

Uso de un Léxico y Escenarios para Mitigar Amenazas a Requisitos en el Desarrollo Global de Software

Juan P. Mighetti¹, Graciela D. S. Hadad²

¹DIIT, Universidad Nacional de La Matanza, Argentina

²Escuela de Informática, Universidad Nacional del Oeste, Argentina

jmighetti@unlam.edu.ar, ghadad@uno.edu.ar

Abstract. El desarrollo global de software se está difundiendo cada vez más en las organizaciones. Si bien brinda ventajas competitivas al aumentar la velocidad en las entregas del software y reducir costos por aprovechar recursos calificados más baratos, es una forma de trabajo compleja, donde amenazas provenientes principalmente de distancias geográficas y culturales pueden impactar negativamente en actividades y artefactos del proceso. La calidad de los requisitos suele deteriorarse debido a estas amenazas latentes no tratadas, propagando defectos en subsiguientes fases del desarrollo. Estas dificultades fueron estudiadas en un proyecto real de gran envergadura, desarrollado bajo esta modalidad de trabajo. Debido a ellas, se elaboró una propuesta utilizando un modelo Léxico y Escenarios para mitigar amenazas que afectan a los requisitos en esta modalidad distribuida. La propuesta fue aplicada en un nuevo proyecto real de similares características y, al compararlos en calidad y tiempos en requisitos, se obtuvieron resultados suficientemente promisorios.

Keywords: Desarrollo Global de Software, Amenazas, Comunicación, Proceso de Requisitos, Léxico Extendido del Lenguaje, Escenarios.

1 Introducción

Las especificaciones de requisitos son la entrada para las etapas siguientes del proceso de desarrollo de software, por lo que la calidad de las mismas es vital para obtener el producto de software apropiado y cumplir además con los plazos y costos establecidos. La producción de requisitos no es una actividad trivial ni monolítica [1], debiéndose encarar siguiendo un proceso definido que contemple las circunstancias propias del proyecto y del contexto de aplicación [2]. Un caso particular son los proyectos de desarrollo global de software (DGS), los cuales suelen estar afectados por diversas amenazas, tales como distancia geográfica y temporal, diferentes culturas e idiomas, y dificultades de comunicación, entre otros factores [3][4][5]. Estas amenazas latentes suelen impactar negativamente sobre los requisitos [4]. 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 proyectos de software [6].

Es importante conocer el nivel de exposición a amenazas con el que se trabaja bajo el DGS; dicho nivel dependerá de aspectos tales como el conocimiento del equipo de trabajo sobre el dominio del problema, la confianza existente entre todos los involucrados, y los intereses comunes o individuales que prevalecen. Conocer tardíamente estas amenazas y/o subestimarlas podría traer consecuencias nefastas para los requisitos, en primer lugar, y para todo el proyecto en general, tanto en calidad como en tiempos y costos. Anticiparlas y tomar las medidas adecuadas para su mitigación puede ayudar a conseguir los resultados esperados [5].

En tal sentido, se propone un proceso de requisitos basado en modelos escritos en lenguaje natural (LN), acompañado de un repositorio de gestión de conocimiento, con el fin de atacar principalmente problemas de comunicación, distancias, diferencia idiomática y conocimiento fragmentado. Esta propuesta se elaboró a partir de las dificultades detectadas y cuantificadas en un proyecto de DGS donde no se habían tomado medidas respecto a las amenazas propias de esta modalidad de trabajo. El proceso fue aplicado en un proyecto real en el mismo dominio y comparando los resultados alcanzados sobre la calidad de los requisitos y tiempos de las actividades involucradas se pudo determinar la mitigación de amenazas a requisitos lograda.

En la siguiente sección se describe un proceso de requisitos basado en modelos en LN y sus beneficios, y se presentan brevemente las amenazas potenciales que pueden afectar a un DGS. En la sección 3 se describen las amenazas identificadas en un proyecto real de DGS tras un análisis post-mortem del mismo. En la sección 4 se presenta la propuesta de mitigación de amenazas a requisitos y su aplicación en otro proyecto de similares características, analizando comparativamente la mitigación de amenazas a requisitos lograda. Al final, se exponen conclusiones y trabajos futuros.

2 Marco Teórico

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 [7]. Si ocurre un defecto en esta fase, impactará en cascada en las demás fases del proceso de software deteriorando el producto en su totalidad [7]. Si a ello se suma que el DGS posee características distintivas que complejizan el proceso de software, esto compromete aún más las actividades de definición de requisitos [8].

