados pudieran tener el mismo punto de vista sobre la complejidad de las especificaciones, a fin de proporcionar un trato adecuado para cada una de ellas. En la Figura 10 se presenta un ejemplo de cálculo de punto de función para una especificación de requisito del proyecto Venezuela, mientras que en las Figuras 11, 12 y 13 se muestran cálculos intermedios.

Insurance Company Effort Estimation					
<i>1. Identification and summary</i>					
Requirement ID	XXXXXX	Main Workstream	Policy Administration		
Demand Tool ID	XXXXXX	Products	Motor Personal Line & Fleet		
Requirement Title	Create Claims Borderaux				
Requirement high level description	<p>Problem Description: It is necessary to have information regarding claims transactions (general ledger entries, status and dates). The table XXXX_EVENT_LOG show transaction record (Payment date and Claims Reserve date), what are needed for some claims reports and Claims Borderoux report.</p> <p>Solution Description: Create the new Claims Bordereaux Report, following the layout example: The fields related to Reinsurance (Risk cession) will be not available until next release. Even that, the business require these fields in order to ensure the Layout report.</p>				
IT Responsible	XXXXXXXX - Project manager	Business/ SME related	XXXXXXXX - Chief Actuary - Venezuela XXXXXXXX - Venezuelan Claims Controller		
Date	01 June 2013	Drop	TBD		
Complexity	High				
				<i>2. Effort Detail</i>	
Total (Hour):	550		550	Total Effort in MD	Total Desing in MD
Phase	<i>1. From Estimator</i>	<i>2. Adjusted (exceptions)</i>	<i>3. Final number</i>	Total Effort in hours	Total Desing in hours
Design	195,3		195,3		24
Transition	24,8		24,8		
Build	178,8		178,8		
System Test	41,3		41,3		
Integrated Test	60,5		60,5		
User Acceptance Test	46,8		46,8		
Deploy	2,8		2,8		
				550	195

Figura 10 - Ejemplo de estimación de especificación del proyecto Venezuela

En la Figura 11 se procede a la obtención de los puntos de función sin ajustar (TUFP: “Total Unadjusted Function Points”), los cuales se obtienen de la sumatoria de puntos de los ILF y los EIF.

Module	Files	Internal Logical Files (ILF)				External Interface Files (EIF)					
		DET	RET	Comp.	Points	DET	RET	Comp.	Points		
Claims Borderaux	New Claim Borderaux report	36	7	H	15					15	
	Get "Estado patrimonial"					10	2	L	5	5	
	Get Claims information					10	6	M	7	7	
	Get Policy Payment Information					14	3	L	5	5	
	Generate Intermediate borderaux table	19	4	L	7					7	
TUFP Totals:					22				17		39

Figura 11 - Cálculo de Puntos de Función para ILF y ELF

En la Figura 12 se procede a la obtención de los puntos de función sin ajustar (TUFP), los cuales se obtienen de la sumatoria de puntos de los EI, EO y EQ.

Functionality	Requirement	External Inputs (EI)				External Outputs (EO)				External Inquires (EQ)				
		DET	FTR	Comp.	Points	DET	FTR	Comp.	Points	DET	FTR	Comp.	Points	
Claims Borderaux	FE PopUp NativeNon Transational					5	2	L	4					4
	FE PopUp CustomNon Transational					5	2	L	4					4
	Borderaux Report									40	6	H	6	6
TUFP Totals:					0				8				6	10

Figura 12 - Cálculo de Puntos de Función para EI, EO y EQ

En la Figura 13 se procede al cálculo de los FP “Function Points” mediante el ajuste de los TUFP utilizando la siguiente fórmula (3):

$$FP = TUFP * VAF \quad (3)$$

En donde VAF es el factor de ajuste de los TUFP (“Value Adjustment Factor”) y se calcula de la siguiente forma (4):

$$VAF = 0,65 + (0,01 * TDI) = 1,12 \quad (4)$$

En donde TDI es el grado total de influencia (“Total degree of influence”), el cual surge del análisis de las 14 principales características funcionales de los sistemas, las cuales son evaluadas (entre 0 y 5) y dan como resultado un valor entre 0 y 70.

Insurance Company	
Estimation phase:	XXXXX
Module/Component	Claims
Autor:	Juan Pablo Mighetti
Total unadjusted function points:	49
Total Degree of Influence (TDI)	47
Value Adjustment Factor (VAF)	1,12
Total functional points	55
HS/PF MIBI (PL/SQL) Technology	10
Effort (Hours)	550

Figura 13 - Resultado de la estimación por puntos de función para un reporte de ejemplo

De lo anterior resulta que la aplicación de la formula (3) combinada con la (4) da como resultado para el ejemplo de la Figura 13:

$$FP = 49 * 1,12 = 54,88 = \sim 55 \quad (5)$$

Ahora bien, para poder calcular el esfuerzo necesario en horas de trabajo para llevar adelante las tareas respectivas, se utiliza una tasa de productividad (considerando un profesional medio) necesario para implementar un punto de función. Esta tasa depende en parte de la complejidad del lenguaje seleccionado y de las características del equipo de trabajo.

De lo cual resulta el cálculo del esfuerzo, (“Effort”), en horas necesarias, según la formula (6) y su aplicación en el ejemplo da:

$$Effort = FP * \frac{HS}{PF} = 55 * 10 = 500 \text{ Horas} \quad (6)$$

3.4 Resultados de métricas obtenidas del Proyecto Venezuela

A continuación, se describen las métricas calculadas en el proyecto Venezuela, que ayudan a dimensionar los problemas encontrados.

3.5 Tiempo de creación de especificaciones de requisitos para el proyecto Venezuela

Uno de los primeros indicios de problemas encontrados al analizar la información disponible fue el tiempo que se requería para el cierre de las especificaciones desde el punto de vista de los analistas.

Este valor representado por la métrica DFE01 “Entrega de especificación de requisitos”, que establece el tiempo promedio necesario para la creación de una especificación de un requisito determinado, para cada una de las complejidades, puede verse en la Figura 14.

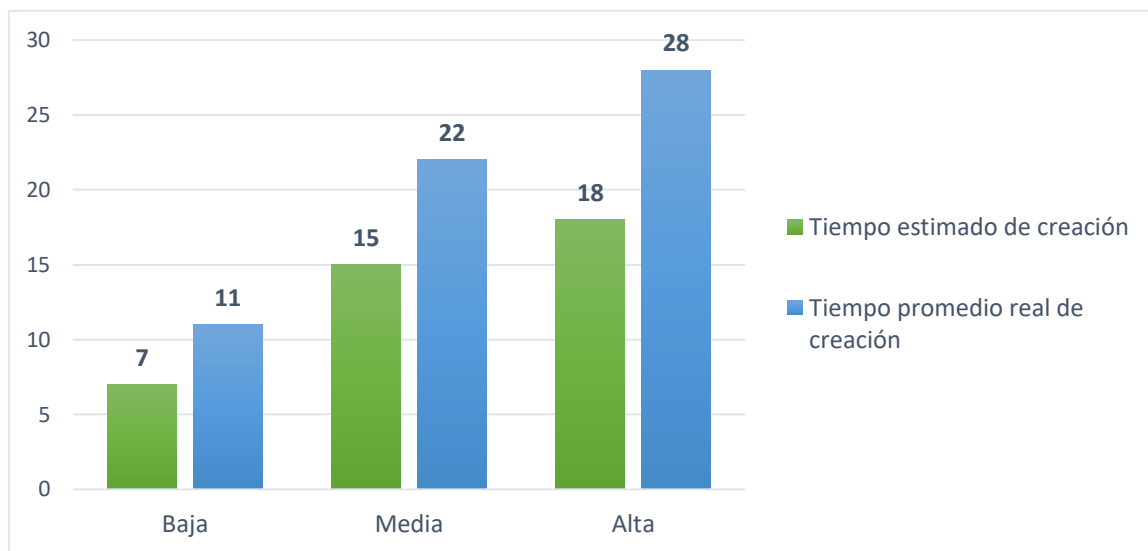


Figura 14 – Tiempo estimado y real para crear especificaciones de requisitos según su complejidad para el proyecto Venezuela (días)

El tiempo real promedio fue de 11 días hábiles para una especificación de complejidad baja, 22 días para una especificación de complejidad media, y 28 días para una de complejidad alta. Las diferencias entre tiempos estimados y reales fueron notorias.

En la Figura 15 se presenta la comparativa de los tiempos estimados y reales para analistas con y sin conocimiento del contexto de aplicación. Para aquellos analistas con conocimiento (denominados “expertos”), el tiempo de construcción fue de 7 días para las especificaciones de complejidad baja, 13 días para las de complejidad media, y 22 días para las de alta complejidad; en general estuvieron acorde al tiempo estimado para cada

complejidad. Mientras que para los que no tenían conocimientos previos del dominio (denominados “inexpertos”), los tiempos fueron de 14, 29 y 32 días respectivamente, muy superiores a los tiempos insumidos por los expertos en el dominio. Como puede apreciarse, estos tiempos superaron las estimativas previas respecto al tiempo necesario para la creación de los documentos. Asimismo, la comparativa entre el tiempo promedio real y el tiempo estimado muestra claramente la amplia diferencia entre lo estimado y lo real para los analistas inexpertos.

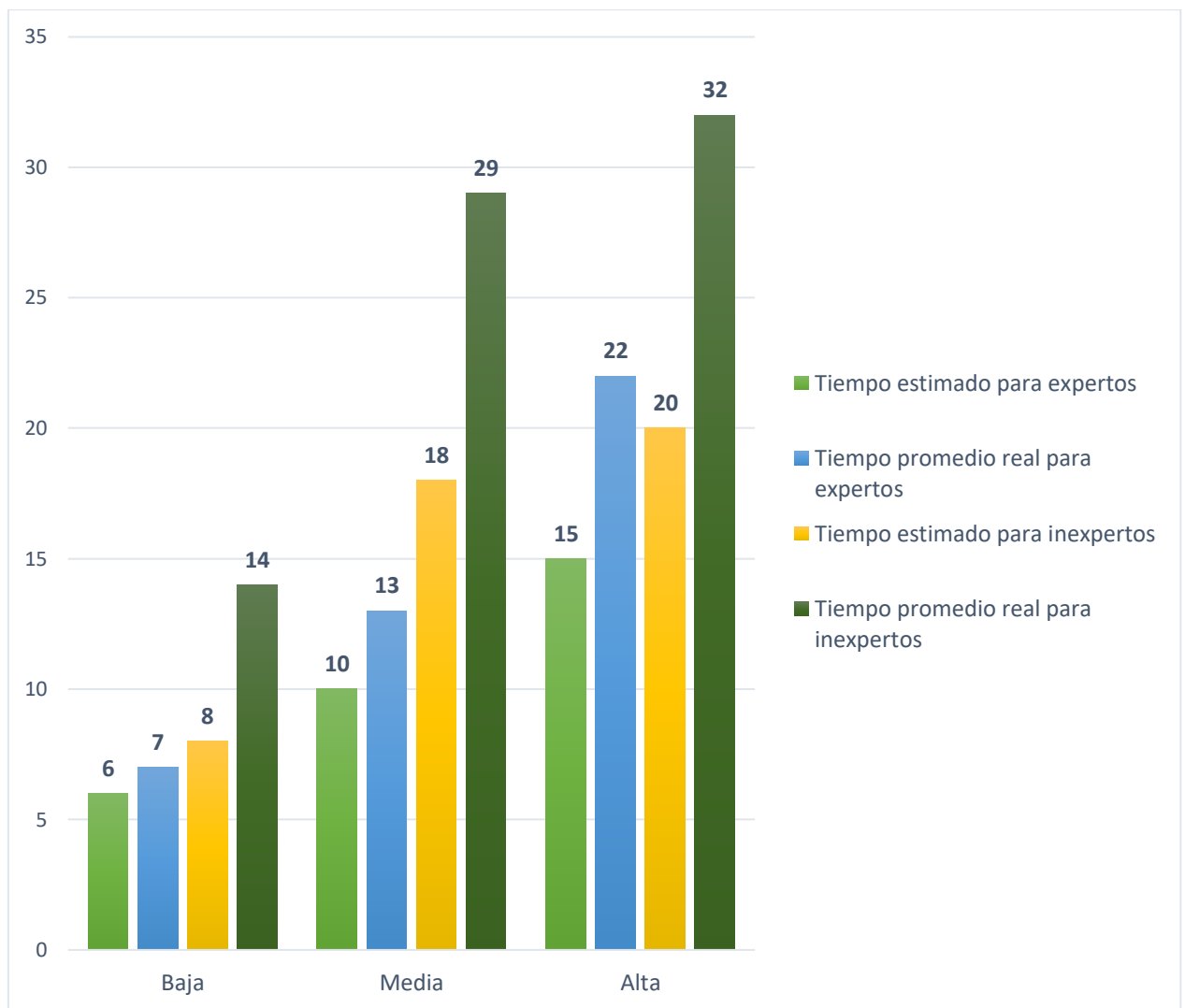


Figura 15 - Tiempo estimado y tiempo real para la creación de especificaciones según analistas con y sin conocimiento del contexto para el proyecto Venezuela (días)

En la Tabla 12 se presenta la desviación en el tiempo real de creación de las

especificaciones de requisitos en función de los tiempos estimados, donde se aprecia que los incrementos principales fueron aportados por los analistas funcionales que no tenían experiencia en el contexto de la aplicación.

Tabla 12 - Variación de tiempo real respecto al estimado para crear especificaciones para el proyecto Venezuela (días)

Complejidad	Tiempo Estimado		Tiempo Promedio Real		Variación de tiempos	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	6	8	7	14	16,7%	75,0%
Media	10	18	13	29	30,0%	61,1%
Alta	15	20	22	32	46,7%	60,0%
Variación promedio					30,9%	65,36%

3.6 Aprobaciones y rechazos de especificaciones de requisitos para el proyecto Venezuela

Los indicadores anteriores sobre tiempos excedidos en la creación de especificaciones se complementaron con los indicadores basados en la métrica DFA01 relacionada a la “Aprobación de especificaciones de requisitos” y descripto como el tiempo medio necesario para la validación de una especificación de requisito y el indicador DFR01 “Rechazo de especificación de requisitos”, definido como el promedio de cantidad de rechazos por especificación durante la aprobación de las mismas por parte de los usuarios. En la Figura 16, se puede evidenciar que el tiempo promedio real obtenido para la aprobación de las especificaciones de requisitos, métrica DFA01, por parte de los analistas funcionales fue excesivamente alto respecto al tiempo estimado, en promedio más del 60% de lo estimado.

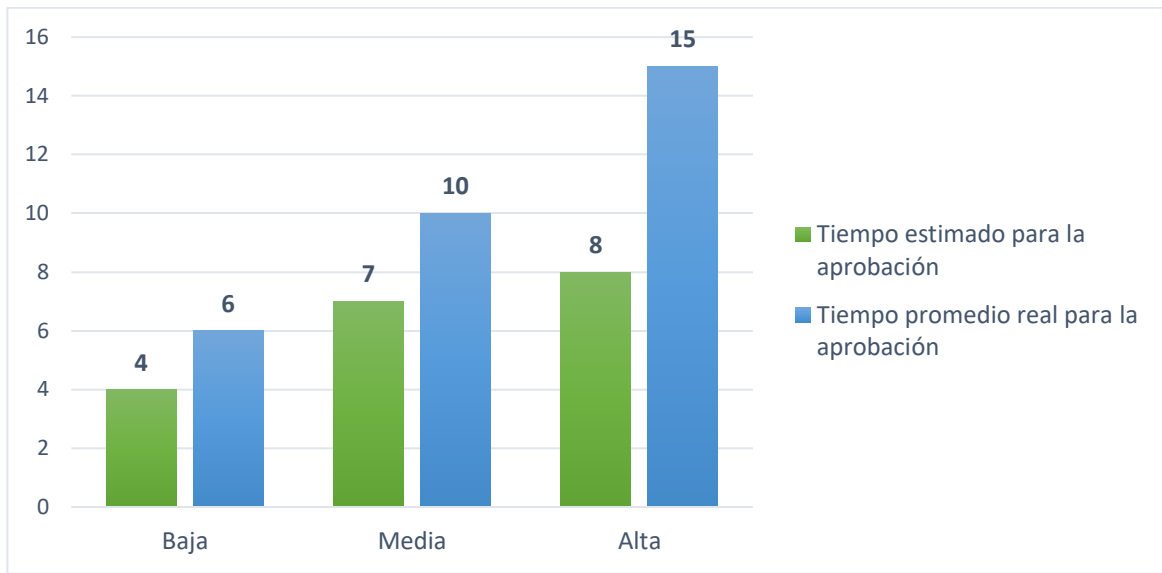


Figura 16 - Tiempo estimado y tiempo real para la aprobación de especificaciones para el proyecto Venezuela (días)

Se asumieron 4 días para la aprobación de cada especificación de baja complejidad y el tiempo promedio real fue de 6 días hábiles, mientras que en el otro extremo el tiempo estimado para aquellos de alta complejidad fue de 8 días, siendo el tiempo promedio real insumido de 15 días. En este último caso, el tiempo de aprobación casi se duplicó respecto de lo estimado.

Si bien no se dispone de una métrica para determinar la causa exacta del retraso en la aprobación, el comentario de los analistas funcionales fue que era difícil obtener el tiempo y el compromiso de los usuarios para el análisis y aprobación de las especificaciones finales de requisitos.

En la Figura 17 se observan los tiempos estimados en forma independiente para cada categoría de analista, y los tiempos promedio reales para la aprobación de las especificaciones de requisitos según el tipo de complejidad.

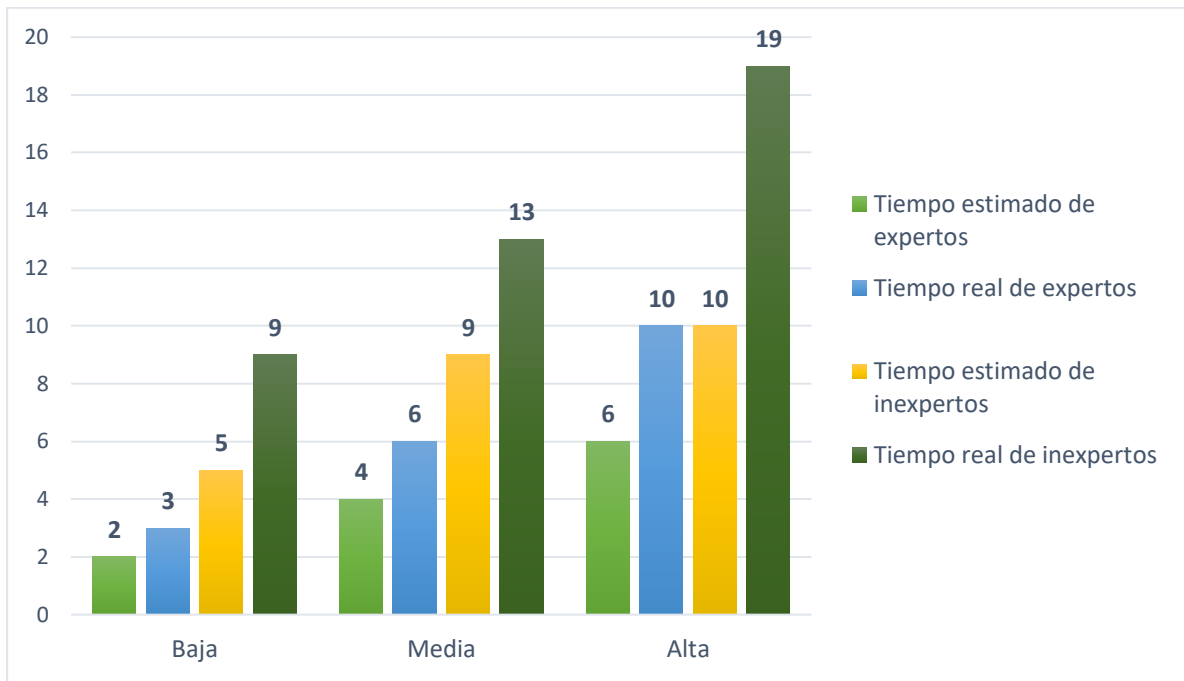


Figura 17 - Tiempo estimado y tiempo real de expertos e inexpertos para aprobar especificaciones de requisitos para el proyecto Venezuela (días)

Estos resultados reflejan que los analistas con conocimiento del contexto se desvían en menor medida de los tiempos estimados, mientras que los tiempos de los analistas que carecen de este conocimiento se ven altamente incrementados, evidenciando un mejor entendimiento de los analistas expertos con los usuarios, lo cual es lógico que esto suceda, aunque el tiempo de aprobación promedio para los analistas, en general, superó ampliamente lo estimado, principalmente para las especificaciones de complejidad alta. Estos resultados se pueden apreciar en la Tabla 13.

Tabla 13 - Variación de tiempo real respecto al estimado para aprobar especificaciones para el proyecto Venezuela (días)

Complejidad	Tiempo Estimado		Tiempo Real		Variación de tiempos	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	2	5	3	9	50,0%	80,0%
Media	4	9	6	13	50,0%	44,4%
Alta	6	10	10	19	66,7%	90,0%
Variación promedio					55,56%	71,46

Por el lado de los rechazos por parte de los usuarios durante el proceso de aprobación de las especificaciones de requisitos, identificado con la métrica DFR01, se observa que la cantidad de rechazos de los documentos creados por los analistas expertos en el contexto es considerablemente menor que aquellos que fueron creados por analistas inexpertos. Esta proporción puede verse en la Tabla 14, donde los rechazos a los analistas sin experiencia en el dominio fueron en promedio 132% superior al de los analistas con experiencia.

Tabla 14 – Variación promedio de rechazos por especificación creada por categoría de analista para el proyecto Venezuela

Complejidad	Promedio de rechazos por especificación		Variación de rechazos
	Analistas Expertos	Analistas Inexpertos	
Baja	2	5	150,0%
Media	4	9	125,0%
Alta	5	11	120,0%
Variación Promedio			131,7%

Cabe mencionar que no fueron registradas las causas de rechazos de las especificaciones de requisitos dado que no era habitual hacerlo.

3.7 Transición de especificaciones de requisitos a la fábrica de software para el proyecto Venezuela

El subproceso de Transición a Fábrica de Software y Desarrollo comienza con la

asignación de las especificaciones de requisitos a los desarrolladores en Pune. Se analizaron las métricas TRATPT01 “Tiempo promedio de transición a fábrica” que expresan el tiempo promedio de transición de las especificaciones a la fábrica de software.

Se debe tener en cuenta que el solapamiento que hay entre Argentina y Pune es solo de media hora dado que se encuentra en UTC/GMT +5:30 horas y el idioma utilizado para las transiciones fue el inglés, el cual no es la lengua madre en ninguno de las dos locaciones.

Una de las acciones que se tomaron fue la de adelantar el ingreso de los recursos humanos en Argentina media hora (08:30) y retrasar la salida de los desarrolladores de India 1 hora, logrando tener 2 horas de solapamiento sincrónico para las transiciones.

En este subproceso, el desarrollador llevaba adelante una lectura previa donde se familiarizaba con la especificación de requisito asignada, y luego se agendaba una sesión donde el analista funcional presentaba la especificación y se esclarecían dudas del desarrollador. Las herramientas utilizadas en este subproceso fueron:

- ✓ *Conferencia telefónica mediante números telefónicos contratados*
- ✓ *Video Conferencia*
- ✓ *Herramienta de escritorio compartido*
- ✓ *Herramienta de gestión de requisitos*
- ✓ *Herramienta de gestión de preguntas y respuestas*

Luego de esta presentación, limitada a 2 horas como máximo por la diferencia horaria, el desarrollador trabajaba sobre la especificación y mediante la herramienta de gestión disponible registraba las preguntas necesarias, las que eran respondidas también mediante la misma herramienta por el analista funcional y eran aclaradas durante las próximas sesiones y a lo largo de todo el proceso de desarrollo.

