

Hacia una nueva plantilla para la especificación de requisitos en lenguaje natural semi-estructurado

Raúl Mazo^{1,2}, Carlos Andrés Jaramillo³

¹CRI, Universidad Panthéon Sorbonne, Paris, Francia,

²GiDITIC, Universidad Eafit, Medellín, Colombia

³Gerencia de innovación, SQA S.A., Medellín, Colombia

^{1,2}raul.mazo@univ-paris1.fr, ³carlos.jaramillo@sqasa.com

Resumen. La Ingeniería de requisitos es un enfoque sistemático y disciplinado para la especificación y gestión de requisitos de software; uno de sus objetivos es transformar las necesidades de los interesados en especificaciones formales con el fin de implementar un sistema. Estas necesidades por lo general se manifiestan y articulan en lenguaje natural, esto debido a la universalidad y facilidad que presenta el lenguaje natural para la comunicación de los requisitos. Para facilitar los procesos de transformación y para mejorar la calidad de los requisitos resultantes, varios autores han propuesto plantillas para la escritura de requisitos en lenguaje natural estructurado. Sin embargo, esas plantillas no permiten escribir ciertos requisitos funcionales, no funcionales y restricciones, y no se adaptan correctamente a ciertos tipos de sistemas como los adaptativos, los basados en líneas de productos y los embebidos. Este artículo (i) presenta algunas evidencias de las debilidades de la plantilla recomendada por el IREB® (International Requirements Engineering Institute) conocida como plantilla Rupp, y (ii) sienta las bases para la construcción de una mejor plantilla que facilite el trabajo de los ingenieros de requisitos y que a la larga mejore la calidad de los nuevos productos. Para ello, la plantilla Rupp fue empleada en la especificación de requisitos de un producto de software para la gestión de recursos humanos de una compañía de tecnología en la ciudad de Medellín-Colombia. En esta experiencia encontramos que algunos requisitos no podrían ser escritos correctamente usando dicha plantilla. Basados en la evidencia recolectada en este caso, este artículo presenta una versión mejorada de la plantilla Rupp que permite escribir correctamente todos los requisitos analizados. A pesar de los resultados prometedores de la validación de esta nueva plantilla, es aún un trabajo preliminar y no concluyente con respecto a su cobertura o al nivel de calidad de los requisitos que se pueden escribir con ella.

Palabras Clave: Requisitos, Ingeniería de requisitos, lenguaje natural, plantilla.

1. Introducción

Los requisitos son quizás la base más importante en la construcción de productos de software porque a través de éstos se puede lograr un entendimiento común entre las partes interesadas sobre el sistema que se va a implementar. Según Wieggers & Beatty [1], los dos objetivos más importantes en la especificación de un requisito son (i) que cuando varias personas lean el requisito lleguen a la misma interpretación; y (ii) la interpretación de cada lector coincida con lo que el autor del requisito intentaba comunicar.

En este sentido, Pohl [2] sostiene que el Lenguaje Natural (LN) es la forma más común de comunicar y documentar los requisitos de un sistema, puesto que el LN es universal y está al alcance de cualquier individuo en cualquier ámbito; además, no requiere ningún tipo de entrenamiento especial en la interpretación de notaciones o símbolos como ocurre cuando se utiliza un lenguaje de ingeniería como el UML (*Unified Modeling Language*). Sin embargo, estas ventajas se ven opacadas por las desventajas mismas del lenguaje natural [3]. De acuerdo con Mavin et al. [4] algunos de los problemas que pueden aparecer en la especificación de requisitos en LN son: (i) **ambigüedad**: una palabra o frase tiene dos o más significados diferentes; (ii) **vaguedad**: falta de

precisión, estructura o detalle; (iii) **complejidad**: requisitos compuestos que contienen sub-cláusulas complejas o varias declaraciones interrelacionadas; (iv) **omisión**: requisitos faltantes, particularmente los requisitos para manejar el comportamiento no deseado; (v) **duplicación**: repetición de requisitos que definen la misma necesidad; (vi) **verbosidad**: uso de un número innecesario de palabras; (vii) **implementación**: declaraciones de cómo debe construirse el sistema, en lugar de qué debería hacer el sistema; y (viii) **falta de capacidad de ser probado**: requisitos que no se pueden probar (verdaderos o falsos) cuando se implementa el sistema

Para reducir la ocurrencia de esos problemas en las especificaciones de los requisitos de un sistema, varios autores han definido lo que se conoce como plantillas, moldes o patrones¹ [3]. Estas plantillas definen la estructura que deben tener los requisitos escritos en LN; esa estructura es flexible para que los requisitos resultantes tengan la ventaja de estar en LN y la ventaja de seguir una estructura bien definida. A este LN acotado por las posibilidades y las restricciones de la plantilla se le conoce como lenguaje natural semi-estructurado.

Las notaciones en lenguaje semi-estructurado, hacen posible construir requisitos siguiendo una plantilla y asignando una estructura similar a cada requisito. Este enfoque ayuda a evitar errores en las primeras etapas del proceso de desarrollo al especificar requisitos de alta calidad que son eficientes en tiempo y costo [5].

La plantilla propuesta por Rupp [3] también conocida MASTeR (*Mustergültige Anforderungen - die SOPHIST Templates für Requirements*) [5], ha sido aceptada como un estándar² para la especificación sintáctica de requisitos de un sistema. Además, el uso de esta plantilla ha sido reconocido como una valiosa herramienta de ayuda para que los requisitos sean más precisos y tengan una estructura sintáctica estándar que facilite su comprensión [3]. Sin embargo, toda persona que haya usado la plantilla de Rupp en proyectos reales se ha podido dar cuenta de que algunos requisitos no pueden ser expresados con esa estructura sin que quede algún grado de ambigüedad o inconsistencia. Es por eso que este artículo se enfoca en investigar la siguiente pregunta: **¿Cuáles son los vacíos de la plantilla usada como estándar para la escritura de requisitos y cómo llenar esos vacíos?**

Para buscar respuesta a esta pregunta de investigación, hemos diseñado un experimento inspirado en el método de investigación *action science* o *action research* [6] que llamaremos en español: investigación-acción. Este método fue usado para analizar los requisitos de un proyecto industrial en una compañía colombiana. Por lo tanto, esta investigación pretende **analizar** la plantilla de Rupp **con el fin de** (i) evaluar su capacidad para representar requisitos de productos industriales de manera semi-estructurada, y (ii) proponer eventuales mejoras a la plantilla; **desde el punto de vista de** un académico y un experimentado ingeniero de requisitos **en el contexto de** una empresa de tecnología.

