



Universidad Politécnica de Valencia

DSIC

TESIS DE MÁSTER

**Obtención de conocimiento empírico
sobre la calidad de Casos de Uso**

Autor: José Antonio Ferrando Moragues

Directores: Óscar Pastor López, Nelly Condori-Fernández

Valencia, Septiembre de 2010

Resumen

El presente trabajo describe un estudio secundario empírico que fue diseñado para identificar qué aspectos de los Casos de Uso son empíricamente evaluados, en qué formato de descripción y en qué dominio son experimentados. Los Casos de Uso son una técnica para especificar requisitos con el principal objetivo de minimizar los errores que suelen tener lugar en las primeras etapas del desarrollo de un sistema software. Esta técnica es normalmente investigada en la Ingeniería de Requisitos para identificar las necesidades de los clientes documentándola de forma que es posible analizar, comunicar e implementar.

En dicho ámbito se lleva a cabo la presente revisión sistemática destinada a sintetizar los conocimientos existentes en la literatura para la mejora de la calidad del proceso de captación de requisitos. Una revisión sistemática establece una respuesta a una pregunta de investigación recogiendo y agregando la información disponible en los estudios publicados junto con los trabajos más destacados realizados hasta el momento. A partir de un protocolo de revisión se llevó a cabo el proceso de revisión sistemática, el cual aportó como resultado 10 estudios primarios que fueron categorizados de acuerdo a los atributos de calidad de especificación de requisitos de software propuestos por Davis [15], y al marco de calidad de Krogstie [16].

En base a esta clasificación realizada, la comprensibilidad (calidad pragmática) y la completitud (calidad semántica) son los aspectos de los Casos de Uso más estudiados en contextos experimentales.

Además, una clasificación por el dominio del problema abordado, y formato de especificación utilizado, fueron también tomadas en cuenta. Los escenarios son el formato de descripción más comúnmente utilizado y que los Sistemas de Gestión de Información son el principal dominio donde los experimentos se llevan a cabo.

De acuerdo a unos criterios de calidad previamente establecidos, los estudios primarios seleccionados y categorizados fueron también evaluados con el fin de

identificar los estudios más relevantes para nuestra investigación, en la que se obtuvieron 6 estudios relevantes.

La investigación en especificación de requisitos no termina aquí, - después de identificar un estudio de replicación y el estudio replicado correspondiente en nuestra selección - se necesitan más replicaciones de estudios empíricos para garantizar las evidencias, más acciones técnicas de investigación para entender los problemas del uso de las técnicas de Casos de Uso y emprender nuevos estudios en otras técnicas de especificación de requisitos para finalmente entender en qué contextos es más adecuado utilizar cada técnica.

CONTENIDO

Capítulo 1	6
Introducción	6
1.1 Descripción del Problema	6
1.2 Motivación	7
1.3 Objetivos	9
1.4 Contexto de la investigación	10
1.5 Estructura de la Tesis de Máster	12
Capítulo 2	13
Ingeniería de Requisitos y las Revisiones Sistemáticas	13
2.1 Ingeniería de Requisitos	13
2.1.1 Tareas de la Ingeniería de Requisitos	14
2.1.1.1 La elicitación	15
2.1.1.2 El modelado	16
2.1.2 Direcciones de investigación	17
2.2 Revisiones Sistemáticas	20
2.3 Revisiones Sistemáticas en la Ingeniería de Software	25
2.4 Revisiones Sistemáticas en la Ingeniería de Requisitos	30
Capítulo 3	34
Calidad de la Especificación de Requisitos de Software	34
3.1 Estándares y marcos de calidad	34
3.2 Aproximaciones para especificar Casos de Uso de calidad	42
Capítulo 4	44
Protocolo de Revisión	44
4.1 Preguntas de investigación	44
4.2 Estructura del protocolo de revisión	46
4.2.1 Estrategia de búsqueda	46
4.2.2 Proceso de selección de estudio	50
4.2.3 Evaluación de la calidad del estudio	51

4.2.4 Extracción de información	51
4.2.5 Síntesis de la información.....	52
4.3 Amenazas de la validez del diseño de la Revisión Sistemática.....	52
Capítulo 5.....	54
Conducción de la Revisión Sistemática	54
5.1 Selección de los estudios.....	54
5.2 Extracción de información.....	62
5.3 Clasificación de los estudios seleccionados	69
5.3.1 Estudios de toda la selección.....	69
5.3.2 Estudios de la selección en Casos de Uso	79
5.4 Evaluación de la calidad de los estudios seleccionados	87
5.5 Síntesis de la información	91
5.5.1 Estudios considerados	91
5.5.2 Tareas y sujetos	91
5.5.3 Heurísticas y esquemas de marcado.....	93
5.5.4 Hipótesis y validaciones	95
5.5.5 Conclusión	100
Capítulo 6.....	103
Conclusiones y trabajos futuros.....	103
Agradecimientos	106
Referencias.....	107
Lista de Tablas	119
Lista de Figuras.....	121

Capítulo 1

Introducción

1.1 Descripción del Problema

Aunque sabemos que, en teoría, el éxito de un sistema software depende de cómo se ajusta a las necesidades de los usuarios [13], en la práctica, los errores ocurridos en las primeras fases continúan siendo la mayor causa de los fallos del software [18], es decir, al término del producto no hay satisfacción por parte del cliente. Para afrontar este problema, investigadores han propuesto numerosos métodos y técnicas de Ingeniería de Requisitos (IR) con el propósito de comprender las necesidades de los clientes y los contextos donde el software desarrollado será utilizado [25] [26] [27]. No obstante, la definición de requisitos es crítica para el éxito en el desarrollo de un sistema software [20]; concretamente los errores en la especificación causan la mayor parte de los problemas del software [41]. Según [43], aproximadamente el 66% de todos los errores en proyectos de software ocurren en las fases de especificación y diseño.

Una técnica de captación y definición de requisitos son los Casos de Uso (CU). Los CU son métodos Orientados a Objetos (OO) que forman parte del Lenguaje de Modelado Unificado (UML) y que se utilizan para representar requisitos de sistemas software. Popularizados por Ivar Jacobson en 1992 [28], hoy en día los utilizan numerosos analistas para especificar requisitos en sistemas software. Varias aproximaciones de desarrollo de software incluido el Proceso de Desarrollo de Software Unificado [43] recomiendan CU para la especificación de requisitos [44]. Gracias a la creciente popularidad de UML¹ [1] y en especial a los CU, éstos han sido

¹ Desde que UML [1] fue adoptado como el lenguaje estándar de OMG para modelados, un gran número de modelos de procesos basados en UML para desarrollos Orientados a Objetos han sido propuestos. Estas aproximaciones son normalmente dirigidas a Casos de Uso, y por tanto, capturar los requisitos funcionales de los sistemas con Casos de Uso, los cuales

aceptados e incluidos en la IR. No obstante, aunque los CU son una herramienta popular en auge para la elicitación y estructuración de requisitos, no es lo bastante conocido el cómo se pueden crear de manera correcta y completa [16], y cómo éstos a su vez, pueden ser usados para generar especificaciones de requisitos correctas y completas.

Los CU han demostrado ser útiles para describir las funcionalidades del futuro sistema, capturando los requisitos a través de la descripción de escenarios involucrando actores, flujos de trabajo (eventos), restricciones, excepciones, etc. Sin embargo, no hay un estándar para tales descripciones que pueden introducir riesgos cuando pensamos en la calidad de CU (que puede impactar en la calidad del sistema). De hecho, no hay una clara definición sobre la calidad de CU y cuáles podrían ser esas características que incrementarían/garantizarían la calidad. En conclusión, no existe una teoría común acordada en cómo construir CU y existen diferentes opiniones sobre qué es lo que constituye la calidad en los modelos de CU.

1.2 Motivación

“Un Caso de Uso es una colección de posibles secuencias de interacciones entre el sistema correspondiente y los actores externos, respecto a un objetivo en particular” [22].

El concepto de modelado de CU fue introducido por Ivar Jacobson en 1992 [1] y además introdujo una aproximación dirigida por CU para el desarrollo de software que provocó que el modelado se hiciera popular y se convirtiera en una técnica ampliamente utilizada para capturar y describir los requisitos funcionales de un sistema software.

Los CU son una conocida manera de capturar procesos de negocio y requisitos de sistemas. Han sido adoptados por la mayoría de metodologías Orientada a Objetos (OO) y por la comunidad de Ingeniería de Procesos de Negocio como una técnica para describir cómo un negocio opera, tanto para los presentes como para los futuros

proporcionan el fundamento para el resto del proceso de desarrollo: planificación, análisis, diseño y pruebas.

modos de operación. Se han utilizado en numerosos proyectos industriales [23], pero también se han generado fuertes debates sobre su utilidad² [24]. En este sentido, la investigación basada en la evidencia juega un importante rol – como veremos más adelante - para determinar si los CU han demostrado realmente ser útiles para entender las necesidades de los clientes.

En la parte técnica, las descripciones de CU (especificaciones) juegan un importante rol para la completitud, corrección, consistencia, no ambigüedad, precisión, comprensibilidad, comunicación, etc., de la especificación del sistema. Se ha dedicado a descubrir problemas sintácticos típicos que comprometen la calidad de los documentos de requisitos, tales como requisitos que han sido abandonados y ya no tienen sentido, descripciones que son demasiado largas y difíciles de leer, o información que ha sido duplicada [60]. Estos defectos dificultan la total comprensibilidad y reusabilidad de documentos de requisitos a través del proceso de desarrollo [53]. Estos problemas pueden ser minimizados con la pronta identificación de los síntomas y la eliminación de sus causas.

A todo esto, se conoce el procedimiento de crear CU a partir de una narración de texto, escenarios o esquemas de interacción, plantillas de descripción o con representaciones gráficas con diagramas. Un modelo de CU puede tener dos partes, el diagrama y las descripciones. El diagrama proporciona una vista amplia (general) de actores y CU, y sus interacciones - un actor representa un rol que el usuario puede jugar respecto al sistema y un CU representa una interacción entre un actor y el sistema. Los CU textuales detallan los requisitos a partir de plantillas o mediante narraciones de manera estructurada o no. En principio, la construcción de CU se trata de una fácil y subjetiva tarea en la que hay que describir y anotar las funcionalidades de un sistema captando los requerimientos a través de escenarios en los que aparecen actores, flujos de trabajo, eventos, restricciones, excepciones, etc. En general, los CU se suelen representar mediante diagramas que pueden soportar diferentes cualidades como interfaces, modulación y dependencias con otros CU, pero solamente con los

² De acuerdo con Moody, un método es percibido como útil si su efectividad real es demostrada y la facilidad de uso es percibida [29].

diagramas no se capturan completamente todos los escenarios que intervienen en el sistema.

A pesar de disponer de varios formatos para construir la especificación de requisitos utilizando la técnica en cuestión, parece ser que no hay una descripción clara de cómo proceder a diseñar CU de calidad y cuál es el procedimiento a seguir para aprovechar al máximo su utilidad. La razón puede ser debido a que no existe ese procedimiento que desemboque en un CU de calidad. Es hora de identificar esas cualidades que permitan detectar esta condición en los CU a partir de demostraciones basadas en evidencias.

Los CU construidos basándose en evidencias juegan un rol importante porque proporcionan los medios para que la mejor evidencia actual desde la investigación pueda ser integrada con experiencia práctica para el proceso de captación de requisitos. El núcleo de este paradigma basado en evidencias es la Revisión Sistemática³ (RS), que ha sido recibida en la Ingeniería de Software (IS) un poco más tarde que en otros campos científico-técnicos [2][3][4]; los estudios realizados sobre ERS, frecuentemente utilizados en otros campos de investigación, han sido descuidados por investigadores de IS [6];**Error! No se encuentra el origen de la referencia..**

1.3 Objetivos

A partir del estudio secundario a emprender se tiene como objetivo averiguar una serie de incógnitas a resolver para tener en cuenta qué influye y qué puede influir en la calidad de los CU. Los CU, como ya hemos visto, pueden ser representados por varios formatos de descripción y se hace interesante saber cuál es el mejor, para qué dominio y cómo construirlos de manera eficaz para desarrollar sistemas software con éxito. Nuestro objetivo en particular es la investigación en estudios primarios y sintetizar la información con el fin de conseguir mejoras en el proceso de la definición de descripciones de CU.

³ Método de investigación secundaria que tiene como objetivo recoger y analizar sistemáticamente todas las evidencias disponibles en un tópico específico en un objetivo [9].

La pregunta principal sería la siguiente:

“¿Qué se debe observar para evaluar la calidad de un Caso de Uso?”

El objetivo de este trabajo es doble: identificar las características (atributos de calidad) que actualmente están siendo consideradas por los investigadores para medir la eficacia real de los CU y verificar la existencia de *formatos de descripción*⁴ que pueden promover la calidad de CU. De esta forma el estudio secundario apunta a las siguientes preguntas de investigación:

- ¿Cuáles son los aspectos de calidad más investigados en el desarrollo de CU? ¿cuáles son las características más importantes de los CU que pueden afectar a la calidad?
- ¿Qué procedimiento habría que seguir para construir el mejor modelo de CU que especifique las necesidades del cliente?
- ¿Cómo se pueden capturar todos los escenarios de un sistema?
- ¿Existe algún formato de descripción que pueda incrementar o garantizar la calidad en especificaciones de CU y que pueda ser así para diferentes tipos de sistemas? ¿en qué dominios son investigados estos aspectos?

1.4 Contexto de la investigación

El presente trabajo forma parte de otro trabajo de investigación que tiene como objetivo verificar la repetitividad de la ejecución de las revisiones sistemáticas realizadas por diferentes centros de investigación. Un grupo de universidades, miembros de ISERN (International Software Engineering Research Network), participaron en la elaboración de un protocolo que brinde las pautas generales para llevar a cabo la presente revisión sistemática.

Este grupo de universidades son: la COPPE/UFRJ de Brasil (representada por Guilherme Horta Travassos), la Universidad Politécnica de Madrid (representada por

⁴ Los CU pueden ser representados de diferentes maneras. Clasificamos las descripciones en: narración de texto, escenarios, plantillas y diagramas.

Oscar Dieste), la Lund University de Suecia (representada por Martin Höst) y Universidad Politécnica de Valencia (representada por Óscar Pastor), centro de desarrollo del presente trabajo. Cabe señalar que han sido involucradas las áreas de investigación de Ingeniería de Requisitos, Medición y Calidad de Software, e Ingeniería de Software Empírica. Todos los participantes definieron por consenso el tópico a tratar: evaluación de la Calidad de Casos de Uso. El protocolo inicial (versión 0.1) fue preparado por Guilherme Horta Travassos y Vitor Faria Monteiro (Experimental Software Engineering Group at COPPE/UFRJ) en diciembre de 2009.

El encuentro de este grupo de investigadores viene justificado porque se ha observado la dificultad de llevar a cabo la síntesis de los estudios referentes a modelos de coste-estimación, pruebas basadas en modelos, métodos de desarrollo web, arquitectura de software, métodos de pruebas, elicitación, inspección y métodos de IR. A pesar de todos los avances en el campo, muchos de los estudios primarios en IS no están reportados de tal manera que ayuden a una síntesis posterior. Como efecto lateral, muchos de los estudios secundarios en IS se enfrentan a problemas similares cuando tratan de agregar resultados primarios. Como ejemplo, los investigadores citaron los siguientes problemas que afectan al campo de la IS:

- Pocos estudios empíricos, dispersos en cuanto a muchos dominios.
- Reportes pobres:
 - Falta de información estadística.
 - Omisión de información importante para revisar/analizar (los autores no tienen en cuenta la posibilidad de una revisión secundaria).
- Replicaciones ejecutadas de manera pobre: las aproximaciones son aplicadas de diversas maneras; las variables de respuesta pueden tener diferentes métricas, etc.
- Ausencia de métodos de evaluación de calidad.
- Las evidencias del conocimiento obtenidas de la agregación no pueden ser clasificadas como fiables.
- Las revisiones son llevadas a cabo en muchas ocasiones por estudiantes; sin embargo se requiere conocimiento experto.

- Falta de un “orden del día” para la investigación.

Por tanto, durante el ISERN 2009 se sugirió en la sesión de Agregación/Generalización de Estudios Empíricos que la interpretación de los resultados - porque los reportes normalmente carecen de información - puede estar afectada por el punto de vista aplicado por los investigadores analizando la información extraída.

1.5 Estructura de la Tesis de Máster

El presente trabajo, objeto de Tesis de Máster, muestra los resultados de un estudio secundario realizado en el campo de la Ingeniería de Software, concretamente en el ámbito de la Ingeniería de Requisitos, con el objetivo de averiguar y sintetizar los conocimientos existentes en la literatura para evaluar la calidad de los Casos de Uso. El trabajo está estructurado de la siguiente manera:

- Capítulo 2: se determina los conceptos más importantes de la Ingeniería de Requisitos y las Revisiones Sistemáticas, conceptos base para llevar a cabo el estudio secundario.
- Capítulo 3: se define el concepto de Especificación de Requisitos de Software (ERS) y se comenta brevemente cómo se identifica la calidad. Además, se presentan algunas aproximaciones existentes en la literatura para los CU en particular.
- Capítulo 4: se presenta el protocolo de revisión utilizado como método de investigación. Se describe el proceso de Revisión Sistemática llevado a cabo.
- Capítulo 5: se presentan los resultados de la Revisión Sistemática, donde aparece la información relevante clasificada, evaluada y sintetizada.
- Capítulo 6: por último las conclusiones de la investigación así como las futuras líneas de investigación y trabajos futuros recomendados.

Capítulo 2

Ingeniería de Requisitos y las Revisiones Sistemáticas

El presente capítulo introduce la disciplina de IR, los distintos tipos de investigación que se han venido llevando a cabo en el pasado, así como las actuales necesidades y desafíos de esta línea de investigación. Además de esto, las revisiones sistemáticas se presentan en detalle, junto con su aplicación en el ámbito de la IR. Finalmente, se discuten las conclusiones de este capítulo.

2.1 Ingeniería de Requisitos

Durante los años 90 se observaron cambios importantes y radicales en la comprensibilidad de IR. En los comienzos de los 90, la IR había emergido como un campo de estudio en dos series de encuentros internacionales – el patrocinado IEEE en conferencia y simposio – y el establecimiento de una revista internacional publicada por Springer. A finales de los 90, el campo creció lo suficientemente para afianzarse en un gran número de encuentros y “workshops” adicionales en varios países.

En términos generales, la IR es el proceso de descubrir ese propósito identificando las necesidades de los clientes y usuarios, y documentándolo de forma que es posible analizar, comunicar e implementar. Existe un número de dificultades inherentes en este proceso. Los clientes pueden ser numerosos y distribuidos, los requisitos pueden variar y generar conflictos, y finalmente el producto puede resultar de un conjunto de funcionalidades no exactamente requeridas en un principio. Los requisitos software comprenden estas necesidades y la IR es el proceso por el cual los requisitos son determinados. Dicho de otra manera, la IR trata de definir de forma precisa el

problema que el software debe tratar de solventar, mientras otras actividades de la IS tratan de definir y refinar una solución software propuesta. La IR puede ser definida como sigue:

“la rama de la Ingeniería de Software ocupada con los objetivos del mundo real para llevar a cabo funciones y restricciones de sistemas software. También se ocupa de las relaciones entre estos factores para precisar las especificaciones de la conducta del software y sus evoluciones sobre el tiempo”.

La IR se ocupa de interpretar y comprender la terminología usada por el cliente, los conceptos, los puntos de vista y los objetivos. Para ello es importante seleccionar una técnica de modelado adecuada, porque la elección de la técnica afecta al conjunto de fenómenos que pueden ser modelados y pueden restringir al cliente aspectos que un ingeniero de requisitos es capaz de observar. Muchos sistemas entregados que no satisfacen a los clientes derivan al menos en cierta medida a un inefectivo IR. Esto se debe básicamente a que el contexto, en el que la IR toma lugar, es normalmente un sistema de actividades humanas y los principales problemas son las mismas personas.

Por tanto, la IR necesita ser sensible a cómo las personas perciben y comprenden el mundo que les rodean, a cómo interactuar y a cómo la sociedad de su entorno de trabajo afecta en sus acciones.

Así mismo, el tipo del producto a desarrollarse, será un factor clave para la elección de un método IR. Por ejemplo: un método IR para sistemas de información es diferente a un método IR para sistemas de control embebidos, o para servicios generales, tales como “networking” y sistemas operativos. En este sentido, la IR basada en evidencias juega un rol importante, ya que gracias a la evaluación de métodos IR utilizando técnicas empíricas, permitirá a los desarrolladores elegir el método más adecuado.

2.1.1 Tareas de la Ingeniería de Requisitos

La IR se puede clasificar por tareas [55], cada una de ellas representando diferentes fases. Estas tareas son:

- Elicitación: comprende actividades que activan la comprensión de los objetivos y motivos para construir un sistema software propuesto.

- Modelado: se utilizan notaciones para formar uno o más modelos gráficos dedicados a representar los requisitos adquiridos en la elicitación.