Los tiempos estimados para transiciones a fábrica de software y los tiempos reales promedio por especificación de requisito pueden observarse en la **¡Error! No se encuentra el origen de la referencia..**

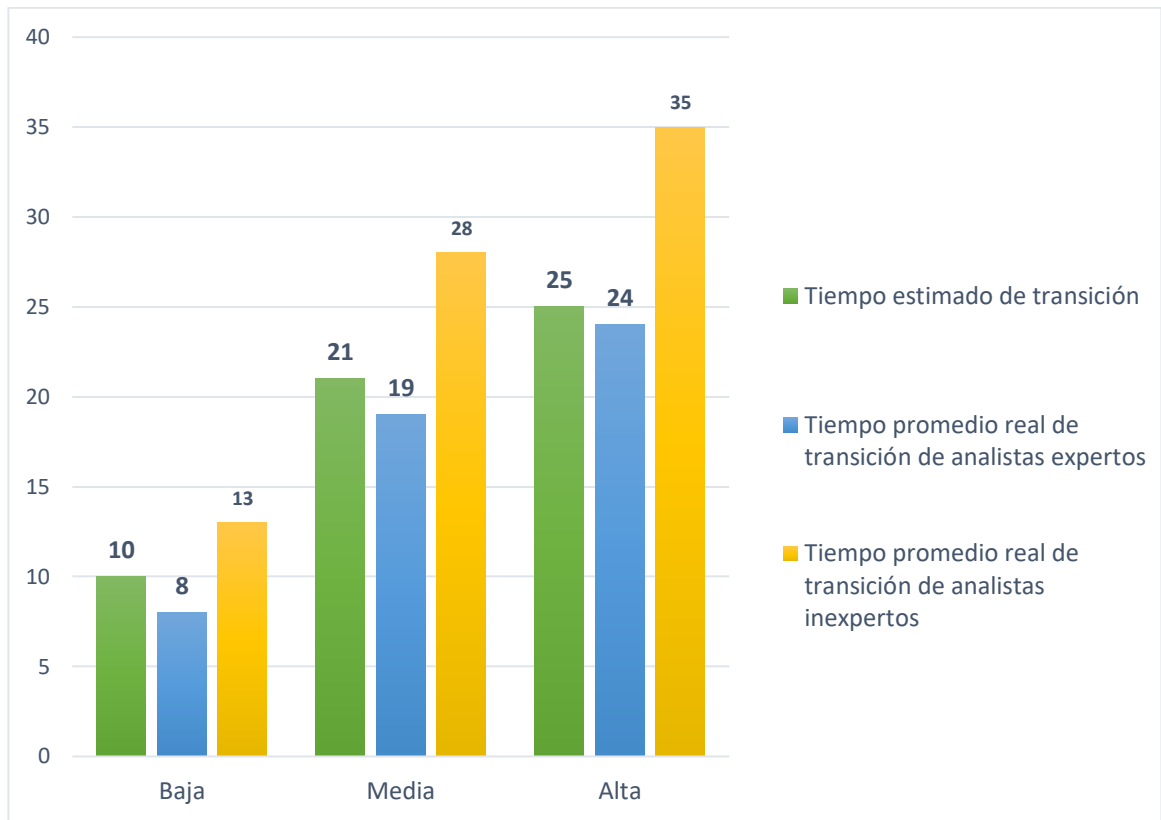


Figura 18 - Tiempo estimado de transición y tiempo promedio real de transición para el proyecto Venezuela (horas)

Del equipo de desarrollo compuesto por 38 recursos, solo 13 contaban con experiencia en el dominio, por lo que se evidenciaron tiempos de transición dispares entre estos dos tipos de recursos. Lo que también puede apreciarse (ver Tabla 15) es que las transiciones de los analistas con manejo del contexto se mantuvieron cerca de los valores estimados, con una variabilidad promedio de alrededor de un 5%. Mientras que para esta misma tarea, llevada a cabo por analistas sin un conocimiento adecuado, los tiempos de transición se vieron afectados con un incremento promedio de casi el 18,5 % respecto a los valores estimados.

Tabla 15 - Variación de tiempo real respecto al estimado en la transición a la fábrica de software por categoría de analista para el proyecto Venezuela (horas)

Complejidad	Tiempo estimado		Tiempo promedio real		Variación de tiempos	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	8	11	8	13	0%	18,2%
Media	18	24	19	28	5,6%	16,7%
Alta	22	29	24	35	9,1%	20,7%
Variación Promedio					4,9%	18,5%

No cabe duda que la distancia geográfica y temporal también jugó un papel importante, por lo que causó el incremento en los tiempos de transición. Probablemente si el solapamiento fuera mayor los tiempos de transición entre ambos grupos de analistas y de desarrolladores podrían ser reducidos, como así también la cantidad de ambigüedades y errores que podrían ser rápidamente solucionados gracias a una mayor comunicación sincrónica.

Esto es una limitación del proyecto y si bien pudo ser mínimamente mitigada modificando los horarios laborales de las locaciones involucradas, existen legislaciones en algunos de los países implicados que impidieron un cambio aún mayor como el que el proyecto hubiera requerido.

El dominio de una misma jerga y el conocimiento del contexto hacen que la comunicación sea mucho más eficiente, dado que las transiciones pueden centralizarse en aspectos más específicos, en detalles, y no tanto en conceptos básicos que consumen el tiempo de transición, el cual en este proyecto era evidentemente escaso.

Esto queda perfectamente evidenciado en los casos particulares en que las transiciones fueron llevadas a cabo entre un analista con conocimiento del contexto y un desarrollador con conocimientos similares, donde hubo una reducción significativa en los tiempos, expresados por la métrica TRATPT01 (ver Figura 19). Por el contrario, los tiempos se ven aún más dilatados sobre aquellas transiciones que se llevaron a cabo entre analistas y desarrolladores inexpertos en el dominio del problema, llegando casi a triplicar el tiempo respecto al caso de ambos recursos expertos para especificaciones de baja complejidad (las

que además representaban el 45% del total de especificaciones). (Ver en la Tabla 16)

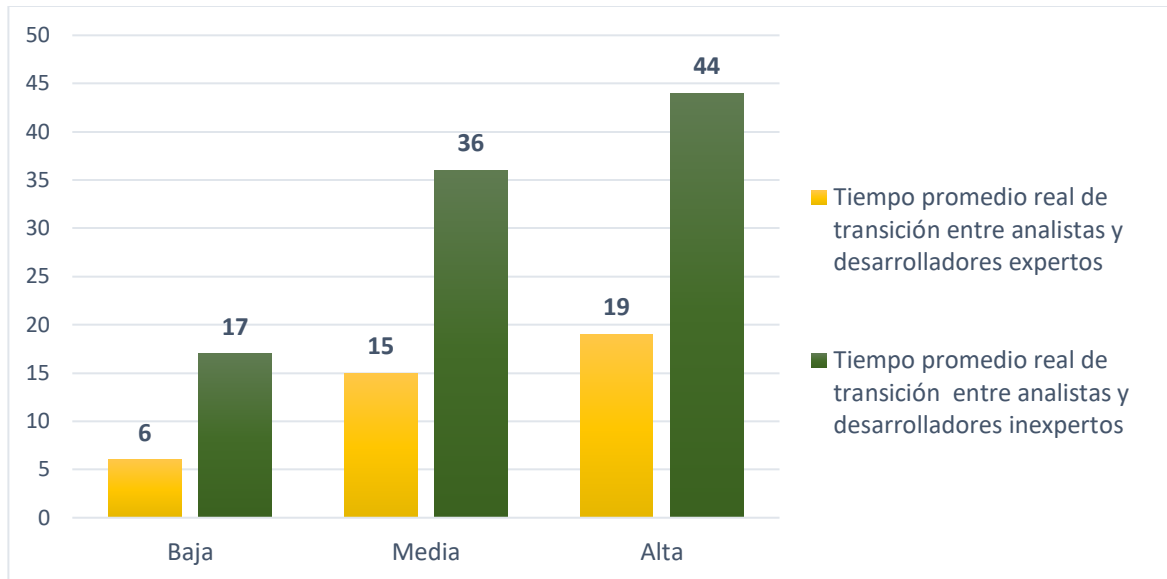


Figura 19 - Tiempo promedio real de transición entre analistas y desarrolladores considerando pares experto-experto e inexperto-inexperto para el proyecto Venezuela (horas)

En la Tabla 16 se muestra la variación respecto a los tiempos de transición de analistas y desarrolladores que compartían el mismo grado de experiencia en el contexto respecto a aquellos analistas y desarrolladores que no lo tenían, según el nivel de complejidad de las especificaciones de requisitos.

Tabla 16 - Variación de tiempos de transición entre analistas y desarrolladores expertos e inexpertos para el proyecto Venezuela

Complejidad	Tiempo real de transición (horas)		Variación entre Inexperto - Inexperto vs Experto - Experto
	Entre analistas y desarrolladores expertos	Entre analistas y desarrolladores inexpertos	
Baja	6	17	183,3%
Media	15	36	140,0%
Alta	19	44	131,6%
		Variación promedio	151,6%

La Tabla anterior evidencia un incremento del tiempo de transición entre analistas y

desarrolladores, ambos inexpertos, del 152 % respecto al par analistas y desarrolladores expertos. Como se presenta en la siguiente sub-sección, también se incrementó la cantidad de preguntas involucradas.

Las herramientas utilizadas permitieron documentar las preguntas asociadas a cada una de las especificaciones, y obtener la métrica TRACPDF01 “Cantidad de preguntas por especificación de requisitos”, expresada como la cantidad promedio de preguntas por especificación de requisitos, durante la transición y desarrollo por parte de la fábrica de software.

La Tabla 17 muestra la cantidad promedio de preguntas realizadas por parte de los desarrolladores a los analistas que efectuaron la transición, teniendo en consideración si eran desarrolladores con y sin experiencia en el contexto.

Tabla 17 - Comparación del promedio de preguntas realizadas durante la transición y desarrollo según el conocimiento en el dominio de los desarrolladores para el proyecto Venezuela

Complejidad	Cantidad de preguntas promedio de desarrolladores por especificación de requisitos		Variación de inexpertos respecto a expertos
	Desarrolladores Expertos	Desarrolladores Inexpertos	
Baja	7	9	28,6%
Media	11	15	36,4%
Alta	15	22	46,7%
		Variación Promedio	37,2%

Las distancias geográfica y temporal limitan el tiempo de comunicación sincrónica entre los involucrados y, por lo tanto, aumenta el número de preguntas, los avances en el trabajo se dilatan y los tiempos totales se incrementan. Durante la transición, los desarrolladores en su conjunto, hicieron en promedio 13,2 preguntas por especificación para esclarecer dudas. Los desarrolladores inexpertos, elaboraron en promedio un 37% más de preguntas que los expertos. Esta carencia de conocimiento del dominio y, por ende, de la jerga quedó evidenciada por la cantidad y naturaleza de las preguntas, dado que gran parte del tiempo sincrónico se debió consumir en esclarecer conceptos básicos.

3.8 No conformidades en la certificación de usuarios para el proyecto Venezuela

Siguiendo con el proceso definido y llevado a la práctica en Venezuela, posterior a la etapa de transición y desarrollo y a las pruebas de integración llevadas adelante por los equipos de la India, se llevó a cabo la certificación del software por parte de los usuarios, actividad en la cual se identificó una alta tasa de rechazos a las funcionalidades entregadas.

Una de las métricas definidas que da un indicio de la calidad de las especificaciones de requisitos es la CERTNCFF01 “Cantidad de no conformidades por errores, cambios o ambigüedades en las especificaciones de requisitos”, definida como el porcentaje de no conformidades sobre las funcionalidades detectados en la etapa de certificación debido a errores, cambios o ambigüedades en las especificaciones de requisitos.

Los tipos de defectos definidos para el reporte de no conformidades en esta actividad, por la métrica CERTNCFF01, son:

- ✓ *Defectos de codificación:*
 - *Son consecuencia de generación de código erróneo.*
- ✓ *Errores de datos:*
 - *Son consecuencia de datos inconsistentes.*
- ✓ *Errores de ambientes:*
 - *Son debido a limitaciones o características de ambiente, tales como falta de memoria, tamaño de disco, entre otros.*
- ✓ *Defectos ya existentes a la inclusión de nuevas funcionalidades:*
 - *No están relacionados con la funcionalidad introducida con la última versión de código promovida.*
- ✓ *Errores en la promoción del código a los ambientes*
- ✓ *Defectos no reproducibles:*
 - *No son posibles de reproducir en el ambiente de pruebas.*

- ✓ *Errores de ejecución de pruebas:*
 - *Son causados por la ejecución deficiente de los test de pruebas, procesamiento manual, test que no es posible ejecutar, entre otros.*
- ✓ *Defectos a errores, cambios o ambigüedades de requisitos:*
 - *Son introducidos en las especificaciones de requisitos, tales como ambigüedades, omisiones, inconsistencia y cambios en requisitos, y no fueron identificados en fases tempranas del proceso de desarrollo.*
- ✓ *Otros:*
 - *Defectos no incluidos en ninguna otra categoría, tales como errores de permisos, de usuarios, entre otros.*

Si bien se hicieron mediciones de todos los tipos de defectos encontrados, al ser el tema de esta tesis las amenazas sobre los requisitos, el foco se ha centrado sobre aquellos defectos relacionados a este tópico.

La Figura 20 muestra todos los tipos y calidad de defectos identificados en la actividad de certificación del software por los usuarios.

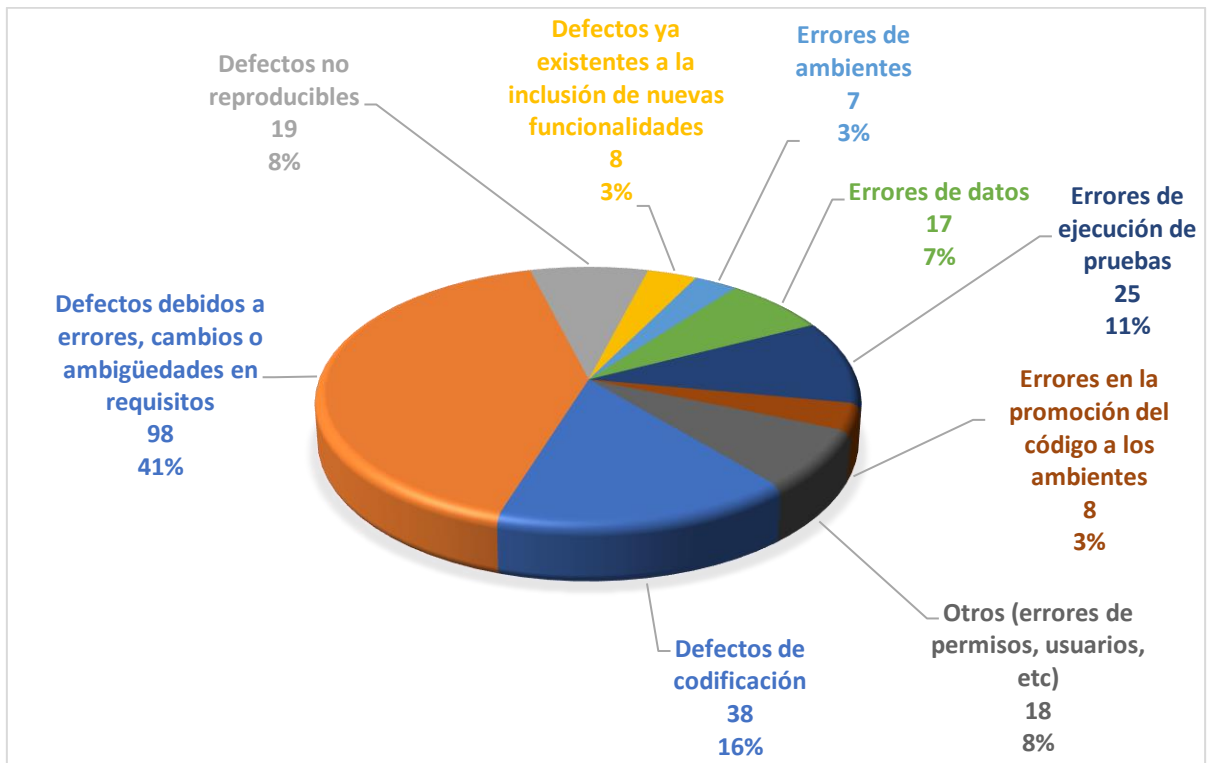


Figura 20 - Defectos encontrados en la certificación en el proyecto Venezuela

Como puede observarse, la cantidad total de 238 defectos reportados por los usuarios fue sumamente alta y esto derivó en el rechazo de los entregables por parte del grupo de usuarios. De los datos anteriores, surge que la tasa más alta es la de 41 % (98 defectos), asociada a la baja calidad de las especificaciones de requisitos, ya sea por la existencia de errores, contradicciones, ambigüedades o por la necesidad de cambios para reflejar el correcto comportamiento requerido por los usuarios.

Los 98 defectos encontrados estaban distribuidos en 75 especificaciones de requisitos de las 104 creadas (siendo el 72,1 % del total de especificaciones), lo cual implica que algunas de ellas tenían más de un defecto asociado.

La métrica CERTINCPER01, “Índice de no conformidades por errores, cambios o ambigüedades por especificaciones de requisitos”, definida en la ecuación (7), proporciona una herramienta útil de comparación entre los dos proyectos evaluados.

$$\text{Indice de defecto de Venezuela} = \frac{98}{104} = 0,94 \quad (7)$$

Si bien el índice arroja un valor de 0,94, la cantidad de especificaciones que fue necesario modificar representa el 72,1 % del total de 104 especificaciones. Un examen más detallado de correcciones necesarias sobre las especificaciones permitió visualizar que la mayor cantidad de modificaciones correspondieron a especificaciones creadas por analistas inexpertos, como puede apreciarse en la Tabla 18, donde de las 75 especificaciones con defectos, el 73 % habían sido creadas por analistas sin experiencia.

Tabla 18 - Comparativa de cantidad de especificaciones de requisitos afectadas por defectos reportados en la certificación para el proyecto Venezuela

Cantidad total de especificaciones con defectos	75
Cantidad de defectos detectados en especificaciones creadas por analistas expertos	27
Cantidad de defectos detectados en especificaciones creadas por analistas inexpertos	71
Variación en la cantidad de defectos encontrados entre especificaciones creadas por expertos frente a inexpertos	163 %

Las no conformidades encontradas en especificaciones creadas por analistas inexpertos fue 163% superior a las detectadas en especificaciones creadas por expertos. Este indicador sigue la misma línea incremental que con los rechazos de especificaciones de requisitos en la actividad de aprobación, donde también se observó un incremento sustancial de rechazos del 131,7% en las especificaciones provenientes de analistas inexpertos frente a las de los expertos (ver Tabla 14).

3.9 Análisis de resultados del Proyecto Venezuela

Uno de los primeros indicios de problemas encontrados fue el tiempo requerido para crear las especificaciones de requisitos. Se estimaron 13 días hábiles para crear una

especificación, sin embargo, el promedio de creación fue de 20 días. Además, los analistas sin conocimiento del dominio duplicaron el tiempo estimado.

En cuanto al tiempo para la aprobación de las especificaciones de requisitos por parte de los usuarios fue excesivamente alto, en promedio más del 60 % de lo estimado. En el caso de los analistas que carecían de experiencia en el dominio, estos tiempos se vieron altamente incrementados, en un 116 % respecto al promedio estimado. Asimismo, la cantidad de rechazos de documentos creados por analistas sin experiencia en el dominio fue considerablemente mayor que la de aquellos creados por analistas con experiencia; los primeros sufrieron un 132 % más de rechazos que los segundos.

En la transición de las especificaciones de requisitos a la fábrica de software en India, los tiempos se vieron más dilatados cuando la transición se llevaba a cabo entre analistas y desarrolladores ambos inexpertos, llegando a duplicar ampliamente el tiempo respecto al caso de ambos recursos expertos. En el caso de transiciones entre analistas y desarrolladores expertos, el tiempo de transición fue un 20 % menos del tiempo promedio, mientras que la transición entre analistas y desarrolladores ambos inexpertos, insumió más de un 50 % sobre el tiempo promedio.

Durante la transición, los desarrolladores hicieron en promedio 13 preguntas por especificación para esclarecer dudas. Los desarrolladores inexpertos, elaboraron un 37% más de preguntas que los expertos. Una carencia que quedó evidenciada por la cantidad y naturaleza de las preguntas.

En las pruebas de certificación de usuarios se tuvo una alta tasa de defectos. El 41 % de ellos provenían de errores, contradicciones y ambigüedades en los requisitos, con lo cual se debieron corregir el 72 % de las especificaciones. La mayor cantidad de defectos 73 % fueron encontrados en especificaciones creadas por analistas inexpertos, siendo esta cantidad 163 % superior a la de analistas expertos. Este indicador siguió la misma línea incremental que con los rechazos de especificaciones en la aprobación, donde la cantidad de rechazos en especificaciones creadas por inexpertos fue 132 % superior a las especificaciones provenientes de analistas expertos.

Cabe mencionar que independientemente de los conocimientos sobre el dominio de los

analistas y desarrolladores, los tiempos reales frente a los tiempos estimados fueron en general bastante superiores, tanto en la creación de especificaciones, en su aprobación y en la transición a la fábrica de software.

3.10 Conclusiones del análisis del Proyecto Venezuela

Como conclusión del análisis de las métricas del proyecto Venezuela, se puede determinar que los problemas encontrados surgen de la inacción y falta de acciones de mitigación adecuadas frente a las amenazas propias del modelo distribuido respecto a los requisitos.

Las amenazas a las que estuvo afectado el proyecto Venezuela, y que probablemente fueron subestimadas, pueden verse en la Tabla 19, identificándose las métricas relacionadas con requisitos asociadas a dichas amenazas.

Se considera que gran parte de estas amenazas impactaron en los requisitos, dado que la mayoría de los defectos detectados en la certificación del software procedían de las especificaciones. Por otro lado, el proyecto sufrió fuertes desviaciones en el costo - incremento del 53 % respecto al presupuesto - y el plazo del proyecto - incremento del 48% respecto de lo planificado -. El mayor costo se vio reflejado como consecuencia del retraso en los tiempos, dado el tipo de contrato con el proveedor era necesario extender la contratación de todo el equipo, repercutiendo en el costo total. La extensión de tiempo surgió de manera indirecta por rechazos en la aprobación de especificaciones, cantidad de preguntas y tiempos de transición. Si bien la cantidad de defectos por especificación puede parecer baja, el gran insumo de tiempo y, por ende costo, ocurrió en las actividades de definición de requisitos.