2.1 Proceso de Requisitos basado en Modelos en Lenguaje Natural

La ingeniería de requisitos tiene un ingrediente social de carácter colectivo, dada la participación de una variedad de involucrados en un proyecto, los que suelen poseer diversas habilidades, conocimientos y vocabularios, que hacen que el entendimiento sea una tarea ardua. Para tener un proceso de ingeniería de requisitos exitoso es vital disponer de una buena comunicación entre las partes y comprender cabalmente el contexto del sistema a construir [9][10]. Es por ello que seguir un proceso de requisitos basado en modelos escritos en LN facilita esta comunicación. Los modelos Léxico Extendido del Lenguaje (LEL) y Escenarios poseen esta particularidad, que

proporciona ventajas no solo al proceso de requisitos, sino a lo largo de todo el ciclo de vida del software, siendo un medio de comunicación de fácil entendimiento entre las partes [9]. El LEL es un glosario que define los términos utilizados en el contexto de aplicación [10]; cada término se define mediante su noción (denotación) y su impacto (connotación en ese contexto). Los escenarios son descripciones de comportamiento en un contexto dado y en un momento particular [9]; ese comportamiento se define mediante un conjunto de episodios, realizados por actores para cumplir un objetivo específico, y las excepciones que impiden ese cumplimiento. Se distinguen dos conjuntos de escenarios: i) los Escenarios Actuales (EA), que describen las situaciones observadas en el contexto de aplicación, y ii) los Escenarios Futuros (EF), que describen situaciones que se proyecta ocurrirán en el contexto con la introducción del software a construir. La Fig. 1 presenta 2 términos del LEL de un proyecto de DGS donde se propuso mitigar amenazas a requisitos. La Fig. 2 muestra un EF creado para dicho proyecto. Ambos ejemplos están escritos en inglés, por ser el idioma común adoptado, debido a las diferencias idiomáticas de los involucrados.

<i>Symbol</i>	<i>Policy / Contract</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, bank transfer.
<i>Symbol</i>	<i>Insured</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.

Fig. 1. Ejemplo de dos Términos del LEL

En general, existen jergas propias del contexto del problema, con palabras o frases que tienen un significado particular. No entender este lenguaje puede conllevar a defectos en las especificaciones de requisitos, que derivarían en re-trabajo por correcciones, reiterando verificaciones y validaciones. Entonces, el LEL mejora la comunicación entre los involucrados, al establecer como terminología común a utilizar la propia de los usuarios, lo que ayuda a entablar una buena relación con ellos, facilita la validación de modelos escritos en LN y minimiza las ambigüedades en los modelos que se generan, entre otros aportes [10].

Los escenarios permiten conocer el contexto de aplicación, unificar criterios sobre cómo se interactuará con el sistema, incrementar el compromiso de clientes y usuarios con el propio proceso de desarrollo, y entrenar a nuevos integrantes del equipo [9].

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).	

Fig. 2. Ejemplo de un Escenario Futuro

El proceso de requisitos basado en modelos en LN se desarrolla en 4 etapas [11]:

- i. Comprender el vocabulario del contexto: crear, verificar y validar el LEL;
- ii. Comprender el contexto actual: crear, verificar y validar Escenarios Actuales;
- iii. Definir el contexto del software: crear, verificar y validar Escenarios Futuros;
- iv. Explicitar los requisitos del software a partir de los Escenarios Futuros: crear, verificar y validar las especificaciones de requisitos (ver Fig. 3).

1. Introduction
2. Business Process and Business Rules
3. Solution Requirements
3.1 Business Function Requirements
3.1.1 Function 1: <description>.
3.1.2 Function 2: <description>.
.....
4 Other Business Process Requirements
4.1 Organizational Requirements
4.2 Location Requirements
4.3 Data Requirements
4.4 Application Requirements
4.5 Technology Requirements

Cabe aclarar que cuando se hace referencia a la creación de un modelo, ello involucra actividades de elicitación, modelado y gestión de los mismos. Por otro lado, debe notarse que todos los modelos utilizan la terminología definida en el LEL con hipervínculos a él, de manera tal de reducir la ambigüedad inherente a las descripciones en LN.