2. Estructura sintáctica de la plantilla de Rupp

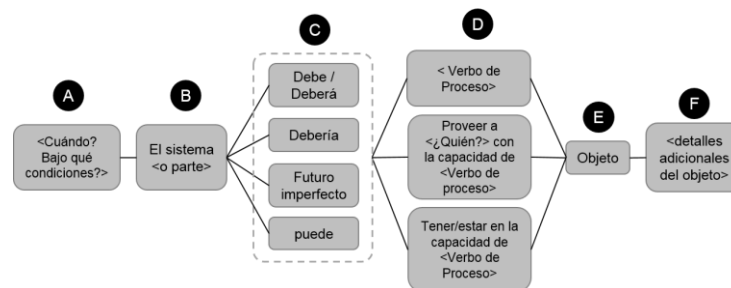


Figura 1. Plantilla de Rupp

¹ En inglés el término utilizado es template o boilerplate

² La plantilla ha sido estandarizada por el IREB® (International Requirements Engineering Institute) en el plan de certificación de ingenieros de requisitos IREB CPRE (Certified Professional for Requirements Engineering).

Como se muestra en la Figura 1 la plantilla está formada por 6 espacios para ir componiendo la sintaxis de un requisito. El resto de esta sección describe brevemente cada una de las partes que componen esta plantilla.

A. Condiciones

El primer espacio es una condición, por lo general opcional, que va al inicio del requisito, esta condición puede ser de dos tipos: condiciones lógicas compuesta por la conjunción “Si”, y condiciones temporales compuesta por la conjunción “tan pronto como” o “después que”.

B. El sistema

El segundo espacio es el nombre del sistema, el subsistema o componente del sistema que se especifica para el requisito.

C. Grado de obligatoriedad

El tercer espacio establece el grado de obligatoriedad³ que puede adquirir el requisito.

D. Actividad funcional

El cuarto espacio caracteriza la actividad funcional que puede asumir el sistema, que incluye el verbo de proceso objeto del requisito.

E. Objeto

El quinto espacio es el objeto para el cual recae el comportamiento especificado en el requisito.

F. Detalles del Objeto

El sexto y último espacio son los detalles adicionales (opcionales) sobre el objeto, los adjetivos que lo califican o las características que el objeto puede poseer.

3. Método de Investigación

El caso industrial estudiado en esta investigación se ejecutó a través del método de investigación llamado investigación-acción [6]. Investigación-acción se define como "la intervención en una situación social con el fin de mejorar esta situación y aprender de ella" [7] [8]. El método investigación-acción tiene como objetivo mejorar la práctica mediante la solución de problemas reales y es conducida con el fin de investigar los fenómenos actuales en su contexto natural [9]. Hemos elegido este método porque nos permite responder a la pregunta de investigación y lograr el objetivo de esta investigación a partir de un experimento empírico en un caso industrial. Además, (i) este método de investigación se puede ejecutar a bajo costo ya que en él los investigadores juegan un rol activo, a diferencia de los estudios de caso (*case study*) en los cuales los investigadores juegan un rol meramente observacional y eso aumenta el riesgo y el costo del experimento puesto que solo se pueden ejecutar en ambientes reales; y (ii) el rigor del método investigación-acción permite reducir las amenazas a la validez del experimento.

Susman [10] ha desarrollado un modelo detallado del método investigación-acción con las diferentes etapas que se deben llevar a cabo en cada ciclo del proceso. La implementación del método investigación-acción se hace en varios ciclos y cinco etapas para cada ciclo, como se muestra en la Figura 2.

³ El grado de obligatoriedad también se puede definir como un atributo del requisito y por eso sólo se puede emplear el verbo “deberá (shall)”[3]

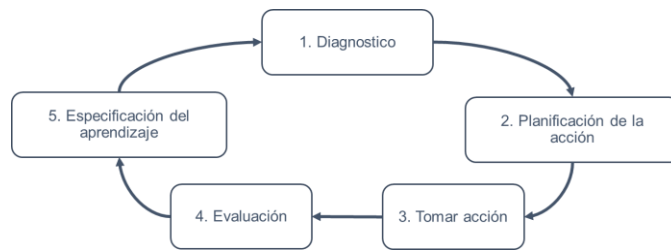


Figura 2. Proceso correspondiente a un ciclo en el método investigación-acción

A continuación resumimos las cinco etapas de cada ciclo del método investigación-acción:

1. **Diagnóstico:** identificar el problema y recopilar los datos requeridos para llevar a cabo un diagnóstico detallado.
2. **Planificación de la acción:** definir las diferentes soluciones posibles que abordan el problema definido en el primer paso.
3. **Tomar acción:** elegir una solución para implementar.
4. **Evaluación:** analizar los datos que corresponden a los resultados del plan de acción elegido.
5. **Especificación del aprendizaje:** interpretar los resultados de este plan de acción aprendiendo de acuerdo con el éxito de la solución o no. Por lo tanto, el problema se vuelve a evaluar y comienza un nuevo ciclo hasta que se resuelva el problema y todos los interesados estén satisfechos con el resultado obtenido.

Para realizar el análisis se tomó como referencia la especificación de requisitos de un proyecto de software para la compañía de tecnología SQA S.A. y se llevaron a cabo 3 ciclos para identificar las carencias, proponer las mejoras y evaluar la aplicabilidad de la plantilla mejorada en dicho proyecto. En este experimento, cada ciclo corresponde al análisis de una forma de especificación de los requisitos para el proyecto industrial. El experimento se llevó a cabo de la siguiente manera: en el **primer ciclo** se especificaron los requisitos sin utilizar ningún tipo de plantilla, estos fueron expresados en forma de prosa, tal como los interesados los manifestaron; en el **segundo ciclo** se especificaron los requisitos utilizando exclusivamente la plantilla Rupp; y en el **tercer ciclo** se especificaron los mismos requisitos utilizando la plantilla mejorada. En cada uno de ellos se ejecutaron las siguientes etapas:

1. **Diagnóstico:** al usar la plantilla de Rupp para representar los requisitos de cada uno de los sistemas que fueron usados del experimento, se identificaron algunos problemas. Esto permitió plantear en primera instancia la pregunta de investigación: ¿Cuáles son los vacíos de la plantilla usada como estándar para la escritura de requisitos y cómo llenar esos vacíos? Esta fase fue conducida a través de varios mini ciclos de especificación de los requisitos a fin de lograr tener un número de requisitos apropiados para lograr responder de forma sistemática a la pregunta de investigación.
2. **Planificación de la acción:** los autores de este artículo consideraron otras propuestas de plantillas de requisitos de otros autores. En particular, se tomó la plantilla de Rupp como referencia y en cada ciclo se evaluaba que la plantilla mejorada (resultante de cada ciclo) fuera consistente con otras plantillas encontradas en la literatura (p.e., EARS [4], Adv-EARS [11][12] e ISO/IEC/IEEE 29148-2011 [13]). Para garantizar que la plantilla mejorada seguía siendo consistente con la plantilla de Rupp y con las otras encontradas en la literatura se planeaba la siguiente estrategia: al principio de cada ciclo de escritura de los requisitos del caso industrial usando la nueva plantilla se usaban las plantillas encontradas en la literatura (no todas fueron encontradas desde el primer ciclo) con el fin de incorporar en la nueva plantilla los elementos de las otras que permitieran mejorar las situaciones en donde la plantilla de Rupp no fuese adecuada.
3. **Tomar acción:** se consideraron, en primer lugar, aquellos requisitos que no se pudieron especificar completamente utilizando la plantilla de referencia (de Rupp). Para estos

requisitos se evaluó en qué forma se podían especificar sintácticamente utilizando el lenguaje natural y las construcciones de otras planillas de especificación sintáctica de requisitos, con el fin de encontrar un patrón repetible en esta especificación. Si se identificaba un patrón repetible en al menos tres requisitos con condiciones similares, entonces se adicionaba este patrón a la nueva plantilla con el fin de enriquecerla.

4. **Evaluación:** al final de cada ciclo se evaluó si la plantilla propuesta permitía especificar por lo menos el 98% de los requisitos del proyecto correspondiente al ciclo en curso y a los requisitos de los proyectos de los ciclos precedentes sin que se presenten problemas de ambigüedad, vaguedad, complejidad, omisión, duplicidad, verbosidad, implementación o falta de capacidad de ser probado.
5. **Especificación del aprendizaje:** al final de cada ciclo los autores realizaron una interpretación de los resultados obtenidos y con base en esos resultados determinaron las fortalezas y limitaciones de la plantilla mejorada en el ciclo correspondiente.

Las diversas fases y la sucesión de ciclos son colaborativos dado que el proceso de investigación y objetivo se han llevado a cabo en colaboración entre los dos autores. Esta es otra característica que nos llevó a elegir la investigación-acción como método de investigación para este trabajo.

4. Problemas identificados en la plantilla de Rupp

Luego de realizar una evaluación sistemática de la plantilla de Rupp, usándola en diferentes en los requisitos del caso industrial de la empresa SQA S.A, se identificaron problemas que generan potenciales ambigüedades. Esos problemas se registraron en un documento, disponible en línea⁴, que contiene cada uno de los requisitos utilizando en el caso industrial y cada uno de los problemas encontrados durante la investigación. En particular, la primera hoja presenta los requisitos utilizando lenguaje natural sin apoyo de ninguna plantilla, la segunda hoja presenta los requisitos utilizando la plantilla Rupp, la tercera hoja presenta los problemas que se identifican al utilizar la plantilla Rupp y la cuarta y última hoja presenta los requisitos utilizando la plantilla mejorada.

A continuación, se describen las dificultades y barreras que se han detectado al trabajar con la plantilla de Rupp. Para cada uno de este tipo de problemas hemos definido un nombre descriptivo, una breve descripción y un ejemplo que permita entender mejor el problema.

4.1. Motivos Faltantes

Algunas veces es necesario expresar el motivo o razón de un requisito. Por ejemplo, en los marcos de desarrollo ágil uno de los aspectos más importantes en la especificación de requisitos por medio de historias de usuarios es especificar el “por qué” o el “para qué” del requisito [14] [15]. Esto le da un mejor contexto a quien implemente la funcionalidad o comportamiento que describa el requisito y le permitirá entender mejor el nivel de importancia o prioridad del requisito. En la plantilla de Rupp no hay un espacio para definir, aunque sea de forma opcional, la razón de un requisito. Por ejemplo, los requisitos “El sistema SMV debe tener la capacidad de interactuar con otros dispositivos de personas cercanas para conocer su actividad vital” y “Si algún sensor sobrepasa los límites tolerables definidos, el sistema de domótica debe encender una sirena para prevenir al dueño de casa” tienen un “para qué” de importancia vital por tratarse de sistemas críticos. Si una plantilla de especificación de requisitos permitiera definir la razón de los requisitos, los desarrolladores podrán entender fácilmente, porque está consignado de manera explícita, lo importante que es implementar esos requisitos con altos niveles de calidad.

⁴ https://docs.google.com/spreadsheets/d/1_xDK1A4JObOsOKDYW3NoT8q5_3BpOyJH-vl1w10OpUs/edit?usp=sharing

4.2. Omisión de cantidades y rangos

En ocasiones los requisitos hacen alusión no solo a un objeto específico, sino a varios objetos o a un rango de objetivos de la misma naturaleza. La plantilla de Rupp no permite explícitamente la posibilidad de especificar rangos o cantidades de objetos en los requisitos. Como lo podemos ver en el siguiente ejemplo de requisito, la omisión de esta cantidad puede dar lugar a ambigüedades o imprecisiones: “El subsistema de punto de venta debe proveer al administrador POS con la habilidad de vincular entre una y 10 bodegas máximo a un punto de venta”

El objeto bodega no hace alusión a una única bodega, sino que deja abierta la posibilidad de vincular un rango de bodegas a un punto de venta. De acuerdo con lo anterior, una plantilla de escritura de requisitos debería ofrecer explícitamente la posibilidad de cuantificar el número de objetos y objetivos.