Tabla 19 - Amenazas identificadas para el proyecto Venezuela

Amenaza	Venezuela	Métrica asociada
Comunicación inadecuada	X	<ul style="list-style-type: none"> Entrega de especificación de requisitos (DFE01) Aprobación de especificación de requisitos (DFA01) Rechazo de especificación de requisitos (DFR01) Cantidad de preguntas por especificación de requisitos (TRACPDF01) Tiempo promedio de transición a fábrica (TRATPT01)
Distancia geográfica y temporal	X	<ul style="list-style-type: none"> Tiempo promedio de transición a fábrica (TRATPT01) Cantidad de preguntas por especificación de requisitos (TRACPDF01)
Diferencias culturales y/o idiomáticas	X	<ul style="list-style-type: none"> Tiempo promedio de transición a fábrica (TRATPT01) Cantidad de preguntas por especificación de requisitos (TRACPDF01)
Gestión del conocimiento	X	<ul style="list-style-type: none"> Tiempo promedio de transición a fábrica (TRATPT01) Cantidad de preguntas por especificación de requisitos (TRACPDF01)
Falta de confianza e involucramiento	X	<ul style="list-style-type: none"> Entrega de especificación de requisitos (DFE01) Aprobación de especificación de requisitos (DFA01) Rechazo de especificación de requisitos (DFR01)
Carencia del dominio del problema	X	<ul style="list-style-type: none"> Entrega de especificación de requisitos (DFE01) Rechazo de especificación de requisitos (DFR01) Cantidad de preguntas por especificación de requisitos (TRACPDF01) Tiempo promedio de transición a fábrica (TRATPT01)
Problemas técnicos o de herramientas		<ul style="list-style-type: none"> N/A
Ambigüedad, contradicciones y/o falta de claridad en las especificaciones	X	<ul style="list-style-type: none"> Rechazo de especificación de requisitos (DFR01) Tiempo promedio de transición a fábrica (TRATPT01) Cantidad de preguntas por especificación de requisitos (TRACPDF01) Cantidad de no conformidades por errores, cambios o ambigüedades en las especificaciones de requisitos (CERTNCF01) Índice de no conformidades por errores, cambios o ambigüedades por especificaciones de requisitos (CERTINCP01)
Objetivos individuales		<ul style="list-style-type: none"> N/A
Rotación del equipo		<ul style="list-style-type: none"> N/A

En virtud de los resultados analizados, se observa que la carencia de conocimiento del dominio del problema por parte de los analistas asignados al proyecto (69% de analistas inexpertos en el dominio), y por ende, el desconocimiento del léxico del contexto, impactó significativamente en los tiempos de creación de las especificaciones de requisitos y en

cascada en los tiempos de aprobación de las especificaciones y la cantidad de rechazos obtenidos.

Esto se encuentra alineado con lo que se menciona en (Hadad, y otros, 1999), respecto a que la comunicación basada en diferentes léxicos tiene una mayor probabilidad de fracasar y evidentemente la relación entre los usuarios y los analistas enviados a Venezuela sin conocimiento del contexto no fue la esperada a la luz de los tiempos necesarios para la aprobación de las especificaciones (véase métrica DFA01).

El lenguaje semi-técnico por parte de los analistas se vio confrontado con el lenguaje propio del contexto que utilizaban los usuarios y, por lo tanto, el mutuo entendimiento se vio afectado, llevando a incrementar no solo los tiempos de elicitación, sino afectando también la calidad de las especificaciones. Esto queda en evidencia cuando se analiza el valor de la métrica DFR01, relacionada a los rechazos por parte de los usuarios, debido a errores y ambigüedades en las especificaciones.

Esto indica que la comunicación y el entendimiento no fueron lo suficientemente eficaces y probablemente tampoco existiera una relación de confianza que permitiera acelerar los tiempos de aprobación y disminuir los rechazos en el proceso de aprobación. Esto está claramente relacionado con lo expuesto en la sub-sección 2.11 sobre las amenazas en proyectos de desarrollo global de software, donde en este punto entran justamente en juego las amenazas respecto a: *Comunicación inadecuada, Carencia del dominio del problema, Ambigüedad, contradicciones y/o falta de claridad en las especificaciones, y Falta de confianza e involucramiento.*

Es más probable que el proceso de elicitación sea exitoso si existe una comunicación efectiva entre los involucrados (Nahar & Student, 2013) y es de vital importancia construir una relación de confianza, la cual puede ser reforzada mediante el manejo de un mismo léxico, contribuyendo con la reducción de defectos y colaborando con la eliminación de las ambigüedades en el proceso (Leite, y otros, 2004).

Las dificultades de comunicación también se vieron reflejadas en los tiempos de transición a fábrica de software y en la cantidad de preguntas entre los analistas y los desarrolladores, principalmente en aquellos con escaso o nulo conocimiento del dominio de seguros.

La *distancia geográfica y temporal* que limitaba a escasas 2 horas diarias la interacción sincrónica de los equipos de analistas y de desarrolladores, varios de estos también inexpertos (69% del equipo de analistas y 66% del equipo de desarrolladores), resintió la comunicación entre los involucrados reduciendo su efectividad y generando falta de confianza e involucramiento entre los miembros del propio equipo del proyecto. En este sentido, es necesario buscar alguna forma alternativa para mitigar esta situación, dado que esta limitación horaria, afecta al vínculo entre los involucrados (Damian, y otros, 2003), potencia el incremento de la duración de las transiciones, afecta la calidad, e impacta directamente en los tiempos y costos del proyecto. En este aspecto, las amenazas no mitigadas se refieren a: *Comunicación inadecuada, Distancia geográfica y temporal, y Falta de confianza e involucramiento.*

Por otro lado, las *diferencias culturales e idiomáticas* entre los involucrados en las transiciones, donde el idioma utilizado no es el nativo de ninguno de los dos países, sumado a la falta de un repositorio común con información sobre el proyecto que permitiese evacuar dudas, mitigar errores, ambigüedades, entre otras consultas, originó que se manifestasen dificultades adicionales. Algo evidente debido a las métricas analizadas fue que muestran una cantidad significativa de ambigüedades, errores y/o contradicciones en las especificaciones, dando como resultado un alto índice de no conformidades en la etapa de certificación de usuarios. Los desarrolladores que no manejaban el mismo léxico, ni tenían conocimiento del contexto, elaboraron más preguntas que aquellos que sí poseían estas características. Una carencia que si bien era conocida, pero que quedó evidenciada aún más por la cantidad y la calidad de las preguntas, fue que no existía un repositorio de información de consulta ni para los analistas sudamericanos ni para los desarrolladores en la India, de forma tal que no podían evacuar en forma personal dudas de índole simple o medianamente complejas, como por ejemplo, cuál era la relación entre un siniestro y un reclamo de un asegurado, cómo era el flujo básico de este último, o cuáles eran las características necesarias para generar uno nuevo. En base a estas observaciones, las amenazas que además afectaron al proyecto fueron: *Diferencias culturales y/o idiomáticas, y Gestión del Conocimiento.*

Las amenazas correspondientes a *Problemas técnicos o de herramientas, Objetivos*

individuales y *Rotación del equipo*, no estuvieron presentes en el proyecto Venezuela. Tal como se mencionó en la sub-sección 3.7, las herramientas tales como: video conferencia, escritorios compartidos, herramientas de gestión y otras, estaban a disposición de los equipos, no evidenciándose problemas técnicos mayores que ocasionasen impactos significativos.

Respecto a la amenaza *Objetivos individuales*, el proveedor de desarrollo y testing seleccionado para los proyectos llevaba varios años trabajando con la empresa y conocía en detalle la estrategia regional. La cúpula gerencial, tanto de la empresa como del proveedor, estaba alineada y contaba con la visibilidad adecuada a fin de tomar las decisiones correctas y lograr el compromiso y alineamiento de los integrantes de cada uno de los equipos con los objetivos comunes del proyecto. Reuniones de comité de dirección se organizaban mensualmente donde no solo se exponían los resultados de los proyectos, sino que tanto la empresa como los proveedores exponían sus preocupaciones, riesgos y acciones de mitigación respectivas a los diferentes gerentes involucrados a fin de alinear los objetivos y prioridades de los equipos del proyecto.

Por el lado de la *Rotación de equipo*, existían penalidades para cada proveedor en caso de que se superara un determinado valor porcentual de rotación del equipo de proyecto asignado, por lo que se habían implementado estrategias y llevado adelante actividades para la retención de recursos durante los proyectos, obteniendo una rotación de recursos menor al 10%, lo cual fue un valor aceptable para el proyecto.

De los 20 millones de dólares destinados al primer año de la transformación regional, se asignaron específicamente a la implementación del primer release de Venezuela 9 millones de dólares. El resto del presupuesto del primer año fue necesario para la creación de la estructura de proyecto, infraestructura para soportar la implementación de toda la región, como ser la creación del Datacenter, licencias, compra de servidores y equipos informáticos, entre otros. El costo final de la implementación del primer release para Venezuela fue de 13,8 millones dólares, lo cual representa un 53,4 % superior al estimado. Respecto a los tiempos estimados, estos eran de 10,5 meses y sufrieron una variación de casi un 47,6 %, extendiendo el tiempo hasta los 15,5 meses.

Las amenazas detalladas son las que debieron ser cuidadosamente evaluadas y consideradas para el proyecto Brasil, a fin de no subestimarlas o ignorarlas, sino actuar proactivamente a fin de mitigar su impacto en los requisitos y, finalmente, evitar las desviaciones en costo y tiempo expuestas anteriormente.

4. Propuesta de Mitigación de Amenazas a Requisitos aplicada en otro Proyecto

Dado que ambos proyectos forman parte de un mismo programa de transformación regional, que ambos persiguen un objetivo común y que las características que se van a presentar en las subsecuentes secciones demuestran sus similitudes, es de suponer que el proyecto Brasil se verá afectado por las mismas amenazas que su predecesor, el proyecto Venezuela. En este sentido es necesario considerar cuales son las amenazas potenciales a las que estaría sujeto y de esta forma, sería posible implementar las mejoras necesarias a los procesos a fin de mitigar o anular dichas amenazas.

4.1 Características del proyecto Brasil

El proyecto Brasil fue de similares características al de Venezuela, si bien el volumen de especificaciones era mayor, el proceso a seguir y la planificación eran similares. En este proyecto se planificó una duración de 16,5 meses con un presupuesto de 14,5 millones de dólares.

El equipo inicial estuvo conformado por 16 analistas funcionales en Argentina (Buenos Aires), 47 desarrolladores en India (Pune), 8 testers en India (Chennai) y 60 usuarios especializados en Brasil (San Pablo). La distribución del equipo de trabajo del proyecto Brasil en cuanto al nivel de experiencia en el dominio puede verse en la Tabla 200. Al igual que en el proyecto Venezuela hubo una proporción similar de analistas y desarrolladores sin conocimiento en el dominio de seguros. Como ya se mencionó en la sección 3.1, existió una carencia de recursos disponibles con conocimiento en el área de seguros, tanto en el mercado laboral de Buenos Aires como en el de Pune.

Tabla 20 - Distribución de competencias del equipo para el proyecto Brasil

Proyecto Brasil		
<i>Roles</i>	<i>Analistas Funcionales</i>	<i>Desarrolladores</i>
Con Conocimiento del Dominio	5	15
Sin Conocimiento del Dominio	11	32
Cantidad total	16	47

4.2 Acciones de mitigación de las amenazas mediante LEL y Escenarios

Al igual que en Venezuela, el proyecto Brasil está inmerso en la modalidad de desarrollo global y, por consiguiente, sujeto a las amenazas ya mencionadas. La no mitigación de estas amenazas implicaría impactos significativos en el proyecto Brasil al igual que ocurrió en el proyecto Venezuela.

En base al análisis post-mortem del proyecto Venezuela, se estableció una propuesta de cómo las amenazas a requisitos identificadas podrían ser mitigadas. Esta propuesta se basa en el uso de modelos en lenguaje natural y puede apreciarse en la Tabla 21, que muestra con “X” aquellas amenazas donde es posible lograr una mitigación en forma directa, y con “/” aquellas que pueden ser mitigadas en forma indirecta.

Tabla 21 - Amenazas y su posible mitigación mediante LEL y Escenarios

<i>Amenaza</i>	<i>Modelos</i>	<i>LEL</i>	<i>Escenarios Actuales</i>	<i>Escenarios Futuros</i>
<i>Comunicación inadecuada</i>		X	X	X
<i>Distancia geográfica y temporal</i>		X	X	X
<i>Diferencias culturales y/o idiomáticas</i>		X	X	X
<i>Gestión del conocimiento</i>		X	X	X
<i>Falta de confianza e involucramiento</i>		X	X	X
<i>Carencia del dominio del problema</i>		X	X	X
<i>Problemas técnicos o de herramientas</i>			/	/
<i>Ambigüedad, contradicciones y/o falta de claridad en las especificaciones</i>		X	X	X
<i>Objetivos individuales</i>				X
<i>Rotación del equipo</i>		X	X	X

X: Mitigación Directa

/: Mitigación Indirecta

A continuación se describen las razones por las que el uso de los modelos LEL, Escenarios Actuales y Escenarios Futuros, como parte de la construcción de requisitos, pueden colaborar en la mitigación de las amenazas que fueron identificadas en el proyecto Venezuela.

I. Mitigación de la comunicación inadecuada:

La comunicación es un elemento complejo pero fundamental en cualquier relación humana (Hadad, y otros, 1999), y lo es mucho más cuando está inmersa en un modelo distribuido, como queda evidenciado en el proyecto anteriormente analizado.

Como herramienta de mitigación, el LEL permite conocer la jerga de los usuarios, lo que a su vez, facilita interactuar con los usuarios, asegurando una buena comunicación y

compartiendo el mismo vocabulario durante todo el proceso de desarrollo (Hadad, y otros, 1997). Al utilizar el mismo lenguaje, mejora la relación de los miembros del equipo, permitiendo al emisor compartir un mensaje claro y comprensible en lenguaje natural, capaz de ser decodificado por el receptor de forma adecuada, minimizando las ambigüedades en las especificaciones de requisitos y dentro de una estructura de conocimiento adecuada, donde los conceptos se relacionan entre sí (Aranda, y otros, 2008).

A su vez, como herramienta de mitigación adicional, los escenarios actuales facilitan una visión adecuada del problema en estudio y proporciona también un medio de comunicación entre los involucrados (Leite, y otros, 2000). Teniendo en cuenta los resultados obtenidos en el proyecto Venezuela, la utilización de ambos modelos colaboraría con la mejora de la comunicación y permitiría unificar un léxico a compartir entre todos los involucrados.

La utilización de lenguaje natural para la descripción de situaciones proyectadas con el nuevo software por medio de los escenarios futuros facilitará la validación de las especificaciones de requisitos por parte de los usuarios y cumpliría con el objetivo de mejorar la comunicación entre los involucrados (Leite, y otros, 2004). Como surge del análisis de las métricas obtenidas en el proyecto Venezuela, el proceso de validación/aprobación de las especificaciones es uno de los puntos clave a mejorar, dado la gran cantidad de rechazos y el impacto que provocaron.

II. Mitigación de la distancia geográfica y temporal:

La distancia geográfica y temporal es un factor que puede ser mitigado parcialmente mediante el planeamiento de viajes frecuentes o modificando los horarios laborales de las locaciones a fin de incrementar el tiempo de solapamiento. Sin embargo, ambas acciones están limitadas por temas de costos y por legislaciones que impiden la modificación sustancial de los horarios laborales. Si bien algunas de estas acciones ya se habían tomado en el proyecto Venezuela, no hubo una mejora significativa evidenciada en las métricas recolectadas, por lo que queda claro que es necesario tomar acciones adicionales para su mitigación.

Como se menciona en (Damian, y otros, 2003), la distancia geográfica y temporal atenta contra la correcta comunicación, limitando o casi anulando la comunicación informal y

cara a cara. Estos dos puntos son de extrema importancia para la rápida evacuación de dudas, ambigüedades o aclaraciones, algo que no fue posible en Venezuela, donde además, la gran diferencia horaria forzó mayormente la utilización de mecanismos de comunicación asincrónicos, como mails o gestión de preguntas mediante diversas herramientas.

En este sentido, el LEL y los escenarios proporcionan una visión de las relaciones entre las tareas y/o flujos del sistema (Leite, y otros, 2000), que permiten evacuar dudas o tomar decisiones rápidamente en forma individual, minimizando la cantidad de preguntas durante las escasas o reducidas sesiones de transición y evitando tener que esperar a los horarios laborales de otras locaciones, siendo muchas veces diferencias temporales considerables que consumen gran parte del tiempo de los proyectos (Fryer, y Gothe 2008).

Adicionalmente, los escenarios actuales describen situaciones del contexto de aplicación (Leite, y otros, 2000) que colaboran o complementan la correcta interpretación de la problemática del cliente o usuario, reduciendo la necesidad de viajes frecuentes y maximizando el aprovechamiento del solapamiento de los horarios laborales, evitando temas superfluos o discusiones básicas respecto a las necesidades o flujos, entre otros aspectos.

III. Mitigación de las diferencias culturales y/o idiomáticas:

El idioma sigue siendo uno de los factores determinantes en los modelos distribuidos, y en el caso de los proyectos de Venezuela y Brasil, el inglés usado como idioma común no es la lengua natural de los analistas argentinos ni de los desarrolladores indios ni de los usuarios venezolanos y brasileños. Aún cuando el idioma puede ser el mismo, dependiendo del país de origen, el significado de ciertas palabras puede ser diferente (hasta incluso ofensivo), pudiendo dar lugar a ambigüedades o malentendidos (Aranda, y otros, 2008). Aún si el país fuese el mismo, es frecuente que el contexto local/regional de las frases, palabras o conceptos puede ser totalmente diferente y dar lugar a malas interpretaciones. Queda en evidencia que las diferencias culturales e idiomáticas, junto con las carencias de algunos analistas y desarrolladores acerca del conocimiento del contexto de aplicación, han afectado al proyecto Venezuela y podrían afectar la confianza y la comunicación del proyecto Brasil.

En este sentido, el LEL proporciona una enorme ayuda, dado que acota y/o unifica el vocabulario para la interacción de los involucrados, proporcionando *símbolos* de uso compartido, que representan las palabras o frases peculiares del contexto, como ya se describió en la sub-sección 2.3.

Por ende, el LEL es esencial para reducir las probabilidades de fracaso de la comunicación entre los miembros del equipo, lo que tiene mayores posibilidades de ocurrir si la comunicación se maneja con léxicos distintos (Hadad, y otros, 1999).

En relación a la cultura de poder, culturas como la India evitan desafiar la autoridad y se limitan a acatar los lineamientos dispuestos, sin cuestionar las decisiones, aún cuando no las entiendan enteramente o identifiquen ciertas contradicciones en las mismas (Fryer, y Gothe 2008). Esto puede dar lugar a la inclusión de defectos que potenciarían el incremento de las no conformidades reportadas por los usuarios en las etapas de certificación, problema identificado en el proyecto Venezuela.

El LEL y los escenarios proporcionan un elemento extra para evacuar preguntas y clarificar dudas por sí mismas, las que probablemente no se harían por esta sumisión o deferencia hacia las figuras de autoridad, de manera tal que facilitarán mitigar potenciales problemas o defectos a futuro.

IV. Mitigación de la falta de confianza e involucramiento:

El LEL y los escenarios utilizan lenguaje natural para su creación (Hadad, y otros, 1999), una característica que es preferida por los usuarios y clientes para la comunicación con los analistas. Esto genera como resultado un mayor compromiso con el proyecto por parte de los usuarios y mejora la confianza entre los miembros (Leite, y otros, 2004). Como se evidenció anteriormente, la falta de confianza era un atributo carente en la relación entre los involucrados del proyecto Venezuela e impactó en varias de las métricas recolectadas, como ser la cantidad de aprobaciones o los tiempos necesarios para crear las especificaciones de requisitos.

La efectividad de la comunicación también se vio afectada y queda en evidencia con las métricas DFE01 “Entrega de especificación de requisitos”, TRATPT01 “Tiempo promedio de transición a fábrica” y TRACPDF01 “Cantidad de preguntas por especificación de

requisitos”, las cuales evidencian valores superiores a los esperados. La utilización de lenguaje natural, además de colaborar con el vínculo de confianza, contribuye adicionalmente con la optimización de los tiempos de transferencia de conocimiento.

Como elemento de mitigación de esta amenaza, se propone el uso del modelo LEL, el cual permite una homogenización del vocabulario permitiendo una comunicación basada en el mismo léxico y logrando que esta sea clara y eficaz entre los involucrados (Hadad, y otros, 1997).

Adicionalmente, como menciona Carroll (Carroll, 2000), dentro de los propósitos de los escenarios se puede mencionar que potencian el involucramiento y/o compromiso por parte de los clientes y usuarios.

V. Mitigación de la carencia del dominio del problema:

En aquellas métricas que permiten un desglose entre analistas y/o desarrolladores que conocen o no el dominio, queda en evidencia la diferencia existente entre ellos, en cuanto a la calidad de lo producido, los tiempos de creación requeridos, la calidad de las comunicaciones o los tiempos de comunicación requeridos.

Para un correcto entendimiento del problema es necesario dominar el contexto del problema, y el inicio de este proceso cognitivo es conociendo la jerga o vocabulario propio de dicho contexto. Este es el principal objetivo del LEL, proporcionar un medio para el entendimiento del contexto de aplicación (Leite, y otros, 2004). La implementación del LEL debería permitir a aquellos recursos que desconocen el contexto, mejorar la calidad y productividad de sus entregables.

La creación de escenarios en lenguaje natural está regida por dos reglas de oro:

- ✓ *Mantener los documentos fáciles de leer.*
- ✓ *Mantener los documentos precisos, concisos, detallados, consistentes y tan completos como sea posible.*

Los escenarios actuales son usados para modelar y comprender las situaciones que ocurren en el contexto actual, mientras que los escenarios futuros colaboran en la comprensión de las situaciones que deben prevalecer una vez que el sistema se ha incorporado a ese

contexto (Leite, y otros, 2004), de esta forma se logra tener una correcta visión del problema, donde estamos y hacia dónde vamos.

La construcción de especificaciones de requisitos como elementos atómicos, sin analizar o considerar su relación con otros requisitos dentro del contexto, trae aparejado omisiones, errores, contradicciones y ambigüedades, que de no ser identificadas en la actividad de validación de especificaciones provocarán una amplia cantidad de rechazos, tal cual ocurrió en el proyecto Venezuela. Al no ser detectadas en esta etapa, desencadenarían una gran cantidad de rechazos en las etapas posteriores, como en la de certificación del software que realizan los usuarios.

La implementación del LEL y de escenarios para la mitigación del desconocimiento del contexto, debería traer como resultado una mejora en los valores de las métricas asociadas a estos aspectos.

VI. Mitigación de los problemas técnicos o de herramientas:

Si bien en el proyecto Venezuela no se evidenciaron problemas relacionados a este punto, la utilización de escenarios es, por sí misma, capaz de mitigar indirectamente la carencia de herramientas adecuadas para trabajar bajo el modelo distribuido.

Al igual que las herramientas, los escenarios colaboran para llevar adelante las actividades relacionadas a la creación de las especificaciones de requisitos. Una correcta utilización de herramientas puede potenciar la productividad para cumplir el objetivo propuesto, aún si estas no son las más idóneas o más avanzadas. El proyecto Venezuela contaba con herramientas de gestión de configuración, como así también herramientas de interacción como chats, teléfonos, números para llamadas grupales, y hasta equipos de video conferencia, lo que reducía los problemas relacionados a este punto.

Aún así, es claro que las herramientas necesitan “nutrirse” o usarse adecuadamente para obtener el mejor rédito posible y ahí es donde los escenarios juegan un papel fundamental, permitiendo un correcto entendimiento de las necesidades, una correcta priorización de los requisitos, y potenciando el uso de las herramientas de comunicación, mitigando los impactos negativos en los requisitos por carencias o problemas relacionadas a las herramientas. Por ejemplo, en algunos proyectos no es posible contar con aplicaciones de

chat como skype, hang out o similares, para mitigar la falta de comunicación informal y lograr, entre otras cosas, un correcto entendimiento de las necesidades o problemas y una mayor confianza entre las partes. Ahí es donde los escenarios podrían colaborar para mitigar la insuficiencia tecnológica.

VII. Mitigación de problemas de gestión del conocimiento:

En la mayoría de los proyectos existen los especialistas que conocen ampliamente el dominio del problema y centralizan dicho conocimiento transformándose en cuellos de botella, lo que afecta al proceso de desarrollo (Richardson, y otros, 2012) y a la calidad de los requisitos. En estos casos, el conocimiento de los procesos actuales o futuros existe, pero no se encuentra disponible a demanda para los integrantes de los equipos que lo requieran. Asimismo, el material que muchas veces se encuentra disponible es parcial o carece de una visión integradora que pueda proporcionar a los involucrados una representación adecuada del sistema o del proceso requerido. Sin el conocimiento sobre este dominio, es muy complejo construir especificaciones de requisitos de calidad, lo cual queda evidenciado principalmente con la métrica DFR01 “Rechazo de especificación de requisitos” en el proyecto de Venezuela.