Fig. 3. Índice de una Especificación de Requisitos

2.2 Amenazas del Desarrollo Global de Software

Si se considera que distribuir los equipos de trabajo es una decisión estratégica que persigue ventajas como reducción de costos y mayor competitividad, ella conlleva una alta inversión y fuertes desafíos [12], por lo que el impacto frente a un fracaso no será menor. Las organizaciones que abordan esta modalidad de trabajo deben contar con procesos estipulados, herramientas adecuadas, medios y formas de comunicación apropiadas, políticas claras, entre otros aspectos, para hacer frente a las diversas amenazas que rodean al DGS [13].

Las principales amenazas a las que está sometido el DGS [3][4][5][12][13] son: i) comunicación inadecuada, ii) diferencias culturales e idiomáticas, iii) distancia geográfica y temporal, y iv) problemas de gestión del conocimiento. Asimismo, se derivan otras amenazas que surgen de la combinación de las anteriores [14], siendo algunas no específicas del DGS pero que deben ser atendidas: v) falta de confianza e involucramiento; vi) carencia de conocimiento del dominio; vii) problemas técnicos o de herramientas; viii) ambigüedad, contradicciones y falta de claridad en las especificaciones; ix) objetivos individuales; y x) rotación del equipo de trabajo. Desconocer o subestimar estas amenazas puede impactar fuertemente en la calidad de los requisitos [14] y cuando se identifican en etapas distantes incrementarán significativamente los costos de resolución [15].

3 Análisis Post-Mortem de un Proyecto de DGS sin Mitigación de Amenazas

La propuesta que se presenta surge como consecuencia de haber identificado, en etapas tardías de un proyecto de software de gran envergadura, un impacto altamente negativo sobre la calidad del software, los plazos estimados y el presupuesto. Este proyecto en el dominio de seguros dispuso de un presupuesto de 9 millones de dólares, un plazo estimado de 10,5 meses y la participación de alrededor de 100 personas. Fue el primero de un programa, cuyos equipos de trabajo estaban dispersos en varios países: analistas en Buenos Aires (Argentina), desarrolladores en Pune (India), testers en Chennai (India), proveedores de configuración en Polonia y proveedores de equipos en Dinamarca. Los destinatarios del software de este primer proyecto estaban localizados en Caracas (Venezuela). El segundo proyecto, cuyos destinatarios estaban en San Pablo (Brasil), tuvo la misma distribución geográfica de equipos aunque con otra composición y otros integrantes. Sobre este segundo proyecto se aplicó la propuesta para mitigar amenazas a requisitos, en base a las dificultades detectadas en el proyecto cuyos usuarios estaban en Venezuela.

La estrategia de desarrollo de software para ambos proyectos consistió en que los analistas de Buenos Aires se movilizaban al país donde residían los usuarios para la elicitación y el modelado preliminar. En Buenos Aires se planificaba, especificaban los requisitos y se diseñaba. Posteriormente, se realizaba la transición a la fábrica de software en Pune para su desarrollo, en la cual las especificaciones de requisitos se asignaban a los desarrolladores y estos evacuaban sus dudas con los analistas.

Finalizado el desarrollo, se realizaban las pruebas de integración en Chennai; el equipo de testing despachaba los entregables al país destinatario del software, donde se realizaban las pruebas de certificación del software.

Ambos proyectos utilizaron un conjunto de métricas pre-definidas por la compañía de seguros, las que se calculaban en función de la información recopilada en cada actividad. La Tabla 1 solo presenta el sub-conjunto de métricas relacionadas con requisitos. La estimación de tiempos de las actividades se realizó en base a información histórica propia de la compañía. Las relacionadas con requisitos se estimaron por complejidad de especificación: Baja, Media y Alta, discriminando por analistas o desarrolladores con y sin conocimiento en el dominio de seguros. La complejidad de las especificaciones se estableció con la ayuda de un estimador basado en puntos de función.

Tabla 1. Métricas relacionados con requisitos, aplicadas en los proyectos Venezuela y Brasil

Código	Indicador	Definición	Fase	Unidad de Medida
DFE01	Creación de especificación	Tiempo promedio de creación de una especificación de requisitos	Análisis y Diseño	Día
DFA01	Aprobación de especificación	Tiempo promedio de aprobación de una especificación	Análisis y Diseño	Día
DFR01	Rechazo de especificación	Cantidad promedio de rechazos por especificación durante la aprobación de los usuarios	Análisis y Diseño	Numérica
TRATPT01	Tiempo de transición a fábrica de software	Tiempo promedio de transición a fábrica por especificación	Transición a Fábrica	Hora
TRACPDF01	Preguntas por especificación	Cantidad promedio de preguntas por especificación de requisitos	Transición a Fábrica	Numérica
CERTNCFF01	% de defectos atribuibles a especificaciones	Porcentaje de no conformidades en especificaciones respecto al total de no conformidades	Pruebas: Certificación de usuarios	Porcentaje
CERTNCPER01	Tasa de defectos por especificación	Índice de no conformidades por especificación, debido a defectos en requisitos	Pruebas: Certificación de usuarios	Numérica