4.3. Omisión de bicondicionales

Algunos requisitos exigen que ciertos comportamientos se lleven a cabo sólo si se cumplen ciertas condiciones; de otra manera el comportamiento no se podrá realizar. Llamamos a esto “bicondicional” para expresar que el comportamiento A se realice “si y solo si” se cumple B y viceversa. Por ejemplo, en el requisito “El subsistema de punto de venta debe mostrar las cajas si y solo si se encuentran en estado activo”, un comportamiento se llevará a cabo solo para los objetos que se encuentren en cierto estado y no para todos los objetos dentro del dominio. Aquí hay una condición explícita que el requisito debe efectuar a través del verbo de proceso “mostrar”, por medio del condicional “si y solo si”. Consideremos otro ejemplo de requisito: “Después de que se ha encendido la aspiradora, el sistema Ivaccum debe iniciar el ciclo de limpieza si y solo si la carga de la batería de la aspiradora se encuentra en el 90% o más”. En este caso el comportamiento del objeto depende de una condición sobre la carga de la batería.

Como se puede observar, este tipo de condiciones son comunes al momento de especificar requisitos en casos industriales; sin embargo, la plantilla de Rupp no facilita, explícitamente, su especificación.

4.4. Condicionalidad inapropiada para el requisito

En los requisitos los comportamientos se encuentran condicionados por diferentes factores, que implican interpretaciones diferentes dependiendo de estas condiciones. Por ejemplo, un requisito que especifica “Mientras el control de temperatura esté encendido, el sistema debe balancear la temperatura del ambiente” puede tener una interpretación diferente a la del requisito que especifica “Si el control de temperatura está encendido, el sistema debe balancear la temperatura del ambiente” y también ambos pueden diferenciarse de un requisito que especifique “Tan pronto como el control de temperatura sea encendido, el sistema debe balancear la temperatura del ambiente”. En los tres casos, aunque se utiliza una condición similar, la interpretación es diferente. En la plantilla de Rupp se emplean únicamente dos tipos de condicionales, los cuales son: los condicionales lógicos y los condicionales temporales [3]. Sin embargo, existen otros tipos de condiciones que no están cubiertos por la planilla Rupp, por ejemplo, comportamientos que son desencadenados por eventos y comportamientos que se llevan a cabo mientras el sistema se encuentra en un estado determinado.

4.5. Falta de verificabilidad de los requisitos no funcionales

La plantilla de Rupp fue creada con el objetivo de especificar requisitos funcionales [3]; sin embargo, no fue diseñada propiamente para la especificación de requisitos no funcionales (requisitos de calidad y restricciones). Por ejemplo, estos dos requisitos de calidad “El sistema deberá estar disponible para los usuarios 7x24x364” y “El rendimiento del sistema debe ser óptimo, procurando dar respuesta a los usuarios en menos de 2 segundos” tienen un factor medible y finito para determinar que el requisito satisfará o no la necesidad de los interesados. Sin embargo, la plantilla de Rupp no tiene una estructura explícita para la escritura adecuada de factores medibles y finitos de satisfacción de requisitos no funcionales.

4.6. Falta de referencia a sistemas o dispositivos externos

En caso de que el tipo de actividad del sistema sea un requisito de interfaz, la estructura sintáctica de la plantilla de Rupp no hace referencia explícita a sistemas o dispositivos externos. Por ejemplo, los requisitos “El subsistema de punto de venta debe ser capaz de leer códigos de barras en etiquetas de artículos” y “El sistema deberá ser capaz de obtener la información de un cliente” siguen la estructura sintáctica que propone la plantilla de Rupp; sin embargo, en ninguno de esos requisitos se menciona el nombre del sistema o dispositivo con el cual se intercambia información, ni se establece si la información va desde o hacia el dispositivo/sistema.

4.7. Falta de conceptos para escribir requisitos de líneas o familias de productos

En algunos casos, los requisitos no hacen referencia a un producto sino a varios productos de una misma familia [16]. Las líneas de productos se basan en el concepto de la gestión de la variabilidad para especificar, diseñar y desarrollar de manera intensiva los productos de una misma familia de manera prescrita. Aunque la plantilla de Rupp puede ser usada para especificar requisitos, con diferentes niveles de prioridad, de líneas de productos únicamente cambiando en la plantilla “El sistema” por “La línea de productos”, no es posible definir con ella la variabilidad de los requisitos que pueden cambiar de un producto de la línea a otro. Por ejemplo, en los requisitos “La línea de productos de tiendas virtuales deberá calcular el valor del IVA de cada compra” o “La línea de productos de tiendas virtuales podría calcular el valor del IVA de cada compra” se especifican dos niveles de prioridad, pero no se considera la variabilidad de los requisitos, es decir, no se dice si es para todos los productos de la línea (obligatorio para todos los productos de la línea) o solo para algunos (opcional). Además, restricciones típicas en una línea de productos como por ejemplo “El usuario administrador de cada tienda virtual debe estar validado antes de poder gestionar los productos de la tienda virtual correspondiente” y “Los productos de la línea de tiendas virtuales que tengan la función A no pueden tener la función B”, tampoco pueden ser especificadas con la plantilla de Rupp.

5. Propuesta de una nueva plantilla de especificación de requisitos

Considerando cada uno de los problemas encontrados durante la ejecución del método investigación-acción y ejemplificados en la Sección 4, los autores han mejorado la plantilla de Rupp [3] inspirándose en conceptos de otros trabajos relacionados encontrados en la literatura (p.ej., la plantilla EARS [4]). La nueva plantilla, ver Figura 3, está compuesta de ocho espacios y cada uno se estructuró pensando en una especificación sintáctica simple y a la vez robusta para abarcar la mayor cantidad de tipos de requisitos en varios tipos de sistemas. Los rectángulos en color *amarillo* representan condicionales; los rectángulos color *gris* se usan para representar la familia de sistemas, el sistema o una parte de éste; los rectángulos de color *naranja* representan el grado de obligación; los de color *verde* son las actividades que caracterizan el sistema; los rectángulos color *azul* representan los objetos (sustantivos), con sus respectivas cantidades y complementos; y el rectángulo de color *purpura* describe el criterio medible de verificación del requisito. Este último es de carácter opcional y por ese la unión con el resto de la plantilla se hace a través de una línea punteada.