En este sentido, el LEL y los escenarios son capaces de mitigar esta dificultad al proporcionar una descripción global del contexto de aplicación, y modelar los flujos y procesos del sistema (Leite, y otros, 2004).

Como valor adicional, se puede tener una disponibilidad irrestricta de los modelos, que permita no solo que la propagación del conocimiento se vea potenciada entre las diferentes locaciones, sino además que ésta sea más simple y rápida, reduciendo la necesidad de viajes frecuentes y maximizando el aprovechamiento del solapamiento limitado de los horarios laborales.

VIII. Mitigación de la ambigüedad, contradicciones y/o falta de claridad en las especificaciones:

El LEL posee dos ventajas principales: por un lado, reduce la ambigüedad y, por otro, permite a los analistas y a todos los involucrados entender la jerga de los usuarios (Leite, y

otros, 2004).

Los escenarios al estar especificados en lenguaje natural, usando la jerga definida en el LEL, colaboran con la validación por parte de los usuarios (Leite, y otros, 2000), evitando incorporar inconscientemente errores, ambigüedades y contradicciones a las especificaciones, y mejorando los tiempos planeados. La cantidad de rechazos en el proyecto Venezuela, documentados en la métrica DFR01, los tiempos obtenidos en la creación de las especificaciones, o la cantidad de preguntas según la métrica TRACPDF01 evidencian la falta de claridad, los errores y las contradicciones en las definiciones, las cuales se hicieron más evidentes con las no conformidades reportadas por los usuarios y recopiladas bajo la métrica CERTNCFF01.

El entendimiento de las funcionalidades fundamentales del sistema es primordial, y más aún bajo el modelo de desarrollo global de software donde las limitaciones de comunicación son inevitables.

En este sentido, se puede lograr una representación de estas funcionalidades principales mediante un conjunto de escenarios primarios (Leite, y otros, 2000); según Booch (1994) una docena de escenarios primarios puede ser suficiente para esta representación. Con lo cual las especificaciones pueden ser contextualizadas de forma tal de tener claridad respecto al contexto y las necesidades de los usuarios y clientes.

IX. Mitigación de la amenaza de tener objetivos individuales:

En el proyecto Venezuela no fue posible evidenciar carencias de este tipo entre los involucrados. Sin embargo, es un punto que merece especial atención dado que podría afectar considerablemente el rendimiento de los equipos y puede ser muy difícil de identificar. Por otro lado, es una muy buena práctica presentar claramente los objetivos del proyecto a fin de lograr el compromiso de los involucrados.

El primer paso para lograr tener el mismo objetivo, anteponiéndolo a intereses individuales, es poder proporcionar una clara visión de cuál es el estado actual y cuál es el logro que se quiere alcanzar, proporcionando además los medios para un correcto entendimiento del contexto y así evitar problemas de interpretación.

Muchas veces la distancia geográfica o la distancia temporal generan aislamientos por parte de los equipos que terminan por trabajar desconectados, sin entender ni comprometerse con el mismo objetivo (Fryer, y Gothe 2008), más aún cuando pertenecen a diferentes compañías y/o no existe un rol de gestión común para todos los equipos. Elaborando y/o diseñando pequeñas porciones del sistema que le son solicitadas, sin entender cómo se relacionan o cual es el comportamiento integral del sistema que se está construyendo, reduce el compromiso con el proyecto y hace propensa la aparición de los intereses personales por sobre el objetivo común. Obviamente, esto atenta contra la colaboración y contra la unificación de un mismo objetivo por parte de los diferentes equipos distribuidos.

Los escenarios futuros suministran un objetivo claro al cual apuntar. A su vez, cuando estos están correctamente creados, los objetivos del sistema software quedan mejor definidos (Doorn, Hadad & Kaplan, 2002) y, por lo tanto, son más simples de comprender. Esto implica que cada miembro conozca y comprenda claramente los objetivos del software a desarrollar y esto promueve el logro de un objetivo común.

X. Mitigación del problema de la rotación del equipo:

La rotación de los equipos no fue significativa en el proyecto Venezuela, de cualquier modo, la pérdida de recursos experimentados es algo común y de lo cual ningún proyecto está exento. Los proyectos se ven afectados en mayor o menor medida, dependiendo de los factores sociales, culturales y organizacionales, entre otros.

No solo se pierde el conocimiento que reside en la persona que se retira, sino que es necesario un esfuerzo adicional por parte del equipo remanente para la capacitación del nuevo recurso. En este sentido, el LEL y los escenarios son de suma utilidad, ayudando a reducir la duración de la curva de aprendizaje, y sacando el mayor provecho de las reuniones con los recursos experimentados.

De hecho, una de las principales ventajas del LEL es la de poder ser utilizado para capacitar a nuevos miembros del equipo en la terminología empleada (Hadad, y otros, 1999).

Los escenarios actuales y futuros por su parte proporcionan una ventaja doble; por un lado, proporcionan una descripción del estado actual y el estado futuro al que se quiere llegar y.

por otro, al estar contruidos en lenguaje natural, facilitan acercar al nuevo recurso dicho conocimiento reduciendo sus tiempos de aprendizaje.

4.3 Introducción de mejoras al proceso de software en el proyecto Brasil

Para el proyecto Brasil se siguió el mismo proceso de software en cascada del proyecto Venezuela (Ver subsección 3.1), aunque se adicionaron actividades a algunos de los subprocesos, con el fin de mitigar las amenazas ocurridas anteriormente. En el Anexo B, se describe en detalle el proceso de software completo llevado a cabo en el proyecto Brasil. En dicho Anexo se han incluido y marcado las actividades de requisitos incorporadas, junto con nuevas actividades de gestión asociadas.

En resumen, las modificaciones al proceso de desarrollo del proyecto de Brasil, básicamente fueron el uso de los modelos LEL, Escenarios Actuales y Escenarios Futuros. La creación y gestión de estos modelos requirió actividades adicionales en varias fases del proceso. Además, se incorporó un repositorio para compartir estos modelos.

En la actividad de preparación de infraestructura y herramientas dentro del subproceso de Conceptualización se incluyó la subactividad de selección, instalación y configuración de herramientas de gestión de conocimiento. En la misma actividad se incorporó la subactividad de inducción de los analistas a la creación del LEL, Escenarios Actuales y Escenarios Futuros. La actividad de definición de alcance y estrategia de elicitación dentro de Conceptualización se rediseño agregándole varias subactividades, desarrollándose de la siguiente manera:

- ✓ *Creación del Léxico Extendido del lenguaje.*
- ✓ *Validación del LEL mediante sesiones.*
- ✓ *Creación de los Escenarios Actuales.*
- ✓ *Validación de los Escenarios Actuales mediante sesiones.*
- ✓ *Almacenamiento del LEL y los Escenarios Actuales en el repositorio de gestión del conocimiento.*

- ✓ *Aprobación formal del LEL.*
- ✓ *Aprobación formal de los Escenarios Actuales.*
- ✓ *Elicitación y análisis de los requerimientos de alto nivel generados por el negocio.*
- ✓ *Creación del registro de requerimientos del negocio.*
- ✓ *Creación de la matriz de rastreabilidad.*
- ✓ *Validación de los requerimientos del negocio mediante sesiones.*
- ✓ *Aprobación formal de los requerimientos del negocio.*
- ✓ *Creación de los Escenarios Futuros.*
- ✓ *Validación de los Escenarios Futuros mediante sesiones.*
- ✓ *Almacenamiento de los Escenarios Futuros en el repositorio de gestión del conocimiento.*
- ✓ *Aprobación formal de los Escenarios Futuros.*

Alineado con lo que se propone en (Leite, y otros, 2004), el primer paso fue la creación del Léxico Extendido del Lenguaje, para posteriormente crear los Escenarios Actuales y luego los Escenarios Futuros. La Figura 21 presenta un ejemplo de dos símbolos del LEL creados para el proyecto Brasil y la Figura 22 muestra un ejemplo de escenario futuro creado en el mismo proyecto. Tanto los escenarios actuales y futuros, como las especificaciones de requisito fueron descriptas utilizando la terminología definida en el LEL. Todos los modelos fueron redactados en inglés, debido a la diversidad de idiomas de los equipos de trabajo y se escribieron usando una terminología unificada al disponer de definiciones precisas a través del modelo LEL. Es por ello, que los términos subrayados en las Figura 21 y Figura 22 indican hipervínculos a las definiciones de los correspondientes símbolos del LEL. En la Figura 22, el tratamiento de las excepciones menciona el título de otros escenarios, donde se describe la situación por excepción (ver dichos títulos de escenario en mayúsculas en la figura).

<i>Symbol</i>	<i>Policy / Contract</i>	<i>Type: Object</i>
Notion	<ul style="list-style-type: none"> • A written document containing the terms of the agreement between the insurer and the insured. • Instrument evidencing the agreement between the insured and the insurer. 	
Behavioral Response	<ul style="list-style-type: none"> • It defines the rules in general, particular or special ways that regulate the contractual relationship between the insurer and the insured. • It has a unique policy number. • It has an insured and policyholder and can be these two different entities. • The insured receives a printout after contracting the insurance. • It may be modified, terminated or extended. • Premium could not be collected by Insurance Company directly. • All premiums are collected through agents. • The payment of the policy premium could be done by check, deposit, or bank transfer. 	
<i>Symbol</i>	<i>Insured</i>	<i>Type: Subject</i>
Notion	<ul style="list-style-type: none"> • Person, owner of the interest on which insurance risk is taken. • The person or organization that is covered by the insurance policy. • He may or may not be the beneficiary of the policy. 	
Behavioral Response	<ul style="list-style-type: none"> • He hires the policy with the insurer. • He can negotiate, terminate or extend the policy. 	

Figura 21 - Ejemplos de símbolos del LEL del proyecto Brasil

En el proyecto Brasil, se definieron 46 símbolos del LEL, mientras que se crearon 26 escenarios actuales, 97 escenarios futuros y 154 especificaciones de requisito. La cantidad de páginas por especificación estuvo en proporciones afines al proyecto Venezuela, con una composición similar en cuanto a la complejidad de las mismas.

Title	Pre Processing Incoming Payments	
Goal	The Interface should process the payments associated to policies collected by other entities in name of the Insurance Company and registering the accounting entries into Intermediate Data Base.	
Context	Temporal Location	<ul style="list-style-type: none"> • Every day. • Brazil Time should be used for Interface schedule execution time
	Geographical Location	<ul style="list-style-type: none"> • Brazil
	Preconditions	<ul style="list-style-type: none"> • Existence of the Intermediate Data Base. • Access (user & Password) to the location where the file is stored in order to allow interface to log and read the file. • Access (user & Password) to the Intermediate Insurance Company Data Base to allow interface to record the registers. • Interface execution scheduled in Execution tool.
Actors	Bank entity (File), Insurance Company (Intermediate data base, processing interface)	
Resources	File with the collection, Intermediate Data Base, processing interface	
Episodes	1. The bank lets the file available for interface reading.	
	2. The interface invokes with the product parameter.	
	3. The interface invokes bank entity logging service.	
	4. The interface reads full file from bank entity.	
	5. The interface filters the rows by motor or other products specified.	
	6. The interface inserts data into the Intermediate Insurance Company Data Base taking into consideration the unique policy number .	
	7. The interface confirms the processing of the data and deliver status log.	
	8. The interface logoffs from the service.	
Exceptions	1. Error trying to access File (log error and INFORM EXECUTION PROBLEM).	
	2. Security error trying to access Intermediate Insurance Company Data Base (log error and INFORM SECURITY PROBLEM).	
	3. Error invoking centralized logging service (log error and INFORM EXECUTION PROBLEM).	
	4. Error invoking centralized error handling service (log error and INFORM EXECUTION PROBLEM).	

Figura 22 - Ejemplo de un Escenario Futuro del proyecto Brasil

Durante el subproceso de *Iniciación*, se llevaron a cabo sesiones en cada una de las locaciones a fin de presentar el LEL, los Escenarios Actuales y los Escenarios Futuros a los

miembros de los diferentes equipos participantes, básicamente a los desarrolladores y a los que se sumaron posteriormente al equipo de analistas. Se presentaron detalladamente el conjunto de Escenarios Actuales principales, los cuales proporcionaron una visión de alto nivel de las características esenciales del negocio para luego profundizar en el resto de ellos a fin de tener una correcta visión de la interacción de las diferentes áreas, como por ejemplo, siniestros y administración de pólizas, y transitar hacia los Escenarios Futuros que proporcionaron una visión del objetivo del sistema y sus subobjetivos. Durante estas sesiones se proporcionó toda la información necesaria relacionada al LEL, a Escenarios Actuales y Futuros, tal como la ubicación dentro del repositorio de información y sus accesos, entre otra información.

En el subproceso de Análisis y Diseño, se realizó la creación validación y aprobación formal de las especificaciones de requisitos, aunque éstas fueron creadas teniendo en cuenta el vocabulario definido en el LEL y basados en los Escenarios Futuros creados. Estas especificaciones también se almacenaron en el repositorio de gestión del conocimiento.

Durante los siguientes subprocesos, se agregó una actividad de evolución del LEL, Escenarios Actuales y Escenarios Futuros, alineada con cambios en los requisitos del negocio o cambios correctivos debido a una mejor comprensión del contexto de aplicación.

En el subproceso de Implementación, el LEL, los Escenarios Actuales y los Escenarios Futuros fueron puestos a disposición del equipo de soporte, junto con las especificaciones de requisitos.

4.4 Resultados obtenidos al implementar los modelos LEL y Escenarios en el proyecto Brasil

La cantidad de especificaciones de requisitos creadas para el proyecto Brasil fue de 154, que al igual que para el proyecto Venezuela, se clasificaron en alta, media y baja complejidad, con la distribución reflejada en la Figura 233. La cantidad promedio de páginas de las especificaciones fue de 53, siendo en promedio 87 páginas para las de alta complejidad, 54 páginas para las de media complejidad y 19 páginas para las de baja complejidad.

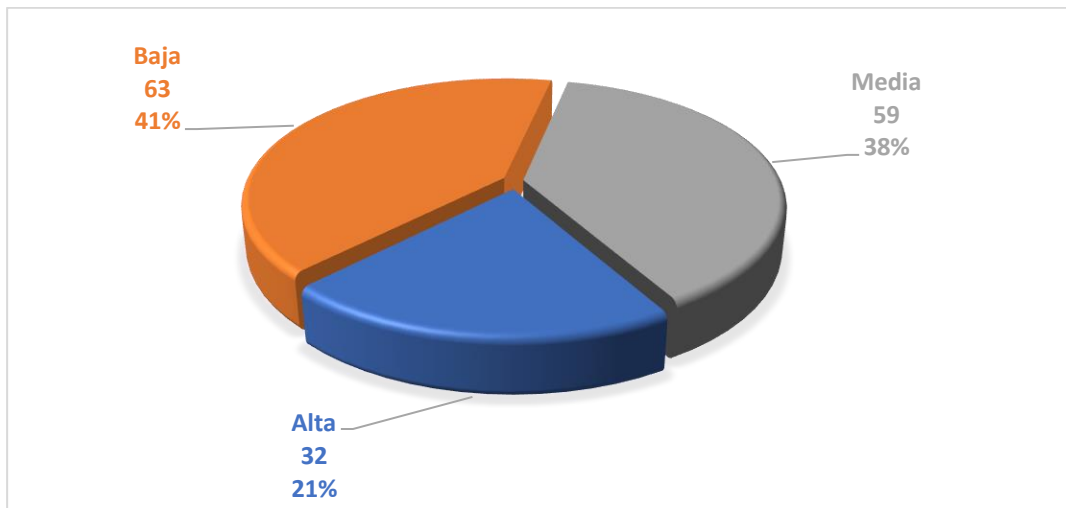


Figura 23 - Cantidad de Especificaciones de requisitos según su complejidad para el proyecto Brasil

Se utilizaron las mismas métricas y modelos de estimación que los definidos para el proyecto Venezuela, tal cual lo requiere el modelo de procesos de la compañía, por lo que para ambos proyectos se usaron los mismos tiempos estimados. A continuación se detallan las métricas obtenidas para el proyecto Brasil.

4.5 Tiempo de creación de especificaciones de requisitos para el proyecto Brasil

Posterior a la recolección de los valores de la métrica DFE01 “Entrega de especificación de requisitos”, se pudo obtener el tiempo real promedio para la creación de las especificaciones de requisitos en el proyecto Brasil, el cual se puede observar en la Figura 24, donde se distinguen los tiempos para los analistas con y sin conocimiento del contexto del problema. En esta Figura se observan mejoras en los tiempos de creación de especificaciones respecto a los tiempos estimados, donde se notan menores desviaciones.

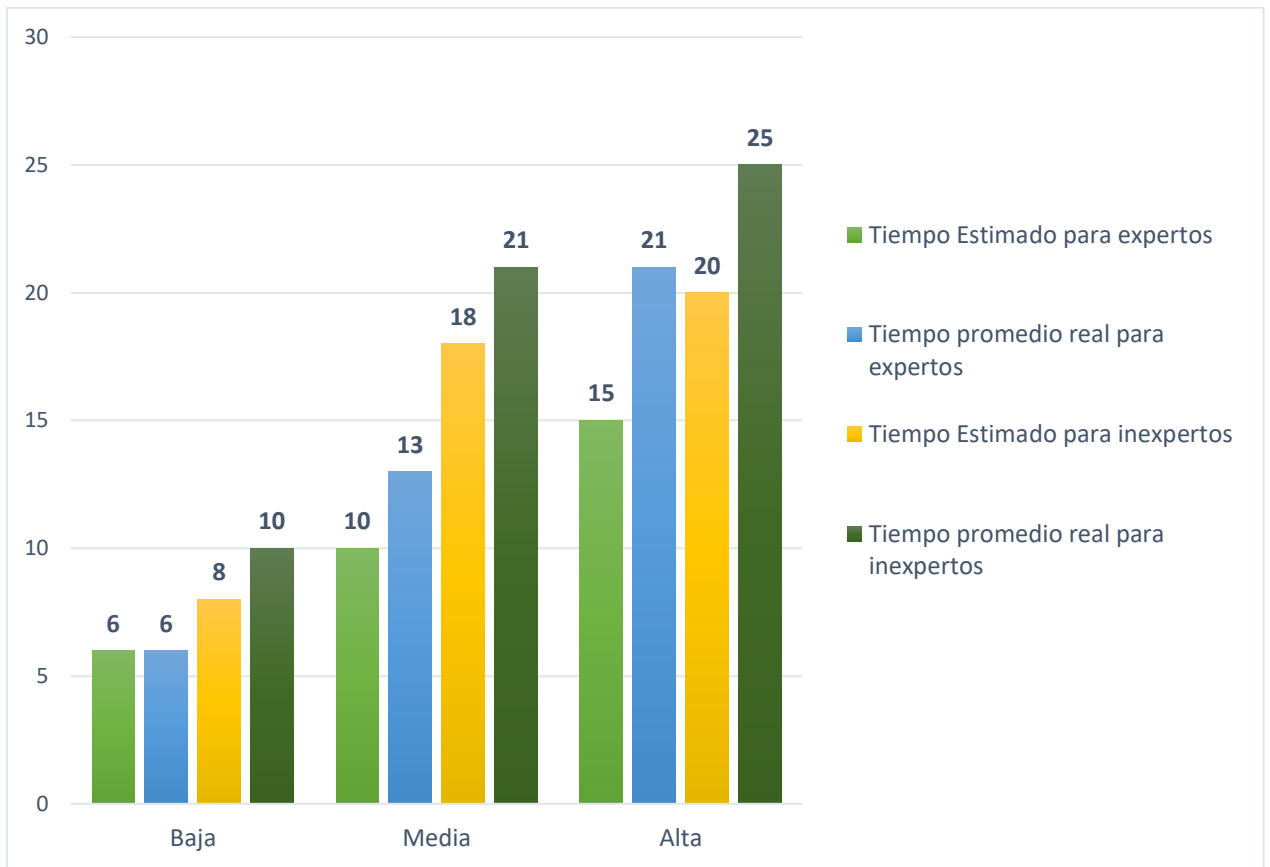


Figura 24 - Tiempo estimado y tiempo real para crear especificaciones para el proyecto Brasil (días)

Si consideramos los valores de la Tabla 12 (proyecto Venezuela) y de la Tabla 22 (proyecto Brasil), se observan mejoras, para los analistas de Brasil, en los tiempos de creación de especificaciones de requisitos respecto a los tiempos estimados. Estas comparaciones se detallan en la subsección 4.9. Tener en consideración que se utilizaron los mismos tiempos estimados para ambos proyectos.

Tabla 22 - Variación de tiempo real respecto al estimado para crear especificaciones para el proyecto Brasil (días)

Complejidad	Tiempo Estimado		Tiempo Promedio Real		Variación de tiempos	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	6	8	6	10	0 %	25 %
Media	10	18	13	21	18 %	17 %
Alta	15	20	21	25	40 %	25 %
Variación promedio					22%	22,33%

4.6 Aprobaciones y rechazos de especificaciones de requisitos para el proyecto Brasil

Las métricas utilizadas para la recolección de información relacionada con la actividad de aprobación de especificaciones fueron la DFA01 “Aprobación de especificaciones de requisitos” (ver Tabla 23) y la DFR01 “Rechazos de las especificaciones de requisitos” (ver Tabla 24).

En la Figura 25 se observan los tiempos estimados y los tiempos promedio reales para la aprobación de especificaciones para el proyecto de Brasil.

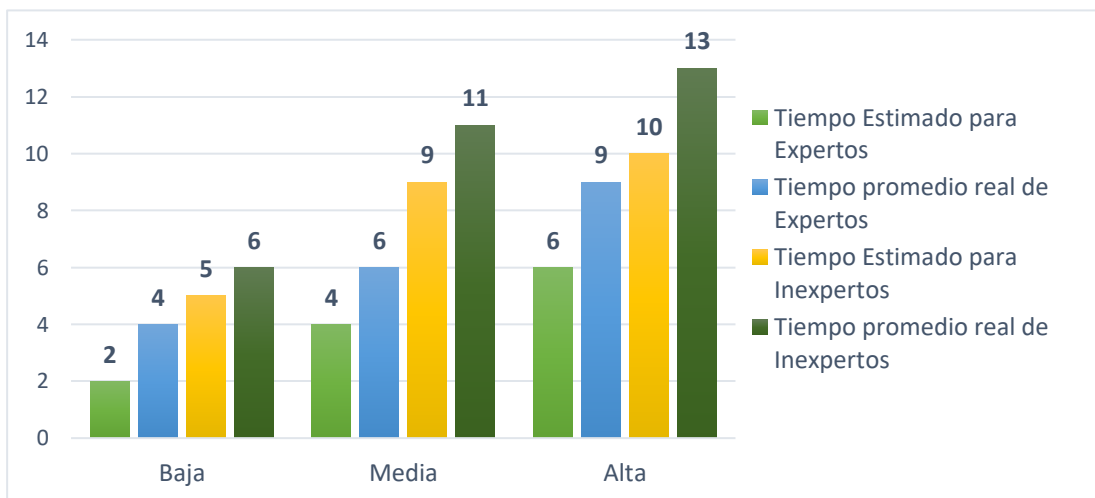


Figura 25 - Tiempo estimado y tiempo real para la aprobación de especificaciones para el proyecto Brasil (días)

En la Tabla 23 respecto a la Tabla 13 (Proyecto Venezuela), se observa una baja importante en los tiempos de aprobación de las especificaciones respecto a los estimados para analistas inexpertos. Por otro lado, cabe notar que hubo una duplicación de tiempo resultante en la aprobación de especificaciones simples creadas por analistas expertos, en cuanto a ello, solo se ha podido concluir que el valor estimado de 2 días era excesivamente optimista.