Durante la certificación del software en el proyecto Venezuela, se identificaron numerosos rechazos por parte de los usuarios, que encendieron alarmas para avanzar con el siguiente proyecto, que se encontraba a punto de comenzar y cuyo destinatario era Brasil. Estos rechazos fueron debidamente documentados, los que provenían mayoritariamente de los requisitos. En base al análisis de las métricas procesadas, se elaboró una propuesta para reducir dichos rechazos en el proyecto Brasil.

En el proyecto Venezuela, se dispuso de 20 usuarios en Caracas, 13 analistas en Buenos Aires, 38 desarrolladores en Pune y 8 testers en Chennai. El 69% de los analistas no tenían conocimiento en el dominio de seguros, y el 66% de los desarrolladores tampoco. Se crearon 104 especificaciones: 43% de complejidad Baja, 35% Media y 23% Alta. Las especificaciones de baja complejidad tenían en promedio 18 páginas, las de media complejidad 48 páginas y las de alta complejidad 82 páginas. En cuanto al proceso de requisitos utilizado involucraba las siguientes actividades:

- Elicitación y análisis de los requerimientos del negocio.

- Creación del registro de requerimientos del negocio.
- Creación de la matriz de trazabilidad.
- Validación usando sesiones y aprobación formal de requerimientos del negocio.
- Priorización de requerimientos del negocio.
- Creación de las especificaciones de requisitos y su validación mediante sesiones.
- Aprobación formal de las especificaciones de requisitos.

3.1 Métricas calculadas en el proyecto Venezuela relacionadas con requisitos

A continuación se presentan las métricas obtenidas en el proyecto Venezuela. El valor de la métrica DFE01 presenta diferencias sustanciales entre los tiempos estimados y reales de creación de especificaciones. La Tabla 2 muestra, además, que el tiempo de creación de una especificación por un analista con experiencia en el dominio es claramente inferior al tiempo de creación por un analista sin experiencia.

Tabla 2. Variación de tiempo real respecto al estimado para crear especificaciones (días)

Complejidad	Tiempo Estimado		Tiempo Real		Variación	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	6	8	7	14	17 %	75 %
Media	10	18	13	29	30 %	61 %
Alta	15	20	22	32	47 %	60 %

El tiempo promedio real para la aprobación formal de cada especificación de requisitos por parte de los usuarios (métrica DFA01) fue excesivamente alto respecto al estimado, en promedio más del 60% de lo estimado (ver Tabla 3).

Tabla 3. Variación de tiempo real respecto al estimado para aprobar especificaciones (días)

Complejidad	Tiempo Estimado		Tiempo Real		Variación	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	2	5	3	9	50 %	80 %
Media	4	9	6	13	50 %	44 %
Alta	6	10	10	19	67 %	90 %

La cantidad promedio de rechazos durante la aprobación de las especificaciones de requisitos (métrica DFR01) fue considerablemente mayor para aquellas especificaciones creadas por analistas inexpertos frente a los expertos (ver Tabla 4), en promedio un 132% superior.

Tabla 4. Variación promedio de rechazos por especificación creada por categoría de analista

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

En la fase de transición a la fábrica del software, el solapamiento del horario laboral entre Buenos Aires y Pune era de solo 2 horas diarias y el idioma utilizado fue el inglés, el cual no era la lengua madre en ninguna de las dos locaciones. Cada desarrollador realizaba una lectura previa para familiarizarse con la especificación de

requisitos asignada, para luego esclarecer dudas con el analista en una sesión limitada a esas 2 horas. Las herramientas de comunicación utilizadas fueron conferencia telefónica, video conferencia y una herramienta de gestión de preguntas y respuestas.