- Análisis de requisitos: se construyen o mejoran técnicas para evaluar la calidad de los requisitos registrados, con la finalidad de eliminar errores de ambigüedad, inconsistencia o incompletitud, entre otros.

- Validación y verificación: esta tarea trata de asegurar que los modelos y la documentación expresa exactamente las necesidades del cliente.

- Gestión de requisitos: se ocupa de identificar y documentar los enlaces entre los diversos documentos y artefactos de requisitos construidos desde un principio con la finalidad de llevar una organización.

Como el objetivo de nuestro trabajo está centrado en CU y esta técnica está involucrada principalmente en la tarea de elicitación y modelado, a continuación describimos en mayor detalle estas dos tareas de la IR.

2.1.1.1 La elicitación

La elicitación de requisitos es quizás la actividad que más frecuentemente se considera como el primer paso del proceso de IR - el término “elicitación” es preferible al de “capturación”, para evitar la insinuación de que los requisitos están fuera para ser recogidos de manera sencilla preguntando las cuestiones correctas. La información recogida durante la elicitación de requisitos en ocasiones tiene que ser interpretada, analizada, modelada y validada antes de que el ingeniero de requisitos sienta que han sido recogidos de manera completa todo el conjunto de requisitos del sistema.

Uno de los más importantes objetivos de la elicitación es averiguar qué problemas necesitan ser solucionados y por tanto identificar los límites del sistema. Estos límites definen un alto nivel, donde el sistema final entregado se ajustará al actual entorno de operaciones, esto afectará a todos los esfuerzos de elicitación consecuentes. La

identificación de clientes y clases de usuario, objetivos y tareas, y de escenarios (y Casos de Uso) depende de cómo los límites han sido escogidos.

La elección de la técnica de elicitación depende del tiempo y recursos disponibles para el ingeniero de requisitos y por supuesto, del tipo de información que necesita para ser elicitado. Se distinguen varias técnicas de elicitación. La técnica de Model-driven proporciona un modelo específico del tipo de información recogida para dirigir el proceso de elicitación. Estos modelos incluyen métodos basados en objetivos tales como KAOS y I*, y métodos basados en escenarios como CREWS [88].

2.1.1.2 El modelado

El modelado – la construcción de descripciones abstractas que son sujetas a interpretación – es una actividad fundamental en IR. Los modelos pueden ser utilizados para representar un rango total de productos del proceso de IR. Además, muchas aproximaciones de modelados son utilizadas como herramientas de elicitación, donde la notación del modelado y modelos producidos parcialmente son utilizados como conductores para obtener más información.

Los modelos resultantes pueden ser utilizados para comunicar los requisitos entre los desarrolladores. Las notaciones del modelado ayudan a incrementar el nivel de abstracción en descripciones de requisitos proporcionando un vocabulario y unas reglas estructurales que son más cercanas a las entidades, relaciones, comportamientos y restricciones del problema que está siendo modelado. Cada notación de modelado está diseñada para elicitar o registrar detalles específicos sobre los requisitos, tales como qué información debe mantener el software, funciones, respuestas, propiedades o comportamiento.

Los modelos basados en escenarios - muy importantes en el presente trabajo, tal como se refleja en los resultados – han sido el foco de la reciente investigación; parcialmente porque los escenarios son más fáciles para los practicantes sin conocimientos técnicos, pero quizás también porque los escenarios son incompletos por naturaleza y entonces se prestan a definir un “sin fin” de problemas de investigación.

2.1.2 Direcciones de investigación

En su momento se trató de analizar la investigación existente en IR, Cheng y Atlee identificaron futuras direcciones de investigación sugeridas por las necesidades emergentes del software [55]. La investigación está considerada con respecto a las tecnologías desarrolladas para dirigir tareas de requisitos específicas, tales como la elicitación, el modelado y el análisis. El campo de la IR es rico y el alcance de las posibles direcciones de investigación es bastante amplio. Una revisión permite identificar áreas maduras de investigación, así como garantizar una mayor investigación. También, se revisan varias estrategias para realizar y ampliar los resultados de la revisión de la IR, para ayudar a trazar el ámbito de las futuras direcciones de investigación. Se consideran los actuales y futuros asuntos de investigación cuyo objetivo es dirigir las necesidades de la IR para sistemas emergentes del futuro.

Cheng y Atlee se centran en nueve “hotspots” (puntos calientes) de investigación, cuyas soluciones tienen probablemente el mayor impacto en la investigación y en la práctica de la IS. La lista de “hotspots” no está intencionada a ser exhaustiva. Es destinada a destacar algunas de las necesidades más urgentes y grandes desafíos en la investigación de la IR. Dichos puntos son los siguientes:

- Escalabilidad: los sistemas software están creciendo en tamaño y los sistemas de gran escala ya no hacen referencia solamente al tamaño significativo como en líneas de código; incluye complejidad, grado de heterogeneidad y número de nodos de decisión descentralizados.
- Seguridad: los sistemas software aumentan debido al incremento de los objetivos de ataques de seguridad, que están constantemente cambiando.
- Tolerancia: el software está incrementando el uso de aplicaciones críticas automatizadas y de servicios, tales como vehículos de transporte y sistemas, decisiones financieras y transacciones, asuntos médicos, militares, etc.
- Aumento de la dependencia en el entorno: el incremento en la escala es parcialmente debido al aumento de sistemas que consisten en software,

hardware y personas, los cuales pueden estar libremente o estrechamente acoplados.

- Auto-gestión: el sistema software es consciente de que debe acomodarse a las variaciones en tiempo de ejecución de incertezas, de incompletitudes y de la evolución de requisitos; así que debe ser capaz de reaccionar y adaptarse a los cambios en el entorno o en los requisitos.
- Globalización: el desarrollo de software global es un paradigma emergente que se mueve hacia equipos de desarrollo distribuidos globalmente. Este movimiento está motivado por el deseo de explotar las 24 horas del día, aprovecharse de recursos globales, disminuir costes y estar geográficamente cerca del consumidor.
- Metodologías, patrones y herramientas: la transferencia de tecnologías de IR desde la investigación a la práctica beneficiaría desde los mejores informes en cómo aplicar las tecnologías más sistemáticamente para mejorar la productividad de los analistas de requisitos y mejorar la calidad de los resultantes artefactos de requisitos.
- Requisitos de reusabilidad: se facilitaría la reusabilidad de los artefactos existentes con una aproximación en tareas de IR más prescriptivas y sistemáticas. El llamado “product lining” es la mayor estrategia en reusabilidad, donde los productos relacionados son tratados como una familia de productos y su co-desarrollo está planeado desde el principio.
- Efectividad de las tecnologías de IR: el impacto de la investigación en IR depende en cómo son de relevantes los resultados en la industria en cuanto a sus necesidades.

De los nueve “hotspots” mencionados, seis surgen de las necesidades del software debido al predicho incremento en la escala, en seguridad, en tolerancia, en dependencias entre el software y su entorno, auto-gestión y globalización. Los otros tres “hotspots” se focalizan en extender y madurar tecnologías existentes para mejorar las metodologías de IR y reusabilidad de requisitos y en el creciente volumen de la investigación basada en evaluaciones. Algunas de estas direcciones son extensiones naturales de trabajo ya realizado por investigadores de IR, mientras otros son

discontinuos debido a los cambios fundamentales en las necesidades de la informática. Todos estos problemas descritos requerirán un esfuerzo sustancial para progresar hacia soluciones efectivas.

Existen cinco recomendaciones en las que la comunidad de IR podría aprovechar para empezar mejorando en la madurez de las actuales tecnologías existentes:

- Los investigadores deberían trabajar con los profesionales.
- Los investigadores de IR deberían trabajar con otros investigadores de IS y profesionales, para establecer fuertes enlaces entre sus respectivos artefactos.
- Los investigadores de IR no deberían descuidar las evaluaciones y la investigación empírica.
- Las organizaciones industriales deberían proporcionar un esfuerzo industrial (respecto a los datos) a los investigadores.
- Investigadores y profesionales deberían juntos establecer repositorios de artefactos de IR.

Las acciones comentadas anteriormente deberían ayudar a la comunidad de investigación de IR para encaminar a un progreso inmediato en mejorar conocimiento existente y técnicas. Además, hay algunas acciones a largo plazo que mejorarían la infraestructura de la comunidad de investigación y su capacidad de confrontar los desafíos planteados por los sistemas emergentes:

- Las necesidades de la comunidad de IR a ser proactiva en identificar los problemas de investigación en IR que surgen de nuevos desafíos informáticos.
- Las necesidades de los investigadores en pensar en más allá de los actuales conocimientos y capacidades del IR y IS para progresar significativamente y dirigir los desafíos planteados por los sistemas emergentes.
- Las necesidades académicas en IR para educar a la siguiente generación de desarrolladores en problemas de IR y tecnologías.

En conclusión, la comunidad de investigadores en IR ha hecho progresos significantes. Al mismo tiempo, las demandas situadas en informática y en ciberinfraestructura han incrementado dramáticamente, surgiendo numerosas nuevas críticas para las preguntas de investigación de IR. Por estas razones, es un momento

excitante para estar involucrado en investigación de IR. Existen tecnologías que avanzan significativamente para resolver estos problemas y que implicarán cambios en el paradigma que impactarán a muchas futuras generaciones de desarrolladores, a sistemas de información y a futuros clientes.

2.2 Revisiones Sistemáticas

El desarrollo de cualquier dominio de conocimiento que invoca principios científicos, incluyendo las ciencias naturales, la ingeniería, la medicina y las ciencias sociales, involucra de manera implícita un diálogo entre teoría y observación [66]. En dominios donde el humano es participante en estudios empíricos, el contexto en el cual los estudios individuales son desempeñados puede influir en sus resultados. Una manera de limitar el efecto del contexto es emprender estudios secundarios sistemáticos. Un estudio secundario establece una respuesta a una pregunta de investigación recogiendo y agregando la información disponible en los estudios publicados (estudios primarios). De este modo, el objetivo es identificar dónde los diferentes estudios soportan conclusiones particulares y también identificar los estudios que difieren y las razones del por qué pueden ocurrir. Su robustez como proceso es determinado por el rigor con el que la información es seleccionada, extraída y agregada. En los últimos años el uso de estudios secundarios ha sustentado el paradigma basado en evidencias, que ha tenido una importante influencia sobre la investigación y la práctica en muchas disciplinas, principalmente en medicina.

Los estudios secundarios utilizan información de anteriores estudios publicados con el propósito de sintetizar tratando de resumir, integrar y donde sea posible combinar las conclusiones de estudios diferentes en un tópico de pregunta de investigación. Tales síntesis pueden identificar también áreas cruciales y preguntas que no han sido dirigidas adecuadamente en la anterior investigación empírica. La síntesis de investigación se refiere, en ocasiones indistintamente, a “revisión sistemática” (RS) y a “revisión de literatura sistemática” (RLS).

Una RS es un medio de evaluación e interpretación de toda investigación relevante disponible respecto a una pregunta de investigación en particular, área temática o fenómeno de interés. Las RS están dirigidas a presentar una evaluación justa de un

asunto de investigación usando una metodología fiable y rigurosa. También se pueden definir como un estudio secundario a partir de estudios individuales o estudios primarios.

¿Para qué introducirse en una RS? Hoy en día nos encontramos en un nivel en el que parece que todo ya se ha estudiado, investigado o inventado, para ello también es necesario investigar sobre el material pertinente existente. Una RS también se podría definir como un estudio del estado del arte sobre un fenómeno de interés.

Las RS son una herramienta clave para activar evidencias basadas en la práctica como traen y combinan juntas las conclusiones desde múltiples estudios. La fuerza de estos métodos se encuentra en sus intentos explícitos de minimizar las oportunidades de presentar incorrectas o engañosas conclusiones como resultado de estudios primarios sesgados. Éstos son esenciales para tomar parte en investigación y en la práctica, y muchos de los intereses actuales en ERS dentro de la IS y otras disciplinas se originan desde revisiones de la efectividad de intervenciones reflejadas en iniciativas. Realizando una RS se demuestra la poca fiabilidad que puede existir dependiendo solamente de los resultados de los estudios individuales. Los ingenieros de software pueden tomar decisiones incorrectas adoptando nuevas técnicas si no se consideran las evidencias científicas para determinar la eficacia de éstas. Mientras tanto, la calidad de los estudios primarios en IS es en ocasiones pobre. Una importante cuestión es cuánta confianza podemos obtener de las conclusiones y recomendaciones que surgen de las revisiones sistemáticas. Es importante que los revisores evalúen críticamente los métodos de los estudios primarios.

Las ERS son solo tan buenas como la evidencia en la que esté basada. Dyba y Dingsoyr se presentan una vista general de algunos de los sistemas más influyentes para evaluar la calidad individual de estudios primarios para clasificar la fuerza total de la evidencia [7]. Para evaluar una RS se debe tener en cuenta primordialmente la fiabilidad de los experimentos, se requiere un medio fiable para evaluar la calidad de los experimentos realizados en IS. En dicho trabajo se confirma la usabilidad de un

*checklist*⁵ de evaluación de calidad que determina cuántos revisores se necesitan por artículo especificándose un proceso apropiado para evaluar la calidad. En los resultados se refleja que la fiabilidad en evaluaciones individuales (de revisores) es bastante pobre, mientras que realizándose por parejas son mucho mejores. Desde el punto de vista de llevar a cabo ERS, en la literatura existen sugerencias para el uso de *checklists* de calidad que pueden ser utilizados para evaluar la calidad de estudios empíricos en IS. En particular, Dyba y Dingsoyr desarrollaron un cuestionario que utilizaron para estudiar los métodos ágiles:

1. ¿Los autores indican claramente el objetivo de la investigación?
2. ¿Los autores describen las muestras y las unidades experimentales (materiales experimentales y participantes)?
3. ¿Los autores describen el diseño del experimento?
4. ¿Los autores describen los procedimientos de colección de datos y definen sus medidas?
5. ¿Los autores definen los procedimientos de análisis de datos?
6. ¿Los autores discuten el potencial del experimento?
7. ¿Los autores discuten sobre las limitaciones de su estudio?
8. ¿Los autores indican claramente las conclusiones?
9. ¿Existe evidencia en que los E/QE (Engineering/Quality Experiment) pueden ser utilizados por otros investigadores o practicantes?

Como se muestra en la Figura 1 presentada por Charters y Budgen [66], la investigación basada en evidencias no es un proceso de peso ligero. Llevar a cabo una revisión sistemática requiere tiempo y esfuerzo.

⁵ Un *checklist* es un listado de preguntas o procedimientos que permite comprobar o evaluar una serie de factores para la consecución de un objetivo, es muy utilizado para cuidar la calidad en Ingeniería de Software.

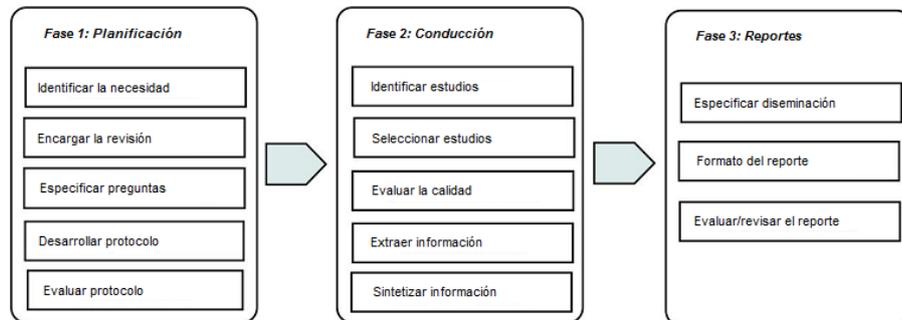


Figura 1. Proceso de Revisión Sistemática

En [17] Kitchenham explica las diferentes etapas del proceso de la RS. Estos son, de manera general, los principales puntos a la hora de llevar a cabo una RS.

1) Planificación de la revisión.

- a. Identificación de la necesidad para una revisión: el requisito de los investigadores de resumir toda la información existente sobre algún fenómeno de una manera completa y exhaustiva. Los investigadores deben identificar y revisar cualquier revisión sistemática sobre el fenómeno de interés para el criterio de evaluación apropiado.
- b. Desarrollo de un protocolo de revisión: un protocolo de revisión especifica los métodos que serán usados para sobrellevar una específica revisión sistemática. Un protocolo predefinido es necesario para reducir la posibilidad de parcialidad del investigador.

2) Realización de la revisión.

- a. Identificación de la investigación: el objetivo de una revisión sistemática es encontrar tantos estudios primarios como sea posible relacionados con la pregunta de investigación utilizando una estrategia de búsqueda.

- b. Selección de los estudios primarios: una vez que se han obtenido los estudios primarios potencialmente relevantes éstos necesitan ser evaluados para conocer su relevancia.
 - c. Evaluación de la calidad de los estudios: para el criterio de inclusión/exclusión a la hora de seleccionar los estudios, es generalmente considerado importante evaluar la “calidad” de los estudios primarios.
 - d. Extracción de información: el objetivo de esta etapa es diseñar formas de extraer la información para un preciso registro de los datos obtenidos por los investigadores desde los estudios primarios.
 - e. Síntesis de datos: involucra resumen y cotejo de los resultados de los estudios primarios incluidos. La síntesis puede ser descriptiva (no cuantitativa). Sin embargo, a veces es posible complementar la síntesis descriptiva con un resumen cuantitativo.
- 3) Informe de la revisión: es importante comunicar los resultados de la revisión sistemática de manera efectiva.

Existen otros trabajos dedicados a construir y a evaluar revisiones sistemáticas (y a evaluar experimentos). Travassos [19] ilustra el proceso de conducción de ERS, presentando el modelo que guía su ejecución. Este artículo describe un procedimiento sistemático para ejecutar revisiones de literatura en IS presentando una plantilla de protocolo y un proceso correspondiente capaz de dirigir a los investigadores a través de un proceso de revisión. El artículo introduce una propuesta que se basa en una plantilla de RS y un proceso de conducción en la realización de estudios secundarios. Para apoyar la evaluación de tal propuesta se siguió un estudio piloto [11] para verificar su viabilidad. Además, también ha sido posible observar cómo los investigadores llevan sus procesos. Los resultados de este estudio permitieron mejorar en la estructura de la plantilla presentada en el trabajo de Travassos.

2.3 Revisiones Sistemáticas en la Ingeniería de Software

Las revisiones sistemáticas están actualmente ganando popularidad en IS, con revisiones publicadas en diversos tópicos incluyendo elicitación de requisitos, ingeniería web, estimación de costes, desarrollo de software con métodos ágiles, así como varios aspectos relacionados con experimentos de IS, tales como estadísticas, efecto del tamaño y la quasi-experimentación [7]. Tales revisiones son importantes como el volumen de la investigación que necesita ser considerada por los practicantes de IS e investigadores. En [9] Sjoberg presenta el enfoque de que para todos los campos de la IS los métodos empíricos deberían impulsar el desarrollo de conocimiento científico sobre la facilidad de usar diferentes tecnologías de IS para diferentes tipos de actores, realizando diferentes tipos de actividades, en diferentes tipos de sistemas. Es parte de la visión que tal conocimiento científico guiará el desarrollo de nueva tecnología de IS y su mayor aporte de decisiones importantes en la industria de IS. Los mayores desafíos en la búsqueda de este enfoque son: más investigación en IS debería ser basado en el uso de métodos empíricos; la calidad, incluida la relevancia, de los estudios utilizando dichos métodos deberían ser incrementados; debería haber más y mejor síntesis de evidencia empírica; y más teorías deberían ser construidas y probadas. Los métodos para afrontar los anteriores son: incremento de la competencia en cuanto a cómo aplicar y combinar métodos empíricos alternativos, ajuste de vínculos entre el ámbito académico y el industrial, el desarrollo de agendas para una investigación común con un enfoque a métodos empíricos, y más recursos para la investigación empírica.

La Evidencia Basada en Medicina (EBM) surgió porque los practicantes (involucrados) de Medicina estaban saturados por el gran número de estudios científicos. Los involucrados en software todavía tienen dificultades para evaluar la calidad de las evidencias y evaluar qué evidencias son importantes para las circunstancias específicas. La IS no posee todavía muchas evidencias y replicaciones llevadas a cabo por experimentos rigurosos, así que los estudios empíricos son mucho menos fiables científicamente.