Tabla 23 - Variación de tiempo real respecto al estimado para aprobar especificaciones para el proyecto Brasil (días)

Complejidad	Tiempo Estimado		Tiempo Real		Variación	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	2	5	4	6	100 %	20 %
Media	4	9	6	11	50 %	22 %
Alta	6	10	9	13	50 %	30 %
Variación promedio					66,66%	24%

Al igual que en el proyecto Venezuela se documentaron los rechazos sobre las especificaciones de requisitos al realizar la actividad de aprobación formal, bajo la métrica DFR01 “Rechazo de especificación de requisitos” (ver Tabla 24), donde se observa una disminución sustancial en el promedio de rechazos para especificaciones de media y alta complejidad creadas por analistas inexpertos frente a expertos.

Tabla 24 - Variación promedio de rechazos por especificación creada por categoría de analista para el proyecto Brasil

Complejidad	Promedio de rechazos por especificación		Variación de rechazos
	Analistas Expertos	Analistas Inexpertos	
Baja	2	5	150 %
Media	5	8	60 %
Alta	6	9	50 %
Variación Promedio			87%

La cantidad de rechazos para las especificaciones de requisitos creadas por analistas sin conocimiento del contexto tanto de Brasil como de Venezuela (ver Tabla 14), se mantiene

igual o con una leve disminución en aquellas de mediana y alta complejidad. Lo que se puede notar en el proyecto Brasil es que se acercaron los valores entre la cantidad de rechazos de analistas expertos frente a la de inexpertos, siendo en promedio que los analistas inexpertos tuvieron 87 % más rechazos que los expertos, valor muy inferior al 132% de Venezuela.

4.7 Transición de especificaciones de requisitos a la fábrica de software para el proyecto Brasil

Los tiempos reales para las transiciones a la fábrica de software, recopilados según la métrica TRATPT01 “Tiempo promedio de transición de las especificaciones de requisitos a la fábrica de software”, se observan en la Figura 26, y la Tabla 25 muestra las variaciones de tiempo entre analistas inexpertos y expertos por especificación de requisitos.

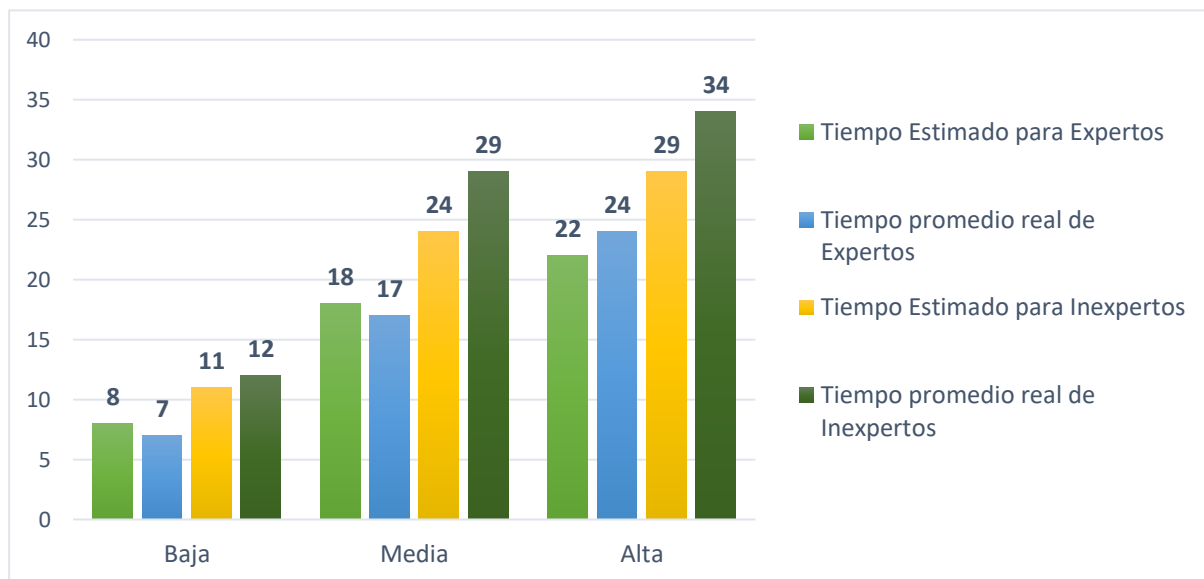


Figura 26 - Tiempo estimado y tiempo real para la transición de especificaciones a la fábrica de software para el proyecto Brasil (horas)

Tabla 25 - Variación de tiempo real respecto al estimado en la transición a la fábrica de software por categoría de analista para el proyecto Brasil (horas)

Complejidad	Tiempo estimado		Tiempo promedio real		Variación de tiempos	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	8	11	7	12	-13 %	9 %
Media	18	24	17	29	-6 %	21 %
Alta	22	29	24	34	9 %	17 %
Variación Promedio					-3,33%	15,6%

Como se puede apreciar en la Tabla 25, no hay prácticamente variaciones en los tiempos de transición respecto al proyecto Venezuela (ver Tabla 15), lo cual en un primer momento fue desconcertante para el equipo de proyecto, pero al tener en cuenta el poco tiempo de interacción sincrónico entre analistas y desarrolladores debido a las escasas horas de solapamiento, es más comprensible que así sea. Sin embargo, es de notar (ver Tabla 25) que los analistas con experiencia en el dominio de la aplicación lograron hasta reducir los tiempos estimados de transición para especificaciones de baja y media complejidad en el proyecto Brasil.

En el proyecto Brasil del equipo de desarrollo compuesto por 47 recursos, solo 15 contaban con experiencia en el dominio, por lo que se evidenciaron tiempos de transición dispares entre estos dos tipos de recursos, similar a lo que ocurrió en el proyecto de Venezuela, ver Figura 27.

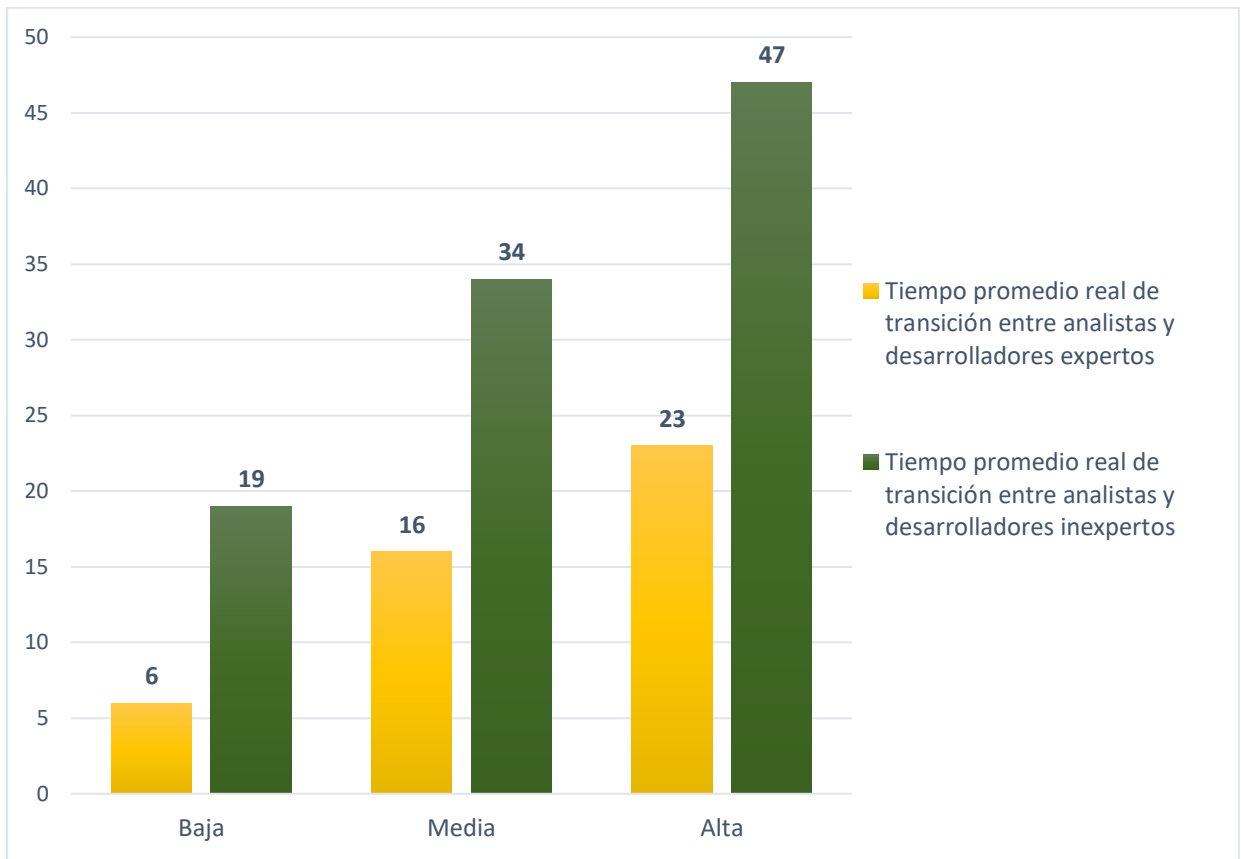


Figura 27 - Tiempo promedio real de transición entre analistas y desarrolladores considerando pares experto-experto e inexperto-inexperto para el proyecto de Brasil (horas)

La Tabla 26 muestra la variación de tiempos de transición entre los analistas y desarrolladores que compartían un grado similar de experiencia en el contexto respecto a aquellos analistas y desarrolladores que no tenían esta experiencia.

Tabla 26 – Variación de los tiempos de transición entre analistas y desarrolladores expertos e inexpertos para el proyecto de Brasil (horas)

Complejidad	Tiempo real de transición (horas)		Variación entre Inexperto – Inexperto vs Experto - Experto
	Entre analistas y desarrolladores expertos	Entre analistas y desarrolladores inexpertos	
Baja	6	19	216.7 %
Media	16	34	112.5 %
Alta	15	47	104.3 %
		Variación Promedio	144,5 %

Si bien los tiempos de transición no variaron significativamente respecto al anterior proyecto, sí existió una menor cantidad de dudas a nivel general como se puede contemplar en Tabla 27 respecto a la Tabla 17.

La cantidad de consultas que realizaron los desarrolladores a los analistas durante la transición y desarrollo se obtuvieron según la métrica TRACPDF01 “Cantidad promedio de preguntas por especificación de requisitos”, y se reflejan en la Tabla 27, teniendo en consideración si eran desarrolladores con y sin experiencia en el contexto.

Se redujo notoriamente la cantidad de preguntas de los desarrolladores respecto al proyecto Venezuela (comparar con Tabla 17), siendo que en el proyecto Brasil los desarrolladores de India hicieron en promedio 8,5 preguntas por especificación de requisito.

Tabla 27 - Comparación del promedio de preguntas realizadas durante la transición y desarrollo según el conocimiento en el dominio de los desarrolladores para el proyecto Brasil

		Cantidad de preguntas promedio de desarrolladores por especificación		
Complejidad	Desarrolladores Expertos	Desarrolladores Inexpertos	Variación de inexpertos respecto a expertos	
Baja	5	5	0,0 %	
Media	9	9	0,0 %	
Alta	9	14	55,6 %	
		Variación Promedio	18,5 %	

4.8 No conformidades en la certificación de usuarios para el proyecto Brasil

Durante la etapa de certificación del proyecto Brasil, al igual que en el de Venezuela, se identificaron las no conformidades de parte de los usuarios. La métrica utilizada fue la CERTNCFF01 “Cantidad de no conformidades por errores, cambios o ambigüedades en las especificaciones de requisitos”, la cantidad de no conformidades detectadas fue de 82, lo cual representa el 32 % de la cantidad total de defectos, que fue de 254. En la Figura 28 puede verse la distribución total de defectos con sus cantidades y porcentajes.

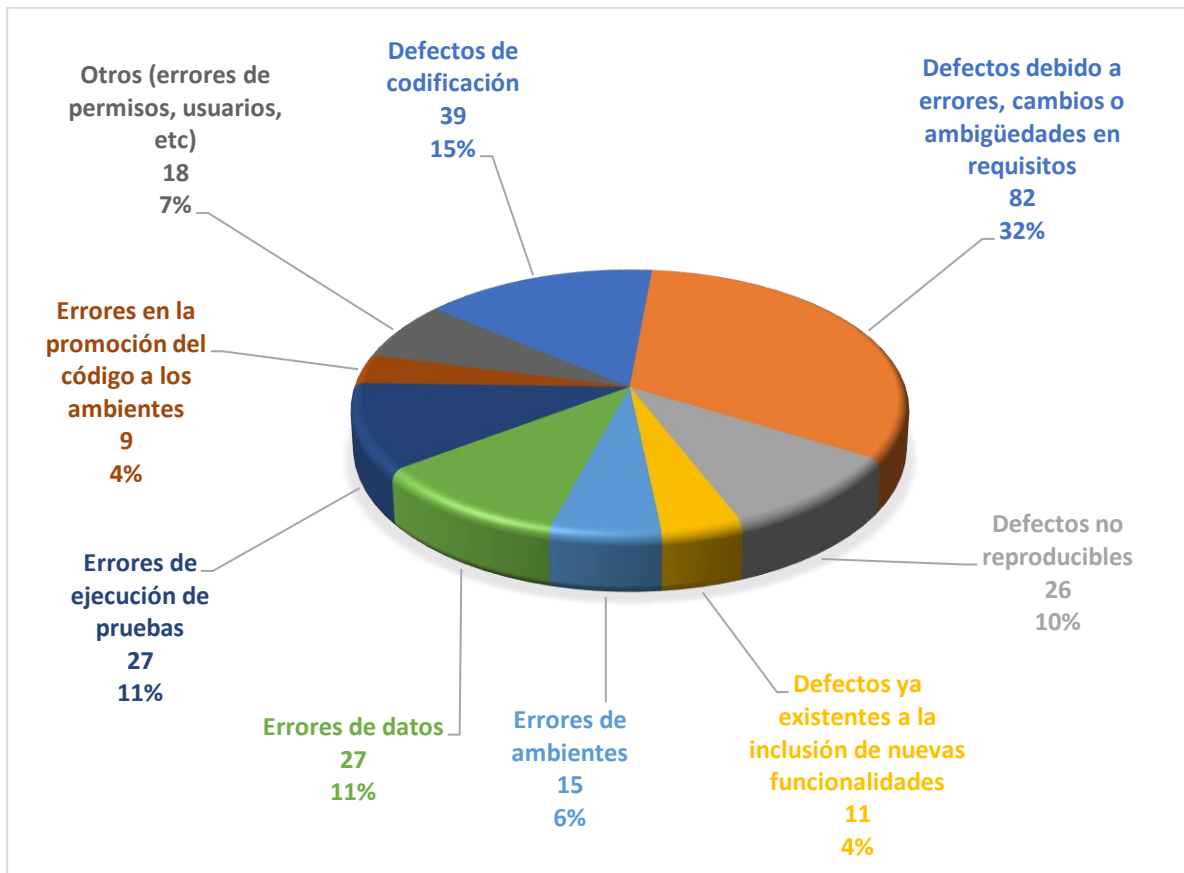


Figura 28 - Defectos encontrados en la certificación en el proyecto Brasil

Como se observa en la Figura 28, el 32 % de los defectos reportados por los usuarios en esta etapa están asociados a la mala calidad de las especificaciones de requisitos, ya sea porque existían errores, contradicciones, ambigüedades o eran necesarios cambios para reflejar el correcto comportamiento requerido por los usuarios. Como puede apreciarse, los defectos en los requisitos sigue siendo el rubro con mayor cantidad de defectos identificados. Si bien la cantidad de no conformidades reportadas sobre las especificaciones es menor que para el proyecto Venezuela (ver Figura 20), se debe tener presente que en el proyecto Brasil existe una cantidad mucho mayor de especificaciones de requisitos (104 especificaciones en el proyecto de Venezuela frente a 154 en Brasil).

La métrica CERTINCPER01, “Índice de no conformidades por errores, cambios o ambigüedades por especificaciones de requisitos”, definida en la formula (8) permite realizar una comparación sumamente valiosa.

Cálculo del índice de defecto para el proyecto Brasil:

$$\text{Índice de defecto Brasil} = \frac{82}{154} = 0,53 \quad (8)$$

Este índice expone una tasa de 0,53 defectos por cada especificación de requisitos, aproximadamente la mitad que para el proyecto Venezuela aún cuando en el proyecto Brasil se realizaron 48% más de especificaciones de requisitos.

Como se puede apreciar en la Tabla 28, de las 154 especificaciones de requisitos, se debieron corregir 69 de ellas (45 % del total), siendo la mayor cantidad de no conformidades correspondientes a especificaciones creadas por analistas sin conocimiento en el dominio del problema. Las no conformidades encontradas en especificaciones creadas por analistas inexpertos fue casi un 93 % superior a las detectadas en especificaciones creadas por analistas con experiencia en el dominio. Respecto al proyecto Venezuela (ver Tabla 18), hubo una baja importante en esta proporción.

Tabla 28 - Comparativa de cantidad de especificaciones de requisitos afectadas por defectos reportados en la certificación en el proyecto Brasil

Cantidad total de especificaciones con defectos	69
Cantidad de defectos detectados en especificaciones creadas por analistas expertos	28
Cantidad de defectos detectados en especificaciones creadas por analistas inexpertos	54
Variación en la cantidad de defectos encontrados entre especificaciones creadas por expertos frente a inexpertos	92,9 %

4.9 Análisis comparativo de métricas entre ambos proyectos

A continuación se hará un estudio comparativo de los resultados de las métricas obtenidas para cada proyecto, con el fin de poder establecer el impacto real que tuvieron la

introducción de los modelos Léxico Extendido del Lenguaje, Escenarios Actuales y Escenarios Futuros en el proyecto Brasil, en función de las amenazas a requisitos identificadas en el proyecto Venezuela.

En principio pondremos en contexto esta comparación teniendo en consideración la composición similar de recursos humanos, en cuanto a su conocimiento o no del dominio de seguros, siendo un alto porcentaje de recursos sin conocimientos de dicho dominio para ambos proyectos. El proyecto de Brasil tuvo un 50% de personal que el proyecto de Venezuela, debido principalmente al número de usuarios de ambos países: 20 usuarios en Venezuela, frente a 60 en Brasil, resultando esto en un trabajo más arduo en la elicitación, validación y posteriormente en la certificación del software por los usuarios.

4.10 Comparación de tiempos de creación de especificaciones de requisitos

En la Figura 29 puede observarse la comparativa para ambos proyectos del promedio de tiempos reales incurridos por los analistas con conocimiento del contexto y aquellos que no lo tenían en la creación de especificaciones de requisitos. En ella puede observarse la reducción significativa de tiempos de creación de especificaciones para el caso de analistas sin experiencia en el dominio para el proyecto Brasil respecto al de Venezuela, mientras que se mantuvieron los tiempos de creación para los analistas expertos.

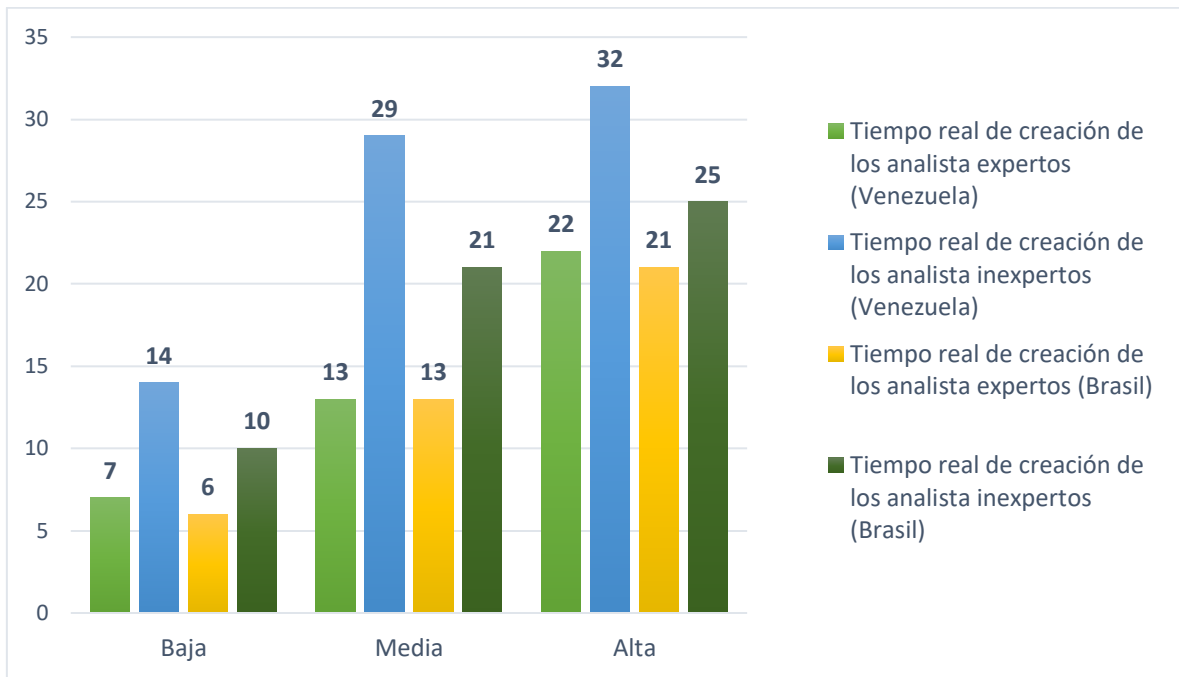


Figura 29 - Tiempos reales de creación de especificaciones para ambos proyectos (días)

Como se observa con más detalle en la Tabla 29, tanto los analistas con y sin experiencia en el dominio redujeron sus tiempos de creación de especificaciones de requisitos en el proyecto Brasil, independiente de la complejidad de la especificación. En esta **Error! No se encuentra el origen de la referencia.**, se muestra cómo los analistas en el proyecto Brasil con conocimiento del contexto, lograron reducir en aproximadamente un 6% los tiempos respecto al proyecto Venezuela, mientras que los analistas sin conocimiento del contexto redujeron aún más sus tiempos de creación de especificaciones, en promedio redujeron un 26 % los tiempos de creación.

Tabla 29 - Variación de tiempos de creación de especificaciones entre ambos proyectos

Tiempos en días	Tiempo real de creación para Venezuela		Tiempo real de creación para Brasil		Variación de tiempos de Brasil respecto a Venezuela	
Complejidad	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos
Baja	7	14	6	10	-14 %	-29 %
Media	13	29	13	21	0 %	-28 %
Alta	22	32	21	25	-5 %	-22 %
Variación Promedio					-6 %	-26 %

4.11 Comparación de aprobaciones y rechazos de especificaciones de requisitos

En la Figura 30 puede verse la comparativa para ambos proyectos, entre los tiempos reales promedio incurridos para la aprobación formal de las especificaciones de requisitos creadas por los analistas con conocimiento del contexto y por aquellos que no lo tenían.