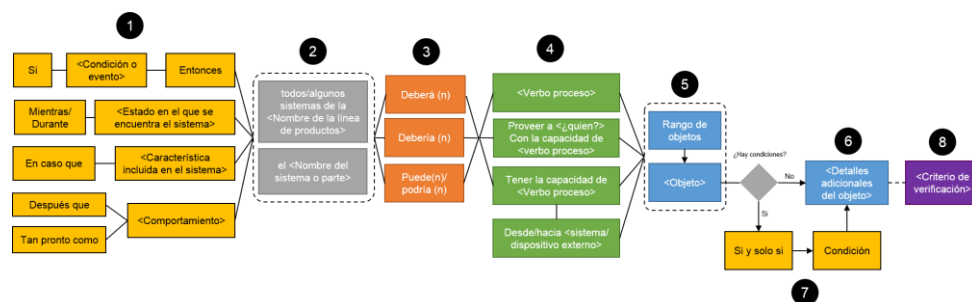


Figura 2. Nueva plantilla propuesta

A continuación, describiremos cada uno de los componentes de la plantilla resultante al final de los tres ciclos de investigación-acción en el caso del sistema PeopleQA de la empresa SQA S.A.

5.1. (1) Condiciones bajo las cuales se presenta un comportamiento

Algunos requisitos no describen comportamientos continuos sino comportamientos que se realizan o proporcionan solo bajo ciertas condiciones; por ejemplo, lógicas o temporales, como lo mostramos a continuación.

- a. Requisitos con condiciones lógicas. Se utilizan para describir comportamientos que se desencadenan únicamente cuando se cumple una condición lógica [3] o cuando ocurre un evento inesperado [4]. La forma de esta especificación es: **SI** <condición o evento> **ENTONCES**, [(todos/ algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>)] deberá(n)/debería(n)/podría(n).
- b. Requisitos guiados por estado. Se utilizan para describir un comportamiento que se debe dar en el sistema mientras el sistema se encuentra en un estado específico. Esta condición fue propuesta por [4]. La forma de esta especificación es: **MIENTRAS (o DURANTE)** <Estado de activación> [(todos/ algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>)] deberá(n)/debería(n)/podría(n).
- c. Requisitos con característica opcional. Se utilizan para describir un comportamiento que se debe dar únicamente si se incluye una característica particular [4]. La forma de esta es especificación es: **EN CASO QUE** <la característica incluida>, [(todos/ algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>)] debe(n)/debería(n)/podría(n).
- d. Requisitos con condiciones temporales. Se utilizan para describir un comportamiento que se debe dar después que ocurre otro comportamiento y se da de forma secuencial, es decir el comportamiento A se efectúa después de B. Esta condición fue propuesta por [3]. La forma de esta especificación es:

DESPUÉS QUE (o TAN PRONTO COMO) <comportamiento>, [(todos/ algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>)] debe/debería/podría.

La palabra "**DESPUÉS QUE**" significa que el sistema debe haber terminado un comportamiento en ejecución antes de iniciar otro comportamiento. La palabra "**TAN PRONTO COMO**" significa que el sistema no necesariamente tiene que haber terminado un comportamiento en ejecución antes de iniciar otro comportamiento.

Los requisitos guiados por estado, requisitos con característica opcional fueron tomados de EARS [4]; los requisitos con condiciones lógicas y temporales fueron tomados de la plantilla de Rupp [3].

A continuación, presentamos un ejemplo de requisito con condicionales, utilizando la plantilla mejorada:

“Mientras no se haya confirmado el pago de una factura de un cliente, el subsistema deberá enviar un mensaje de texto diario al número celular que tenga registrado el cliente”.

5.2. (2) Familia de sistemas, sistema o parte del sistema

Este espacio en la plantilla está reservado para el nombre de la línea de productos, o del sistema, subsistema o componente del sistema. En el caso de que se trate de un requisito de una línea de productos, se debe especificar si el requisito es válido para todos o solo para

algunos sistemas. La estructura de esta segunda parte de la nueva plantilla es la siguiente: (todos/algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>). El símbolo | significa que en un mismo requisito se debe usar sólo una de las dos partes entre paréntesis.

Completamos la segunda parte de la plantilla de Rupp con la posibilidad de especificar requisitos de líneas de productos puesto que dicha plantilla no se adaptaba correctamente para poderlos escribir en LN semi-estructurado.

A continuación, un ejemplo de requisito para líneas de productos de software, utilizando la plantilla mejorada:

“Si la aplicación de automatización es de tipo web, algunos sistemas de la línea de productos Automatización de pruebas deberán proveer al tester con la habilidad de seleccionar el tipo de navegador donde se ejecutará la prueba (ya sea Chrome, Firefox o Safari)”.

5.3. (3) El grado de prioridad

En la plantilla de Rupp, este espacio tradicionalmente está reservado para especificar el grado de obligatoriedad del requisito; sin embargo, el concepto de “obligatoriedad” lo hemos tenido que cambiar por el de “prioridad” para no confundirlo con el concepto de “obligatoriedad” de las líneas de productos. El concepto de “obligatoriedad” usado para especificar ciertos requisitos de las líneas de productos como lo presentamos en el componente 2 de la nueva plantilla. La prioridad de los requisitos hemos usado la técnica MoSCoW [17], en la cual se establecen tres grados de prioridad: esenciales, recomendados y deseables.

- a. Requisitos esenciales. Son requisitos que se deben implementar para lograr el éxito del producto o de la línea de productos. Se utiliza la palabra “deberá(n)”.
- b. Requisitos recomendados; es decir, que son importantes, pero no necesarios para alcanzar el éxito del producto o de la línea de productos. Se emplea la palabra “debería(n)”.
- c. Requisitos deseables (pero no necesarios) porque podrían mejorar la experiencia del usuario y la satisfacción del cliente. Se emplea la palabra “puede(n)” o “podría(n)”. El plural se usa para requisitos de líneas de productos.

5.4. (4) La actividad

El cuarto espacio, el mismo de la plantilla de Rupp, especifica la caracterización de la actividad que es conducida por el sistema o por los sistemas de la línea correspondiente. Existen tres tipos de actividades que se pueden llevar a cabo:

- a. Actividad autónoma. En esta actividad el(los) sistema(s) inicia(n) y ejecuta(n) el comportamiento de forma autónoma, no hay ningún usuario involucrado. La forma de esta este tipo de actividad es: ...[(todos los/algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>)] deberá(n)/debería(n) /podría(n) <Verbo de proceso>
- b. Interacción de usuario. En esta actividad el (los) sistema(s) provee(n) a un usuario con la capacidad o habilidad de usar cierto comportamiento que es iniciado o estimulado por un usuario (actor) que interactúa con el sistema. La forma de este tipo de actividad es: ...[(todos los/algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>)] deberá(n)/debería(n)/podría(n) Proveer a <¿quién?> con la capacidad de <verbo proceso>. Donde “quién” es el actor o usuario que debe tener la capacidad de usar la funcionalidad. Dicho usuario debe estar caracterizado correctamente y no incurrir en el uso indebido de sustantivos sin índice de referencia [3]; es decir, indicar “el usuario” sería un error que conduciría a una ambigüedad en la especificación.