La Tabla 5 presenta los tiempos estimados y reales para la transición a fábrica por especificación y por categoría de analista (métrica TRATPT01). Las transiciones de analistas con manejo del dominio se mantuvieron cerca de los valores estimados, mientras que para los analistas sin conocimiento se incrementó en promedio casi 19% respecto al estimado. Los desarrolladores hicieron en promedio 13,2 preguntas por especificación para esclarecer dudas (métrica TRACPDF01). Los desarrolladores inexpertos elaboraron un 37% más de preguntas que los expertos. Esta falta de conocimiento del dominio y de la jerga hizo que gran parte del tiempo sincrónico se consumiera en esclarecer conceptos básicos.

Tabla 5. Variación de tiempo real respecto al estimado en la transición a fábrica (horas)

Complejidad	Tiempo Estimado		Tiempo Real		Variación	
	Expertos	Inexpertos	Expertos	Inexpertos	Expertos	Inexpertos
Baja	8	11	8	13	0 %	18 %
Media	18	24	19	28	6 %	17 %
Alta	22	29	24	35	9 %	21 %

En las pruebas de certificación del software, se reportaron 238 defectos clasificados según 9 orígenes, 98 de los cuales provenían de los requisitos (41% del total, métrica CERTNCFF01). Estos 98 defectos se originaban en 75 especificaciones de requisitos de las 104 creadas (72% del total de especificaciones), lo cual implicaba que algunas de ellas tenían más de un defecto asociado. De estas 75 especificaciones, el 73% habían sido creadas por analistas sin experiencia. La tasa de defectos en requisitos (métrica CERTINCPER01) fue 0,94 defectos por especificación.

En la Tabla 6 se han resumido las amenazas a las que estuvo afectado el proyecto Venezuela, 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.

La comunicación basada en diferentes léxicos tiene una mayor probabilidad de fracasar [9], lo que en este proyecto provino, por un lado, de la carencia de conocimiento del dominio por una gran parte del equipo de trabajo y, por otro, de las diferencias idiomáticas (castellano, hindú, inglés) y culturales, usando jergas locales particulares. Las consecuencias de la carencia del dominio se observaron, por ejemplo, en la interacción entre usuarios y analistas inexpertos, provocando un

incremento sustancial en los tiempos de creación y aprobación de especificaciones y, por ende, en la cantidad de rechazos a las mismas. Asimismo, la comunicación sincrónica restringida afecta el vínculo entre involucrados en el DGS [4], evidenciado en el aumento de los plazos previstos de transición de analistas a desarrolladores.

Tabla 6. Amenazas identificadas en el proyecto Venezuela

Amenaza	Impactó en:
Comunicación inadecuada	<ul style="list-style-type: none"> • Creación de especificación de requisitos (DFE01) • Aprobación de especificación de requisitos (DFA01) • Rechazo de especificación de requisitos (DFR01) • Preguntas por especificación de requisitos (TRACPDF01) • Tiempo de transición a fábrica de software (TRATPT01)
Distancia geográfica y temporal	<ul style="list-style-type: none"> • Tiempo de transición a fábrica de software (TRATPT01) • Preguntas por especificación de requisitos (TRACPDF01)
Diferencias culturales e idiomáticas	<ul style="list-style-type: none"> • Tiempo promedio de transición a fábrica (TRATPT01) • Preguntas por especificación de requisitos (TRACPDF01)
Gestión del conocimiento	<ul style="list-style-type: none"> • Tiempo promedio de transición a fábrica (TRATPT01) • Preguntas por especificación de requisitos (TRACPDF01)
Falta de confianza e involucramiento	<ul style="list-style-type: none"> • Creación de especificación de requisitos (DFE01) • Aprobación de especificación de requisitos (DFA01) • Rechazo de especificación de requisitos (DFR01)
Carencia del dominio del problema	<ul style="list-style-type: none"> • Creación de especificación de requisitos (DFE01) • Rechazo de especificación de requisitos (DFR01) • Preguntas por especificación de requisitos (TRACPDF01) • Tiempo de transición a fábrica de software (TRATPT01)
Ambigüedad, contradicciones y/o falta de claridad en las especificaciones	<ul style="list-style-type: none"> • Rechazo de especificación de requisitos (DFR01) • Tiempo de transición a fábrica de software (TRATPT01) • Preguntas por especificación de requisitos (TRACPDF01) • % de defectos atribuibles a especificaciones (CERTNCFF01) • Tasa de defectos por especificación (CERTINCPER01)