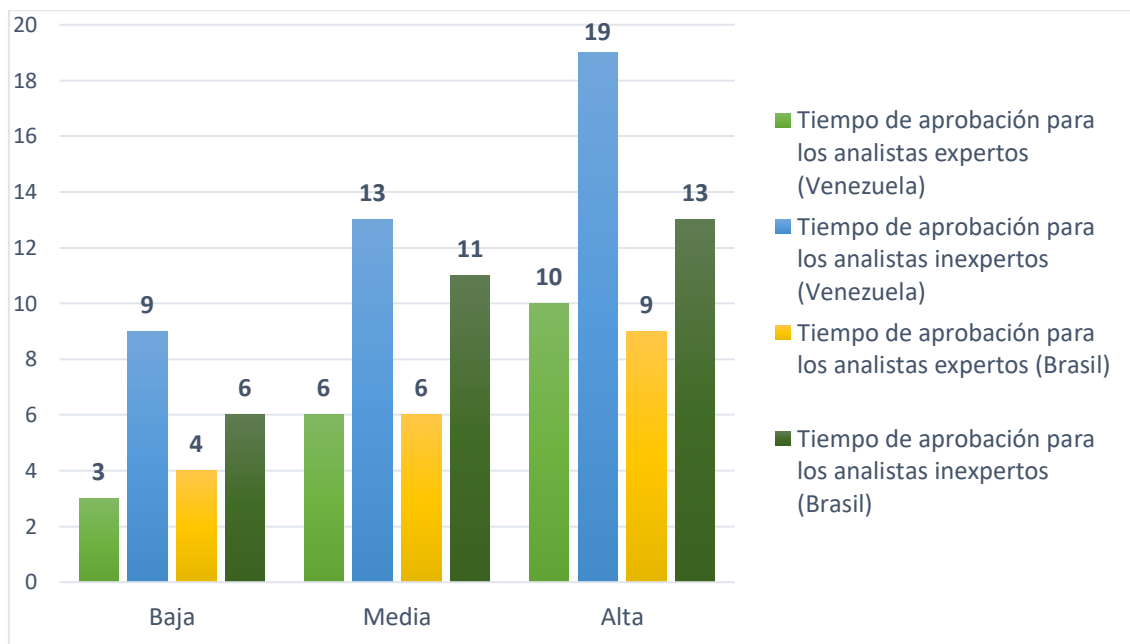


Figura 30 - Tiempos reales de aprobación de especificaciones para ambos proyectos (días)

Lo tiempos de aprobación para aquellos analistas que conocían el dominio casi no variaron

entre ambos proyectos, manteniéndose incluso cerca de los tiempos estimados. Mientras que sí se observa una reducción de tiempos para analistas sin conocimiento previo del dominio.

En la Tabla 30, pueden apreciarse más claramente los porcentajes de mejora de tiempos en el proyecto Brasil respecto del proyecto Venezuela. En el caso de las especificaciones de complejidad baja y alta creadas por analistas inexpertos, se observa una mejora en los tiempos de aprobación en el proyecto Brasil respecto al de Venezuela, de alrededor del 30% mientras que para aquellas de complejidad media se observa una leve mejora de un 15%. En general, en la aprobación formal de las especificaciones se redujeron los tiempos, excepto para las especificaciones de baja complejidad creadas por analistas expertos, que aumentó un 33%, lo que indica un incremento de 1 día el tiempo de aprobación por especificación, pudiendo considerarse poco significativo. Debe notarse que este mismo valor 33% pero de reducción de tiempo para analistas inexpertos en especificaciones de baja complejidad (en Tabla 30), implicó en promedio en valores absolutos 3 días menos para aprobación respecto a los mismos tiempos en el proyecto de Venezuela.

Tabla 30 - Variación de tiempos de aprobación de especificaciones entre ambos proyectos

Tiempos en días	Tiempo de aprobación para Venezuela		Tiempo de aprobación para Brasil		Variación de tiempos de Brasil respecto a Venezuela	
	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos
Baja	3	9	4	6	33 %	-33 %
Media	6	13	6	11	0 %	-15 %
Alta	10	19	9	13	-10 %	-32 %
Variación Promedio					8 %	-27 %

Por otro lado, se redujo la brecha en el tiempo de aprobación de las especificaciones entre los analistas expertos y los inexpertos en el proyecto Brasil.

Respecto a los rechazos de las especificaciones de requisitos al realizar la actividad de aprobación formal, la comparativa entre los dos proyectos se muestra en la Tabla 31

Tabla 31 - Variación de la cantidad de rechazos sobre las especificaciones de requisitos entre ambos proyectos

Complejidad	Promedio de rechazos para Venezuela		Promedio de rechazos para Brasil		Variación de rechazos de Brasil respecto a Venezuela	
	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos
Baja	2	5	2	5	0,0 %	0,0 %
Media	4	9	5	8	25,0 %	-11,1 %
Alta	5	11	6	9	20,0 %	-18,2 %
Variación Promedio					15,0 %	-9,8 %

La cantidad de rechazos para las especificaciones de requisitos creadas por analistas sin conocimiento del contexto tanto de Brasil como de Venezuela, se ha mantenido igual o con una leve disminución. Lo que se puede notar en el proyecto Brasil es que se acercaron los valores entre la cantidad de rechazos de analistas expertos frente a la de inexpertos. Se puede apreciar que la cantidad de rechazos no sufrió variaciones significativas en las especificaciones creadas por inexpertos, aun cuando los tiempos de aprobación se redujeron en aproximadamente un 27 % para los analistas inexpertos. Lo que se podría suponer es que esta reducción en los tiempos se pudo deber a que los rechazos provenían de defectos menores que no implicaban modificaciones significativas a las especificaciones de requisitos.

Aunque puede observarse una leve suba en rechazos de aquellas especificaciones de mediana y alta complejidad, creadas por analistas expertos en el proyecto Brasil, este incremento de rechazos corresponde en promedio a 1 defecto más por especificación. Se puede presumir que, en general, este incremento se ha debido a defectos menores. Sin embargo, este no puede afirmarse dado que la compañía de seguros no cuantificaba los defectos identificados según su severidad.

4.12 Comparación en la transición de especificaciones de requisitos a la fábrica de software

Se disminuyeron levemente los tiempos de transición a la fábrica de software en el

proyecto Brasil respecto al proyecto Venezuela (ver Tabla 32), mientras que hubo una reducción significativa en la cantidad de preguntas de los desarrolladores en India a los analistas en Argentina (ver Tabla 33). Lo primero se debe básicamente a la limitación de horas de solapamiento entre Buenos Aires y Chennai para la comunicación sincrónica. Los desarrolladores hicieron en promedio 13,2 preguntas por especificación en el proyecto Venezuela mientras que en el proyecto Brasil el promedio fue de 8,5 preguntas.

Tabla 32 - Variación de tiempos de transición a la fábrica de software entre ambos proyectos

Tiempos en horas	Tiempo de transición para Venezuela		Tiempo de transición para Brasil		Variación de tiempos de Brasil respecto a Venezuela	
	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos
Baja	8	13	7	12	-12,5 %	-7,7 %
Media	19	28	17	29	-10,5 %	3,6 %
Alta	24	35	24	34	0,0 %	-2,9 %
			Variación Promedio		-7,7 %	-2,3 %

Tabla 33 - Variación de cantidad de preguntas realizadas por los desarrolladores de ambos proyectos

Complejidad	Promedio de preguntas para Venezuela		Promedio de preguntas para Brasil		Variación de preguntas de Brasil respecto a Venezuela	
	Desarrolladores Expertos	Desarrolladores Inexpertos	Desarrolladores Expertos	Desarrolladores Inexpertos	Desarrolladores Expertos	Desarrolladores Inexpertos
Baja	7	9	5	5	-28,6 %	-44,4 %
Media	11	15	9	9	-18,2 %	-40,0 %
Alta	15	22	9	14	-40,0 %	-36,4 %
			Variación Promedio		-28,9 %	-40,3 %

Durante el subproceso de transición y desarrollo en el proyecto Brasil, se aprecia una importante reducción de las preguntas por parte de los desarrolladores inexpertos de India, del orden del 40 % en promedio respecto a lo medido para Venezuela. Por otro lado, se observa que en el proyecto Brasil se redujo la brecha entre la cantidad de preguntas realizadas por desarrolladores inexpertos frente a los expertos (Ver Tabla 33).

4.13 Comparación de no conformidades en la certificación de usuarios

Como puede apreciarse en la Tabla 34, si bien la cantidad de no conformidades reportadas sobre las especificaciones es ligeramente menor que para el proyecto Venezuela en las pruebas de certificación de usuarios, se debe considerar que en el proyecto Brasil se elaboraron una cantidad mucho mayor de especificaciones de requisitos, un 48 % más que en el proyecto Venezuela. Esto se traduce en una tasa de defectos por especificación en el proyecto Brasil de casi la mitad de la del proyecto Venezuela.

Tabla 34 - Índice de defectos por especificación de requisitos para ambos proyectos

	Venezuela	Brasil	Variación de Brasil respecto a Venezuela
Cantidad de especificaciones	104	154	48,1 %
Cantidad de defectos detectados	98	82	-16,3 %
Índice de defecto	0,94	0,53	-43,6 %

En estas pruebas de certificación de los usuarios en el proyecto Brasil, se identificó que el 44,8 % del total de especificaciones de requisitos creadas (154) tuvieron defectos, un valor mucho menor que para el proyecto Venezuela, donde el 72,1 % de las especificaciones presentaban defectos. Estos valores pueden observarse en la Tabla 35.

Tabla 35 - Comparativa de especificaciones con defectos reportados en la certificación para ambos proyectos

	Venezuela	Brasil
Cantidad de especificaciones creadas	104	154
Cantidad de especificaciones con defectos	75	69
Porcentaje de especificaciones con defectos	72,1 %	44,8 %

En ambos proyectos, la mayor cantidad de especificaciones con defectos fueron las creadas

por los analistas inexpertos. En la Tabla 36, se observa que se redujo la cantidad de defectos en especificaciones creadas por analistas inexpertos en el proyecto Brasil frente al proyecto Venezuela.

Tabla 36 - Comparativa de defectos reportados en la certificación según especificaciones creadas por categoría de analistas para ambos proyectos

	Venezuela	Brasil
Cantidad de defectos detectados en especificaciones	98	82
Cantidad de defectos detectados en especificaciones creadas por analistas expertos	27	28
Cantidad de defectos detectados en especificaciones creadas por analistas inexpertos	71	54
Variación en la cantidad de defectos detectados entre especificaciones creadas por expertos frente a inexpertos	163,0 %	92,9 %

En el proyecto Brasil, las no conformidades encontradas en especificaciones creadas por analistas inexpertos fue de 92,9 % superior a las detectadas en especificaciones creadas por expertos, mientras que en el proyecto Venezuela este valor fue del 163 %. Es decir, hubo una reducción importante en esta variación entre el proyecto Brasil respecto al de Venezuela.

4.14 Resumen comparativo de métricas

Los resultados obtenidos en ambos proyectos se han resumido en la Tabla 37.

Tabla 37 - Comparación general de mediciones entre los proyectos Venezuela y Brasil

Métrica	Venezuela	Brasil
Días promedio de creación de cada especificación de requisito	20	16
Días promedio para la aprobación de cada especificación de requisito	10	8,16
Cantidad de rechazos en la aprobación de las especificaciones de requisitos	6	5,83
Horas promedio de transición de especificaciones a la fábrica de software	21,16	20,5
Cantidad promedio de preguntas por especificación (transición a fábrica de software)	13,16	8,5
Tasa de defectos por especificación (certificación)	0,94	0,53
Proporción de defectos en requisitos respecto al total de defectos (certificación)	41 %	32 %
Proporción de especificaciones con defectos respecto al total de especificaciones (certificación)	72,1 %	44,8 %
Desviación en el costo total estimado del proyecto	53,4 %	26,2 %
Desviación en el tiempo total estimado del proyecto	47,6 %	17,9 %

Cada especificación de requisitos para el proyecto Brasil, se creó en 16 días en promedio frente a los 20 días promedio que llevó en el proyecto Venezuela. Los analistas expertos redujeron en aproximadamente un 6 % estos tiempos respecto al proyecto Venezuela, mientras que los analistas inexpertos los redujeron en un 26 % en promedio.

La reducción de tiempos en la creación de las especificaciones de requisitos se debe a múltiples factores, los que se potenciaron por el uso de los modelos LEL y Escenarios.

Los tiempos de aprobación de las especificaciones variaron levemente en términos generales pasando en promedio de 10 días en el proyecto Venezuela a 8,2 días en el proyecto Brasil. Aquellas especificaciones de requisitos creadas por analistas que tenían

experiencia en el dominio casi no variaron, mientras que sí hubo una reducción de tiempo del 27 % para los analistas sin experiencia previa.

La cantidad de rechazos en la aprobación de especificaciones no se vio afectada significativamente respecto al proyecto Venezuela, hubo tan solo una leve disminución para las creadas por analistas inexpertos. Es decir, los tiempos de aprobación se redujeron, aunque no así la cantidad de rechazos en la aprobación; esta reducción en los tiempos se pudo deber a que los rechazos provenían de cambios o errores menores.

En los tiempos promedio de transición a la fábrica de software, no hubo prácticamente variaciones entre ambos proyectos. Considerando el poco tiempo de interacción sincrónica entre analistas y desarrolladores es comprensible que así sea. Lo que sí existió es una menor cantidad de preguntas de los desarrolladores hacia los analistas. Se redujo de 13 preguntas en promedio por especificación en Venezuela a 8.5 preguntas, reduciendo en un 40 % las preguntas de desarrolladores inexpertos y en un 29 % las de desarrolladores expertos.

Durante las pruebas de certificación, la cantidad de defectos asociados a la mala calidad de las especificaciones de requisitos fue del 32 % del total de defectos frente al 41 % de Venezuela. La tasa de defectos por especificación de requisitos fue de 0,53, aproximadamente la mitad que para el proyecto Venezuela con 0,94. Fue necesario modificar el 44,8 % de las especificaciones, valor mucho menor que el 72,1 % en el proyecto Venezuela.

En el proyecto Brasil hubo una desviación respecto del presupuesto del 26,2 %, mientras que en el proyecto Venezuela esta desviación se había duplicado, siendo del 53,4 %. En relación a los tiempos estimados, hubo una desviación del 17,9 % en el proyecto Brasil, mientras que en el proyecto Venezuela esta desviación había sido del 47,6 %. En otras palabras, se observa una baja significativa en los costos y tiempos en el proyecto Brasil respecto al de Venezuela.

En base al análisis comparativo de resultados entre Venezuela y Brasil, se observan indicios de mejoras importantes sobre ciertos aspectos y en otros casos mitigaciones parciales de amenazas a requisitos identificadas. Mediante el uso del LEL, que ancla el

vocabulario del contexto manejado durante todo el proceso de software, se mejoró la calidad de la comunicación, se disminuyó la ambigüedad, contradicciones y/o falta de claridad en las especificaciones, y se mitigó parcialmente las diferencias culturales e idiomáticas. Mediante los Escenarios Actuales y Escenarios Futuros, se mejoró claramente las amenazas referidas a carencia del dominio, y a ambigüedad, contradicciones y/o falta de claridad en las especificaciones, presentando estas una tasa menor de defectos, debido principalmente a que los Escenarios Futuros habían sido previamente validados y dichas validaciones se facilitaron por estar dichos escenarios futuros escritos en el lenguaje propio de los usuarios. Al utilizarse modelos escritos en lenguaje natural con un vocabulario unificado, se motivó la confianza e involucramiento de los equipos de trabajo, dado que se mejoró la comunicación. Estos modelos LEL, Escenarios Actuales y Escenarios Futuros permitieron mitigar parcialmente la amenaza referida a distancia geográfica y temporal, ya que fueron una fuente validada para crear las especificaciones de requisitos y, además, durante la transición a fábrica permitieron evacuar dudas o tomar decisiones rápidamente en forma individual, minimizando la cantidad de preguntas. Asimismo, un repositorio compartido con estos modelos facilitó la gestión del conocimiento.

Deben remarcarse algunos puntos respecto a la validez de los resultados presentados:

Aunque ambos proyectos se desarrollaron para la misma compañía de seguros, ellos fueron ejecutados independientemente uno del otro.

Los analistas, desarrolladores y testers fueron contratados independientemente para cada proyecto en cada locación, dado que había un cierto solapamiento en el desarrollo de ambos proyectos. Las mismas actividades del proceso fueron realizadas en la misma locación, por lo tanto, los equipos en cada locación tenían la misma cultura. Obviamente los usuarios de cada proyecto era del país destino del software, con su propia cultura en ambos casos.

El mismo tipo de proceso de desarrollo se realizó en ambos proyectos, a pesar de las actividades adicionales incluidas en el proyecto de Brasil para ayudar a mitigar los amenazas a requisitos previamente identificados.

La estimación de tiempos de cada actividad, dependiendo de la complejidad de las

especificaciones de requisitos, se realizó usando puntos de función con su estimador común, independiente de las diferentes locaciones, proveedores y proyectos.

Las mismas métricas se aplicaron en ambos proyectos, y as medidas fueron obtenidas de la misma manera.

Los tiempos y costos estimados para el proyecto de Brasil no incluían las actividades de creación y validación de los modelos LEL, Escenarios Actuales y Escenarios Futuros, pues el presupuesto y planeamiento fue realizado antes de conocer las amenazas que luego afectaron al proyecto Venezuela.

5. Conclusiones

El mejoramiento de la infraestructura global, a través por ejemplo de conectividad internacional de alta velocidad, y demás ventajas tecnológicas y económicas, han hecho que las compañías opten por el desarrollo global de software (Fryer & Gothe, 2008).

Existe una amplia variedad de trabajos en la literatura que tienen como eje el desarrollo global de software y que señalan múltiples amenazas que atentan contra la calidad del proceso y del producto (Ul Haq et al., 2011), (Aranda y otro., 2008), (Richardson et al., 2012), (Damian & Zowghi, 2003), (Fryer & Gothe, 2008), (Prikladnicki y otro., 2004), (Nonyelum Ogwueleka, 2012), (Zowghi, 2007), (Niazi y otro., 2014).

Muchas de estas amenazas afectan en mayor o menor medida a los requisitos, provocando sobre ellos efectos indeseados que no siempre son identificados tempranamente, y que se propagan en cadena a las siguientes fases, impactando en los diversos artefactos del proceso software (Nahar & Student, 2013). El proceso de construcción de requisitos, de por sí ya desafiante (Nuseibeh & Easterbrooks, 2000) (Leite et al., 2004), puede ser afectado adicionalmente debido a las amenazas no mitigadas en el desarrollo global de software.

La identificación de las amenazas existentes, nuevas o potenciadas, sobre los requisitos al trabajar bajo el modelo multi-site es el primer paso hacia una estrategia proactiva de mitigación con el fin de eludir los impactos negativos sobre la calidad del producto software, los tiempos y los costos del proyecto.

La propuesta presentada, utilizando el Léxico Extendido del Lenguaje y los Escenarios (Leite et al., 2004), (Hadad, 2008), intenta aportar beneficios en la mitigación de amenazas sobre los requisitos en proyectos de desarrollo global de software.

El programa regional donde se puso a prueba la propuesta constaba de diferentes implementaciones en varios países de América Latina, siendo el primero de el proyecto Venezuela y continuando con el proyecto Brasil. Si bien inicialmente se tomó como supuesto el contar con el 100% de recursos con conocimientos en la industria de seguros, esto no resultó viable de cumplir ni para la propia empresa de seguros ni para los proveedores involucrados, por lo que se conformaron equipos mixtos, con recursos expertos e inexpertos en dicho dominio.

Durante el proyecto Venezuela, se subestimaron las amenazas propias del modelo distribuido y los resultados obtenidos no fueron los esperados, observándose un fuerte impacto en la calidad de los requisitos. El análisis de las métricas obtenidas para este proyecto permitió identificar las amenazas no consideradas y, a partir de ello, elaborar una adaptación al proceso de software respecto a las actividades de construcción de requisitos, a fin de mitigar dichas amenazas.

Las principales dificultades que sufrió el proyecto Venezuela fueron la carencia de conocimiento del dominio y de su léxico, evidenciado en los analistas inexpertos, lo que afectó significativamente a los tiempos de creación de las especificaciones de requisitos y la cantidad de rechazos a dichas especificaciones durante la actividad de aprobación.

El desconocimiento del vocabulario del contexto obstaculizó la comunicación entre los involucrados (Hadad et al., 1999). El lenguaje propio del contexto no fue correctamente interpretado y decodificado por los analistas, quienes utilizaban un lenguaje más técnico que los limitaba en lograr una correcta comprensión del problema y, por consiguiente, afectando negativamente a los tiempos de la creación de especificaciones de requisitos.

Las deficiencias en la comunicación y en el entendimiento del problema probablemente hayan colaborado además con la fragilidad del vínculo de confianza entre los usuarios y analistas, aspectos esenciales si se pretende que el proceso de elicitación sea exitoso (Nahar & Student, 2013).

El manejo de un mismo léxico hubiese contribuido a reducir ambigüedades y errores que afectaron a los requisitos y hubiese reforzado la relación de confianza entre los involucrados (Leitey otros, 2004). Estas carencias quedaron en evidencia por la cantidad de no conformidades originadas en los requisitos durante la certificación de usuarios.

Las amenazas encontradas en el proyecto Venezuela fueron en su mayoría propias del desarrollo global de software. Si se analiza el trabajo de Hanisch y Corbitt (2007) referido al proyecto Sapphire Software House, pueden apreciarse similitudes respecto a las complicaciones enfrentadas por el proyecto Venezuela. El equipo descrito en dicho trabajo se encontraba disperso en varias locaciones, Reino Unido y Nueva Zelanda, y sumado a la existencia de recursos inexpertos, dio como resultado interpretaciones

inexactas por parte de los analistas, quienes por ende introdujeron diversos defectos en las especificaciones. Es decir, se observaron consecuencias similares a las sufridas en el proyecto Venezuela. Las diferencias culturales y la distancia geográfica y temporal en dicho proyecto también afectaron la calidad de los requisitos e impactaron en las subsiguientes fases del proyecto. La consultora de software contratada para llevar adelante el proyecto Sapphire consideró que la mayor amenaza enfrentada estuvo relacionada con la comunicación, alineado con lo ya planteado en (Damian & Zowghi, 2003). Esta carencia en la comunicación, además de generar problemas de interpretación, afectó la generación del vínculo de confianza entre los involucrados, siendo una conclusión similar a la arribada en el proyecto Venezuela. Como acción de mitigación optaron por relocalizar a los recursos más experimentados en las oficinas del cliente en el Reino Unido, lo cual requirió por parte del resto del equipo más tiempo para interpretar y comprender las especificaciones de requisitos que llegaban desde Reino Unido, generando un incremento considerable en la cantidad de preguntas; consecuencia bastante similar a lo ocurrido en el proyecto Venezuela. Para mitigar este nuevo problema incorporaron nuevas herramientas para mejorar la comunicación y la gestión del conocimiento, ya que el correo electrónico era la herramienta de interacción más utilizada. Las herramientas adicionales fueron las comunicaciones telefónicas y las video-conferencias, lo que les permitió llevar adelante la tarea en forma más eficiente, aunque no se aclara específicamente en qué consistieron las mejoras. Para los proyectos de Venezuela y Brasil, las herramientas como teléfono, video conferencias, y otras, ya estaban implementadas.

En resumen, los problemas detectados en ambos proyectos, Sapphire y Venezuela, fueron similares, pero las opciones de mitigación implementadas fueron diferentes. Mientras que en el proyecto Sapphire se optó por el desplazamiento de los recursos más experimentados a fin de mejorar la comunicación e incrementar el vínculo de confianza, en el proyecto Brasil se utilizaron los modelos LEL, Escenarios Actuales y Escenarios Futuros, complementados con la incorporación de un repositorio conteniendo estos modelos. Estos modelos implementados permitieron mejorar no solo la comunicación sino también el conocimiento de los involucrados, otorgando herramientas útiles para la ejecución de sus tareas en forma remota asincrónica. Asimismo, en el proyecto Brasil se cuantificó y

comparó las mejoras obtenidas con motivo de la incorporación de estos modelos en lenguaje natural LEL y Escenarios y el repositorio para la gestión del conocimiento, respecto al proyecto Venezuela.