- c. Requisito de interfaz. En esta actividad, el sistema lleva a cabo un comportamiento con dependencia de otra entidad (que puede ser otro sistema o un dispositivo físico). Este espacio fue mejorado en la nueva plantilla adicionando explícitamente el nombre de la entidad externa con la cual interactúa el sistema y la dirección de la relación. La forma de este tipo de actividad es: ...[(todos los/algunos sistemas de la <Nombre de la línea de productos>)|(el <Nombre del sistema o parte>)] debe(n)/ debería(n)/podría(n) Tener la capacidad de <Verbo proceso>. Además, esta estructura se completa con la entidad con la cual interactúa el sistema:
- Si el comportamiento es ejecutado por el sistema externo que transmite datos a la interfaz del sistema receptor, entonces se complementará la especificación agregando: **Desde** <nombre del sistema o dispositivo externo>.
 - Si el comportamiento es efectuado por el sistema e interactúa o afecta a otro sistema o dispositivo externo entonces se complementará la especificación agregando: **Hacia** <nombre del sistema o dispositivo externo>

5.5. (5) El objeto o los objetos

Este espacio está reservado para el objeto o los objetos que componen al sistema. En la nueva plantilla hemos incorporado el concepto de “rango”, puesto que los objetos se pueden afectar en distintos rangos. Los rangos en la nueva plantilla se especifican de la siguiente manera:

- a. Un único objeto: Un (uno, una) <objeto>
- b. Un objeto específico: el (la) <objeto>
- c. Cada objeto de un conjunto: cada <Objeto>
- d. Varios objetos: X <objetos>, siendo X el número de objetos
- e. Rango de objetos: Entre A y B <objetos>, siendo A el rango inferior y B el rango superior de objetos
- f. Todos los objetos de un conjunto: Todos los (Todas las) <objetos>.

A continuación, un ejemplo de requisito con rango de objetos, utilizando la plantilla mejorada:

“El subsistema de inventario debe proveer al administrador de inventario con la capacidad de asociar a una caja registradora entre 1 y 3 pistolas lectoras de código de barras”.

5.6. (6) Los detalles complementarios al objeto

En el sexto espacio de la plantilla se especifican de forma opcional los detalles complementarios del objeto, que pueden ser uno o varios adjetivos, así como una descripción más enriquecida sobre el objeto, sin el riesgo de llegar a alterar el significado propio de la especificación del requisito y centrándose únicamente en describir los detalles relacionados al objeto en cuestión. Este espacio de la plantilla se conservó de la plantilla de Rupp y no fue modificado por los autores.

5.7. (7) Condicionalidad en el objeto

Es posible que en algunos casos el comportamiento del requisito sea condicionado por el estado en el cual se encuentre un objeto. En la nueva plantilla hemos reservado el séptimo espacio para especificar un comportamiento que el sistema debe llevar cabo si y solo si, el objeto cumple cierta condición, en este caso el requisito se completa adicionando la siguiente expresión: **SI Y SOLO SI** <condición>. Es importante aclarar que esta condición es opcional y solo se da de forma explícita si el objeto puntual del requisito

requiere especificar la condición, por lo tanto, no es obligatoria en la especificación del requisito.

A continuación, presentamos un ejemplo de requisito con condicionalidad en el objeto, utilizando la plantilla mejorada:

“El subsistema de inventarios podría proveer al administrador de bodega con la capacidad de eliminar una orden de compra, si y solo si, la orden de compra no ha sido despachada”.

5.8. (8) El criterio de verificación (ajuste) del requisito

En algunos tipos de requisitos, especialmente en los requisitos no funcionales, es necesario establecer el grado en el cual el requisito se debe alcanzar. Robertson & Robertson [18] recomiendan incluir criterios de ajuste; es decir, incluir "una cuantificación del requisito que demuestre el estándar que debe alcanzar el producto" como parte de la especificación de cada requisito, funcional y no funcional. Los criterios de ajuste describen una forma medible de evaluar si cada requisito se ha logrado satisfacer correctamente.

Para tal fin, en la nueva plantilla, hemos agregado en el último espacio la opción de establecer un criterio medible u observable para determinar el grado de verificabilidad del requisito, esto con el fin de asegurar que el requisito se puede comprobar ya sea por una persona o una máquina. Este criterio se define a discreción del autor del requisito y es de carácter opcional, aunque se recomienda emplearlo siempre en los requisitos de calidad.

A continuación, un ejemplo de requisito no funcional, utilizando la plantilla mejorada:

“Si ocurre una falla que produce que el sistema se detenga, el sistema Oktopus deberá reiniciar todos los sensores en un tiempo inferior a 20 segundos”.

6. Evaluación preliminar de la plantilla propuesta

Para la evaluación preliminar de la plantilla propuesta en este artículo se establecieron 2 grupos; cada uno compuesto por un analista de negocios con experiencias similares (esto es, mismo número de años en la empresa y participación en proyectos comparables en tema, duración y tamaño) en ingeniería de requisitos y con un entrenamiento previo del uso de la plantilla Rupp; y un revisor técnico de requisitos. El analista de cada grupo debía especificar los requisitos en la plantilla Rupp (para el caso del primer grupo) o en la plantilla mejorada (para el caso del segundo grupo) y el segundo rol inspeccionaba el trabajo del primero.

Como se mencionó en la Sección 3, se hicieron 3 ciclos para esta investigación. En el primer ciclo se especificaron los requisitos en prosa tal como los manifestaron los interesados. De esta actividad fueron especificados 46 requisitos. En el segundo ciclo, el primer grupo reescribió los requisitos utilizando para ello la plantilla Rupp y con un acompañamiento de los autores para apoyarlos en la resolución de dudas. Para este ciclo se obtuvieron 138 requisitos. Una vez concluida la especificación, el revisor técnico del primer grupo identificó 48 requisitos con problemas de especificación que no se adhieren al estándar propuesto por Rupp es decir que el 34.8% de los requisitos tuvieron problemas de adherencia con la plantilla de Rupp. Estos problemas fueron categorizados según siete tipos de criterios como se muestra en la Tabla 1 (para una explicación más detallada sobre cada uno de estos tipos de problemas y algunos ejemplos que se puedan presentar en una especificación de requisitos basada en la plantilla Rupp ver la Sección 4).