La Ingeniería de Software Basada en Evidencias (ISBE) confía en la investigación de la Ingeniería de Software Empírica pero enfatiza en la necesidad de

encontrar y agregar la mejor evidencia disponible en un tópico específico y utiliza estudios secundarios tales como las revisiones sistemáticas y los “mapping studies” como marco de trabajo metodológico para encontrar y agregar evidencias [57]. La diferencia entre revisiones sistemáticas y “mapping studies” es el tipo de pregunta que se contestan. Una revisión sistemática pregunta una específica y completa cuestión como “¿Cuál de las métricas OO son mejores en la predicción de fallos propensos?”, mientras un “mapping study” pregunta cuestiones más generales como “¿Qué sabemos acerca de las métricas relacionadas con OO?”. Las revisiones sistemáticas y los “mapping studies” son conocidos como estudios secundarios porque apuntan a agregar o sintetizar la investigación de otros estudios que son atribuidos como estudios primarios. Debe ser enfatizado que la ISBE no simplemente equipara a la metodología, la finalidad es ayudar a los profesionales en la industria. Sin embargo, los profesionales piden algo diferente y normalmente cuestiones muy específicas en su contexto, por ejemplo: “¿será la métrica X buena para predecir fallos propensos en mi gran sistema evolutivo OO?”. Basarse en los resultados de una revisión sistemática para contestar a una pregunta orientada desde el ámbito profesional no es trivial.

Budgen et al. examina las experiencias y resultados de 6 “mapping studies”, de los cuales cuatro están publicados [62] (ver los 6 primeros de la Tabla 1). Desde éstos apuntamos a un tema recurrente sobre los problemas de clasificación y a una preponderación de los “vacíos” en el conjunto de los estudios empíricos. Las etapas tempranas de un “mapping study” son generalmente muy similares a aquellos de una revisión sistemática, aunque la pregunta de investigación en sí misma es probablemente más extensa, con el fin de abordar adecuadamente la extensión del ámbito del estudio. Estas tres etapas son:

1. La identificación de estudios primarios que contienen resultados de investigaciones relevantes (búsqueda);
2. La selección de estudios primarios apropiados desde éstos después de una exhaustiva exploración;
3. Donde sea apropiado, realizar una evaluación de calidad de los estudios seleccionados.

Antes de poder sobrellevar RLS, necesitamos tener una mejor comprensión de qué evidencia está disponible dentro de un dominio - y los “mapping studies” son particularmente útiles para este propósito. Son útiles para identificar las áreas donde hay suficiente información para que un RLS sea efectivo, así como aquellas áreas donde se necesitan más estudios primarios.

Kitchenham et al. resume el estado de la ISBE (como revisión de literatura terciaria) [10]. El propósito de este estudio es revisar el actual estado de la ISBE desde 2004 usando un estudio terciario para revisar artículos relacionados con la ISBE y en particular se concentra en artículos describiendo revisiones sistemáticas de literatura (RLS). Aunque RLS no es sinónimo de ISBE, la agregación de los resultados de investigación son una importante parte del proceso de ISBE y además, es la parte del proceso de ISBE que puede ser fácilmente observada en la literatura científica. Las preguntas de investigación son: ¿cuánta actividad de RLS ha habido desde 2004?, ¿qué tópicos de investigación están siendo dirigidos?, ¿quién está liderando la investigación RLS? y ¿cuáles son las limitaciones de la actual investigación?

Las compañías de software en ocasiones adoptan tecnologías bajo presión, inmaduras en el mercado [51]. Se sugiere que los practicantes consideren Ingeniería de Software basada en la Evidencia (ISBE) [56]. Esto significa que las decisiones tomadas sean más acertadas respecto al desarrollo del software y mantenimiento integrando la mejor evidencia actual desde la investigación con experiencia práctica, evaluando desde el punto de vista de las circunstancias específicas. Además el rigor es condición necesaria, esto implica que la IS no debe confiar únicamente en experimentos de laboratorio y debe intentar adquirir evidencias de proyectos industriales (reales). Se debe evaluar la evidencia para asegurar que es apropiada para lo que se busca.

La idea básica detrás de la ISBE es establecer una fructífera cooperación entre los investigadores y profesionales, debería existir un vínculo cercano entre investigadores y profesionales ya que la investigación es relevante para las necesidades de los profesionales y los profesionales están dispuestos a participar en la investigación. Los investigadores en IS podrían proporcionar más ayuda a los practicantes si se

empresarios y se publicaran más revisiones sistemáticas sobre asuntos de interés en IS.

Kitchenham propone una guía para ERS dirigida a investigadores de IS [17]. El objetivo de la guía es introducir el concepto de revisiones rigurosas de evidencia empírica actual a la comunidad de IS. Está basada en una revisión de tres existentes guías para ERS:

1. *The Cochrane Reviewer's Handbook.*
2. *Guidelines prepared by the Australian National Health and Medical Research Council.*
3. *CRD Guidelines for those carrying out or commissioning reviews.*

Estas guías presentadas han sido destinadas a ayudar a los investigadores en medicina. La guía propuesta por Kitchenham intenta adaptar las guías de medicina a las necesidades de los investigadores de IS. Se debate un gran número de asuntos donde la investigación en IS difiere de la investigación en medicina. En particular, la investigación en IS goza relativamente de poca investigación empírica comparada con la gran cantidad de material empírico disponible en medicina. Además, los métodos de *investigación* usados por los ingenieros de software no son tan rigurosos. Nuestra revisión sistemática parte desde el trabajo de Kitchenham [17], propuesta por el ISERN.

Un ejemplo de RS en IS es el trabajo de Racheva et al. que presenta cómo el valor de negocio es creado por proyectos de métodos ágiles [64]. En este artículo se toma una mirada cercana en las maneras en las que las prácticas de software en métodos ágiles crean valor en proyectos ágiles. Se trata de establecer respuesta a tres preguntas de investigación: ¿qué conceptos de valor de negocio son usados en el contexto ágil?, ¿en qué manera los proyectos ágiles crean valor de negocio? y ¿en qué manera las prácticas específicas o individuales tienen influencia en la creación de valor del negocio? Las conclusiones son que en la literatura de la IS ágil no existe una definición elaborada de valor de negocio, que los practicantes ofrecen definiciones las cuales traducen el valor del negocio en el valor del dólar y que la noción del valor de negocio es altamente volátil.

Tabla 1. "Mapping studies" en Ingeniería de Software

<i>Autores y Año Publicado</i>	<i>Periodo de búsqueda</i>	<i>Tópico</i>	<i>Nº de estudios primarios</i>	<i>Forma de Análisis</i>	<i>Fuentes de búsqueda</i>
Juristo (2006)	Desde 2005	Técnicas de Pruebas Unitarias	24	Clasificar por técnicas para la generación y selección de pruebas establecidas	IEEE, ACM
Jorgensen & Shepperd (2007)	Desde 04/2004	Esfuerzo de Desarrollo de Software y Estimación de Coste	54 (204)	Clasificar por tópico, aproximación estimada, el contexto del estudio para la aproximación de la investigación y el conjunto de datos	76 revistas
Sjoberg et al. (2007) [9]	1993 – 2002	Uso de Estudios Experimentales	103	Clasificar por tópicos, formas de estudio, participantes, etc.	9 revistas, 3 conferencias
Bailey et al. (2007) [8]	Desde 2007	Diseño Orientado a Objetos	138	Clasificar por formas de estudio y formas de las intervenciones	IEEE, ACM, Science Direct, Web of Science, Google Scholar
Cheng (n/a)	1995 – 2008	Patrones de Diseño de Software	185 (55 expts)	Clasificar frente a 8 afirmaciones derivadas de patrones de literatura y por la forma de estudio	IEEE, ACM, Science Direct, Web of Science, CiteSeer, Google Scholar
Pretorius (2008) [6]	1994 - 2008	Características UML	33	Clasificar por formas de estudio y aspectos de UML	IEEE, ACM, ScienceDirect, Athens
Racheva et al. (2009) [64]	Desde 2000	Valor de Negocio en Proyectos Ágiles	50	Comparar los puntos de convergencia y divergencia sobre los términos utilizados al discutir el valor de negocio	ACM, IEEE Xplore, ISI Web of Science, Kluwer Online ScienceDirect – Elsevier, SpringerLink, Wiley InterScience, Scopus
Dyba y Dingsor (2008) [58]	Desde 2005	Desarrollo de Software Ágil	36	Clasificar por introducción y adopción, factores humanos y sociales, percepciones del desarrollador y del cliente, y estudios comparativos	ACM, Compendex, IEEE Xplore, ISI Web of Science, Kluwer Online, ScienceDirect – Elsevier, SpringerLink, Wiley Inter Science Journal Finder

2.4 Revisiones Sistemáticas en la Ingeniería de Requisitos

Existen estudios secundarios publicados en referencia a IR destinados a la mejora de la calidad del proceso de captación de requisitos. Algunos de ellos se comentan en esta sección.

Davis et al. llevaron a cabo un análisis detallado de la literatura de IR [67], existente desde años atrás, para adquirir en profundidad el conocimiento de su disciplina. El artículo proporciona un detallado análisis demográfico cuantitativo de las publicaciones dentro de la IR abarcando los últimos 40 años. Trata y consigue de entender los siguientes puntos:

- Entender las tendencias cuantitativas en publicaciones de IR que ayudarán a determinar si hay un creciente o un menor interés en el campo de IR.
- Entender cuándo los artículos tempranos fueron escritos permitirá empezar el proceso de entender las raíces de la disciplina.
- Entender las salidas (outlets) de las publicaciones de IR puede ser de ayuda cuando al elegir una salida para un nuevo artículo.
- Entender las salidas (outlets) de las publicaciones de IR permitirá a los practicantes a localizar soluciones potenciales más fácilmente a los problemas de requisitos.
- Entender qué partes están publicando artículos en IR permitirá entender mejor si IR es una solución en la búsqueda de un problema o un problema en la búsqueda de una solución.
- Entender las tendencias organizacionales y nacionales podrían ser de ayuda para los investigadores cuando buscan socios para investigar.

Las preguntas de investigación son:

- ¿Cuáles son las publicaciones más tempranas que llevan cuestiones de IR?
- ¿Cuántas publicaciones (y páginas) han sido escritas en el campo del IR?
¿cómo ha cambiado la producción sobre los años?

- ¿Cuáles han sido las salidas más populares para las publicaciones de IR?
¿cómo ha cambiado esto en el tiempo?
- ¿Cuántos autores diferentes son responsables de la salida de publicaciones de IR? ¿cuál es la media de publicaciones por autor?
- ¿Cuántos autores se asocian juntos para escribir artículos?
- ¿Difieren los autores académicos en alguna sus características con los autores no académicos? ¿esto ha cambiado con el paso del tiempo?
- ¿Qué organizaciones han sido responsables de escribir artículos de IR? ¿cómo ha cambiado esto con el paso del tiempo?

Como conclusión destacar que las publicaciones más tempranas en IR toman lugar en 1963-64, el origen de los términos “Ingeniería de Requisitos” y “Gestión de Requisitos” está en los años 70 y 80 respectivamente, y también señalar la enorme cantidad de publicaciones de IR (4.089), autores únicos (4.547) y organizaciones afiliadas únicas (1.450).

Condori et al. [63], en base a 46 estudios primarios identificados y categorizados, encontró que la comprensibilidad es el aspecto más evaluado de las Especificaciones de Requisitos Software (ERS, ver capítulo 3), los experimentos son los métodos de investigación más comunes y el entorno académico es donde se han efectuado la mayoría de evaluaciones empíricas. El artículo gira alrededor de las siguientes preguntas de investigación: ¿cuáles son los aspectos de calidad más investigados en las técnicas de ERS?, ¿en qué tipos de estudios y en qué dominios han sido estos aspectos investigados? y ¿qué método de investigación se utilizó para la evaluación de los aspectos más estudiados? A partir de los resultados obtenidos se indica que cada uno tiene su propia interpretación para qué termino usar para referirse a aspecto de calidad, que no fue posible encontrar una lista de posibles dominios donde los usuarios pueden decidir utilizar sistemas software y que el método de investigación más destacado es el experimento, entre otros estudios empíricos.

Desde el trabajo de Davis et al. [3] se parte que las entrevistas preferentemente estructuradas parecen ser una de las más efectivas técnicas de elicitación; muchas técnicas ocasionalmente citadas en la literatura, como “card sorting”, “ranking” o “thinking aloud” tienden a ser menos efectivas que las entrevistas; la experiencia de

los analistas no parece ser un factor relevante. Estos resultados se obtienen de un estudio donde a través de una revisión sistemática se profundiza en la efectividad de las técnicas de elicitación, aunque, los estudios de los cuales estos resultados han sido agregados no han sido replicados, y por tanto, no se puede decir que las afirmaciones anteriores sean absolutamente ciertas.

Hermann et al. [2] estudió la priorización de requisitos, que necesariamente se basa en los requisitos especificados y en predicciones de beneficios y coste de los requisitos individuales. El trabajo de Henmann et al. presenta los resultados de una revisión sistemática de literatura, la cual investiga cómo los métodos existentes enfocan el problema de la priorización de requisitos basados en beneficios y costes. ¿En qué manera está la priorización de requisitos basada en beneficios y estimación de coste actualmente soportada y por qué método de publicación? Como respuesta se identifica una agenda de investigación para este tópico y para refinarla se plantean dos puntos: saber qué métodos han sido validados empíricamente y cómo, y tratar preguntas que afecten al método de soporte.

Tabla 2. "Mapping studies" en Ingeniería de Requisitos

<i>Autores y Año Publicado</i>	<i>Periodo de búsqueda</i>	<i>Tópico</i>	<i>Nº de estudios primarios</i>	<i>Fuentes de búsqueda</i>
Davis et al (2007) [67]	1963 - 2006	Ingeniería de Requisitos	4089	IEEE Xplore, ACM Digital Library, ScienceDirect, Kluwer Online, Engineering Village 2, Wiley Interscience, SpringerLink
Condori et al (2009) [63]	n/a	Especificación de Requisitos Software	46	Scopus, IEEE Digital Library, ACM Digital Library, manual search
Dieste et al. (2008) [68]	n/a	Técnicas de Elicitación de Requisitos Software	564	Scopus, IEEE Xplore, ACM DL databases, Google
Davis et al. (2006) [3]	n/a	Técnicas de Elicitación de Requisitos Software	26	Scopus, IEEE Xplore, ACM DL databases
Hermann et al. (2008) [2]	n/a	Priorización de Requisitos basados en Estimación y Coste	240	ACM, Compendex, IEEE Xplore, ISI Web of Science, Kluwer Online

				ScienceDirect – Elsevier, SpringerLink, Wiley InterScience, manual search
Yue et al. (2010) [59]	1996 -2008	Transformación de Modelos desde los Requisitos	20	IEEE Xplore, ACM Digital Library, Compendex, Inspec, and SpringerLink, manual search
Svahnberg et al. (2009) [60]	n/a	Modelos de Planificación Estratégica de Liberación	24	Compendex, Inspec, IEEE Xplore, ACM Digital Library, Springer Link
Mellado et al. (2009) [61]	Desde 2004	Requisitos de Seguridad	22	ACM digital library, IEEE digital library, Google Scholar, 9 conferencias, 3 revistas

Como ya se ha comentado anteriormente, el presente trabajo trata de una RS en el ámbito de la IS que se adentra en la IR y en particular en las necesidades existentes actualmente de los UC. Dicha técnica interviene principalmente en las tareas de elicitación y modelado para buscar, recoger, especificar y diseñar gráficamente los requisitos. En dicho estudio secundario se establecen respuestas a preguntas de investigación recogiendo y agregando la información disponible en los estudios publicados. Cabe destacar que la de “mapping study” es la definición más adecuada para el presente trabajo ya que nuestra investigación se basa en preguntas generales sobre la calidad de CU. De las RS expuestas señalar que la presente RS está muy relacionada con el estudio de Condori et al. [63], en la que estudian la calidad en ERS en general mientras aquí se estudia en CU (considerado como ERS o parte de una ERS) ya que parece ser que no hay una descripción clara de cómo proceder a diseñar CU de calidad. En el capítulo 4 se muestra el protocolo de revisión diseñado para llevar a cabo la RS y en el capítulo 5 la conducción de la misma.

Capítulo 3

Calidad de la Especificación de Requisitos de Software

En este capítulo se provee información sobre trabajos realizados en el pasado en lo que respecta a calidad en especificación de requisitos y trabajos llevados a cabo sobre calidad de CU en particular. Teniendo en cuenta que las descripciones de CU se incluyen dentro del conjunto de especificaciones de requisitos es importante analizar dentro de éstos el rendimiento atribuido hasta ahora, ya que si se mejora la calidad en especificaciones de requisitos puede que se mejore la calidad del modelo de CU y si se mejora la calidad del modelo de CU seguro que se mejora la calidad del documento de especificación de requisitos que lo incluye.

El máximo objetivo para clientes, programadores y toda persona involucrada al inicio en el desarrollo de un producto software es describir el qué debe hacer el sistema final detallando sus diferentes funcionalidades.

3.1 Estándares y marcos de calidad

Existen plantillas y documentos estándar con la finalidad de plasmar estas descripciones, por ejemplo, se presentan propuestas para identificar calidad en Especificaciones de Requisitos Software (ERS) por el estándar-830 de IEEE⁶

⁶ Los documentos de los estándares IEEE (*The Institute of Electrical and Electronics Engineers*) son desarrollados por los *Technical Committees of the IEEE Societies* y los *Standards Coordinating Committees of the IEEE Standards Board*. Los estándares desarrollados en IEEE representan un consenso entre los expertos de la materia dentro del Instituto y aquellas actividades de fuera del IEEE las cuales han expresado el interés en participar en el desarrollo del estándar. El uso del estándar IEEE es totalmente voluntario. La existencia del estándar IEEE no implica que no existan otras maneras de producir pruebas, medidas u otras buenas maneras y servicios relacionados con ese mismo ámbito.

[30][31], cuya finalidad es tratar de conseguir una descripción que conduzca al éxito el producto software final.

No obstante, en el ERS no solo se describe el proceso dirigido a especificar requisitos de software, sino también puede ser aplicado para ayudar en la estructuración dentro de la misma organización y en la distribución de productos software comercial. La finalidad máxima del ERS es que el resultado sea un documento completo y no ambiguo en el que: los clientes sean más precisos a la hora de describir lo que quieren obtener, los proveedores entiendan más exactamente lo que los clientes quieren, - y a nivel interno de la organización - desarrollar un estándar de ERS, definir el formato y contenido, y desarrollar otros soportes como *checklists* de calidad [33][34] o guías para escribir ERS [38].

También el ERS cumple con otras funciones como: establecer el acuerdo entre clientes y proveedores sobre qué tiene que hacer el sistema, reducir el esfuerzo de desarrollo, proporciona la base para estimar los costes y planificación, proporciona la base para su validación y verificación, facilita la transferencia entre clientes y usuarios, y sirve de base para futuras mejoras (actualización del ERS).

En el estándar-830 se hace referencia a ciertas características de calidad, recomendadas para un correcto ERS:

Tabla 3. Atributos de calidad recomendados por el estándar 830

Corrección.	Clasificación por importancia y/o estabilidad.
No ambigüedad.	Verificable.
Compleitud.	Modificable.
Consistencia.	Trazabilidad.

Parece importante apoyarse en ciertas características o atributos de calidad para conseguir o detectar una buena descripción, pero no es lo suficientemente importante si no sabemos cómo se calculan o cómo se miden estos atributos. Así que, saber definir y medir documentos de requisitos se convierte en requisitos imprescindibles para obtener características de calidad y por tanto detectar descripciones de calidad.

Existen métodos de evaluación de calidad de requisitos [35]. Durante muchos años se han revisado numerosos documentos para identificar y eliminar defectos que se

puedan propagar a lo largo del ciclo de vida del desarrollo, que pueden suponer un elevado coste. Una inspección es una técnica de revisión conocida para ser eficientes e identificar defectos, pero como cualquier otra técnica no se garantiza que los defectos sean encontrados. Trudel [32] presenta un proyecto en el que se investiga en la contribución de medidores encontrando defectos en documentos de requisitos. El artículo describe un experimento donde el mismo documento de requisitos fue inspeccionado por un número de inspectores así como por un número de medidores, el número y tipo de defectos encontrado por los inspectores y medidores fueron comparados y discutidos. Los resultados muestran un aumento significativo en la identificación de defectos cuando tanto en la inspección como en la medición del tamaño funcional son utilizados para encontrar y reportar defectos.

Pero más allá de detectar defectos nos interesa primero saber a partir de qué se pueden detectar los defectos. Como se ha mencionado anteriormente creemos importante hacer uso de los atributos de calidad y saber cómo conseguir llegar hasta ellos. Davis et al. explora completamente el concepto de calidad en ERS, define los atributos que contribuyen a dicha calidad y sugiere métricas para estos atributos. Davis considera que en un ERS de calidad se exhiben las siguientes cualidades [15]:

Tabla 4. Atributos de calidad propuestos por Davis

No ambiguo	Guardado electrónicamente
Completo	Ejecutable/Interpretable
Correcto	Clasificado por Importancia Relativa
Comprensible	Clasificado por Estabilidad Relativa
Verificable	Clasificado por Versión
Internamente consistente	No Redundante
Externamente consistente	A nivel correcto de detalle
Realizable	Preciso
Conciso	Reusable
Diseño independiente	Rastreado
Traceable	Organizado
Modificable	Con referencias cruzadas

La consigna de Davis es que si los atributos “se pueden comprender mejor, reorganizar y medir la calidad en un ERS, estaremos mejor equipados para detectar errores en el ERS” y por tanto contribuir a un desarrollo más libre de defectos. Davis asegura que un ERS de calidad es un ERS que cumple con los 24 atributos listados. El artículo define cada atributo, proporciona ideas para medir los atributos, proporciona los atributos con un recomendado peso relativo a otros atributos (dependencia) y describe tipos de actividades que pueden ser usados para optimizar la presencia de atributos.