4 Propuesta para Mitigar Amenazas a Requisitos en el DGS

El proceso de desarrollo de software seguido en el proyecto Brasil fue similar al de Venezuela, salvo que se incorporaron otras actividades de requisitos para ayudar en la mitigación de amenazas a requisitos identificadas en el proyecto Venezuela. La propuesta de utilizar modelos en LN: LEL, EA y EF, está fundamentada en la subsección 2.1, donde se resaltan los aportes de estos modelos a la comunicación entre involucrados, reducción de ambigüedades e imprecisiones, comprensión del problema y comprensión de qué debe hacer el futuro software en dicho contexto. El proceso de requisitos utilizado constó de (en *cursiva* se identifican las nuevas actividades):

- *Creación del Léxico Extendido del Lenguaje.*
- *Creación de los Escenarios Actuales.*
- *Almacenamiento del LEL y los EA en el repositorio de gestión del conocimiento.*
- *Validación usando sesiones y aprobación formal del LEL y EA.*
- Elicitación y análisis de los requerimientos del negocio.

- Creación del registro de requerimientos del negocio.
- Creación de la matriz de trazabilidad.
- Validación usando sesiones y aprobación formal de requerimientos del negocio.
- *Creación de los Escenarios Futuros.*
- *Almacenamiento de los EF en el repositorio de gestión del conocimiento.*
- *Validación usando sesiones y aprobación formal de los EF.*
- Priorización de requerimientos del negocio.
- Creación de las especificaciones de requisitos y validación usando sesiones.
- Aprobación formal de las especificaciones de requisitos.

Adicionalmente, se incorporaron actividades referidas a la evolución de estos modelos escritos en LN, y se utilizó un repositorio compartido por todos los involucrados, conteniendo dichos modelos.

El equipo de trabajo estuvo conformado por: 60 usuarios en San Pablo, 16 analistas en Buenos Aires, 47 desarrolladores en Pune y 8 testers en Chennai. Al igual que en el proyecto Venezuela hubo una proporción similar de analistas y desarrolladores sin conocimiento en el dominio de seguros. Esto se debió, en ambos proyectos, a la falta de recursos disponibles en los mercados laborales de Buenos Aires y Pune. Se crearon un LEL con 46 términos, 26 Escenarios Actuales, 97 Escenarios Futuros y 154 especificaciones de requisitos. 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. Todos los modelos fueron redactados en inglés, dado la diversidad de idiomas de los equipos y con una terminología unificada al disponer de definiciones precisas a través del modelo LEL.

4.1 Métricas calculadas en el Proyecto Brasil

A continuación las Tablas 7 a 10 presentan los valores calculados para el proyecto Brasil, siendo Tablas equivalentes a las descritas en la sub-sección 3.1. Para ambos proyectos se usaron los mismos tiempos estimados. Comparando las Tablas 2 y 7, se observan mejoras en los tiempos de creación de especificaciones respecto a tiempos estimados.

Tabla 7. Variación de tiempo real respecto al estimado para crear especificaciones (días)

Complejidad	Tiempo Estimado		Tiempo Real		Variación	
	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 %

En la Tabla 8 respecto a la Tabla 3, se observa una baja importante en los tiempos de aprobación de especificaciones respecto a los estimados para analistas inexpertos. Del análisis de la duplicación de tiempo resultante en la aprobación de especificaciones simples creadas por analistas expertos, solo se ha podido concluir que el valor estimado de 2 días era realmente optimista. Por otro lado, se observa una disminución sustancial en el promedio de rechazos para especificaciones de media y alta complejidad creadas por analistas inexpertos frente a expertos (ver Tablas 4 y 9).

Tabla 8. Variación de tiempo real respecto al estimado para aprobar especificaciones (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 %

Tabla 9. Variación promedio de rechazos por especificación creada por categoría de analista

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 %

En la transición a fábrica en el proyecto Brasil, los tiempos reales frente a los estimados se redujeron aunque con poca significancia respecto al proyecto Venezuela (ver Tablas 5 y 10). Se redujo notoriamente la cantidad de preguntas de los desarrolladores respecto a Venezuela, siendo 8,5 preguntas en promedio por especificación. Se logró igualar la cantidad de preguntas de desarrolladores inexpertos frente a expertos, salvo para las especificaciones de alta complejidad.