Existen varias similitudes entre el trabajo de esta tesis y el análisis hecho por (Aranda et al., 2008b) donde exponen que la principal problemática enfrentada en el desarrollo global de software es la carencia de una adecuada comunicación, haciendo mención a la falta de comunicación cara a cara. Adicionalmente, mencionan también como principales amenazas la diferencia horaria, las diferencias culturales y la gestión del conocimiento. Un punto que destacan especialmente se relaciona con la necesidad de tener un entendimiento común sobre el dominio; para lo cual recurren al uso de ontologías como facilitadores de la comunicación. Dicha propuesta es relativamente similar a las ventajas que proporciona el modelo LEL, aunque éste se enfoca estrictamente en el vocabulario empleado en dicho contexto, y se complementa con el modelo de Escenarios Actuales que permite una comprensión acabada del comportamiento en dicho contexto, en forma clara y precisa (Leite et al., 2004). Debe considerarse que en algunos casos los glosarios como el LEL pueden brindar un mejor soporte que una ontología dado que el LEL solo incluye los términos internalizados por los actores en el contexto de la aplicación, en lugar de definiciones generales en un dominio (Kaplan y otros, 2008)

Aranda et al. (2008b) mencionan la necesidad de seleccionar cuidadosamente el grupo de herramientas de trabajo, dependiendo de las características cognitivas de los involucrados, considerando especialmente la incapacidad del modelo de desarrollo global en permitir una comunicación totalmente sincrónica. Si bien para los proyectos analizados en esta tesis las herramientas ya estaban implementadas y a disposición de los equipos, es interesante el enfoque mencionado por Aranda et al. (2008b) sobre la afinidad de los involucrados a determinadas formas de comunicación, tales como visuales y verbales, y utilizar este análisis para definir los instrumentos más idóneos. Dicha afinidad depende de las características cognitivas de cada uno de ellos, y los beneficios de conocerlo anticipadamente a fin de mitigar las carencias y potenciar las fortalezas. Mediante un experimento controlado dichos actores han podido determinar el grado de satisfacción de grupos usando herramientas groupware afines con sus características cognitivas. Respecto a

esta propuesta, debería evaluarse cómo manejarse frente a la diversidad de características cognitivas de los involucrados en equipos de trabajo de cierto tamaño, que podría desencadenar en el uso de una gran variedad de herramientas, las que por otro lado, deben utilizarse coordinada y colaborativamente. Debe tenerse presente la dificultad que esto acarrearía en un proyecto como el de Brasil donde participaron alrededor de 130 personas, donde en la elicitación se involucraron 60 usuarios y 16 analistas. En resumen, la propuesta de Aranda et al. (2008b) se enfoca en el uso de ontologías y herramientas groupware, con la finalidad principalmente de mejorar la comunicación, el conocimiento del problema y la gestión de conocimiento.

Ambos proyectos, Venezuela y Brasil, poseían características similares desde la perspectiva técnica, pues se reutilizaron los mismos procesos, métricas, modelo de estimación y herramientas. A lo largo del proyecto Brasil, pudo evidenciarse que la incorporación del LEL, los Escenarios Actuales y los Escenarios Futuros, junto con un repositorio común conteniendo estos modelos, permitió de alguna manera reducir las amenazas identificadas previamente en el proyecto Venezuela y ya mencionadas por diversos autores en el tema.

La evidencia ha mostrado que la propuesta presentada y probada en un caso real de gran envergadura, ha provisto beneficios importantes para aquellos recursos humanos (analistas y desarrolladores) que no poseían conocimientos en el dominio del problema.

Esta propuesta está alineada con lo planteado por Zowghi (2007), quien expone la necesidad de un proceso de ingeniería de requisitos diferente para los proyectos inmersos en el desarrollo global de software.

El análisis de los resultados comparativos de los proyectos de Venezuela y de Brasil han proporcionado los siguientes indicios:

- ✓ *Mejoras en la calidad de la comunicación, mediante el uso del Léxico Extendido del Lenguaje, que ancló el vocabulario del contexto del problema manejado durante todo el proceso de software. Los escenarios también proporcionaron un medio de comunicación de fácil comprensión para los involucrados. Adicionalmente, estos modelos Escenarios Actuales y Futuros estaban escritos usando la terminología del contexto de la aplicación.*

- ✓ *Mejoras en la confianza e involucramiento de los equipos de trabajo, mediante el uso de modelos escritos en lenguaje natural, como son el LEL y los Escenarios, dado que los usuarios prefieren la comunicación en la propia jerga y, por ende, se manifiestan más abiertamente cuando pueden expresarse y ser comprendidos por la otra parte.*
- ✓ *Un mejor manejo del contexto del problema, mediante los escenarios que permiten tener una acabada visión del problema, dónde estamos (Escenarios Actuales) y hacia dónde vamos (Escenarios Futuros). También conocer el vocabulario propio del contexto es el inicio del proceso cognitivo. Esto fue observado principalmente en aquellos recursos con pobre o nulo conocimiento del dominio.*
- ✓ *Una disminución de las ambigüedades, contradicciones y/o falta de claridad en las especificaciones, mediante los escenarios, ya que por estar descriptos en lenguaje natural colaboran en la validación con los usuarios, evitando incorporar defectos en las especificaciones y mejorando los tiempos planeados. Además, la creación de las especificaciones usando los términos del LEL ayuda a reducir la ambigüedad. Estas mejoras en las especificaciones, se tradujeron a su vez en un sistema software con menos defectos, evidenciado en la certificación de los usuarios con una tasa de defectos notoriamente menor.*
- ✓ *Mejoras en la gestión del conocimiento, mediante el aprovechamiento del repositorio de información compartida conteniendo las descripciones en lenguaje natural del vocabulario (LEL), del problema bajo estudio (Escenarios Actuales) y del sistema a construir (Escenarios Futuros).*
- ✓ *Mitigación parcial de los problemas de diferencia geográfica y temporal, mediante el LEL y los escenarios, que proporcionaron una visión de las relaciones entre las tareas y/o flujos del sistema, y permitieron resolver dudas o tomar decisiones rápidamente en forma individual, minimizando la cantidad de preguntas de los desarrolladores a los analistas y focalizándose las preguntas en consultas más profundas y de difícil comprensión.*

- ✓ *Mitigación parcial de los problemas de diferencias culturales e idiomáticas, mediante el LEL que unificó el vocabulario en la interacción de los involucrados, proporcionando términos de uso compartido con un significado preciso. Esto involucra la interacción oral, como la interacción mediante documentos escritos utilizando dicho vocabulario unificado.*

Como queda en evidencia la adecuación del proceso de requisitos mediante la incorporación de los modelos LEL y Escenarios para mitigar las amenazas del desarrollo global de software aporta como resultado una mejora general en la calidad de las especificaciones de requisitos, permitiendo mejorar por lo tanto los costos y los tiempos de los proyectos. En principio se puede observar, como valores netamente asociados a los requisitos, que en el proyecto Brasil se redujeron en un 9 % los rechazos causados por requisitos en la certificación del software (bajaron del 41% del total de rechazos asociados a requisitos en el proyecto Venezuela al 32% en Brasil) y se redujeron en un 27% las especificaciones con defectos asociadas a rechazos en la certificación (del 72% en Venezuela se redujo al 45% en Brasil). De igual manera se redujo casi a la mitad la tasa de defectos por especificación de requisito.

Cabe mencionar que en el marco de esta tesis, se realizó una publicación con una propuesta preliminar del trabajo, que fue presentada en el 3° Congreso Nacional de Ingeniería Informática/Sistemas de Información (Mighetti & Hadad, 2015). Posteriormente, se publicó un artículo en el congreso iberoamericano CIBSE 2016 19th Workshop on Requirements Engineering (Mighetti & Hadad, 2016) realizado en Quito, Ecuador, el cual fue seleccionado para una versión extendida para la revista CLEI Electronic Journal y la misma fue publicada en (Mighetti & Hadad, 2016b).

Un punto no analizado en el presente trabajo, y propuesto a futuro, es cómo el LEL y los Escenarios podrían colaborar en el entendimiento de la jerga y los flujos del sistema por parte del equipo de pruebas, tarea que muchas veces es encargada a un proveedor distinto al que lleva adelante el desarrollo en un entorno multi-site. Una alternativa viable para facilitar el testing es la generación de casos de prueba a partir de escenarios futuros, actividad que ya ha sido desarrollada utilizando casos de uso (Heumann, 2002), (González Palacio, 2009) y también utilizado en el LEL y escenarios (Sarmiento y otros 2014). Se

puede suponer que las mejoras logradas en el proyecto Brasil podrían ser potenciadas en las etapas de prueba.

Otro tema que puede abordarse referido a las pruebas, es estudiar cómo los escenarios futuros pueden colaborar en la selección de un pequeño grupo de flujos o funcionalidades clave del sistema para llevar a cabo certificaciones acotadas por parte de los usuarios en aquellos casos que exista un grupo reducido de ellos para afrontar las pruebas de certificación, o el tiempo disponible sea limitado.

Las características que presentaban ambos proyectos, Venezuela y Brasil, son afines a muchos otros proyectos en la modalidad de trabajo distribuido, lo que haría suponer que las mejoras alcanzadas en la calidad de los requisitos utilizando el LEL, los Escenarios y el repositorio compartido con estos modelos, pueden ser alcanzadas en proyectos similares, logrando mejoras como las presentadas en este trabajo.

Se espera, como próximo paso de la propuesta presentada, recopilar las causas de rechazos durante la aprobación de las especificaciones de requisitos, dado que hubo solo una leve mejora en la cantidad de rechazos de especificaciones creadas por inexpertos e, inversamente, un leve aumento en las especificaciones de expertos en el proyecto Brasil.

Otra posible mejora a la propuesta presentada es la creación de un LEL adicional que refleje el vocabulario utilizado en los Escenarios Futuros y en las especificaciones de requisitos, debido a que estos pueden requerir el uso de nuevos términos por cambios en los procesos del negocio o cambios tecnológicos al incorporar el software (Kaplan et al., 2013). Este léxico adicional favorecería la reducción de ambigüedades en dichos modelos, por la aparición de terminología no definida en el LEL inicial o la modificación de términos existentes pero con cambios en su definición.

6. Referencias

6.1 Referencias Bibliográficas

- Alrajeh, D., Ray, O., Russo, A., & Uchitel, S. (2007). *Extracting Requirements from Scenarios with ILP*. Berlin, Springer-Verlag, Alemania: S. Muggleton, R. Otero, A. Tamaddoni-Nezhad (Eds.), *Inductive Logic Programming, Lecture*. pp. 64-78. ISBN: 978-3-540-738.
- Antonelli, L., & Oliveros, A. (2002). *Fuentes utilizadas por desarrolladores de software en Argentina para elicitación de requerimientos*. 5th Workshop on Requirements Engineering. pp. 106-116.
- Aranda, G. N., Vizcaíno, A., Cechich, A., & Piattini, M. (2008b). *A Methodology for Reducing Geographical Dispersion Problems during Global Requirements Elicitation*. 11th Workshop on Requirements Engineering. pp.117-127.
- Aranda, G. N., Vizcaíno, A., Cechich, A., & Piattini, M. (2008). Strategies to Minimize Problems in Global Requirements Elicitation. *CLEI electronic Journal*, Vol.11, N°1.
- Basili, V. R., & Weiss, D. (1981). *Evaluation of a Software Requirements Document by Analysis of Change Data*. Los Alamitos, California, EEUU: IEEE Computer Society Press, Fifth International Conference on Software Engineering. pp. 314-323.
- Bass, M., & Paulish, D. (2004). *Global Software Development Process Research at Siemens*. Third International Workshop on Global Software Development, Edinburg, Scotland. pp. 8-11.
- Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs, New Jersey, EEUU: Prentice-Hall.
- Boland, D., & Fitzgerald, B. (2004). *Transitioning from a Co-Located to a Globally-Distributed Software Development Team : A Case Study at Analog Devices Inc*. Third International Workshop on Global Software Development, Edinburg, Scotland. pp.4-7.
- Booch, G. (1994). *Scenarios*. Report on Object Analysis and Design. Vol.1, N°3, pp. 3-6.
- Brooks, F. P. (1987). *No Silver Bullet: Essence and Accidents of Software Engineering*.

IEEE Computer Society Press, Vol.20, N°4, pp.10-19.

- Cardona Arboleda, O. D. (2001). Estimación holística del riesgo sísmico utilizando sistemas dinámicos complejos, Conceptos de Amenazas, Vulnerabilidad y Riesgo. Universitat Politècnica de Catalunya, Barcelona, España, Documento N° 14888, 2001, 322 pág.
- Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*. New Jersey, USA: Prentice Hall. ISBN:0-13-924218-X. pp. 269.
- Carroll, J. M. (2000). Five reasons for scenario-based design. *Thirty-Second Annual Hawaii International Conference on System Sciences (HICSS '99)*, Vol. 3. IEEE Computer Society, Washington, DC, USA.
- Cherry, S., & Robillard, P. N. (2004). *Communication Problems in Global Software Development: Spotlight on a New Field of Investigation*. Third International Workshop on Global Software Development, Edinburg, Scotland. pp. 48-52.
- Christel, M. G., & Kang, K. C. (1992). *Issues in Requirements Elicitation*. Software Engineering Institute, Carnegie Mellon University. Technical Report: CMU/SEI-92-TR-012.
- Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley Professional.
- Damian, D. E., & Zowghi, D. (2003). Requirements Engineering challenges in Multi-site software development organizations. *Requirements Engineering Journal*, Vol.8, N°3, pp. 149-160.
- Davis, C. J., Fuller, R. M., Chiarini Tremblay, M., & Berndt, D. J. (2006). *Communication Challenge Requirements Elicitation and the use of the repertory Grid Technique*. *Journal of Computer Information Systems*. Vol.46, N°5, pp. 78-86.
- Dharmadas, M. (2008). *Global Software Development: A Case Study of Knowledge Management Challenges and Industry Approaches*. Tesis de Maestría. Norwegian University of Science and Technology Department of Computer and Information Science. 156 pág.
- Doorn, J.H., Hadad, G.D.S., & Kaplan, G.N. (2002, Noviembre) Comprendiendo el Universo de Discurso Futuro con Escenarios. *WER '02 - V Workshop on*

- Requirements Engineering*, Universidad Politécnica de Valencia, Valencia, España, pp.117-131.
- Ebert, C., & De Neve, P. (2001). *Surviving Global Software Development*. Alcatel, Antwerpen, Belgium: IEEE, Vol.18, N°2, pp. 62-69.
- Fryer, K., & Gothe, M. (2008). *Global Software Development and Delivery. Trends and Challenges. IBM Developer Works, The Rational Edge*. Accedido 12/06/2016 en: http://www.ibm.com/developerworks/rational/library/edge/08/jan08/fryer_gothe/
- García Guzmán, J., Saldaña Ramos, J., Amescua Seco, A., & Sanz Esteban, A. (2011). *Success Factors for the management of Global Virtual Teams for Software Development*. International Journal of Human Capital and Information Technology Professionals, Vol.1, N°2, pp. 48-59
- Vol., Issue 2, pp. 48-59.
- Gereffi, G., Castillo, M., & Fernandez-Star, K. (2009, Diciembre). *The Offshore Services Industry: A New Opportunity for Latin America*. Inter-American Development Bank. Center on Globalization, Governance & Competitiveness, Duke University.
- Gil, G. D. (2002). *Herramientas para Implementar LEL y Escenarios (TILS)*. Tesis de Magíster en Ingeniería de Software, Facultad de Informática, Universidad Nacional de La Plata.
- Goguen, J. A., & Linde, C. (1993). *Techniques for Requirements Elicitation*. IEEE First International Symposium on Requirements Engineering (RE'93). IEEE Computer Society Press, Los Alamitos, CA, 1993, pp.152-164.
- González Palacio, L. (2009). Método para generar casos de prueba funcional en el desarrollo de software. *Revista Ingenierías Universidad de Medellín*. Colombia, Vol.8, N°15-1, pp. 29-36
- Hadad, G. D.S. (2008). *Uso de Escenarios en la Derivación de Software*. Doctoral Thesis, Facultad de Ciencias Exactas, Universidad Nacional de La Plata, Argentina.
- Hadad, G.D.S., Doorn, J.H., & Kaplan, G.N. (2009). *Creating Software System Context Glossaries*. In Mehdi Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology, Second Edition*, Vol. II, pp.789-794. Hershey, PA: IGI Global,

Information Science Reference.

- Hadad, G. D., Kaplan, G., Oliveros, A., & Sampaio do Prado Leite, J. C. (1999). Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requerimientos: Aplicación a un caso real. *Revista de Informática Teórica y Aplicada (RITA)*, Brasil, Vol.6, No.1, pp.77-103. ISSN 0103-4308.
- Hanisch, J., & Corbitt, B. (2004). Requirements Engineering During Global Software Development: Some Impediments to the Requirements Engineering Process-A Case Study. 13th European Conference on Information Systems 2004, p.68-80.
- Hanisch, J., & Corbitt, B. (2007). Impediments to requirements engineering during global software development. *European Journal of Information Systems*. Vol. 16, pp. 793-805.
- Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. *2007 Future of Software Engineering (FOSE '07)*, IEEE Computer Society, Washington, pp.188-198.
- Heumann, J. (2002, Septiembre). Generating Test Cases from Use Cases. *Journal of Software Testing Professionals*. EEUU.
- IEEE_29148. (2011). Systems and software engineering — Life cycle processes — Requirements engineering. Piscataway, NJ 08854, USA: IEEE.
- Jackson, D. (2009). A Direct Path to Dependable Software. *Communications of the ACM*, Vol. 52 No. 4, pp. 78-88.
- Kaindl, H. (2000). A design process based on a model combining scenarios with goals and functions. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 30 No. 5, pp. 537-551.
- Kaindl, H., Brinkkemper, S., Bubenko Jr, J. A., Farbey, B., Greenspan, S. J., Heitmeyer, C. L., y otros. (2002). *Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda*. Requirements Engineering Journal, Springer-Verlag London Ltd., Vol. 7, No. 3, pp.113-123.
- Kaplan, G.N., Hadad, G.D.S., Doorn, J.H., Leite, J.C.S.P. (2000). Inspección del Léxico Extendido del Lenguaje. WER'00 – III Workshop on Requirements Engineering, Río

- de Janeiro, Brazil, pp. 70-91.
- Kaplan, G.N., Doorn, J.H., & Gigante, N.C. (2013). Evolución semántica de glosarios en los procesos de requisitos. *XVIII Congreso Argentino de Ciencias de la Computación*. Mar del Plata, Argentina.
- Kaplan, G.N., Doorn, J.H., & Hadad, G.D.S. (2008) Una visión cognitiva de modelos de requisitos de software. *X Workshop de investigadores de Ciencias de la computación*, La Pampa, Argentina.
- Kearney, A.T. (2009). *The Shifting Geography of Offshoring*. The 2009 A.T.Kearney Global Service Location Index. 24 pág.
- Kovitz, B. L. (1998). *Practical software requirements: a manual of content and style*. Greenwich, CT: Manning Publications Co.
- Kruchten, P. (2004). Analyzing Intercultural Factors Affecting Global Software Development – A Position Paper. Vancouver, Canadá. *The International Workshop on Global Software Development, International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotland*. pp. 59-62.
- Kumar, A. S., & Kumar, A. T. (2011). Study the impact of Requirements Management Characteristics in Global Software Development Projects: An Ontology Based Approach. *International Journal of Software Engineering & Applications*, Vol. 2, No. 4, pp.107.
- Leffingwell, D., & Widrig, D. (2003). *Managing Software Requirements - A unified approach, 2nd. Edition*. Addison-Wesley Professional.
- Leite, J. C. S. P. (1989). *Application Languages: A Product of Requirements Analysis*. Rio, Brasil: Computer Science Department of PUC Rio, Brasil.
- Leite, J.C.S.P., & Franco, A.P.M. (1993). A strategy for conceptual model acquisition. *IEEE International Symposium on Requirements Engineering*. San Diego, CA: IEEE Computer Society Press, pp. 243-246.
- Leite, J.C.S.P., & Franco, A.P.M. (1990). O Uso de Hipertexto na Elicitação de Linguagens da Aplicação, IV Simpósio Brasileiro de Engenharia de Software, SBC, Brasil, pp.134-149.

- Leite, J.C.S.P., & Freeman, P. A. (1991). *Requirements Validation Through Viewpoint Resolution*. IEEE Transactions on Software Engineering, Vol. 17, N° 12, pp.1253-1269.
- Leite, J.C.S.P., Oliveros, A., Rossi, G., Balaguer, F., Hadad, G.D.S., Kaplan, G.N., Maiorana, V. (1996). *Léxico Extendido del Lenguaje y Escenarios del Sistema Nacional para la Obtención de Pasaportes*. Documento de Trabajo, Serie Formación en Investigación de Docentes Auxiliares, N° 7, Universidad de Belgrano, Buenos Aires, 44 pág.
- Leite, J.C.S.P., Doorn, J.H., Kaplan, G.N., Hadad, G.D.S., & Ridaio, M.N. (2004). Defining System Context using Scenarios. J. S. C. P. Leite, & J. H. Doorn (Eds.), *Perspectives on Software Requirements*. Norwell, MA: Kluwer Academic Press, pp. 169–199.
- Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., & Kaplan, G.N. (2000). A Scenario Construction Process. *Requirements Engineering Journal*, Vol. 5, No. 1, pp.38-61.
- Mighetti, J.P., & Hadad, G.D.S. (2015, Noviembre). *Proceso de requisitos adaptado para mitigar amenazas en el Desarrollo Global de Software*. Memorias de 3er Congreso Nacional de Ingeniería Informática / Sistemas de Información (CONAIISI 2015). ISBN: 978-987-1896-47-9.
- Mighetti, J.P., & Hadad, G.D.S. (2016, Abril). *Uso de un Léxico y Escenarios para Mitigar Amenazas a Requisitos en el Desarrollo Global de Software*. WER 2016 – 19th Workshop on Requirements Engineering, anales del XIX Congreso Iberoamericano en Ingeniería de Software (CIBSE 2016), Quito, Ecuador, pp.407-420. ISBN: 978-9978-301-81-4.
- Mighetti, J.P., & Hadad, G.D.S. (2016b, Diciembre). *A Requirements Engineering Process Adapted to Global Software Development*. *CLEI Electronic Journal*, Vol. 19, No. 3, paper 7.
- Nahar, N., & Student, P.G. (2013). Managing Requirement Elicitation Issues Using Step-Wise Refinement Model. *International Journal of Advanced Studies in Computers, Science and Engineering*, Vol. 2, No. 5, pp.27-33.
- Niazi, M., Mahmood, S., Alshayeb, M., & Hroub, A. 2014 Challenges of the Existing Tools

- Used in Global Software Development Projects. ICSEA 2014 - The Ninth International Conference on Software Engineering Advances, pp. 385-389. ISBN: 978-1-61208-367-4.
- Nonyelum Ogwueleka, F. (2012). Requirement elicitation problems in software development – A case study of a GSM service provider . Indian Journal of Innovations & Developments, Nigeria, Vol. 1, No. 8, pp.599-605.
- Nuseibeh, B., & Easterbrook, S. (2000). Requirements Engineering: A Roadmap. Future of SE Track 2000, Limerick, Ireland, pp.35-46
- Oberg, R., Probasco, L., & Ericsson, M. (2000). *Applying Requirements Management with Use Cases*. Rational Software Corporation. Technical Paper TP505 (Version 1.4), 24 pág.
- Potts, C. (1995). Using schematic scenarios to understand user needs. The 1st conference on Designing interactive systems: processes, practices, methods, & techniques, ACM, pp. 247-256.
- Pressman, R. (2014). *Ingeniería de Software - Un enfoque Práctico* (Séptima Edición ed.). México: McGrawHill.
- Prikladnicki, R., Audy, J.L., & Evaristo, R. (2004). *An empirical study on Global Software Development: Offshore Insourcing of IT Projects*. Third International Workshop on Global Software Development, Edinburg, Scotland. pp.53-58.
- Prikladnicki, R., Audy, J.L., & Evaristo, R. (2003). *Global Software Development in Practice Lessons Learned. Software Process: Improvement and Practice*, Vol. 8, No. 4, pp.267-281.
- Regnell, B., Kimbler, K., Wesslén, A. (1999). Improving the Use Case Driven Approach to Requirements Engineering. Requirements Engineering with Use Cases – a Basis for Software Development, Technical Report 132, Paper I, Department of Communication Systems, Lund University, pp. 43-63.
- Richardson, I., Casey, V., McCaffery, F., Burton, J., & Beecham, S. (2012). *A process Framework for Global Software Engineering Teams. Information and Software Technology*, 54, 11, pp.1175-1191.