Además, según sus exploraciones, un ERS perfecto es imposible. No es fácil construir un ERS de calidad porque la mejora de un atributo puede afectar a la desmejora de otro. Por ejemplo, si se eliminan todas las ambigüedades se añade demasiada formalidad que no sería comprensible para un inexperto en software; si eliminamos toda redundancia, se hace más difícil de leer; si perfeccionamos la completitud, perdemos concisión. Todo ello a partir de las métricas utilizadas.

Un ERS puede contener una mezcla de modelos conceptuales y texto en lenguaje natural, para ello es necesario incluir recursos de calidad para ambos tipos de modelos combinados. Así, al igual que Davis, Krogstie [14] aplica calidad para modelos textuales basándose en un aspecto muy importante: en que los modelos de requisitos deben ser comprendidos por muchas personas con diferentes conocimientos. Se considera que la construcción del modelado conceptual es en parte la externalización de la realidad que una persona interpreta en el interior y actúa como realidad organizacional para la audiencia del modelo. El uso de modelos conceptuales construidos como parte del desarrollo y del mantenimiento de sistemas de aplicaciones son discutidos por varios investigadores [42][43][44][45][46].

Krogstie describe un marco de trabajo para entender calidad en modelado conceptual, pero como ya hemos dicho calidad es una noción difícil de entender, ¿qué entendemos por calidad? - se han propuesto muchas aproximaciones respecto a calidad en el campo de los sistemas de información, Denning [21] propuso lo que se consideró como el primer nivel de calidad. El modelo de Krogstie profundiza en el desarrollo del marco y de los lenguajes de modelado, en los que también agrupa atributos de calidad como algunos apuntados por Davis. Existen otros marcos que evalúan la calidad de modelos como:

- Lindland [40]: considera la principal estructura del marco calidad de Krogstie. La idea básica es evaluar la calidad de modelos a través de tres dimensiones comparando conjuntos de estados. Los conjuntos de estados son: el modelo externalizado, la extensión del lenguaje, el dominio de modelado y la interpretación de la audiencia. Las dimensiones – que representan la relación entre los estados - son: la calidad sintáctica, la calidad semántica y la calidad pragmática.

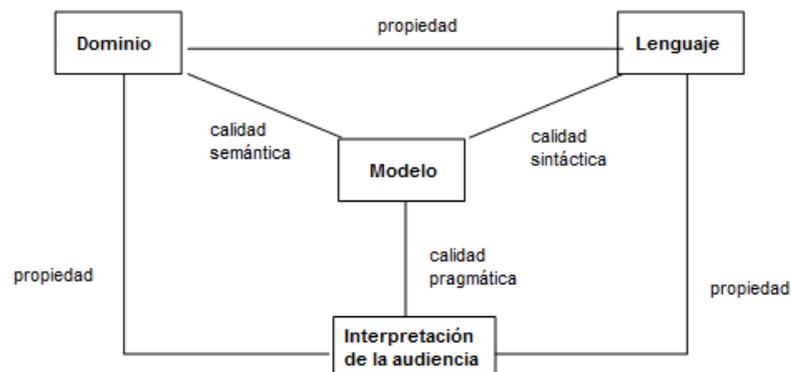


Figura 2. Modelo de marco de calidad de Lindland et al.

- NATURE [41]: corresponde al marco de trabajo de Phol donde el proceso de especificación de requisitos, que en ocasiones incluye modelado conceptual, es extendido a través de tres dimensiones: la dimensión de especificación (grado de comprensión de requisitos), la dimensión de representación (grado de formalidad) y la dimensión de acuerdo (grado de acuerdo entre interesados/clientes).

- FRISCO [39]: en él se identifican los medios de comunicación y las áreas relacionadas que pueden ser examinadas en un marco semiótico (signos). Se distinguen diferentes capas de comunicación: física, empírica, sintáctica, semántica, pragmática y social. Éstas las podemos dividir en dos grupos para diferenciar entre métodos técnicos (física, empírica y sintáctica) y métodos sociales (semántica y pragmática).

En el marco de Krogstie se distinguen objetivos y medios, de manera que los objetivos son el qué conseguir y los medios el cómo se consigue. Se basa en una vista real, reconociendo que estos modelos se suelen crear como parte del diálogo entre los

participantes involucrados en el modelado. Está fuertemente relacionado con conceptos lingüísticos y conceptos semióticos (signos). En particular, el núcleo del marco que incluye la discusión en la sintaxis, en la semántica y en la pragmática, es paralelo al uso de estos términos semióticos. Tales niveles semióticos afectan a la calidad en diferentes niveles, de ahí que se incluyan las dimensiones: física, empírica y social.

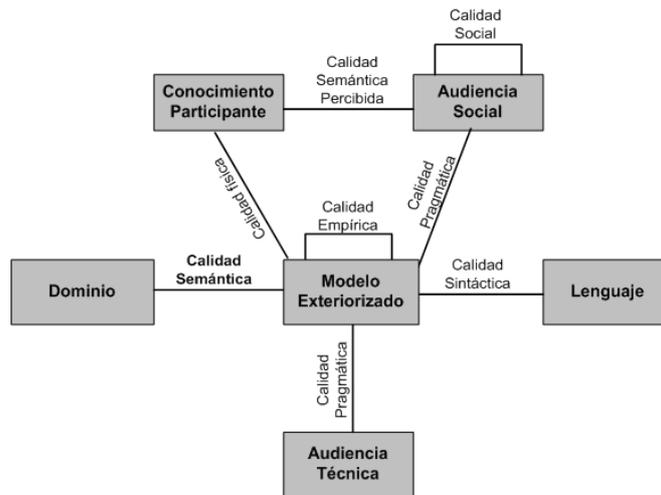


Figura 3. Marco de calidad de Krogstie para el Modelado Conceptual (simple)

En [16] Krogstie presenta un nuevo marco para evaluar UML (versión 1.4) profundizando en las áreas que más cree que pueden necesitar ser mejoradas, para ello añade la dimensión de *calidad organizacional* y divide la dimensión de *calidad pragmática* en *calidad pragmática social* y *calidad pragmática técnica*.

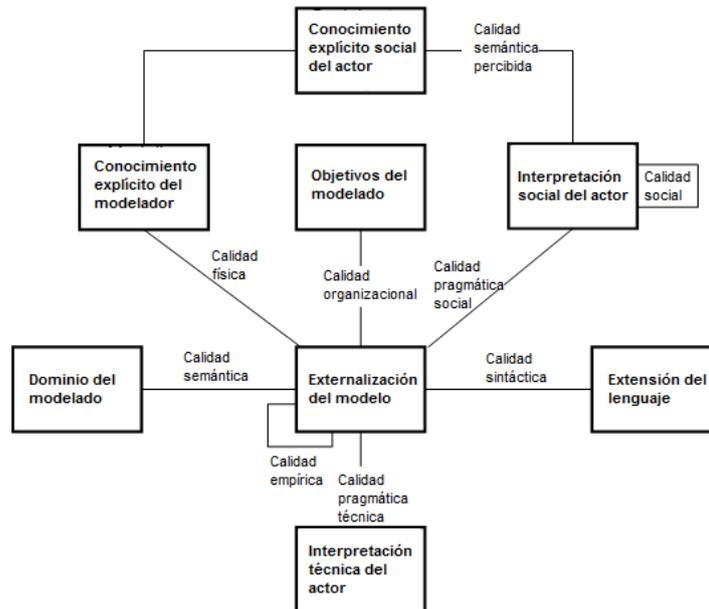


Figura 4. Marco de calidad de Krogstie para el Modelado Conceptual (completo)

Los conceptos de las dimensiones de calidad propuestas en el marco de Krogstie son los siguientes:

1.- Calidad Física: capacidad de externalizar el conocimiento del dominio en un lenguaje de modelado o de texto (descripción) actualizable. Se accede a las descripciones (partes que se necesitan) de manera eficiente.

2.- Calidad Organizacional: se refiere a que todos los estados del modelo contribuyen a satisfacer los objetivos del modelado (validación) y todos los objetivos de modelado son dirigidos a través del modelo (completitud). Ayuda a crear un uso eficiente del lenguaje de modelado (partes a utilizar) y a la trazabilidad.

3.- Calidad Empírica: se manejan errores frecuentes y previsibles de cuando un modelo es escrito o leído (legibilidad) por varios usuarios. Para ello se utilizan

*guidelines*⁷ y modelos gráficos estéticamente agradables para mejorar la comprensión de los modelos.

4.- Calidad Pragmática (técnica): corresponde a la comprensión de los modelos por parte de las personas y a partir de herramientas (se pueden incluir índices, referencia, glosarios, etc.).

5.- Calidad Social: es el acuerdo entre varias interpretaciones de personas en conexión con el lenguaje de modelado y sus extensiones. Una guía de notación es importante.

6.- Calidad Semántica: sus objetivos son la validación del modelo en cuanto a que todos sus aspectos (estados) son correctos (descripciones consistentes) en referencia al dominio y la completitud, respecto a que el modelo contiene todos los aspectos que se encuentran en el dominio.

7.- Calidad Sintáctica: para el lenguaje en el que se escribe el modelo. Todos los estados del modelo deben ser sintácticamente correctos. Los errores comunes son la invalidez sintáctica por términos que no pertenecen al lenguaje y la no completitud, porque faltan partes en el modelo o texto necesarios para seguir la gramática del lenguaje.

Los atributos de calidad se clasifican de la siguiente manera:

Tabla 5. Clasificación de los atributos de calidad a partir del marco de Krogstie

Dimensión de calidad	Atributos de calidad
Calidad semántica	Completitud, consistencia, corrección, precisión, ambigüedad, concisión
Calidad pragmática	Comprensibilidad
Calidad social	Comunicación
Calidad física	Granularidad

⁷ Las *guidelines* son guías que forman un conjunto de instrucciones, directrices, afirmaciones o recomendaciones, desarrolladas de forma sistemática cuyo propósito es ayudar a realizar alguna tarea o llegar a algún objetivo. En el presente trabajo se incide notablemente en *guidelines* para construir CU, principalmente para comprobar su efectividad.

Calidad organizacional	Reusabilidad, trazabilidad, mantenibilidad
------------------------	--

3.2 Aproximaciones para especificar Casos de Uso de calidad

Existen varios trabajos con la finalidad de ayudar a construir CU de calidad. Entre ellos tenemos el de Phalp [36], Cockburn [47], Rolland [48], Lilly [49] y el de Ramos [37].

En el trabajo de Phalp et al. [36] se describe el uso del *checklist* como propuesta para comprobar la calidad de los CU que pueden ser utilizados para examinar guías y para verificar que se cumplen los atributos de calidad deseados. El artículo está dirigido a examinar la amplitud de las guías de CU presentadas, para explorar la razón de ser de las guías, y considerar la comprensión de textos para detectar calidad en CU. El *checklist* resultante ha sido creado llevando una investigación conjunta en IR e intentando producir un conjunto coherente de cualidades deseables de CU.

Cockburn et al. [47], a sabiendas de que no existen guías fiables para escribir o revisar CU lanza un libro donde presenta una guía para construir CU. En el libro se explican puntos como conceptos básicos, conceptos relacionados, recordatorios (frases clave para recordar a la hora de qué se debe hacer y qué no se debe hacer para un buen caso de uso), ejemplos y plantillas, respuestas a ejercicios, otras opiniones, respuestas a e-mails de consulta y trabajo relacionado.

Rolland et al. [48] ayuda a descubrir expresiones incompletas, elementos que faltan, casos excepcionales y partes de la especificación del caso de uso por medio de unas reglas de conocimiento. Estas reglas de conocimiento se derivan de una guía que permite construir CU bajo el formato de escenarios, texto en lenguaje natural y con el compromiso de construir descripciones sin ambigüedad. Se trata de un proceso guiado donde la progresiva transformación parcial y total del lenguaje bien estructurado integra especificaciones de CU. La guía de construcción toma la forma de reglas sobre tipos de acciones de dependencia, relaciones entre acciones y condiciones de flujo, propiedades de objetos y agentes, etc.

Lilly [49] presenta 10 problemas de proyectos reales utilizando CU, problemas más bien que suelen tener personas con no mucha experiencia. La lista de los 10 son: los límites del sistema están indefinidos o inconsistentes, los CU están escritos bajo el punto de vista del sistema (y no de los actores), los nombres de los actores son inconsistentes, hay demasiados CU, las relaciones entre actores y caso de uso parecen una tela de araña, las especificaciones de CU son demasiado largas, son confusas, no describen correctamente la funcionalidad, el cliente no los comprende, no terminan nunca.

En el trabajo de Ramos et al. [37] se trata de mejorar la calidad de los documentos de requisitos basados en CU. Para evaluar la calidad se incorporan guías para que los participantes (analistas) puedan medir de manera efectiva, y por tanto, mejorar la calidad de los documentos de requisitos. Para ello se usa AIRDoc; se trata de un proceso en el que se soporta la evaluación y mejora de los documentos basados en CU usando refactorización de requisitos adecuados o patrones. Se demuestra como el proceso de mejora es satisfactorio aplicado en un documento de requisitos industrial.

En nuestro estudio se han utilizado los atributos propuestos por Davis y el marco de calidad de Krogstie para el modelado conceptual. Estas propuestas estudiadas en su momento para identificar la calidad en la ERS las aprovechamos para clasificar la información obtenida. Respecto a las aproximaciones presentadas para especificar Casos de Uso llegamos a la conclusión de que pueden servir como guía pero no se garantiza el éxito de su uso, ya que no está demostrado empíricamente que aporten calidad, motivo por el que no se han incluido como estudios primarios en la presente investigación.

En el siguiente capítulo se explica en detalle el método de investigación utilizado para adquirir la mayor cantidad de información posible - basada en evidencias - y así identificar las características (atributos de calidad) que actualmente están siendo consideradas por los investigadores para medir la eficacia real de los CU y verificar la existencia de *formatos de descripción*⁸ que pueden promover la calidad de CU.

⁸ Los CU pueden ser presentados de diferentes maneras. Clasificamos las descripciones en: narración de texto, escenarios, plantillas y diagramas.

Capítulo 4

Protocolo de Revisión

Los siguientes puntos indican el procedimiento utilizado para ejecutar el estudio secundario empezando por observar el protocolo de revisión diseñado, seguido de las partes en las que está estructurado dicho protocolo, y las amenazas del diseño escogido.

4.1 Preguntas de investigación

Existen diferentes guías para diseñar protocolos de revisión en numerosos campos de investigación en forma de *guidelines* y de *checklists*. En la red se encuentran, sobre todo, guías relacionadas con el campo de las humanidades.

En esta sección se presenta el protocolo utilizado para llevar a cabo nuestra revisión sistemática, el cual se basa en la guía aportada por Kitchenham [17].

Este protocolo fue inicialmente diseñado por Travassos et al. [65], en la que se establecía las pautas generales para guiar a la selección, evaluación de calidad, y extracción de información. Sin embargo, este protocolo se ha ido refinando en el transcurso de la ejecución de la misma (por ejemplo, para la definición de criterios de búsqueda).

La revisión sistemática apunta a responder una serie de incógnitas qué influye y qué puede influir en la calidad de los Casos de Uso. La pregunta principal es la siguiente:

“¿Qué se debe observar para evaluar la calidad de un Caso de Uso?”

A partir de esta pregunta surgen las siguientes:

- ¿Cuáles son los aspectos de calidad más investigados en el desarrollo de CU?
¿cuáles son las características más importantes de los CU que pueden afectar a la calidad?
- ¿Qué procedimiento habría que seguir para construir el mejor modelo de CU que especifique las necesidades del cliente?
- ¿Cómo se pueden capturar todos los escenarios de un sistema?
- ¿Existe algún formato de descripción que pueda incrementar o garantizar la calidad en especificaciones de CU y que pueda ser así para diferentes tipos de sistemas? ¿en qué dominios son investigados estos aspectos?

Una revisión sistemática implica llevar a cabo un análisis de la información existente dentro del conjunto de la población, de intervenciones, comparaciones y resultados.

Tabla 6. Componentes de la revisión del problema

Población	Conjunto de artículos que describen estudios empíricos llevados a cabo por el ámbito industrial, académico y público con la finalidad de experimentar con Casos de Uso.
Intervenciones	En el caso de comparar evaluaciones, se trata de conocer los diferentes factores o formatos bajo estudio que pueden conducir a que afecten a los resultados, en este caso a la calidad de Casos de Uso.
Resultados	Formatos y procedimientos a seguir para identificar y obtener calidad en Casos de Uso.
Diseños de estudio	Experimentos, casos de estudio e informes empíricos.

4.2 Estructura del protocolo de revisión

4.2.1 Estrategia de búsqueda

La estrategia de búsqueda comprende la selección de palabras clave y los motores de búsqueda a utilizar. Cabe destacar que se hará uso únicamente de la lengua inglesa.

Palabras clave

La selección de palabras consiste en dos partes. La primera parte comprende el tipo de estudio que se desea incluir en la revisión sistemática. La segunda parte está relacionada con la tecnología específica a revisar. Las dos partes se deben combinar para establecer la cadena de palabras final. La primera parte está formada por una cadena de palabras clave mientras la segunda está compuesta por cuatro cadenas.

Primera cadena de palabras

Con relación al estudio empírico a emprender para llevar a cabo la revisión, las palabras consideradas son:

1. Experiment,
2. Quasi-experiment,
3. “Case Study”,
4. “Experience Reports”.

A fin de incrementar el número de experimentos relevantes, se debe considerar algunos sinónimos de la palabra “experiment”; tales como: *experimental study*, *experimental comparison*, *experimental análisis*, *experimental evidence*, *empirical study*, etc.

De acuerdo a recuperar el material con la información más relevante, se decide usar los sinónimos más comunes:

5. “Experimental study”,

6. "Experimental comparison",
7. "Experimental analysis",
8. "Experimental evidence",
9. "Empirical study".

Combinando dichos términos, se obtienen resultados más precisos.

Segunda cadena de palabras

Con relación al objeto a analizar en el estudio, es necesario encontrar los términos apropiados para profundizar en dicho objeto, calidad de CU. En un segundo nivel se decide investigar en una población, esta población es la de Ingeniería de Software - o Ingeniería de Requisitos para ser más exactos. Por tanto, los siguientes términos son considerados para la cadena de búsqueda:

10. "Requirements engineering",
11. "Requirements specification",
12. "Requirements elicitation",
13. "Modeling language",
14. "UML-based development",

Tercera cadena de palabras

Además, de dichos términos se deben añadir los siguientes y así incrementar el nivel de detalle:

15. UML,
16. "Use Case".

Cuarta cadena de palabras

En cuarta instancia, encontrar los términos que se refieren a las características que pueden afectar a la calidad de CU. Por tanto, los siguientes términos son considerados para la cadena de búsqueda:

17. Quality,
18. Procedure,
19. Template,
20. Diagram,
21. Description,
22. Guideline,
23. Outline,
24. Scenario,
25. Feature,
26. "Specification style".

Una vez concluido este punto se construye la siguiente combinación para extraer el material de estudio primario. Usamos internamente los operadores OR para enlazar las palabras clave de la misma cadena y el operador AND para unir las cadenas, y así obtener la cadena de búsqueda final:

Tabla 7. Combinación para la obtención de la cadena de búsqueda

$([1-9] \text{ OR } *) \text{ AND } ([10-14] \text{ OR } *) \text{ AND } ([15-16] \text{ OR } *) \text{ AND } ([17-26] \text{ OR } *)$
--

Motores de búsqueda

El motor de búsqueda a utilizar es el de SCOPUS (www.scopus.com), donde se escribe la siguiente cadena de búsqueda construida finalmente para adquirir los artículos. A partir de dicha cadena, el SCOPUS rastrea los campos Título, Abstract y Palabras Clave de la literatura indexada.

((experiment OR quasi-experiment OR "Case study" OR "Experience Reports" OR experimental AND (study OR comparison OR analysis OR evidence)) OR "Empirical

study") AND ((requirements AND (engineering OR specifications OR elicitation)) OR "modelling language" OR "UML-based development") AND (UML OR "Use case") AND (quality OR procedure OR template OR diagram OR description OR guideline OR outline OR scenario OR feature OR "specification style")

Como se ha mencionado, a través de esta herramienta se presentan los artículos que ha sido capaz de encontrar. Sin embargo, no todos los artículos están disponibles para su adquisición. A parte de SCOPUS hemos tenido que usar otros recursos como el Polibuscador (UPV), ACM digital library, IEEE Xplore, CiteSeer y Google. Aún así, muchos artículos no están accesibles en Internet.