Problema identificado	Cantidad de requisitos usando la plantilla de Rupp	Cantidad de requisitos usando la plantilla propuesta
Condicionalidad inapropiada	6	0
Falta de referencia a sistemas o dispositivos externos	4	0
Omisión de bicondicionales	7	0

Omisión de cantidades y rangos	18	0
Falta de verificabilidad de requisitos No funcionales	3	0
Motivos Faltantes	3	0
Otros	7	3

Tabla 1 problemas identificados en la especificación de requisitos con las plantillas comparadas

Para el tercer y último ciclo, el segundo grupo se enfocó en especificar los mismos requisitos del ciclo número 2 utilizando la plantilla mejorada en este ciclo. Los resultados fueron interesantes. En este ciclo fue posible especificar satisfactoriamente el 98% de los requisitos utilizando la plantilla propuesta en este artículo. Solo 3 requisitos no lograron ser especificados completamente utilizando la plantilla mejorada, pero es de resaltar que 135 lograron una especificación que se adhiere a la plantilla mejorada, sin ningún tipo de observación por parte del revisor técnico.

El factor principal por el cual algunos requisitos continúan con algunos problemas cuando se utiliza la plantilla mejorada es porque se detectó durante el ciclo 3 que algunos requisitos cuando tienen un comportamiento restrictivo, es decir, cuando el requisito especifica lo que no debe hacer el sistema, en lugar de lo que debe hacer, la plantilla no se adhiere apropiadamente. De acuerdo con [1] este tipo de requisitos se conocen como **requisitos negativos** y serán parte de una investigación posterior sobre como especificar este tipo de requisitos.

La plantilla mejorada ha sido implementada para ser usada en la herramienta VariaMos⁵ con el fin de facilitar la escritura de requisitos de dominio, de aplicaciones y de (auto) adaptación (para, por ejemplo, sistemas ciberfísicos auto-adaptables).

7. Trabajos relacionados

Aparte de la plantilla de Rupp hay otras propuestas de plantillas para la especificación de requisitos en LN semi-estructurado. Por ejemplo, EARS (*Easy Approach to Requirements Syntax*) [4] es una plantilla que considera varios patrones condicionales a partir de los cuales se tipifican varios requisitos. Por ejemplo, **requisitos ubicuos** (no tienen una condición previa que desencadenen comportamientos, pero siempre está activo), **requisitos guiados por eventos** (describen un comportamiento que se da en el sistema cuando se desencadena un evento) y **guiados por estados** (describen un comportamiento que se encuentra activo mientras el sistema está en un estado definido). La plantilla EARS se enfoca en una sintaxis orientada a patrones condicionales bajo los cuales se presentan los requisitos; adicionalmente la investigación en EARS se centra en la industria aeronáutica. A diferencia de ésta, nuestra propuesta pretende ser independiente del dominio de aplicación (ver Sección 4) y pretende soportar la escritura de requisitos no sólo funcionales sino también no funcionales y restricciones.

Alexander & Stevens [19] proponen una plantilla para escribir requisitos funcionales desde la perspectiva del usuario; ya que es más natural formular los requisitos en términos de la acción de un usuario, no desde la perspectiva del sistema [1]. La estructura de la plantilla es la siguiente:

El [Tipo de usuario o Nombre de actor] tiene la capacidad de [Verbo de proceso] [para algún objeto] [Criterio de medición o calificación, tiempo de respuesta o declaración de calidad]

A diferencia del trabajo propuesto por Alexander & Stevens, nuestro enfoque está orientado a la especificación de los requisitos desde la perspectiva del comportamiento de los sistemas y no de las necesidades propias de los interesados.

⁵ <https://variamos.com/variamosweb>

Adv-EARS [11] [12] (*Advanced EARS*) propone una sintaxis en lenguaje semi-estructurado para especificar requisitos funcionales, de manera tal que se da un soporte automatizado para la derivación de casos de uso y actores en modelos de casos de uso. A diferencia de Adv-EARS nuestra plantilla se enfoca en requisitos funcionales y no funcionales, mientras que Adv-EARS se enfoca únicamente en requisitos funcionales. Esta sintaxis es una versión avanzada de [EARS], por lo cual se podría incorporar a futuro algunos elementos de Adv-EARS en nuestro trabajo.

Algunos artículos complementan nuestro trabajo; por ejemplo, Tjong et al. [20] y Denger et al. [21] presentan dos propuestas para reducir la ambigüedad de los requisitos por medio de patrones y reglas lingüísticas; aunque parte de nuestro trabajo también es reducir la ambigüedad del LN, nuestro trabajo se enfoca en mejorar las plantillas de especificación de requisitos, partiendo de una plantilla estándar. Arora et al. [22], [23] proveen una visión adicional al apoyar la conformidad automática de los requisitos usando técnicas de procesamiento de LN para la comprobación de requisitos, algo que nuestro trabajo no plantea y que será parte de un trabajo posterior sobre la comprobación automática de los requisitos. Además, [22] presenta también una plantilla flexible para especificar requisitos que se puede adaptar a diferentes estilos de escritura de requisitos y otras propuestas como la de Souag et al. [24] van aún más lejos permitiendo la generación automática de requisitos no funcionales (de seguridad en particular) en LN semi-estructurado gracias al uso de una ontología de seguridad [25] y a una propia a cada dominio. En contraste, nuestra propuesta es agnóstico al tipo de requisitos no funcionales, pero deberá a futuro inspirarse de los trabajos relacionados y de las ontologías de requisitos y dominios, para facilitar la escritura de requisitos.