Tabla 10. Variación de tiempo real respecto al estimado en la transición a fábrica (horas)

Complejidad	Tiempo Estimado		Tiempo Real		Variación	
	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 %

En la certificación del software se detectaron 254 defectos, correspondiendo el 32% a defectos en requisitos. Estos defectos se originaron en 69 especificaciones de un total de 154, es decir, el 45% del total de especificaciones contenían defectos, valor muy inferior al de Venezuela. Al igual que en dicho proyecto, la mayor cantidad de especificaciones con defectos fueron creadas por analistas inexpertos. La tasa de defectos en requisitos por especificación fue de 0,53, casi la mitad de Venezuela.

4.2 Análisis Comparativo de Métricas de ambos Proyectos

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 (ver Tabla 11). En el caso de los analistas inexpertos la reducción de tiempos de creación respecto a Venezuela fue del 26%, mientras que para los expertos fue del 6%. En la aprobación formal de las especificaciones también se redujeron los tiempos, excepto para especificaciones de baja complejidad creadas por analistas expertos, que en valores absolutos se incrementó en 1 día, pudiendo considerarse poco significativo; algo similar se observa en la cantidad de rechazos de especificaciones de expertos. Hubo solo una mínima reducción en la cantidad de rechazos de especificaciones creadas por analistas inexpertos, aunque debe notarse que los rechazos en el proyecto Brasil se originaron en cambios o errores menores.

Tabla 11. Variación de mediciones de Brasil respecto a Venezuela en la creación y aprobación de especificaciones de requisitos

Complejidad	Tiempo de Creación		Tiempo de Aprobación		Rechazos en Aprobación	
	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos	Analistas Expertos	Analistas Inexpertos
Baja	-14 %	-29 %	33 %	-33 %	0 %	0 %
Media	0 %	-28 %	0 %	-15 %	25 %	-11 %
Alta	-5 %	-22 %	-10 %	-32 %	20 %	-18 %

Se disminuyeron levemente los tiempos de transición a fábrica de software, mientras que hubo una reducción significativa en la cantidad de preguntas (ver Tabla 12). 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 12. Variación de mediciones de Brasil respecto a Venezuela en la Transición a Fábrica

Complejidad	Tiempo de Transición		Cantidad de Preguntas	
	Analistas Expertos	Analistas Inexpertos	Desarrolladores Expertos	Desarrolladores Inexpertos
Baja	-13 %	-8 %	-29 %	-44 %
Media	-11 %	4 %	-18 %	-40 %
Alta	0 %	-3 %	-40 %	-36 %

La Tabla 13 resume las métricas obtenidas para ambos proyectos, observándose mejoras significativas en varias de dichas métricas. Es notoria la disminución en el desvío del costo y plazo del proyecto Brasil, como la reducción casi a la mitad de la tasa de defectos por especificación en la certificación del software. En promedio hubo mejoras menos significativas en cantidad de rechazos en la aprobación y en tiempos de transición, siendo más sustanciales para los recursos sin experiencia en el dominio.

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 EA y EF, 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 EF habían sido previamente validados y dichas validaciones se facilitaron por estar los EF escritos en el lenguaje propio de los usuarios. Al utilizarse modelos escritos en LN con un vocabulario unificado, se motivó la confianza e involucramiento de los equipos de trabajo. Estos modelos LEL, EA y EF 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.

Tabla 13. Comparación de mediciones entre los proyectos Venezuela y Brasil

Métrica	Venezuela	Brasil
Días promedio de creación de cada especificación	20	16
Días promedio de aprobación de cada especificación	10	8,2
Cantidad de rechazos en aprobación de especificaciones	6	5,8
Horas promedio de transición a la fábrica de software	21,2	20,5
Cantidad de preguntas por especificación de requisitos	13,2	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 %	45 %
Desviación en el costo total estimado del proyecto	53 %	26 %
Desviación en el tiempo total estimado del proyecto	48 %	18 %

5 Conclusiones

El DGS se ha extendido ampliamente en la industria del software, y es por ello que debe prestársele la debida atención dado que, aún cuando presenta grandes beneficios, estos pueden reducirse por desconocer o desestimar sus características particulares, las que deben ser consideradas apropiadamente en el proceso de desarrollo. Esas particularidades pueden convertirse en amenazas que atentan contra la calidad del proceso y del producto [3][4][5][8][12][13]. Muchas de ellas afectan en mayor o menor medida a los requisitos, por lo que, su identificación es el primer paso hacia una estrategia proactiva de mitigación de las mismas. La propuesta que se ha presentado utilizando los modelos LEL, EA y EF [10][11] intenta aportar beneficios en la mitigación de amenazas sobre los requisitos en proyectos de DGS.