Criterios de inclusión

- 1.1.1. Se habla sobre calidad de CU.
- 1.1.2. Presenta características de diagramas de CU.
- 1.1.3. Presenta características de especificaciones de CU.
- 1.1.4. Analiza la búsqueda o modelado de requisitos respecto a CU.
- 1.1.5. Presenta experimentos con CU.
- 1.1.6. Presenta comparaciones con descripciones, modelos o guías para representar CU.
- 1.1.7. El artículo debe de estar publicado después del año 1986, cuando Ivar Jacobson - importante colaborador en UML y en RUP - formuló por primera vez la técnica de modelado visual especificando CU.

Criterios de exclusión

- 1.1.8. Se habla sobre CU sin profundizar en ellos.
- 1.1.9. No se tiene en cuenta ninguno de los criterios citados en el punto anterior.

4.2.2 Proceso de selección de estudio

El proceso de selección de estudio es un proceso previo a la revisión, consiste en seleccionar un artículo en particular como estudio primario para la revisión sistemática. Como se muestra en la Tabla 8, el proceso comprende cuatro iteraciones: las tres primeras iteraciones son llevadas a cabo por dos revisores y la última iteración es llevada a cabo por un experto en IS. Los participantes para la selección del estudio son:

- Revisores: José A. Ferrando y Nelly Condori.
- Experto: Óscar Pastor.

Tabla 8. Proceso de selección de estudio

Proceso de selección de estudio			
Revisores (mínimo 2)	Primera iteración	<p>Revisar todos los resultados de la búsqueda</p> <hr/> <p>Título y Resumen</p>	<p>- Incluido - Excluido - Indeterminado</p> <p>Estado</p>
	<i>Si existen artículos donde la naturaleza del estudio no está clara</i>		
	Segunda iteración	<p>Revisar artículos clasificados como indeterminados</p> <hr/> <p>Introducción y Conclusiones</p>	<p>- Incluido - Excluido - Indeterminado</p> <p>Estado</p>
Expertos	Tercera iteración	<p>Consenso entre revisores</p> <hr/> <p>Título, Resumen, Introducción y Conclusiones</p>	<p>- Incluido - Excluido</p> <p>Estado potencial</p>
	Última iteración	<p>Revisión del estado potencial asignado</p> <hr/> <p>Artículo completo y justificaciones dadas por los revisores</p>	<p>- Incluido - Excluido</p> <p>Estado final</p>

4.2.3 Evaluación de la calidad del estudio

La calidad de un estudio es juzgada a partir de un conjunto de preguntas presentadas a continuación:

- Criterio relacionado con las características de calidad de CU.
 - Q1: ¿Existe alguna descripción de cómo los atributos de calidad han sido identificados? (1 punto).
 - Q2: ¿Existe alguna descripción sobre las restricciones y condiciones donde la calidad de los atributos fueron observados? (1 punto).
 - Q3: ¿Existe alguna descripción sobre cómo la calidad de los atributos puede ser medida? (1 punto).
- Criterio relacionado al formato de descripción.
 - Q4: ¿Describe el artículo alguna adaptación/evolución de una pre-existente aproximación? (1 punto).
 - Q5: ¿El formato de descripción es referenciado/evaluado/utilizado en otros trabajos? (0,5 puntos x número de referencias).
 - Q6: ¿Describe el artículo una aplicación del formato de descripción utilizado para su evaluación? (0,5 puntos).
 - Q7: ¿Existe algún resultado empírico/experimental respecto al formato de descripción? (2 puntos).
 - Q8: ¿Es posible identificar para qué tipos de sistemas puede ser usado el formato de descripción? (0,5 puntos x cada tipo de sistema).
 - Q9: ¿Es posible evaluar qué atributos de calidad promueve el formato de descripción? (0,5 puntos x atributo de calidad).

4.2.4 Extracción de información

Para cada artículo seleccionado se extrae la siguiente información.

Tabla 9. Información a extraer

Campo	Descripción
-------	-------------

Título	El título del artículo
Autores	Lista de autores del artículo
Año de Publicación	El año cuando el artículo fue publicado
Fuente de la Publicación	Nombre de la revista, conferencia o lugar donde el artículo fue publicado
Resumen	Resumen completo del artículo
Atributos de Calidad de CU	La lista de atributos de calidad
Medidas de los atributos de Calidad de CU	Medidas normalmente tomadas que pueden ser utilizadas para valorar el significado/interpretación de los atributos
Formato de Descripción de CU	El formato utilizado para describir un Caso de Uso
Tipos de Sistema donde el CU ha sido aplicado.	Lista de tipos de sistema donde el formato de descripción ha sido utilizado/experimentado

4.2.5 Síntesis de la información

La agregación o síntesis de la información extraída se construye de manera descriptiva. La estructura en esta sección se compone a partir de los: estudios considerados, las tareas y sujetos que las llevan a cabo, heurísticas y esquemas de marcado, hipótesis y validaciones, y la conclusión de la síntesis.

4.3 Amenazas de la validez del diseño de la Revisión Sistemática

Desde una institución académica y pública, como es la Universidad Politécnica de Valencia, se ha llevado a cabo el estudio secundario cuyo protocolo de revisión acabamos de ver. Por tanto, el trabajo se ha realizado desde un ámbito estrictamente académico sin ningún acceso ni contacto con la parte profesional. Una amenaza para los profesionales puede ser la capacidad limitada para acceder a los estudios y emprender una revisión. También señalar la amenaza por parte nuestra de no poder contar con los estudios que pueden haber realizado las mismas empresas.

Por otra parte, el recurso utilizado (Scopus) y la cadena de búsqueda formada pueden llevar a limitaciones en los resultados. De los recursos existentes unos serán más eficaces que otros, incluso existirán estudios inaccesibles para este tipo de

herramientas. En cuanto a la cadena de búsqueda, en cualquier estudio se pueden utilizar diferentes términos para referirse al mismo campo en el que estamos involucrados. Además todo esto teniendo en cuenta que el medio utilizado ha sido la “red de redes” (Internet), con la posibilidad de la existencia de estudios no incluidos en la red.

Capítulo 5

Conducción de la Revisión Sistemática

Entramos en el núcleo del presente trabajo para focalizar en cada punto seguido desde el protocolo visto en el capítulo anterior. En este capítulo se expone el resultado de emprender la revisión sistemática: los estudios seleccionados, la información extraída, las diferentes clasificaciones, la evaluación final y la síntesis de la información obtenida. Todo ello con las diversas curiosidades y dificultades encontradas en el proceso.

5.1 Selección de los estudios

Una vez diseñado nuestro protocolo de revisión y ejecutar la estrategia de búsqueda presentada en el capítulo anterior, un total de 123 artículos fueron obtenidos. Siguiendo el proceso de selección, en una primera iteración, 41 fueron seleccionados como relevantes, 16 estudios considerados como indeterminados (marcado de esta manera porque no han sido encontrados para comprobarse) y 66 estudios considerados como excluidos. De los 41 que se han incluido, en una segunda iteración se detectó que 20 estudios contenían algún tipo de evidencia empírica. También, cabe tener en cuenta que solamente 10 estudios se refieren concretamente a la mejora de Casos de Uso, mientras 5 se refieren a UML en general (sin centrarse en mejorar los Casos de Uso en concreto) y otros 5 estudios se refieren a mejorar las especificaciones de requisitos en general (sin centrarse en mejorar los Casos de Uso en concreto), a esta categoría la denominamos: Especificaciones de Requisitos Software (ERS).

Los criterios en los que nos basamos para su división en ERS son los siguientes:

- Un ERS puede incluir CU o estar representado por los mismos.
- Si se mejora la calidad en un ERS puede que también la mejore en CU, pero no siempre.
- Los CU se consideran una técnica de especificación de requisitos.
- Si se consigue incrementar la calidad de los CU se logrará mejorar la calidad del ERS que lleve consigo CU.

En cuanto a UML consideramos que:

- UML es un lenguaje de modelado que está formado por un número de diagramas, entre ellos los de CU.
- Se podría afirmar que si se mejora UML se podría mejorar el diagrama de CU, pero solamente el diagrama. Un modelo de casos de uso incluye su diagrama y su especificación textual que puede ser representado por medio de plantillas estructuradas o no. Cuando se menciona a UML, solo se verá la notación gráfica del modelo de casos de uso.
- Si se consigue mejorar la calidad en los diagramas de CU estaremos de acuerdo que se estará mejorando una parte de UML.

Por estos motivos en este trabajo clasificamos los artículos por tipos, según la técnica de especificación que se tiene como objetivo evaluar. Por lo tanto, según la finalidad del artículo mencionamos que es de tipo UML⁹, de tipo ERS o de tipo CU¹⁰.

Tabla 10. Distribución de la literatura encontrada

123 En Total		
41 ID	16 UD	66 ED

⁹ Aunque UML no es realmente una técnica - es un lenguaje de modelado utilizado para especificar requisitos -, la incluimos como técnica para proceder a la clasificación de los artículos.

¹⁰ Hay algún artículo como [84] que se basa en construcción de CUs pero al no referirse a calidad en CU no se tiene en cuenta para que sea de tipo CU, se considera de tipo ERS.

33 F			8 NF
20 EE		13 NEE	
10 CU	5 UML	5 ERS	

ID = Incluidos
UD = Indeterminados
ED = Excluidos
F = Encontrados
NF = No Encontrados
EE = Evidencias Empíricas
NEE = No Evidencias Empíricas

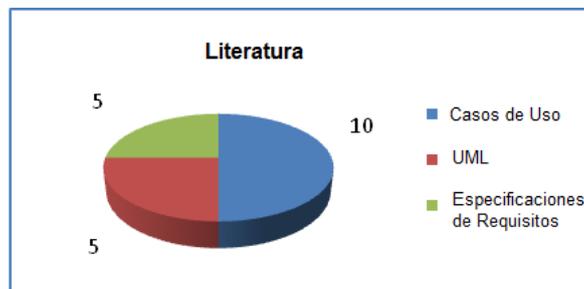


Figura 5. Distribución de la literatura incluida

Como se ve en los gráficos, todos los artículos considerados en un principio no tratan de efectuar evaluaciones empíricas para mejorar la construcción de descripciones de CU – aunque mencionen o utilicen CU tienen objetivos diferentes. Se han incluido artículos que no evalúan directamente la especificación de CU, sino que evalúan la calidad de las ERS y de UML en general donde normalmente se utilizan los CU (los CU aparecen en segundo plano). Aprovechando que tales artículos forman parte de los resultados obtenidos en nuestra búsqueda hemos pensado que indirectamente pueden mejorar los CU.

Por ejemplo Kamsties et al. [81] evalúa un experimento para comparar dos estilos diferentes de ERS, el de caja negra y el de caja blanca. A través del estilo de caja negra un sistema es descrito por su comportamiento visible en el exterior y se pretenden omitir los detalles internos; con el estilo de caja blanca un sistema es descrito por el comportamiento de sus entidades (objetos, módulos, etc.). El uso de

uno de estos estilos puede beneficiar a las descripciones de CU o no (una importante tarea sería la de investigar si estos estilos incorporan CU, en principio parece que no). Los autores concluyen que el estilo de caja negra resulta más comprensible desde el punto de vista del cliente. Además, señalan como una de las futuras propuestas para otros investigadores tratar de saber cuál de los dos estilos se debería seguir cuando se usa UML. El artículo mencionado lo clasificamos de tipo ERS.

España et al. [86] evalúa la calidad de ERS en términos de completitud y granularidad comparando el uso de CU con el Análisis de Comunicación, en el que el Análisis de Comunicación obtiene mejores resultados. Pero no se trata en ningún caso de mejorar la construcción en CU. También lo clasificamos como artículo de tipo ERS.

Gravino et al. [94] lleva a cabo un estudio empírico dirigido a evaluar si el modelado dinámico proporciona precisión a la hora de adquirir los requisitos del cliente. Además, efectúa un experimento controlado para evaluar la comprensibilidad de dichos requisitos. Como el modelado dinámico afecta a UML decimos que el artículo es de tipo UML.

Bernardez et al. [80] propone una técnica basada en métricas (MBR) para identificar tipos específicos de defectos en CU. Inspecciona si el proceso es más efectivo y eficiente en comparación con la técnica de *checklists*. Este trabajo estudia mejorar la calidad en descripciones de CU, por tanto, es un artículo de tipo CU.

Después de presentar algunos ejemplos de cómo clasificamos los artículos según la técnica, en la Tabla 11 se exponen los artículos seleccionados identificados por un código junto con la información: título, autores, finalidad (de manera muy resumida) y la técnica que el estudio pretende mejorar (UML, ERS o CU). Respecto a la finalidad, cabe destacar que existen 3 artículos en los que se preocupan por estudiar la detección de defectos (uno en CU y dos en UML) y 6 que resaltan la construcción de CU a partir de la especificación de *guidelines*. El resto de artículos se pueden considerar como “independientes” ya que no se pueden comparar con otros por finalidad.

La nomenclatura utilizada para obtener el código se ha generado de la siguiente manera:

- Se han ordenado por título los artículos obtenidos en primera instancia (incluidos, excluidos e indeterminados).
- Se han numerado.

El código se ha formado a partir del número marcado seguido por las tres primeras palabras del título y por el apellido del primer autor.

Tabla 11. Estudios seleccionados

Código	Título	Autores	Finalidad	Técnica
5_A_controlled_experiment___Bernardez [80]	A controlled experiment for evaluating a metric-based reading technique for requirements inspection	Bernardez B., Genero M., Duran A., Toro M.	Detección de defectos en Casos de Uso	CU
7_A_controlled_experiment___Kamsties [81]	A controlled experiment to evaluate how styles affect the understandability of requirements specifications	Kamsties E., Von Knethen A., Reussner R.	Comparación de dos estilos de Especificación de Requisitos	ERS
17_An_approach_to___Pow-Sang [82]	An approach to determine software requirement construction sequences based on use cases	Pow-Sang J.A., Nakasone A., Imbert R., Moreno A.M.	Uso de Diagramas de precedencia de Casos de Uso	UML
20_An_experimental_investigation___Briand [83]	An experimental investigation of formality in UML-based development	Briand L.C., Labiche Y., Di Penta M., Yan-Bondoc H.	Mejora de UML con el uso de OCL	UML
29_Athena:_A_collaborative___Laporti [84]	Athena: A collaborative approach to requirements elicitation	Laporti V., Borges M.R.S., Braganholo V.	Se analiza la herramienta Athena para construir ERS	ERS
40_Effects_of_defects___Lange [85]	Effects of defects in UML models - An experimental investigation	Lange C.F.J., Chaudron M.R.V.	Detección de defectos en UML	UML
44_Evaluating_the_completeness___Espana [86]	Evaluating the completeness and granularity of functional requirements specifications: A controlled	Espana S., Condori-Fernandez N., Gonzalez A., Pastor O.	Comparación de Casos de Uso con Análisis de Comunicación para analizar calidad en ERS funcionales	ERS
46_Experimental_comparison_of___Laitenberger [87]	Experimental comparison of reading techniques for defect detection in UML design documents	Laitenberger O., Atkinson C., Schlich M., El Emam K.	Detección de defectos en UML	UML
52_Guiding_use_case___Ben [88]	Guiding use case authoring: results of an empirical study	Ben Achour C., Rolland C., Maiden N.A.M., Souveyet C.	Se presentan guías para la construcción de Casos de Uso	CU

73_On_the_comprehension___Hogganvik [89]	On the comprehension of security risk scenarios	Hogganvik I., Stolen K.	Se presenta el lenguaje CORAS para documentar escenarios de riesgo de seguridad	ERS
78_Prioritizing_scenario_evolution__Moisiadis [90]	Prioritizing scenario evolution	Moisiadis Frank	Expansión de escenarios	CU
84_Replicating_the_CREWS___Cox [91]	Replicating the CREWS use case authoring guidelines experiment	Cox K., Phalp K.	Se presentan guías para la construcción de Casos de Uso	CU
92_Software_requirements_retrieval___Udomchaiporn [92]	Software requirements retrieval using use case terms and structure similarity computation	Udomchaiporn A., Prompoon N., Kanongchaiyos P.	ERS y Descripciones de Casos de Uso	ERS
97_Systematic_use_case___Islam [93]	Systematic use case interviews for specification of automotive systems	Islam S., Omasreiter H.	Proceso para crear Casos de Uso	CU
118_An_empirical_investigation___Carmine [94]	An empirical investigation on dynamic modeling	Carmine Gravino, Giuseppe Scanniello, Genoveffa Tortora	Evaluación de la precisión de los requisitos a través de modelado dinámico	UML
119_Quality_and_understandability___Bente [95]	Quality and understandability of use case models	Bente Anda, Dag Sjoberg, Magne Jorgensen	Se presentan guías para la construcción de Casos de Uso	CU
120_Comparing_Use_Case___Cox [96]	Comparing Use Case Writing Guidelines	Cox Karl, Phalp Keith, Shepperd Martin	Se presentan guías para la construcción de Casos de Uso	CU
121_Improving_the_quality___Phalp [97]	Improving the quality of use case descriptions: empirical assessment of writing guidelines	Phalp K.T., Vincent J., Cox K.	Se presentan guías para la construcción de Casos de Uso	CU
122_Guiding_the_process___Mustapha [98]	Guiding the process of requirements elicitation trough scenario analysis: results of an empirical study	Mustapha Tawbi, C. Ben Achour, Fernando Vélez	Se presentan guías para la construcción de Casos de Uso	CU

123_An_Empirical_Study___Irit [99]	An Empirical Study of Requirements Model Understanding: Use Case vs. Tropos Models	Irit Hadar, Tsvi Kuflik, Anna Perini	Comparación entre Casos de Uso y Tropos	CU
------------------------------------	--	---	--	----

5.2 Extracción de información

Nuestra estrategia en la extracción de la información para artículos relevantes sobre CU está dividida en dos partes: información descriptiva e información técnica. La información descriptiva se refiere a: código del artículo, autores, año de publicación, fuente y resumen. La información técnica incluye: atributos de calidad de CU, atributos de medida de calidad de CU, dimensión de calidad, formato de descripción y dominio al que pertenecen los sistemas en los cuáles se han experimentado.

Para cada artículo seleccionado se ha extraído la siguiente información (Tabla 12) y la hemos almacenado y estructurado apoyándonos en el programa de código abierto JabRef, adquirido desde <http://jabref.sourceforge.net/>.

Tabla 12. Información a extraer

Campo	Descripción
Título	El título del artículo
Autores	Lista de autores del artículo
Año de Publicación	El año cuando el artículo fue publicado
Fuente de la Publicación	Nombre de la revista, conferencia o lugar donde el artículo fue publicado
Resumen	Resumen completo del artículo
Atributos de Calidad de CU	La lista de atributos de calidad
Medidas de los atributos de Calidad de CU	Medidas normalmente tomadas que pueden ser utilizadas para valorar el significado/interpretación de los atributos
Formato de Descripción de CU	El formato utilizado para describir un Caso de Uso
Tipos de Sistema donde el CU ha sido aplicado.	Lista de tipos de sistema donde el formato de descripción ha sido utilizado/experimentado

Entre los atributos de calidad encontrados (Davis) [31], tenemos los siguientes: completitud, consistencia, corrección, precisión, ambigüedad, comprensibilidad, comunicabilidad, concisión, granularidad, reusabilidad y legibilidad. Además, también consideramos la eficiencia como atributo de calidad – como la satisfacción de llegar al objetivo en el menor tiempo posible- porque es muy común para llevar a

cabo mejoras, aunque no lo incorporamos en ninguna dimensión de calidad. Por tanto, clasificamos las dimensiones de calidad de Krogstie [30], con sus respectivos atributos, frecuencia y porcentaje con respecto a la literatura incluida.

En la Tabla 14 se muestran la lista de artículos junto con la información extraída respecto a la Fuente, Atributos de Calidad, Dimensión de calidad, Formato de descripción y Dominio del sistema.

La información referente a medidas de los atributos de calidad (o métricas) no se ha considerado relevante por la carencia de datos en la mayoría de los artículos. Por ejemplo en el trabajo de Achour et al. [88] se presentan las *CREWS guidelines* que indican cómo se tienen que construir los CU para que este procedimiento sea más efectivo y eficiente. Esta finalidad se basa en términos de corrección, ambigüedad, completitud, etc., pero no indican unas métricas explícitamente aunque sí se da una explicación de cómo se miden los atributos para su evaluación de la calidad. Otro artículo que podemos mencionar es el de Phalp et al. [97] en el que se presentan las heurísticas (las 7 Cs) para evaluar el uso de unas guías construidas en términos de comprensibilidad a partir de las guías CREWS más simplificadas, pero no se especifican métricas ni se indica cómo se hace la medición.

Para el formato de descripción nos podríamos referir solamente a los artículos de tipo CU, aunque si son de tipo ERS o UML también se puede en ocasiones deducir en qué medida afectarían a los CU, es decir, en qué formato de descripción se verían aludidos. Por ejemplo si el artículo es de tipo UML, sabemos que en caso de afectar a los CU lo hará en el formato de diagrama.