8. Conclusiones y Trabajos Futuros

La plantilla de Rupp ha sido establecida como un estándar de facto para la especificación de requisitos individuales por parte del IREB, sin embargo, al momento de representar ciertos requisitos a nivel industrial esta plantilla se ve limitada. Es por eso que decidimos estudiar la pregunta de investigación relacionada con (i) los vacíos de la plantilla usada como estándar para la escritura de requisitos y (ii) cómo llenar esos vacíos. Este estudio lo fundamentamos en el método experimental de investigación llamado investigación-acción. Este método nos permitió usar de manera coherente la experiencia propia de los autores en el ámbito de la ingeniería de requisitos, y la información disponible a nivel industrial y en la literatura (otras plantillas y trabajos relacionados). Como resultado de este experimento logramos identificar los vacíos de la plantilla de Rupp y basados en esos vacíos propusimos una plantilla más robusta que, a diferencia de la otra, permitió representar en LN semi-estructurado la cuasi-totalidad de los requisitos y las restricciones del caso industrial.

A través de esta investigación pudimos observar que la plantilla de referencia debe ser mejorada y que es posible mejorarla. También constatamos que la nueva plantilla puede ser utilizada en casos industriales; sin embargo, nuestra investigación aún no es concluyente considerando que solo experimentamos con un solo caso industrial.

Algunos aspectos que quedan pendientes y requieren aún más trabajo, tanto teórico como experimental, son por ejemplo: recomenzar el experimento partiendo de una plantilla diferente a la de Rupp y comparar la plantilla resultante con la reportada en este artículo; e implementar una herramienta de software que permita la comprobación automática de requisitos con base en la plantilla mejorada resultado de este trabajo de investigación, ya que otras herramientas se basan en la plantilla de Rupp y otras plantillas para realizar esta comprobación automática. También es necesario estudiar la plantilla mejorada en otros casos y en otros tipos de proyectos, p. ej., para sistemas complejos, sistemas pervasivos, líneas de productos dinámicos, sistemas autoadaptables, y sistemas inteligentes y de analítica. Adicionalmente se podrían definir patrones de lenguaje natural, verbos de proceso estandarizados y modelos que complementen la plantilla mejorada.

Adicional a lo anterior esperamos fortalecer la evidencia empírica respecto a las ventajas que tiene la plantilla mejorada, realizando por lo menos otros dos experimentos en empresas de sectores

de actividad diferentes con el fin de tener observaciones más concluyentes sobre la facilidad de uso, la completitud y la precisión de la plantilla propuesta en otros contextos. Otro trabajo futuro consiste en complementar la plantilla propuesta en este artículo para el tratamiento de requisitos negativos, ubicuos, entre otros.

Referencias

- [1] Wiegers K., Beatty J. *Software Requirements*, Third Edition. Microsoft Press, 2013.
- [2] Pohl K. *Requirements Engineering - Fundamentals, Principles, and Techniques* Springer, 2010.
- [3] Rupp C. *Requirements-Engineering und -Management* Hanser, 2007.
- [4] Mavin A., Wilkinson P., Harwood A., Novak M. EARS: Easy Approach to Requirements Syntax. *Requirements Engineering Conference*, 2009.
- [5] Rupp C. *Requirements Templates - The Blueprint of your Requirement*. SOPHIST GmbH, 2014.
- [6] O'Brien R. An Overview of the Methodological Approach of Action Research. In Roberto Richardson (Ed.), *Theory and Practice of Action Research*. Brazil: Universidade Federal da Paraíba, 2001.
- [7] Wieringa R., Morali A. Technical action research as a validation method in information systems design science. *Design science research in information systems. Advances in theory and practice*, 7286, 220-238. Springer Berlin Heidelberg, 2012.
- [8] Susman G.I., Evered R.D. An Assessment of the Scientific Merits of Action Research. *Administrative Science Quarterly*: Vol. 23 (pp. 582-603), 1978.
- [9] Koshy E., Koshy V., Waterman H. *Action research in healthcare*, Thousand Oaks, CA: Sage, 2011.
- [10] Susman G. I. *Action Research: A Sociotechnical Systems Perspective*, Ed. G. Morgan. London: Sage Publications, 1983
- [11] Majumdar D., Sengupta A., Kanjilal A., Bhattacharya S., Adv-EARS: A Formal Requirements Syntax for Derivation of Use Case Models, *First International Conference on Advances in Computing and Information Technology*, pp 40-48, 2011.
- [12] Dipankar Majumdar, Sabnam Sengupta, Ananya Kanjilal, Swapan Bhattacharya, Automated Requirements Modeling with Adv-EARS. *International Journal of Information Technology Convergence and Services*, pp 57-67, 2011.
- [13] ISO/IEC/IEEE 29148, *Systems and software engineering — Life cycle processes — Requirements engineering*, 2011.
- [14] Cohn M. *User Stories Applied for Agile Software Development*, Addison Wesley, 2004.
- [15] Beck K. Embracing change with extreme programming. *IEEE Computer*. 32 (10): 70–77, 1999
- [16] Mazo R. *Guía para la adopción industrial de líneas de productos de software*. Editorial Eafit, Medellín-Colombia, 2018.
- [17] Clegg D., Barker R. *Case Method Fast-Track: A Rad Approach*, Addison-Wesley, 1994.
- [18] Robertson S., Robertson J. *Mastering the Requirements Process: Getting Requirements Right*, 3rd ed. Addison-Wesley, 2013.
- [19] Alexander I., Stevens R. *Writing Better Requirements*, Addison-Wesley, 2002.
- [20] Tjong S. F., Hallam N., Hartley M. Improving the quality of natural language requirements specifications through natural language requirements patterns. In *CIT'06*, pp 199-205, 2006.
- [21] Denger C., Berry D., Kamsties E. Higher quality requirements specifications through natural language patterns. In *SwSTE'03*, pp. 80-90, 2003.
- [22] Arora C., Sabetzadeh M., Briand L. C., Zimmer F., Gnaga R. Automatic checking of conformance to requirement boilerplates via text chunking: An industrial case study, *ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 35-44, 2013.
- [23] Arora C., Sabetzadeh M., Briand L., Zimmer F. Automated checking of conformance to requirements templates using natural language processing, *TSE*, vol. 41, no. 10, pp. 944-968, 2015.
- [24] Souag A, Mazo R, Salinesi C, Comyn-Wattiau I. Using the AMAN-DA method to generate security requirements: a case study in the maritime domain. *Requirements Engineering*. 2018.
- [25] Souag A, Salinesi C, Mazo R, Comyn-Wattiau I. A security ontology for security requirements elicitation. In *International symposium on engineering secure software and systems* (pp. 157-177). Springer, Cham, 2015.