Ambos proyectos presentados, Venezuela y Brasil, poseían características similares desde la perspectiva técnica, pues se utilizaron los mismos procesos, métricas, modelo de estimación y herramientas. En base a las métricas calculadas para el proyecto Brasil, pudo evidenciarse que la incorporación de dichos modelos junto con un repositorio común conteniéndolos, permitió reducir las amenazas identificadas previamente en el proyecto Venezuela. Esta propuesta está alineada con lo planteado en [8], que expone la necesidad de un proceso de ingeniería de requisitos diferente para proyectos inmersos en el DGS. Las características de los proyectos Venezuela y Brasil son afines a muchos otros proyectos de DGS, según se evidencia en la literatura, lo que haría suponer que las mejoras alcanzadas en la calidad de las especificaciones de requisitos utilizando el LEL, los Escenarios y el repositorio compartido, también podrían obtenerse en proyectos similares. La propuesta de [3] se enfoca en el uso de ontologías y herramientas groupware para mejorar la comunicación, la comprensión del problema y la gestión de conocimiento. Recurrir al uso de ontologías es relativamente similar a las ventajas que proporciona el LEL,

aunque éste se centra en el vocabulario específico del contexto, complementado con EA para una comprensión acabada del comportamiento en ese contexto.

Una posible mejora en la propuesta presentada es la creación de un LEL adicional que refleje el vocabulario utilizado en los EF y en las especificaciones de requisitos, debido a que en ellos pueden aparecer términos no definidos en el LEL inicial o modificar términos existentes, por cambios en los procesos del negocio o cambios tecnológicos al incorporar el software. Un punto propuesto a futuro es cómo el LEL y los EF podrían colaborar en entender la jerga y los flujos del sistema para el equipo de pruebas, tarea que muchas veces es encargada a otro proveedor.

Referencias

1. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: Future of SE Track 2000, pp. 35-46. Limerick, Ireland (2000)
2. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering. In: Situational method engineering: fundamentals and experiences, pp.33-48. Springer (2007)
3. Aranda, G.N., Vizcaíno, A., Cechich, A., Piattini, M.: A Methodology for Reducing Geographical Dispersion Problems during Global Requirements Elicitation. In: 11th Workshop on Requirements Engineering, pp.117-127 (2008)
4. Damian, D.E., Zowghi, D.: RE challenges in multi-site software development organisations. Requirements Engineering Journal, 8(3),149-160 (2003)
5. Richardson, I., Casey, V., McCaffery, F., Burton, J., Beecham, S.: A Process Framework for Global Software Engineering Teams. Information and Software Technology, 54(11),1175-1191 (2012)
6. Niazi, M., Shastry, S.: Role of requirements engineering in software development process: an empirical study. In: 7th International Multi Topic Conference, pp. 402-407 (2003)
7. Nahar, N., Student, P.G.: Managing Requirement Elicitation Issues Using Step-Wise Refinement Model. International Journal of Advanced Studies in Computers, Science and Engineering, 2(5),27-33 (2013)
8. Zowghi, D.: Does Global Software Development Need a Different Requirements Engineering Process? In: International Workshop on Global Software Development - ICSE 2002, Orlando, Florida, pp.53-55 (2002)
9. Hadad, G.D.S, Kaplan, G.N., Oliveros, A., Leite, J.C.S.P.: 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), 6(1),77-103, Brasil (1999)
10. Leite, J.C., Doorn, J.H., Kaplan, G.N., Hadad, G.D., Ridao, M.: Defining System Context using Scenarios. Perspectives on Software Requirements, pp.169-199. Springer (2004)
11. Hadad, G.D.S.: Uso de Escenarios en la Derivación de Software. Tesis Doctoral, Facultad de Ciencias Exactas, Universidad Nacional de La Plata (2008)
12. Fryer, K., Gothe, M.: Global Software Development and Delivery. Trends and Challenges. IBM Developer Works, The Rational Edge (2008)
13. ul Haq, S., Raza, M., Zia, A., Khan, M.N.A.: Issues in global software development: A critical review. Journal of Software Engineering and Applications, 4(10),590--595 (2011)
14. Bhat, J.M., Gupta, M., Murthy, S.N.: Overcoming requirements engineering challenges: Lessons from offshore outsourcing. IEEE Software, 23(5), 38-44 (2006)
15. Westland, J.C.: The cost of errors in software development: evidence from industry, Journal of Systems and Software, Vol.62, pp.1-9 (2002)