En la Tabla 14 se observan celdas vacías, esto es debido a que no se especifica tal información. Le sucede en ocasiones al dominio del sistema y al formato de descripción. El dominio del sistema a veces no se indica como en el trabajo de Lange et al. [85] y el formato de descripción se puede suponer en ocasiones pero en otras no.

Existen situaciones en las que se trata de mejorar todos los formatos de descripción o de mejorar las descripciones de CU de forma general. Bernardez et al. [80] y Udomchaiporn et al. [92] presentan aproximaciones de ERS en base a una estructura de CU, sin especificar formato de descripción. No obstante, artículos como el de Kamsties et al. [81] no nombran para nada ningún formato de descripción, no se

mencionan CU, tratándose en general de ERS (no se puede deducir ningún formato de descripción ya que ni siquiera se tienen en cuenta los CU).

Los CU pueden estar representados por diferentes formatos. Nosotros hemos sacado en conclusión 4 tipos de formato: diagramas, escenarios, plantillas y narraciones de texto. Los diagramas son modelos gráficos, los escenarios son descripciones de funcionalidades individuales sin estructura, las plantillas son descripciones de funcionalidades individuales o generales con una estructura dividida por campos como *actores, descripción, disparador, precondiciones, postcondiciones*, etc., y las narraciones son “historias” narradas en lenguaje natural sin ninguna estructura o formato.

Los dominios de sistemas se han adquirido del artículo de Kotonya et al. [54], el cual describe un modelo de clasificaciones en el que se encuentra una lista de dominios de sistemas. Tales dominios se presentan en la Tabla 13.

Tabla 13. Lista de dominios

Dominios	
Aviónicos	Industrial
Comandos y control	Ingeniería de software
Sistemas embebidos	Científica
Comercio electrónico	Simulación
Computación empresarial	Telecomunicaciones
Finanzas	Transportación
Salud	Utilidades
Tiempo real	General (MIS)

De los anteriores dominios se han encontrado en nuestra literatura: sistemas embebidos, finanzas, comercio electrónico, tiempo real y por supuesto de dominio general. Los sistemas de dominio general se refieren a sistemas de información y gestión (MIS, Sistemas de Información y Gestión).

Tabla 14. Información descriptiva y técnica por estudio seleccionado

Código	Fuente/Año de Publicación	Atributos de calidad	Dimensión de calidad	Formato de descripción	Dominio del sistema
5_A_controlled_experiment___Bernardez [80]	10th IEEE International Software Metrics Symposium (METRICS 2004), 11-17 September 2004, Chicago, IL, USA, IEEE Computer Society	Corrección, Completitud, Ambigüedad, Comprensibilidad, Concisión	Calidad Semántica	Todos	Dominio general
7_A_controlled_experiment___Kamsties [81]	Information and software technology, ISSN 0950-5849, Vol. 45, N° 14, 2003, pags. 955-965	Comprensibilidad	Calidad Pragmática, Calidad Empírica		Sistemas embebidos
17_An_approach_to___Pow-Sang [82]	Proceedings of the 2008 Advanced Software Engineering and Its Applications, Pages: 17-22, Year of Publication: 2008, ISBN:978-0-7695-3432-9, IEEE Computer Society	Precisión	Calidad Semántica	Diagramas	Dominio general
20_An_experimental_investigation___Briand [83]	IEEE Transactions on Software Engineering Volume 31, Issue 10 (October 2005), Pages: 833 – 849, ISSN:0098-5589, IEEE Press	Ambigüedad, Precisión, Comprensibilidad, Mantenibilidad	Calidad semántica Calidad organizacional	Diagramas	Dominio general

29_Athena:_A_collaborative___Laporti [84]	Computers in Industry, Volume 60, Issue 6 (August 2009), ISSN:0166-3615, Elsevier Science Publishers B. V.	Efectividad, Trazabilidad, Comunicación, Ambigüedad, Consistencia, Corrección	Calidad organizacional Calidad Semántica Calidad Social	Narración / Escenarios	Dominio general
40_Effects_of_defects___Lange [85]	28th International Conference on Software Engineering (ICIS 2006), Shanghai, China, May 20-28, 2006. ACM	Consistencia, Compleitud, Comprensibilidad	Calidad semántica	Diagramas	
44_Evaluating_the_completeness___Espana [86]	IR 2009, 17th IEEE International Requirements Engineering Conference, Atlanta, Georgia, USA, August 31 – September 4, 2009. IEEE Computer Society	Granularidad, Compleitud	Calidad Semántica	Diagramas	Dominio general
46_Experimental_comparison_of___Laitenberger [87]	Journal of Systems and Software Volume 53 , Issue 2 (August 2000), Pages: 183 – 204, ISSN:0164-1212, Elsevier Science Inc	Efectividad, Corrección, Consistencia, Compleitud	Calidad Semántica	Diagramas	Dominio general
52_Guiding_use_case___Ben [88]	Proceedings IEEE Symposium on Requirements Engineering, IEEE Comput. Soc, Los Alamitos, CA, USA, 1999	Compleitud, Corrección, Consistencia, Precisión, Eficiencia	Calidad Empírica, Calidad Semántica	Escenarios	Sistema embebido
73_On_the_comprehension___Hogganvik [89]	13th International Workshop on Program Comprehension (IWPC 2005), 15-16 May 2005, St. Louis, MO, USA IEEE Computer Society	Comprensibilidad	Calidad Empírica, Calidad Pragmática, Calidad Sintáctica	Diagramas	Dominio de finanzas, Dominio general

78_Prioritizing_scenario_evolution__Moisiadis [90]	Proceedings of the 4th International Conference on Requirements Engineering (ICIR'00), Page: 85, ISBN:0-7695-0565-1, IEEE Computer Society	Reusabilidad, Completitud, Eficiencia, Granularidad	Calidad Física Calidad Social	Escenarios	Sistema general
84_Replicating_the_CREWS__Cox [91]	Software Engineering Journal, Vol. 5, No. 3, pp. 245-268, 2000	Completitud, Corrección, Ambigüedad, Consistencia, Legibilidad	Calidad Semántica	Escenarios	Sistema embebido
92_Software_requirements_retrieval__Udomchaiporn [92]	13th Asia-Pacific Software Engineering Conference (APISC 2006), 6-8 December 2006, Bangalore, India. IEEE Computer Society	Eficiencia, Efectividad	Calidad física y calidad organizacional (pragmática en Nelly)	Todos	Dominio general, Comercio electrónico, Dominio de
97_Systematic_use_case__Islam [93]	12th Asia-Pacific Software Engineering Conference (APISC 2005), 15-17 December 2005, Taipei, Taiwan. IEEE Computer Society	Eficiencia, Comprensibilidad, Completitud, Concisión, Corrección	Calidad física y calidad organizacional (pragmática en Nelly)	Plantillas	Sistema embebido, Sistema de tiempo real
118_An_empirical_investigation__Carmine [94]	Model Driven Engineering Languages and Systems, 11th International Conference, MoDELS 2008, Toulouse, France, September 28 - October 3, 2008.	Comprensión, Precisión	Calidad pragmática, Calidad semántica	Diagramas	Dominio general
119_Quality_and_understandability__Bente [95]	ECOOP 2001 - Object-Oriented Programming, 15th European Conference, Budapest, Hungary, June 18-22, 2001, Proceedings. Springer	Comprensibilidad, Legibilidad, Completitud, Corrección, Consistencia, Ambigüedad	"Calidad Empírica", Calidad Pragmática	Plantillas	Dominio general

120_Comparing_Use_Case___Cox [96]	Proceedings of IRFSQ'2001 - 7th International Workshop on Requirements Engineering: Foundations for Software Quality, Essener Informatik Beitrage, Interlaken, Switzerland, June 4-5, p. 101-112	Eficiencia, Comprensibilidad, Completitud, Granularidad, Legibilidad	Calidad Empírica, Calidad Semántica, Calidad Social	Escenarios	Comercio electrónico
121_Improving_the_quality___Phalp [97]	14th Intl Software Quality Management Conference, Southampton, 10-12 April, 2006, The British Computer Society	Comprensibilidad, Precisión, Comunicabilidad	Calidad Empírica Calidad Social	Escenarios	
122_Guiding_the_process___Mustapha [98]	10th International Workshop on Database and Expert Systems Applications, 1-3 Proccedings (http://computer.org/cspress) ISBN: 0-7695-0281-4	Corrección, Precisión, Consistencia, Efectividad	Calidad Empírica, Calidad Semántica	Escenarios	
123_An_Empirical_Study___Irit [99]	Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26, 2010. ACM	Comprensibilidad, Consistencia. Modificabilidad, Efectividad, Comunicabilidad	Calidad pragmática	Diagramas	Dominio general

En la siguiente sección se presenta la información estructurada en Atributos de Calidad, Dimensiones de Calidad, Formato de descripción, Dominio del sistema, etc. de tal manera que se puede observar si existe o puede existir relación entre dichos campos.

5.3 Clasificación de los estudios seleccionados

La clasificación aportada en el presente capítulo se basa en la estrategia de extracción de información apuntada en la sección anterior, concretamente en la información técnica. En primer lugar se mostrarán los datos referentes a Dimensión y Atributos de Calidad, en segundo lugar hablaremos del Formato de descripción y por último, de los Dominios afectados de entre los que se han considerado. El punto 5.3.1 describe la clasificación de todos los artículos seleccionados y el punto siguiente sigue el mismo proceso para los de tipo CU.

5.3.1 Estudios de toda la selección

La literatura seleccionada presenta numerosos atributos que a partir de sus mediciones pretenden mejorar los ERS, UML y los CU. De entre todos estos atributos algunos son más comunes durante el estudio en la IS.

En la Tabla 15 se presenta las dimensiones de calidad consideradas a partir del modelo de Krogstie [14] con los atributos de calidad correspondientes adquiridos desde el artículo de Alan Davis [15], tal como se ha explicado en el capítulo 3. Además, en la tabla se exponen los artículos referenciados que aportan calidad en la dimensión y atributos correspondientes.

Tabla 15. Distribución de dimensiones y atributos de calidad junto con los estudios correspondientes

Dimensión de calidad	Atributos de calidad	Estudios primarios
Calidad semántica	Completitud, consistencia, corrección, precisión,	[80], [82], [83], [84], [85], [86], [87], [88], [91], [94], [96], [97],

	ambigüedad, concisión	[98]
Calidad pragmática	Comprensibilidad	[81], [82], [83], [88], [89], [92], [93], [94], [95], [96], [99]
Calidad social	Comunicación	[84], [90], [96], [97]
Calidad física	Granularidad	[80], [86]
Calidad organizacional	Reusabilidad, trazabilidad, mantenibilidad	[83], [84], [90], [99]

Tanto en la Tabla 15 como en el gráfico de la Figura 6 podemos observar que la calidad *semántica* es la más destacada seguida de la *pragmática*, la calidad *social* y *organizacional*, y por último la calidad *física*.

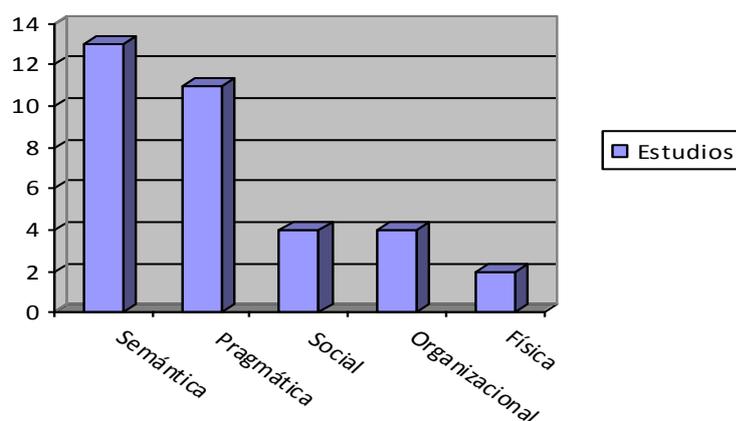


Figura 6. Relación de estudios por dimensión de calidad

Los artículos [94] y [96] son un ejemplo de estudio en Calidad semántica y en Calidad pragmática. Respecto a [94] se lleva a cabo un trabajo empírico dirigido a evaluar si el modelado dinámico proporciona *precisión* a la hora de adquirir los requisitos del cliente, además se efectúa un experimento controlado para evaluar la *comprensibilidad* de dichos requisitos. En [96] se comparan las guías CREWS con las

guías simplificadas CP de tal manera que las guías CP se aprovechan de la *completitud* que aportan las CREWS y de la *comprensibilidad* evaluada para las mismas CP. En Tabla 16 se distribuyen los artículos y los atributos de calidad. Se marcan con el símbolo “smile” (☺) las celdas donde el atributo es señalado por el artículo correspondiente.

Tabla 16. Distribución de atributos de calidad detectados y estudios seleccionados

Código	Compleitud	Consistencia	Corrección	Precisión	Ambigüedad	Comprensibilidad	Reusabilidad	Comunicación	Concisión	Trazabilidad	Mantenibilidad	Granularidad	Eficiencia
5_A_controlled_experiment___Bernardez [80]	☺				☺	☺			☺				
7_A_controlled_experiment___Kamsties [81]						☺							
17_An_approach_to___Pow-Sang [82]				☺									
20_An_experimental_investigation___Briand [83]				☺	☺	☺					☺		
29_Athena: A_collaborative___Laporti [84]			☺		☺			☺		☺			
40_Effects_of_defects___Lange [85]	☺	☺											
44_Evaluating_the_completeness___Espana [86]	☺											☺	
46_Experimental_comparison_of___Laitenberger [87]	☺	☺	☺										
52_Guiding_use_case___Ben [88]	☺		☺										☺
73_On_the_comprehension___Hogganvik [89]						☺							
78_Prioritizing_scenario_evolution___Moisiadis [90]	☺							☺					☺
84_Replicating_the_CREWS___Cox [91]	☺												
92_Software_requirements_retrieval___Udomchaiporn [92]													☺
97_Systematic_use_case___Islam [93]													☺
118_An_empirical_investigation___Carmin [94]				☺		☺							

119_Quality_and_understandability___Bente [95]						😊							
120_Comparing_Use_Case___Cox [96]	😊					😊		😊					😊
121_Improving_the_quality___Phalp [97]				😊		😊		😊					
122_Guiding_the_process___Mustapha [98]			😊										
123_An_Empirical_Study___Irit [99]				😊							😊		

Tanto en la Tabla 16 como en el gráfico de la Figura 7 se muestra que la *completitud* y la *comprensibilidad* son de los atributos considerados, los atributos más estudiados. De lo contrario, la *reusabilidad*, la *concisión*, la *trazabilidad* y la *granularidad* son los menos tenidos en cuenta.

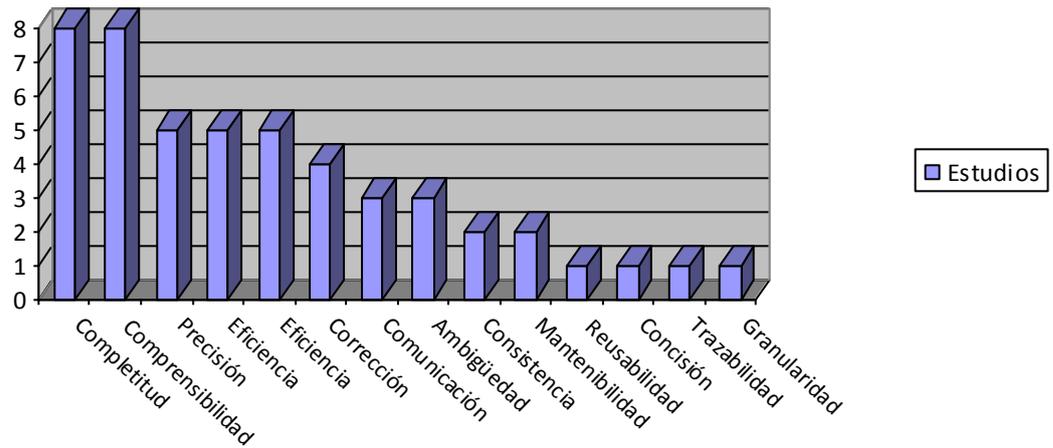


Figura 7. Relación de atributos de calidad con estudios seleccionados

Un atributo a destacar es la *eficiencia*, que como ya hemos dicho lo tratamos como un atributo especial, ya que no es extraído del trabajo de Davis y no incorporamos en ninguna dimensión de calidad.

Un artículo donde se evalúa la eficiencia es el [93], en el cual se desarrolla y se valida un nuevo concepto para elicitar y especificar requisitos representados por CU. Se trata de un sistema experto de entrevistas en el que desarrolla una aproximación para que el proceso de creación de ERS sea más sistemático y por tanto más eficiente en cuanto a reducir tiempo y coste.

En la Tabla 17 se distribuyen los formatos de descripción y los artículos afectados correspondientes.

Tabla 17. Distribución de formatos de descripción con los estudios primarios correspondientes

Formato	Estudios primarios
Diagramas	[80], [83], [85], [86], [87], [89], [92], [94], [99]
Narración	[80], [84], [92]
Escenarios	[80], [84], [88], [90], [91], [92], [96], [97], [98]
Plantillas	[80], [84], [92], [93], [95]

Tanto en la Tabla 17 como en el gráfico de la Figura 8 se indica que el formato más referenciado es el *diagrama* (debido en parte a los estudios de tipo UML), seguido por los *escenarios* (por los estudios sobre especificaciones de *guidelines*), plantillas y el menor referenciado es la narración de texto.

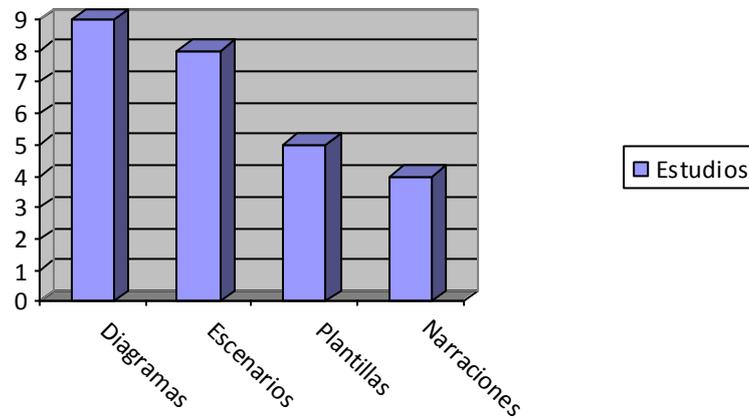


Figura 8. Relación de formatos de descripción con estudios seleccionados

El artículo [83] presenta los resultados de dos experimentos controlados en el que se usa el lenguaje de restricción OCL para tratar de proporcionar modelados UML más precisos y menos ambiguos. Por tanto, se trata de un experimento sobre diagramas de UML en general, que puede servir para diagramas de CU.

En [84] se utiliza una herramienta para la elicitación y especificación de requisitos a partir de historias narradas por los clientes que se transforman en CU. Estas historias narradas se convierten en escenarios y de escenarios a plantillas de CU - así que lo hemos considerado en narraciones, en escenarios y en plantillas.

Tal como hemos visto en la segunda sección del presente capítulo se han considerado los dominios listados en [54], de los cuales se han identificado los siguientes en los respectivos estudios.

Tabla 18. Distribución de dominios con los estudios primarios correspondientes

Dominio	Estudios primarios
General	[80], [82], [83], [84], [86], [87], [89], [90], [92], [94], [95], [99]
Sistemas embebidos	[81], [88], [91], [93]
Comercio electrónico	[92], [96]
Finanzas	[89], [92]
Tiempo real	[93]

Tanto en la Tabla 18 como en el gráfico de la Figura 9 se indica claramente el dominio *general* como el más distinguido de entre todos los demás.

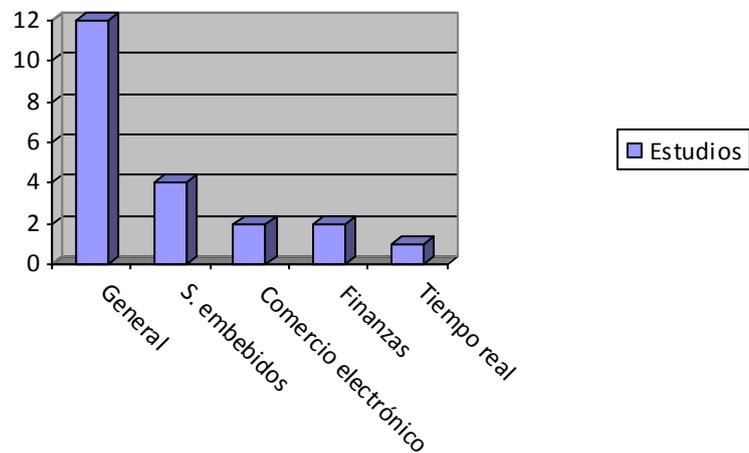


Figura 9. Relación de dominios con los estudios seleccionados

Existen artículos donde se experimenta en un solo dominio de sistema y hay otros como en [92] donde se experimenta en dominios diferentes. En [92] se presenta una aproximación para la recuperación de especificaciones de requisitos en forma de CU. Se usan 5 sistemas software (de dos dominios diferentes) para la prueba del uso del sistema de recuperación de CU: un sistema de enseñanza (dominio general), un sistema de negocio de productos (dominio de finanzas), un sistema de gestión de clientes (dominio general), un sistema de cálculo de financiación (dominio de finanzas) y un sistema de generación de reportes (dominio general).