- Rolland, C., Ben Achour, C., Cauvet, C., Ralyté, J., Sutcliffe, A., Maiden, M., Jarke, M., Haumer, P., Pohl, K., Dubois, E., & Heymans, P. (1998). A Proposal for a Scenario Classification Framework. *Requirements Engineering Journal*, Springer-Verlag London Ltd., Vol.3, N°1, pp.23-47.
- Royce, W. W. (1970). *Modelo de Ciclo de Vida en Cascada*. Los Angeles, CA, USA: IEEE WESCON.
- Ryan, K. (1993). The Role of Natural Language in Requirements Engineering. *IEEE International Symposium on Requirements Engineering*, San Diego, CA, pp. 240-242.
- Sarmiento, E., Leite, J.C.S.P., & Armentero, E. (2014) C&L: Generating Model based test cases from natural languages, requirements descriptions. *IEEE 1st International WorkShop on Requirements Engineering and Testing (RET 2014)*, IEEE, pp.32-38.
- Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., & Ncube, C. (2009). Exploring how to use scenarios to discover requirements. *Requirements Engineering Journal*, 14(2), 91-111.
- Sutcliffe, A. G., Maiden, N. A., Minocha, S., & Manuel, D. (1998). *Supporting Scenario-Based Requirements Engineering*. *IEEE TSE*, Vol.24, N°12, pp.1072-1088.
- Ul-Haq, S., Raza, M., Zia, A., & Khan, N. A. (2011). Issues in Global Software Development: A Critical View. *Journal of Software Engineering and Applications*, 4,10, pp.590-595.
- Van Lamsweerde, A., & Willemet, L. (1998). Inferring declarative requirements specifications from operational scenarios. In *IEEE Transactions on Software Engineering*, vol. 24, no. 12, pp. 1089-1114.
- Westland, J. C. (2002). The cost of errors in software development: evidence from industry. *Journal of Systems and Software*, 62, 1, pp.1-9
- Zapata, S. (2013). Efectividad de Técnicas Tradicionales de Elicitación de Requisitos de Software Aplicadas en Escenarios Distribuidos de Desarrollo. Tesis de Maestría, Escuela de Posgrado, Universidad Nacional de La Matanza.
- Zowghi, D. (2007). Does Global Software Development Need a Different Requirements

Engineering Process?. International Workshop on Global Software Development (ICSE 2002), Orlando, Florida, pp. 53-55.

6.2 Índice de Figuras

Figura 1 - Distribución geográfica de involucrados.....	9
Figura 2 - Ciclo de vida de los proyectos Venezuela y Brasil.....	10
Figura 3 - Costo relativo de reparación de defectos de requisitos a lo largo del tiempo	16
Figura 4 - Modelo del Léxico Extendido del Lenguaje, extraído de (Kaplan, y otros, 2000).....	20
Figura 5 - Etapas para la construcción del LEL, extraída de (Hadad, y otros, 1999).....	21
Figura 6 - Modelo de Escenario, extraído de (Hadad, 2008)	25
Figura 7 - Principales Amenazas y sus derivadas en el desarrollo global de software	34
Figura 8 - Ejemplo de Estructura de una Especificación de Requisito	50
Figura 9 - Cantidad de Especificaciones de requisitos según su complejidad para el proyecto Venezuela	56
Figura 10 - Ejemplo de estimación de especificación del proyecto Venezuela.....	58
Figura 11 - Cálculo de Puntos de Función para ILF y ELF.....	59
Figura 12 - Cálculo de Puntos de Función para EI, EO y EQ	59
Figura 13 - Resultado de la estimación por puntos de función para un reporte de ejemplo	60
Figura 14 – Tiempo estimado y real para crear especificaciones de requisitos según su complejidad para el proyecto Venezuela (días)	61
Figura 15 - Tiempo estimado y tiempo real para la creación de especificaciones según analistas con y sin conocimiento del contexto para el proyecto Venezuela (días)	62
Figura 16 - Tiempo estimado y tiempo real para la aprobación de especificaciones para el proyecto Venezuela (días)	64
Figura 17 - Tiempo estimado y tiempo real de expertos e inexpertos para aprobar especificaciones de requisitos para el proyecto Venezuela (días).....	65
Figura 18 - Tiempo estimado de transición y tiempo promedio real de transición para el proyecto Venezuela (horas)	68
Figura 19 - Tiempo promedio real de transición entre analistas y desarrolladores considerando pares experto-experto e inexperto-inexperto para el proyecto Venezuela (horas)	70
Figura 20 - Defectos encontrados en la certificación en el proyecto Venezuela.....	74
Figura 21 - Ejemplos de símbolos del LEL del proyecto Brasil	96
Figura 22 - Ejemplo de un Escenario Futuro del proyecto Brasil	97
Figura 23 - Cantidad de Especificaciones de requisitos según su complejidad para el proyecto Brasil	99
Figura 24 - Tiempo estimado y tiempo real para crear especificaciones para el proyecto Brasil (días)...	100
Figura 25 - Tiempo estimado y tiempo real para la aprobación de especificaciones para el proyecto Brasil (días)	101
Figura 26 - Tiempo estimado y tiempo real para la transición de especificaciones a la fábrica de software para el proyecto Brasil (horas)	103
Figura 27 - Tiempo promedio real de transición entre analistas y desarrolladores considerando pares experto-experto e inexperto-inexperto para el proyecto de Brasil (horas)	105
Figura 28 - Defectos encontrados en la certificación en el proyecto Brasil.....	108
Figura 29 - Tiempos reales de creación de especificaciones para ambos proyectos (días)	111
Figura 30 - Tiempos reales de aprobación de especificaciones para ambos proyectos (días)	112

6.3 Índice de Tablas

Tabla 1 - Objetivos y consecuencias del LEL (Hadad, y otros, 1999).....	19
Tabla 2 - Objetivos y consecuencias del uso de Escenarios, extraído de (Hadad, y otros, 1997)	29
Tabla 3 - Amenazas identificadas por los distintos autores en el tema.....	44
Tabla 4 - Distribución de competencias del equipo para el Proyecto Venezuela	46
Tabla 5 - Métrica DFE01: Entrega de especificación de requisitos	52
Tabla 6 - Métrica DFA01: Aprobación de especificación de requisitos	53
Tabla 7 - Métrica DFR01: Rechazo de especificación de requisitos	53
Tabla 8 - Métrica TRATPT01: Tiempo promedio de transición a fábrica	54
Tabla 9 - Métrica TRACPDF01: Cantidad de preguntas por especificación de requisitos.....	54
Tabla 10 - Métrica CERTNCF01: Cantidad de no conformidades por errores, cambios o ambigüedades en las especificaciones de requisitos.....	55
Tabla 11 - Métrica CERTINCPER01: Índice de no conformidades por errores, cambios o ambigüedades por especificaciones de requisitos	55
Tabla 12 - Variación de tiempo real respecto al estimado para crear especificaciones para el proyecto Venezuela (días)	63
Tabla 13 - Variación de tiempo real respecto al estimado para aprobar especificaciones para el proyecto Venezuela (días)	66
Tabla 14 – Variación promedio de rechazos por especificación creada por categoría de analista para el proyecto Venezuela.....	66
Tabla 15 - Variación de tiempo real respecto al estimado en la transición a la fábrica de software por categoría de analista para el proyecto Venezuela (horas)	69
Tabla 16 - Variación de tiempos de transición entre analistas y desarrolladores expertos e inexpertos para el proyecto Venezuela.....	70
Tabla 17 - Comparación del promedio de preguntas realizadas durante la transición y desarrollo según el conocimiento en el dominio de los desarrolladores para el proyecto Venezuela.....	71
Tabla 18 - Comparativa de cantidad de especificaciones de requisitos afectadas por defectos reportados en la certificación para el proyecto Venezuela	75
Tabla 19 - Amenazas identificadas para el proyecto Venezuela.....	78
Tabla 20 - Distribución de competencias del equipo para el proyecto Brasil.....	84
Tabla 21 - Amenazas y su posible mitigación mediante LEL y Escenarios	85
Tabla 22 - Variación de tiempo real respecto al estimado para crear especificaciones para el proyecto Brasil (días)	101
Tabla 23 - Variación de tiempo real respecto al estimado para aprobar especificaciones para el proyecto Brasil (días)	102
Tabla 24 - Variación promedio de rechazos por especificación creada por categoría de analista para el proyecto Brasil.....	102
Tabla 25 - Variación de tiempo real respecto al estimado en la transición a la fábrica de software por categoría de analista para el proyecto Brasil (horas)	104
Tabla 26 – Variación de los tiempos de transición entre analistas y desarrolladores expertos e inexpertos para el proyecto de Brasil (horas).....	106
Tabla 27 - Comparación del promedio de preguntas realizadas durante la transición y desarrollo según el conocimiento en el dominio de los desarrolladores para el proyecto Brasil.....	107
Tabla 28 - Comparativa de cantidad de especificaciones de requisitos afectadas por defectos reportados en la certificación en el proyecto Brasil.....	109
Tabla 29 - Variación de tiempos de creación de especificaciones entre ambos proyectos.....	112

Tabla 30 - Variación de tiempos de aprobación de especificaciones entre ambos proyectos.....	113
Tabla 31 - Variación de la cantidad de rechazos sobre las especificaciones de requisitos entre ambos proyectos	114
Tabla 32 - Variación de tiempos de transición a la fábrica de software entre ambos proyectos	115
Tabla 33 - Variación de cantidad de preguntas realizadas por los desarrolladores de ambos proyectos	115
Tabla 34 - Índice de defectos por especificación de requisitos para ambos proyectos	116
Tabla 35 - Comparativa de especificaciones con defectos reportados en la certificación para ambos proyectos	116
Tabla 36 - Comparativa de defectos reportados en la certificación según especificaciones creadas por categoría de analistas para ambos proyectos	117
Tabla 37 - Comparación general de mediciones entre los proyectos Venezuela y Brasil	118

Anexo A – Proceso de software en el proyecto Venezuela

El proceso de desarrollo seguido en el proyecto Venezuela está compuesto por 7 subprocesos:

- 1) Conceptualización
- 2) Iniciación
- 3) Análisis y diseño
- 4) Transición a Fábrica de Software y Desarrollo
- 5) Pruebas
- 6) Implementación
- 7) Post-Implementación

Cada uno de estos subprocesos está compuesto por sus respectivas actividades que se detallan a continuación.

1. Subproceso de Conceptualización

- a) Actividades de gestión del proyecto
 - Análisis de factibilidad, técnico y económico.
 - Obtención de patrocinadores, y definición de objetivos.
 - Obtención de la aprobación formal del proyecto.
 - Análisis inicial de riesgo.
 - Análisis de impacto de interesados.
- b) Preparación de herramientas e infraestructura
 - Instalación y configuración de herramientas de estimación y planificación.
 - Instalación y configuración de herramientas de seguimiento y control.
 - Instalación y configuración de herramientas de gestión de configuración.
 - Definición de la arquitectura de ambientes.
 - Definición de los parámetros técnicos de arquitectura.

- Preparación y certificación de los ambientes.
- c) Configuración, promociones y generación de versiones
 - Definición de la estrategia y el calendario de promociones.
 - Revisión general, validación y aprobación formal de la estrategia y el calendario de promociones.
- d) Definición del alcance y la estrategia de elicitación de requerimientos
 - Elicitación y análisis de los requerimientos de alto nivel generados por el negocio.
 - Creación del registro de requerimientos del negocio.
 - Creación de la matriz de rastreabilidad.
 - Validación mediante sesiones de los requerimientos del negocio
 - Aprobación formal de los requerimientos del negocio.
- e) Definición de la arquitectura preliminar del sistema
 - Análisis de los requisitos no funcionales de alto nivel.
 - Creación de la arquitectura de alto nivel.
 - Validación mediante sesiones de la arquitectura preliminar.
 - Aprobación formal de la arquitectura preliminar.

2. Subproceso de Iniciación

- a) Actividades de gestión del proyecto
 - Análisis de riesgo detallado.
 - Definición de métricas para el proyecto.
 - Generación de estimaciones de alto nivel.
 - Revisión general y validación de entregables propuestos.
 - Generación de la planificación de alto nivel.

- Aprobación operacional.
- Aprobación financiera.
- Definición del plan de comunicación.

b) Reunión de lanzamiento del proyecto

- Presentación a los interesados de los objetivos del proyecto, metodología, estrategia y plan general.

c) Presentación y entrenamiento del equipo

- Presentación y entrenamiento del equipo de proyecto en la metodología del proyecto.
- Presentación y entrenamiento del equipo de proyecto en las herramientas del proyecto.

3. Subproceso de Análisis y Diseño

a) Actividades de análisis y gestión del proyecto

- Revisión general y validación detallada de los requerimientos del negocio.
- Priorización de requerimientos del negocio.
- Planificación de las transiciones a la fábrica de software.
- Revisión y actualización de la planificación.
- Creación de las especificaciones de requisitos.
- Validación mediante sesiones de las especificaciones de requisitos.
- Aprobación formal de las especificaciones de requisitos.
- Definición de la arquitectura definitiva.
- Definición y diseño del enfoque de pruebas.
- Revisión general y aprobación formal del enfoque de pruebas.

b) Actividades de diseño

- Diseño del modelo de datos y clases del proyecto.
- Revisión y aprobación formal del modelo de datos y clases del proyecto.
- Creación de las especificaciones de diseño de los requisitos no funcionales del sistema.
- Revisión general y aprobación formal de las especificaciones de diseño de los requisitos no funcionales del sistema.
- Creación del plan de pruebas de integración.
- Revisión general y aprobación formal del plan de pruebas de integración.
- Creación del plan de pruebas de certificación de usuarios.
- Revisión general y aprobación formal del plan de pruebas de certificación de usuarios.
- Creación de casos de pruebas.
- Revisión general y aprobación formal de casos de pruebas.

4. Subproceso de Transición a Fábrica de Software y Desarrollo

a) Actividades de construcción

- Ejecución de las transiciones a la fábrica de software.
- Creación del código de la aplicación.
- Ejecución de las pruebas unitarias.
- Desarrollo de los manuales de usuario, guías de instalación, notas de release y guías de operación.

5. Subproceso de Pruebas

a) Actividades de pruebas de integración

- Creación del log de defectos de la etapa de pruebas de integración.
- Seguimiento y control de la etapa de pruebas de integración.
- Generación de reportes de la etapa de pruebas de integración.

- Aprobación formal de la etapa de pruebas de integración.

b) Actividades de pruebas de certificación de usuarios

- Creación del log de defectos de la etapa de pruebas de certificación de usuarios.
- Seguimiento y control de la etapa de pruebas de certificación de usuarios.
- Generación de reportes de la etapa de pruebas de certificación de usuarios.
- Aprobación formal de la etapa de pruebas de certificación de usuarios.
- Creación del plan de implementación.

6. Subproceso de Implementación

a) Actividades de gestión del proyecto

- Planificación de la reunión de lecciones aprendidas.
- Reuniones de cierre de implementación.
- Aprobación formal de los entregables.

b) Actividades de pre-operación

- Disponibilización de las especificaciones de requisitos para el equipo de soporte.
- Entrenamiento a los diferentes grupos de usuarios.
- Creación del plan de soporte de post salida a producción.
- Ejecución de pruebas y simulaciones de pre-operación.
- Aprobación formal del equipo de producción.
- Implementación de la solución en los ambientes de producción y ejecución de las pruebas de verificación.
- Comunicación de la implementación del sistema a los interesados.

c) Actividades de gestión del conocimiento

- Creación, validación y aprobación formal de la documentación de transferencia

de conocimientos del equipo de proyecto al equipo de soporte.

- Ejecución de las actividades de gestión y transferencia del conocimiento del equipo de proyecto al equipo de soporte.

7. Subproceso de Post-Implementación

a) Actividades de gestión del proyecto.

- Reunión de lecciones aprendidas.
- Liberación de los recursos y archivo de la información del proyecto.
- Informar la finalización del proyecto.
- Revisión post-implementación.

Anexo B – Proceso de software en el proyecto Brasil

Para el proyecto Brasil se siguió el mismo proceso utilizado en el proyecto Venezuela, aunque se adicionaron actividades a algunos de los subprocesos, correspondientes a la propuesta de mitigación de amenazas a requisitos. A continuación se enumeran los subprocesos y sus actividades, indicando en cursiva las actividades agregadas.

1. Subproceso de Conceptualización

a) Actividades de gestión del proyecto

- Análisis de factibilidad, técnico y económico.
- Obtención de patrocinadores, y definición de objetivos.
- Obtención de la aprobación formal del proyecto.
- Análisis inicial de riesgo.
- Análisis de impacto de interesados.

b) Preparación de herramientas e infraestructura

- Instalación y configuración de herramientas de estimación y planificación.
- Instalación y configuración de herramientas de seguimiento y control.
- Instalación y configuración de herramientas de gestión de configuración.
- Definición de la arquitectura de ambientes.
- Definición de los parámetros técnicos de arquitectura.
- Preparación y certificación de los ambientes.
- *Selección, instalación y configuración de herramientas de gestión del conocimiento.*
- *Inducción a los analistas sobre la creación del LEL, Escenarios Actuales y Escenarios Futuros.*

c) Configuración, promociones y generación de versiones

- Definición de la estrategia y el calendario de promociones.
- Revisión general, validación y aprobación formal de la estrategia y el calendario de promociones.

d) Definición del alcance y la estrategia de elicitación de requerimientos

- *Creación del Léxico Extendido del lenguaje.*
- *Validación mediante sesiones del LEL.*
- *Creación de los Escenarios Actuales.*
- *Validación mediante sesiones de los Escenarios Actuales.*
- *Almacenamiento del LEL y los Escenarios Actuales en el repositorio de gestión del conocimiento.*
- *Aprobación formal del LEL y los Escenarios Actuales.*
- Elicitación y análisis de los requerimientos de alto nivel generados por el negocio.
- Creación del registro de requerimientos del negocio.
- Creación de la matriz de rastreabilidad.
- Validación mediante sesiones de los requerimientos del negocio.
- Aprobación formal de los requerimientos del negocio.
- *Creación de los Escenarios Futuros.*
- *Validación mediante sesiones de los Escenarios Futuros.*
- *Almacenamiento de los Escenarios Futuros en el repositorio de gestión del conocimiento.*
- *Aprobación formal de los Escenarios Futuros.*

e) Definición de la arquitectura preliminar del sistema

- Análisis de los requisitos no funcionales de alto nivel.
- Creación de la arquitectura de alto nivel.

- Validación mediante sesiones y aprobación formal de la arquitectura preliminar.

2. Subproceso de Iniciación

a) Actividades de gestión del proyecto

- Análisis de riesgo detallado.
- Definición de métricas para el proyecto.
- Generación de estimaciones de alto nivel.
- Revisión general y validación de entregables propuestos.
- Generación de la planificación de alto nivel.
- Aprobación operacional.
- Aprobación financiera.
- Definición del plan de comunicación.

b) Reunión de lanzamiento del proyecto

- Presentación a los interesados de los objetivos del proyecto, metodología, estrategia y plan general.
- ***Presentación a los interesados de los modelos LEL y Escenarios, y gestión del conocimiento.***

c) Presentación y entrenamiento del equipo

- Presentación y entrenamiento del equipo de proyecto en la metodología del proyecto.
- Presentación y entrenamiento del equipo de proyecto en las herramientas del proyecto.
- ***Presentación y entrenamiento del equipo de proyecto en los modelos LEL y Escenarios.***
- ***Presentación y entrenamiento del equipo de proyecto en las herramientas***

de la gestión del conocimiento.

3. Subproceso de Análisis y Diseño

a) Actividades de análisis y gestión del proyecto

- Revisión general y validación detallada de los requerimientos del negocio.
- Priorización de requerimientos del negocio.
- Planificación de las transiciones a la fábrica de software.
- Revisión y actualización de la planificación.
- Creación de las especificaciones de requisitos.
- Validación mediante sesiones de las especificaciones de requisitos.
- Aprobación formal de las especificaciones de requisitos.
- *Almacenamiento de las especificaciones de requisitos en el repositorio de gestión del conocimiento.*
- Definición de la arquitectura definitiva.
- Definición y diseño del enfoque de pruebas.
- Revisión general y aprobación formal del enfoque de pruebas.

b) Actividades de diseño

- Diseño del modelo de datos y clases del proyecto.
- Revisión y aprobación formal del modelo de datos y clases del proyecto.
- Creación de las especificaciones de diseño de los requisitos no funcionales del sistema.
- Revisión general y aprobación formal de las especificaciones de diseño de los requisitos no funcionales del sistema.
- Creación del plan de pruebas de integración.
- Revisión general y aprobación formal del plan de pruebas de integración.

- Creación del plan de pruebas de certificación de usuarios.
- Revisión general y aprobación formal del plan de pruebas de certificación de usuarios.
- Creación de casos de pruebas.
- Revisión general y aprobación formal de casos de pruebas.

4. Subproceso de Transición a Fábrica de Software y Desarrollo

a) Actividades de construcción

- Ejecución de las transiciones a la fábrica de software.
- Creación del código de la aplicación.
- Ejecución de las pruebas unitarias.
- Desarrollo de los manuales de usuario, guías de instalación, notas de release y guías de operación.

5. Subproceso de Pruebas

a) Actividades de pruebas de integración

- Creación del log de defectos de la etapa de pruebas de integración.
- Seguimiento y control de la etapa de pruebas de integración.
- Generación de reportes de la etapa de pruebas de integración.
- Aprobación formal de la etapa de pruebas de integración.

b) Actividades de pruebas de certificación de usuarios

- Creación del log de defectos de la etapa de pruebas de certificación de usuarios.
- Seguimiento y control de la etapa de pruebas de certificación de usuarios.
- Generación de reportes de la etapa de pruebas de certificación de usuarios.
- Aprobación formal de la etapa de pruebas de certificación de usuarios.

- Creación del plan de implementación.

6. Subproceso de Implementación

a) Actividades de gestión del proyecto

- Planificación de la reunión de lecciones aprendidas.
- Reuniones de cierre de implementación.
- Aprobación formal de los entregables.

b) Actividades de pre-operación

- *Disponibilización del LEL, los Escenarios Actuales y los Escenarios Futuros para el equipo de soporte.*
- Disponibilización de las especificaciones de requisitos para el equipo de soporte.
- Entrenamiento a los diferentes grupos de usuarios.
- Creación del plan de soporte de post salida a producción.
- Ejecución de pruebas y simulaciones de pre-operación.
- Aprobación formal del equipo de producción.
- Implementación de la solución en los ambientes de producción y ejecución de las pruebas de verificación.
- Comunicación de la implementación del sistema a los interesados.

c) Actividades de gestión del conocimiento

- Creación, validación y aprobación formal de la documentación de transferencia de conocimientos del equipo de proyecto al equipo de soporte.
- Ejecución de las actividades de gestión y transferencia del conocimiento del equipo de proyecto al equipo de soporte.

7. Subproceso de Post-Implementación

a) Actividades de gestión del proyecto.

- Reunión de lecciones aprendidas.
- Liberación de los recursos y archivo de la información del proyecto.
- Informar la finalización del proyecto.
- Revisión post-implementación.