Una vez hemos clasificados los artículos por atributos de calidad, dimensiones de calidad, formato de descripción y dominio, resulta interesante exponer dos comparaciones: una entre dimensiones de calidad y formato de descripción, y otra entre dimensiones de calidad y dominio. El análisis en estas relaciones puede ser fructífero porque nos puede llevar a averiguar si existe un formato o un dominio que promueva o que resulte ser mejor para una dimensión de calidad.

En la Tabla 19 se presenta la relación entre Dimensión de Calidad y Formato extraída de la presente literatura. Como conclusión cabe señalar que para obtener calidad semántica los autores se han centrado en el uso de los escenarios (35%) y de

los diagramas (30%). Otro dato a destacar puede ser el aporte de calidad pragmática mediante diagramas (25%) y tal vez la obtención de calidad social a partir de escenarios (20%). En lo que respecta al resto de resultados la información es más regular.

Tabla 19. Distribución de estudios en relación a la dimensión de calidad y el formato de descripción

Formato Dimensión de calidad	Escenarios	Diagramas	Narraciones	Plantillas
Calidad semántica	7 (35%)	6 (30%)	2 (10%)	2 (10%)
Calidad social	4 (20%)	0	1 (5%)	1 (5%)
Calidad pragmática	3 (15%)	5 (25%)	2 (10%)	3 (15%)
Calidad organizacional	2 (10%)	2 (10%)	1 (5%)	1 (5%)
Calidad física	1 (5%)	2 (10%)	1 (5%)	1 (5%)

En la Tabla 20 se presenta la relación entre Dimensión de Calidad y Dominio extraída de la presente literatura. Esta vez hay un claro destacado por parte del dominio más utilizado, el General (MIS), prácticamente con todas las dimensiones de calidad, con un 40% en calidad pragmática, un 35% en calidad semántica, un 20% en calidad organizacional, un 10% en calidad social y un 10% en calidad física. También se destaca que la calidad pragmática tiene lugar en todos los dominios.

Tabla 20. Distribución de estudios en relación a la dimensión de calidad y al dominio

Dominio Dimensión de calidad	General (MIS)	S. Embebidos	Comercio Electrónico	Finanzas	Tiempo real
---	----------------------	---------------------	-----------------------------	-----------------	--------------------

Dimensión de calidad \ Dominio	General (MIS)	S. Embebidos	Comercio Electrónico	Finanzas	Tiempo real
Calidad pragmática	8 (40%)	3 (15%)	2 (10%)	2 (10%)	1 (5%)
Calidad semántica	7 (35%)	2 (10%)	1 (10%)	0	0
Calidad organizacional	4 (20%)	0	0	0	0
Calidad social	2 (10%)	0	1 (5%)	0	0
Calidad física	2 (10%)	0	0	0	0

5.3.2 Estudios de la selección en Casos de Uso

En esta sección se detalla la información específica solamente para los artículos de tipo CU. Veremos el mismo tipo de información extraída que en el punto anterior, a diferencia de que en este caso la información afecta directamente a los Casos de Uso. Además a partir de esta información se realiza la evaluación y la agregación (síntesis). Se trata de 10 artículos, 6 de ellos sobre especificación de guidelines [88][91][95][96][97][98] para la construcción de CU, uno para detección de defectos en CU [80], uno para expandir escenarios [90], otro es un proceso para crear CU [93] y otro es una comparación entre CU y Tropos [99].

Tabla 21. Lista de estudios seleccionados de Casos de Uso

Código	Título	Autores	Finalidad	Técnica
5_A_controlled_experiment___Bernardez [80]	A controlled experiment for evaluating a metric-based reading technique for requirements inspection	Bernardez B., Genero M., Duran A., Toro M.	Detección de defectos en Casos de Uso	CU
52_Guiding_use_case___Ben [88]	Guiding use case authoring: results of an empirical study	Ben Achour C., Rolland C., Maiden N.A.M., Souveyet C.	Se presentan guías para la construcción de Casos de Uso	CU
78_Prioritizing_scenario_evolution___Moisiadis [90]	Prioritizing scenario evolution	Moisiadis Frank	Expansión de escenarios	CU
84_Replicating_the_CREWS___Cox [91]	Replicating the CREWS use case authoring guidelines experiment	Cox K., Phalp K.	Se presentan guías para la construcción de Casos de Uso	CU
97_Systematic_use_case___Islam [93]	Systematic use case interviews for specification of automotive systems	Islam S., Omasreiter H.	Proceso para crear Casos de Uso	CU
119_Quality_and_understandability___Bente [95]	Quality and understandability of use case models	Bente Anda, Dag Sjoberg, Magne Jorgensen	Se presentan guías para la construcción de Casos de Uso	CU
120_Comparing_Use_Case___Cox [96]	Comparing Use Case Writing Guidelines	Cox Karl, Phalp Keith, Shepperd Martin	Se presentan guías para la construcción de Casos de Uso	CU
121_Improving_the_quality___Phalp [97]	Improving the quality of use case descriptions: empirical assessment of writing guidelines	Phalp K.T., Vincent J., Cox K.	Se presentan guías para la construcción de Casos de Uso	CU
122_Guiding_the_process___Mustapha [98]	Guiding the process of requirements elicitation trough scenario analysis: results of an empirical study	Mustapha Tawbi, C. Ben Achour, Fernando Vélez	Se presentan guías para la construcción de Casos de Uso	CU
123_An_Empirical_Study___Irit [99]	An Empirical Study of Requirements Model Understanding: Use Case vs. Tropos Models	Irit Hadar, Tsvi Kuflik, Anna Perini	Comparación entre Casos de Uso y Tropos	CU

A continuación en la Tabla 22 se presenta las dimensiones de calidad consideradas y los atributos de calidad correspondientes. A partir de la siguiente información obtenemos respuesta a la pregunta:

- 1) *¿Cuáles son los aspectos de la calidad más investigados en el desarrollo de CU?*

Tabla 22. Distribución de dimensiones de calidad junto con sus atributos correspondientes y los estudios de Casos de Uso afectados

Dimensión de calidad	Atributos de calidad	Estudios primarios
Calidad semántica	Complejidad, consistencia, corrección, precisión, ambigüedad, concisión	[80], [88], [91], [96], [97], [98]
Calidad pragmática	Comprensibilidad	[88], [93], [95], [96], [99]
Calidad social	Comunicación	[90], [96], [97]
Calidad física	Granularidad	[80]
Calidad organizacional	Reusabilidad, trazabilidad, mantenibilidad	[90], [99]

Tanto en la Tabla 22 como en el gráfico de la Figura 10 podemos observar que la calidad *semántica* es la más destacada seguida de la *pragmática*, después está la calidad *social* y *organizacional*, y por último la calidad *física*. Hasta en 11 ocasiones tanto la calidad semántica como la pragmática son protagonistas, lo que significa que existe un mayor interés y necesidad en estas calidades.

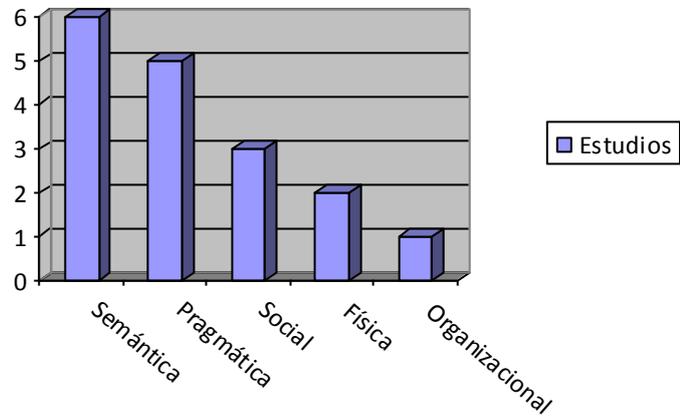


Figura 10. Relación de dimensión de calidad con los estudios de Casos de Uso

En la Tabla 23 se muestra cada atributo de calidad junto al artículo donde es señalado. Se marcan con el símbolo “smile” (☺) las celdas correspondientes. A diferencia del punto anterior no aparecen en la presente literatura los atributos: consistencia, trazabilidad y granularidad.

Tabla 23. Lista de estudios de Casos de Uso con los atributos de calidad correspondientes

Código	Complejidad	Corrección	Precisión	Ambigüedad	Comprensibilidad	Reusabilidad	Comunicación	Concisión	Mantenibilidad	Eficiencia
5_A_controlled_experiment__Bernardez [80]	☺			☺	☺			☺		
52_Guiding_use_case__Ben [88]	☺	☺								☺
78_Prioritizing_scenario_evolution__Moisiadis [90]	☺					☺				☺
84_Replicating_the_CREWS__Cox [91]	☺									

97_Systematic_use_case___Islam [93]										😊
119_Quality_and_understandability___Bente [95]					😊					
120_Comparing_Use_Case___Cox [96]	😊				😊		😊			😊
121_Improving_the_quality___Phalp [97]			😊		😊		😊			
122_Guiding_the_process___Mustapha [98]		😊								
123_An_Empirical_Study___Irit [99]			😊						😊	

Tanto en la Tabla 23 como en el gráfico de la Figura 11 se muestra que la *completitud* y la *comprensibilidad* son de los atributos considerados los atributos más estudiados. De lo contrario, la *reusabilidad*, la *concisión*, la *mantenibilidad* y la *ambigüedad* son los menos tenidos en cuenta.

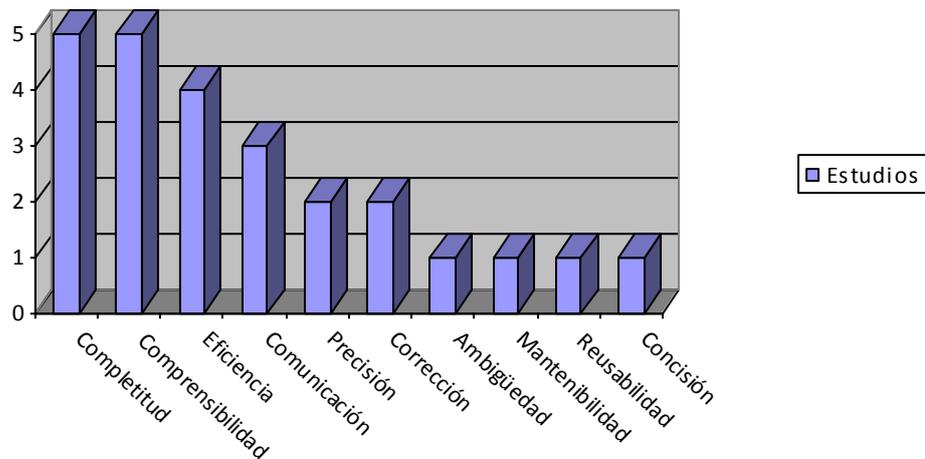


Figura 11. Relación de atributos de calidad con los estudios de Casos de Uso

En la Tabla 24 se distribuyen los formatos de descripción y los artículos correspondientes. A partir de la siguiente información podemos obtener respuesta a la pregunta:

2) ¿Existe un formato de descripción que pueda promover la calidad de la especificación de CU?

Tabla 24. Distribución de los formatos de descripción con los estudios de Casos de Uso correspondientes

Formato	Estudios primarios
Diagramas	[80], [99]
Narración	[80]
Escenarios	[80], [88], [90], [91], [96], [97], [98]
Plantillas	[80], [93], [95]

Tanto en la Tabla 24 como en el gráfico de la Figura 12 podemos comprobar que los escenarios son el formato más destacado con un 70% de los estudios. Esto se debe en gran medida por el interés en la construcción de CU a partir de *guidelines*.

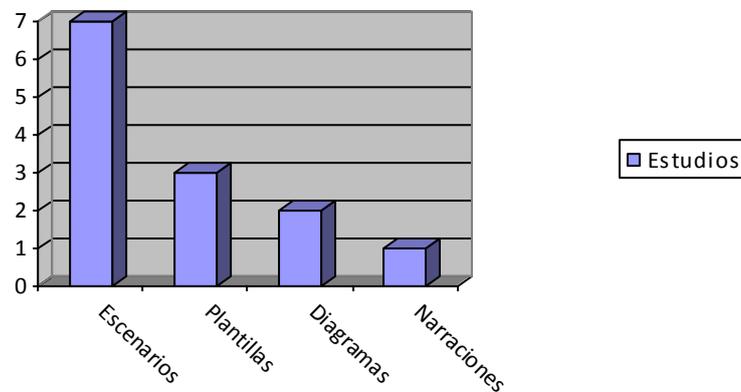


Figura 12. Relación de los formatos de descripción con los estudios de Casos de Uso

En la Tabla 25 se distribuyen los dominios y los artículos correspondientes. A partir de la siguiente información podemos obtener respuesta a la pregunta:

3) ¿En qué dominios son investigados los aspectos de calidad?

Tabla 25. Distribución de dominios con los estudios de Casos de Uso

Dominio	Estudios primarios
General	[80], [90], [95], [99]
Sistemas embebidos	[88], [91], [93]
Comercio electrónico	[96]
Tiempo real	[93]

Tanto en la Tabla 25 como en el gráfico de la Figura 13 se indica que el dominio más utilizado es del dominio general (40%) seguido de los sistemas embebidos (30%).

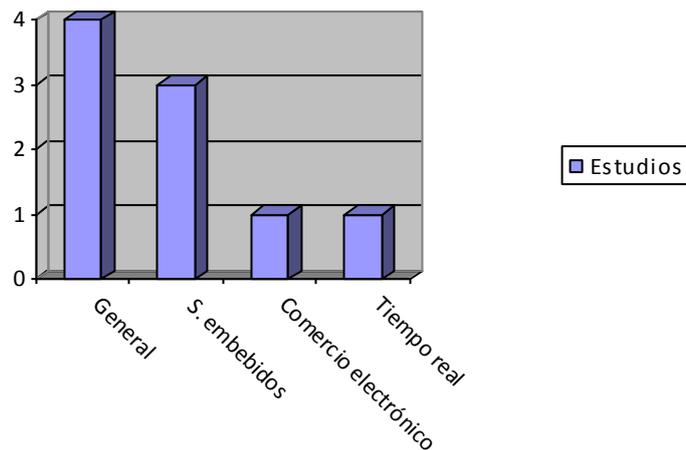


Figura 13. Relación de los dominios con los estudios de Casos de Uso

En la Tabla 26 se compara las dimensiones de calidad con los formatos de descripción. El 60% de la literatura que afecta a la calidad semántica está representada por escenarios. Otro dato a destacar es el 30% que relaciona la calidad social también con escenarios.

Tabla 26. Distribución de los estudios de Casos de Uso en relación de las dimensiones de calidad y los formatos de descripción

Formato Dimensión de calidad	Escenarios	Diagramas	Narraciones	Plantillas
Calidad semántica	6 (60%)	1 (10%)	1 (10%)	1 (10%)
Calidad social	3 (30%)	0	0	0
Calidad pragmática	1 (10%)	1 (10%)	0	2 (20%)
Calidad organizacional	1 (10%)	1 (10%)	0	0
Calidad física	1 (10%)	1 (10%)	1 (10%)	1 (10%)

En la Tabla 27 se presenta la relación entre dimensión de calidad y dominio de la presente literatura seleccionada. Cabe destacar que en todas las dimensiones de calidad se ha experimentado con el dominio general. También cabe señalar que la calidad pragmática ha sido experimentada en los cuatro dominios dominios considerados.

Tabla 27. Distribución de estudios de Casos de Uso en relación a la dimensión de calidad y los dominios

Dominio Dimensión de calidad	General (MIS)	S. Embebidos	Comercio Electrónico	Tiempo real
Calidad pragmática	2 (20%)	2 (20%)	1 (10%)	1 (10%)
Calidad organizacional	2 (20%)	0	0	0
Calidad semántica	1 (10%)	2 (20%)	1 (10%)	0
Calidad social	1 (10%)	0	1 (10%)	0
Calidad física	1 (10%)	0	0	0

5.4 Evaluación de la calidad de los estudios seleccionados

En este punto pasamos a presentar la evaluación de la revisión sistemática llevada a cabo. Para dicha evaluación se ha utilizado los criterios de evaluación de calidad proporcionado por Travassos et al., los cuales han sido establecidos para evaluar concretamente a los artículos seleccionados de tipo CU (las preguntas van dirigidas directamente a los artículos CU). Dicho criterio consiste en un sistema de puntuación de 9 preguntas donde cada una lleva asociada su particular puntuación, esto depende de la respuesta: Sí/No/Varios. Las preguntas son las siguientes:

- Criterio relacionado con las características de calidad de Casos de Uso.
 - Q1: ¿Existe alguna descripción de cómo los atributos de calidad han sido identificados? (1 punto) – *Se deben tener en cuenta todos los atributos de calidad detectados y tratar de encontrar al menos una descripción de la identificación de uno de ellos.*
 - Q2: ¿Existe alguna descripción sobre las restricciones y condiciones donde la calidad de los atributos fueron observados? (1 punto) – *Condiciones del experimento, por ejemplo: tiempo de entrenamiento, conocimientos aplicados para el experimento, etc.*
 - Q3: ¿Existe alguna descripción sobre cómo la calidad de los atributos puede ser medida? (1 punto) – *Si existen métricas o alguna explicación sobre el método de medición.*
- Criterio relacionado al formato de descripción.
 - Q4: ¿Describe el artículo alguna adaptación/evolución de una pre-existente aproximación? (1 punto) – *Por ejemplo una extensión de una aproximación presentada de un formato de descripción de Casos de Uso.*
 - Q5: ¿El formato de descripción es referenciado/evaluado/utilizado en otros trabajos? (0,5 puntos x número de referencias) – *Si existe un estudio referenciado o replicación en otros artículos.*

- Q6: ¿Describe el artículo una aplicación del formato de descripción utilizado para su evaluación? (0,5 puntos) – *Si se utiliza un sistema para evaluar el experimento relacionado con el formato de descripción.*
- Q7: ¿Existe algún resultado empírico/experimental respecto al formato de descripción? (2 puntos) – *Si existe evidencia empírica en algún otro artículo (similar al Q5).*
- Q8: ¿Es posible identificar para qué tipos de sistemas puede ser usado el formato de descripción? (0,5 puntos x cada tipo de sistema) – *Si se identifica el dominio del sistema utilizado para realizar el experimento.*
- Q9: ¿Es posible evaluar qué atributos de calidad promueve el formato de descripción? (0,5 puntos x atributo de calidad) – *Atributos que mejoran la calidad del formato de descripción a partir de su validación (hipótesis).*

En la Tabla 28 se presentan los resultados de las puntuaciones recibidas por cada artículo y en la Tabla 30 se detalla el resultado de cada pregunta para cada artículo. Si analizamos los resultados finales comprobamos que la máxima puntuación obtenida es de 7,5 puntos (artículo más relevante) y la mínima de 2 puntos (artículo menos relevante). Con esto, la puntuación media es de 4,35.

Tabla 28. Puntuaciones de evaluación para los estudios de Casos de Uso

Puntuación									
7,5	7	5,5	5,5	4,5	4	2,5	2,5	2,5	2

Si se tuviera en cuenta que podría existir algún artículo cuyas respuestas a todas las preguntas fueran afirmativas (con un “1” se referencia a las preguntas que necesiten número de referencias), dicho artículo obtendría 8 puntos (1+1+1+1+0,5+0,5+2+0,5+0,5). Proponiendo los 8 puntos como puntuación máxima

señalaríamos dicho artículo como *Muy Relevante*¹¹ a partir de la siguiente clasificación:

Tabla 29. Relevancia de los estudios de Casos de Uso

Relevancia	Puntuación	Artículos
Muy Relevante	De 6,01 a 8	84_Replicating_the_CREWS___Cox [91] 120_Comparing_Use_Case___Cox [96]
Bastante Relevante	De 4,01 a 6	119_Quality_and_understandability___Bente [95] 121_Improving_the_quality___Phalp [97] 78_Prioritizing_scenario_evolution___Moisiadis [90] 52_Guiding_use_case___Ben [88]
Poco Relevante	De 2,01 a 4	122_Guiding_the_process___Mustapha [98] 123_An_Empirical_Study___Irit [99] 97_Systematic_use_case___Islam [93]
Muy poco Relevante	De 0 a 2	5_A_controlled_experiment___Bernardez [80]

Así que, teniendo en cuenta esta clasificación tenemos 2 artículos *Muy Relevantes*, 4 artículos *Bastante Relevantes*, 3 artículos *Poco Relevantes* y 1 artículo *Muy poco Relevante*.

Podemos observar que 5 de los 6 artículos sobre especificación de guías se encuentran entre los *Bastante Relevantes* y *Muy Relevantes*, esto significa que existe una literatura relevante dirigida a construir CU de calidad dedicada a la especificación de guías. También cabe destacar que 5 de los 7 artículos donde se presentaba el formato de descripción de escenarios, se encuentran en dichos 6 artículos relevantes. Así que los escenarios destacan de manera notable en los estudios de calidad de CU por, además del número de artículos, de la puntuación recibida en tales artículos.

En la siguiente sección se presenta la agregación o síntesis del estudio secundario. En este apartado se involucra resumen y cotejo de los resultados de los estudios primarios incluidos.

¹¹ Entendemos por “relevantes” los trabajos que aportan información significativa en lo que se refiere a los objetivos del presente trabajo.

Tabla 30. Puntuaciones por pregunta de evaluación en los estudios de Casos de Uso

Código	Q1 (1pt)	Q2 (1pt)	Q3 (1pt)	Q4 (1pt)	Q5 (x0,5pt)	Q6 (0,5pt)	Q7 (2pt)	Q8 (x0,5pt)	Q9 (x0,5pt)	Puntuación
5_A_controlled_experiment___Bernardez [80]	0	1	1	0	0	0	0	0	0	2
52_Guiding_use_case___Ben [88]	1	1	0	0	0	1	0	1	2	4
78_Prioritizing_scenario_evolution___Moisiadis [90]	1	0	1	1	0	1	0	1	1	4,5
84_Replicating_the_CREWS___Cox [91]	1	1	1	0	1	1	1	1	2	7,5
97_Systematic_use_case___Islam [93]	0	0	0	0	0	1	0	2	2	2,5
119_Quality_and_understandability___Bente [95]	0	1	0	0	2	1	1	1	1	5,5
120_Comparing_Use_Case___Cox [96]	1	1	0	1	1	1	1	1	1	7
121_Improving_the_quality___Phalp [97]	0	1	0	0	4	0	1	0	1	5,5
122_Guiding_the_process___Mustapha [98]	0	0	0	0	1	0	1	0	0	2,5
123_An_Empirical_Study___Irit [99]	0	0	1	0	0	1	0	1	1	2,5

5.5 Síntesis de la información

La agregación o síntesis de la información extraída se construye de manera descriptiva. A continuación se presenta la síntesis estructurada en diferentes secciones.

5.5.1 Estudios considerados

Los artículos considerado para la síntesis una finalidad en común, la exploración en construcción de CU de calidad a partir de guías (*guidelines*). En la Tabla 31 se presentan estos artículos junto con las guías que se estudian.

Tabla 31. Estudios y guías para la síntesis

<i>Artículos</i>	<i>Guías</i>
52_Guiding_use_case___Ben	CREWS
84_Replicating_the_CREWS___Cox	CREWS (replicación)
119_Quality_and_understandability___Bente	Minor G., Template G. y Style G.
120_Comparing_Use_Case___Cox	CP Style, CP Structure y CREWS
121_Improving_the_quality___Phalp	CP Style, CP Structure y CREWS
122_Guiding_the_process___Mustapha	CREWS L'Ecritoire

5.5.2 Tareas y sujetos

En el conjunto de artículos la finalidad es la elicitación y especificación de requisitos a través de escenarios de CU. Las tareas a realizar y las características de los sujetos presentados en la Tabla 32 son muy semejantes.

Tabla 32. Tareas y sujetos por artículo

<i>Artículos</i>	<i>Tarea</i>	<i>Sujetos</i>
------------------	--------------	----------------

52_Guiding_use_case___Ben	60 minutos para escribir CU.	69 sujetos divididos en 4 grupos diferenciados por el material de apoyo facilitado ¹² .
84_Replicating_the_CREWS___Cox	60 minutos para escribir CU.	14 sujetos divididos en 4 grupos diferenciados por el material de apoyo facilitado ¹ .
119_Quality_and_understandability___Bente	2 semanas para la creación de una ERS en la que se incluye un modelo de CU.	139 sujetos divididos en 31 grupos. Cada grupo organizado en 2 parejas, una pareja como cliente de un sistema A y la otra pareja como desarrolladores de un sistema B.
120_Comparing_Use_Case___Cox	Fase 3: escribir los Casos de Uso con <i>guidelines</i> . Una hora. Fase 4: leer Casos de Uso escritos con la ayuda de <i>guidelines</i> con el consiguiente cuestionario sobre los Casos de Uso leídos. 30 minutos.	4 grupos de 6 sujetos cada uno: 2 grupos utilizan guías CP y otros 2 utilizan guías CREWS.
121_Improving_the_quality___Phalp	60 minutos para escribir CU.	4 grupos de 15 sujetos cada

¹² A: El dominio del problema.

B: El dominio del problema con las CREWS SGs.

C: El dominio del problema con las CREWS CGs.

D: El dominio del problema con las CREWS SGs y CGs.

		uno. 2 grupos utilizan CP y los otros 2 CREWS.
122_Guiding_the_process___Mustapha	Realizar un análisis de requisitos.	41 sujetos.

5.5.3 Heurísticas y esquemas de marcado

Los autores reflejan varias heurísticas y esquemas de marcado para realizar sus evaluaciones. En algunas ocasiones se utilizan las mismas que en experimentos anteriores con la adición de otras nuevas.

52_Guiding_use_case___Ben:

Diferentes criterios de evaluación no especificados para cada una de las guías. Se consideran CU individualmente.

84_Replicating_the_CREWS___Cox:

Se consideran CU individualmente. Heurísticas:

1. Plausibilidad: ¿Cómo de realista es el CU?
2. Legibilidad: ¿Hay un flujo en el CU? ¿Sigue unos pasos lógicos?
3. Estructura consistente: ¿La terminología es consistente?
4. Flujo alternativo: marcas para considerar alternativas en el CU

119_Quality_and_understandability___Bente:

Se consideran modelos de CU completos. Se especifica un conjunto de recomendaciones:

- Los CU deben ser fáciles de leer.
- Las descripciones no deben incluir suposiciones de diseño o implementación.
- Las descripciones no deben incluir detalles de interfaz.

- No se deben describir eventos que no están relacionados con el funcionamiento general de la aplicación.
- Las descripciones de acciones deben ser completas.
- La estructura de flujo debe ser correcta y no ambigua.
- La terminología debe ser consistente.

Se consideran los flujos de eventos sugeridos en *52_Guiding_use_case___Ben* y las heurísticas propuestas en *84_Replicating_the_CREWS___Cox*.

1. Plausibilidad: ¿Cómo de realista es el CU?
2. Legibilidad: ¿Hay un flujo en el CU? ¿Sigue unos pasos lógicos?
3. Estructura consistente: ¿La terminología es consistente?
4. Flujo alternativo: marcas para considerar alternativas en el CU.

Añaden otro esquema de marcado:

- Diagrama individual: el modelo de CU debe incluir un diagrama mostrando todos los actores y todos los CU.

- Actores: los actores correctamente identificados. La corrección está determinada en relación a la especificación de requisitos informal.

- Casos de Uso: los Casos de Uso correctamente identificados. La corrección está determinada en relación a la especificación de requisitos informal.

- Contenido: la descripción de cada caso de uso contiene la información requerida por el conjunto de *guidelines*: actores, suposiciones que deben ser válidas antes del inicio del CU, flujos de eventos, variaciones y post-condiciones.

- Nivel de detalle: la descripción de cada caso de uso está en un apropiado nivel de detalle. No deben ser necesarios detalles acerca de la interfaz de usuario o diseño interno. Cada evento debe ser atómico, las sentencias con más de dos cláusulas se deben evitar.

- Realismo: el flujo de eventos es realista, los eventos siguen un orden lógico y completo y se indican claramente dónde las variaciones pueden ocurrir.
- Consistencia: el uso de la terminología es consistente.

120_Comparing_Use_Case___Cox:

Las heurísticas (4Cs, comunicación en CU) utilizadas para evaluar la comprensibilidad son:

- Cobertura.
- Coherencia y legibilidad.
- Estructura consistente.
- Consideración de alternativas.

121_Improving_the_quality___Phalp:

Las heurísticas (7Cs, comunicación en CU) utilizadas para evaluar la comprensibilidad son:

- Cobertura.
- Convincente.
- Coherencia.
- Abstracción consistente.
- Estructura consistente.
- Gramática consistente.
- Consideración de alternativas.

5.5.4 Hipótesis y validaciones

Las hipótesis propuestas se orientan en mejorar la construcción de descripciones de CU. En términos generales se trata de mejorar la corrección, la completitud y la comprensibilidad.

52_Guiding_use_case__Ben:

Algunas hipótesis se deben evaluar al mismo tiempo, ya que son dependientes:
C1/S1, C2/S2, C3/S3, S4.

7 Hipótesis: C1, C2, C3, S1, S2, S3 y S4.

- C1/S1: Completitud en descripciones de acciones.
- C2/S2: Completitud en número de acciones.
- C3/S3: Estructura.
- S4: Corrección terminológica.

¿Se cumplen las hipótesis? Resultados:

- C1: Sí, significativamente, pero en combinación con las SGs.
- S1: Sí, proporcionalmente al número de acciones.
- C2: Sí, significativamente mejorado.
- S2: Sí, significativamente mejorado.
- C3: Sí, significativamente para el flujo interno de la estructura de escenarios en un CU.
- S3: Sí, significativamente para la estructura de CU.
- S4: Sí, ligeramente para sinónimos y significativamente para referencias anafóricas.

84_Replicating_the_CREWS__Cox:

Algunas hipótesis se deben evaluar al mismo tiempo, ya que son dependientes:
C1/S1, C2/S2, C3/S3, S4.

7 Hipótesis: C1, C2, C3, S1, S2, S3 y S4.

- C1/S1: Completitud en descripciones de acciones.
- C2/S2: Completitud en número de acciones.
- C3/S3: Estructura.
- S4: Corrección terminológica.

¿Se cumplen las hipótesis? Resultados:

- C1: Sí.
- S1: No.
- C2: Sí.
- S2: No.
- C3: No.
- S3: Sí, para tiempos no pasivos. No, para el resto.
- S4: No.

119_Quality_and_understandability_____Bente:

Hipótesis:

H1,0 No hay diferencia en comprensibilidad cuando se leen los modelos de CU contruidos con guías diferentes.

H1,1 Existe diferencia en comprensibilidad cuando se leen los modelos de CU contruidos con guías diferentes.

Afirmativo mejor cuando las guías son Template o Style. Cuando las guías son familiares (que ya han sido utilizadas por los lectores) mejor utilizar las Template (para la comprensibilidad) y cuando no lo son mejor usar las Style.

H2,0 No hay diferencia en comprensibilidad de los requisitos cuando se usan guías diferentes en la construcción de modelos de CU.

H2,1 Existe diferencia en comprensibilidad de los requisitos cuando se usan guías diferentes en la construcción de modelos de CU.

Negativo.

H3,0 No hay diferencia en la utilidad de diferentes guías cuando se construyen modelos de CU.

H3,1 Existe diferencia en la utilidad de diferentes guías cuando se construyen modelos de CU.

Afirmativo. Significativamente las Template son más útiles.

120_Comparing_Use_Case___Cox:

Hipótesis:

H0: las guías CREWS producen una mejora significativa de la calidad de CU en términos de eficiencia, efectividad y comprensibilidad.

No se evalúa.

H1: no existe diferencia significativa entre las reglas CP y las guías CREWS en términos de tiempo tomado para escribir CU y su longitud.

Afirmativo. Sin embargo parece que las CP producen CU más compactos, aunque no se sabe si necesariamente es una ventaja.

H2: no existe diferencia significativa entre las reglas CP y las guías CREWS en términos de número de eventos que implementan correctamente las reglas o las guías.

Afirmativo. Sin embargo las CP pueden ser mejores en algún caso.

H3: no existe diferencia significativa entre las reglas CP y las guías CREWS en términos de comprensibilidad de los CU.

Afirmativo. Sin embargo las CP pueden ser incluso mejores en algún caso.

121_Improving_the_quality___Phalp:

Hipótesis:

H1: las construcciones sugeridas por las reglas CP se encuentra en cantidades significativamente mayores que las construcciones sugeridas por las guías CREWS equivalentes cuando ambos conjuntos de guías son aplicados a los mismos problemas.

Negativo (no siempre).

Proposición: las descripciones de CU producidas por las reglas CP son significativamente más comprensibles que las producidas por las CREWS equivalentes. Al no poderse evaluar esta proposición directamente, se adoptan las 7Cs para evaluar la H2.

H2: las descripciones de CU producidas por las reglas CP se puntúan significativamente mejor que las descripciones equivalentes producidas por CREWS, cuando son marcados por las características de calidad de CU de las 7Cs.

Negativo (no siempre).

Con el H1 y el H2 evaluados se confirma que el uso de las guías y la calidad general de los CU están relacionados. En ambos casos tanto con CREWS que con CP se mejora la comunicabilidad. Cuantas más reglas se aplican las descripciones de CU mejoran.

122_Guiding the process___Mustapha:

Hipótesis:

H1: las CREWS-L'Ecritoire ayudan a formular objetivos más precisos.

Afirmativo.

H2: las CREWS-L'Ecritoire soportan el descubrimiento de alternativas de diseño, cuando son usados en nivel contextual, ayudan a visionar mejor el futuro sistema.

Afirmativo.

H3: las CREWS-L'Ecritoire ayudan a la estrategia de alternativas encontrando más variaciones que un proceso ad hoc.

Negativo (en ocasiones).

H4: las CREWS-L'Ecritoire ayudan a la estrategia de alternativas encontrando variaciones que están al mismo nivel de abstracción.

No confirmado

H5: las CREWS-L'Ecritoire ayudan a la estrategia de alternativas separando asuntos de descripciones de comportamiento alternativas.

Afirmativo.

H6: las CREWS-L'Ecritoire ayudan a la estrategia de composiciones encontrando más funciones de sistema que un proceso ad hoc.

Afirmativo.

H7: las CREWS-L'Ecritoire ayudan a la estrategia de composiciones encontrando funciones de sistema en el mismo nivel de abstracción.

Afirmativo.

H8: las CREWS-L'Ecritoire ayudan a la estrategia de composiciones separando funciones de sistema.

No confirmado.

H9: los tres niveles predefinidos de abstracción ayudan a mejorar preservando la consistencia de descripciones de acciones en un escenario.

Afirmativo.

5.5.5 Conclusión

Con las heurísticas y esquemas de marcado presentadas anteriormente, dependientemente de las condiciones en cuanto a tareas, sujetos y guías experimentadas junto con las hipótesis evaluadas, podemos llegar a una conclusión acerca de las mejoras que se pueden aplicar en la construcción de modelos de CU. En algunos casos se han considerado CU individualmente (*52_Guiding_use_case___Ben*, *84_Replicating_the_CREWS___Cox*) y en otros se han considerado modelos de CU completos (*119_Quality_and_understandability___Bente*).

A partir de las guías CREWS tomamos la *84_Replicating_the_CREWS___Cox* como la confirmación de las guías que sí mejoran. En este artículo, se destaca que no

necesariamente se mejora la completitud en CU como dicen en *52_Guiding_use_case__Ben*, sólo que los sujetos implementan un número variable de guías en sus descripciones de CU. Por lo que se afirma que:

- (C1) El uso de CGs (Content Guidelines) conduce a descripciones de CU más correctas en términos de acciones de descripción completas.

- (C2) El uso de CGs conduce menos a descripciones de acciones menos apropiadas.

- (S3) El uso de SGs (Style Guidelines) conduce a descripciones de CU que contienen descripciones más correctas y menos ambiguas para la estructura de flujo en el CU, en tiempos no pasivos.

Concluimos que usando guías Template o Style (*119_Quality_and_understandability __Bente*) para construir modelos de CU mejoramos la comprensibilidad cuando se leen estos modelos. *Cuando las guías son familiares (que ya han sido utilizadas por los lectores) mejor utilizar las Template (para la comprensibilidad) y cuando no lo son mejor usar las Style.*

Respecto a la utilidad de las guías, las guías Template son más útiles para construir modelos de CU.

Hemos dado por hecho que las CP son simplificaciones de las CREWS (*120_Comparing_Use_Case__Cox* y *121_Improving_the_quality__Phalp*). Se justifica el uso de las CP por delante de las CREWS ya que, sabemos que no existe diferencia significativa entre las reglas CP y las guías CREWS en términos de tiempo tomado para escribir CU y su longitud - *sin embargo parece que las CP producen CU más compactos, aunque no se sabe si necesariamente es una ventaja*. No existe diferencia significativa entre las reglas CP y las guías CREWS en términos de número de eventos que implementan correctamente las reglas o las guías. No existe diferencia significativa entre las reglas CP y las guías CREWS en términos de comprensibilidad de los CU - *sin embargo las CP pueden ser incluso mejores en algún caso*. En ambos casos tanto con CREWS que con CP se mejora la comunicabilidad. Cuantas más reglas se aplican en las descripciones de CU mejor.

Finalmente usando la aproximación CREWS-L'Ecritoire (*122_Guiding_the_process___Mustapha*) nos podemos aprovechar de las ventajas de: formular objetivos más precisos, descubrir nuevas de alternativas de diseño (pueden ayudar a visionar mejor el futuro del sistema), mejorar la estrategia de alternativas separando asuntos de descripciones de comportamiento alternativas, mejorar la estrategia de composiciones encontrando más funciones de sistema que un proceso ad hoc, mejorar la estrategia de composiciones encontrando funciones de sistema en el mismo nivel de abstracción, mejorar la estrategia de composiciones separando funciones de sistema y mejorar preservando la consistencia de descripciones de acciones en un escenario (a través de los tres niveles predefinidos de abstracción).

Según *119_Quality_and_understandability___Bente*, cuanto mayor sea el número de actores y CU identificados, junto con el número de eventos y de variaciones, mayor será la calidad de las guías. En *121_Improving_the_quality___Phalp* con el H1 y el H2 evaluados se confirma que el uso de las guías y la calidad general de los CU están relacionados.

Capítulo 6

Conclusiones y trabajos futuros

En esta Tesis de Máster se ha llevado a cabo una revisión sistemática para extraer conocimiento a partir de información encontrada en la literatura sobre calidad de Casos de Uso. Por consiguiente, a través de este trabajo hemos tratado de averiguar cómo mejorar la calidad en las construcciones de CU, directa o indirectamente, desde el conjunto de todos los artículos seleccionados basados en evidencias empíricas.

Durante el proceso se han seleccionado 20 artículos en total que se han clasificado en dos conjuntos: por una parte todos los artículos los cuáles hemos clasificado de tipo UML, ERS y CU (por la finalidad del contenido) y por otra parte solamente los artículos que estudian directamente la mejora de calidad en la construcción de CU, en los cuales se ha sintetizado la información. En base a 10 estudios primarios de tipo UC identificados y categorizados - nuestro estudio ha revelado que existen muy pocos estudios primarios publicados acerca de mejorar la calidad de los CU -, encontramos que la comprensibilidad y la completitud son los aspectos de los CU más evaluados, los escenarios son el formato de descripción más comúnmente utilizado y que los Sistemas de Gestión de Información son el principal dominio donde los experimentos son observados.

En cuanto a la calidad de los estudios primarios seleccionados 2 estudios se han clasificado como muy relevantes (mucha calidad), 4 como bastante relevantes (bastante calidad), 2 poco relevantes (poca calidad) y 1 muy poco relevante (muy poca calidad). Haciendo caso de tal evaluación, solamente 6 estudios han sido considerados para ser sintetizados. Estos estudios, se centran en evaluar la calidad de guías para construir escenarios y CU, siendo las guías CREWS [88] como las guías más utilizadas.

A partir de la síntesis de estos estudios primarios, concluimos:

- El procedimiento que ha demostrado ser más efectivo para la construcción de CU acorde a las necesidades son las guías. El uso de *Content Guidelines* (CGs) conduce a descripciones de CU más correctas en términos de acciones de descripción completas. El uso de CGs conduce menos a descripciones de acciones menos apropiadas. El uso de *Style Guidelines* conduce a descripciones de CU que contienen descripciones más correctas y menos ambiguas para la estructura de flujo en el CU. Las guías Template y Style mejoran la comprensibilidad.

- Los aspectos de calidad más investigados en el desarrollo de los CU son: la completitud, la comprensibilidad y la comunicación. Además, la eficiencia es un factor muy considerado en consecuencia de conseguir el objetivo en el menor tiempo posible.

- La característica considerada más importante es la comprensibilidad, ya que las especificaciones de CU cumplen con la función de elicitación – que comprende actividades que activan la comprensión de los objetivos - con la importancia que conlleva en la IR, ya que el propósito principal de la IR es comprender las necesidades de los clientes y los contextos donde el software desarrollado será utilizado.

- Para capturar todos los escenarios de un sistema, según [91], utilizando las “Content Guidelines” como guías para construir CU se consigue completitud en descripciones de acciones y completitud en número de acciones.

- El formato de descripción más utilizado son los escenarios, pero no se ha llegado a la conclusión de que incrementen o garanticen necesariamente calidad en especificaciones de CU. Sin embargo, sí nos dan a entender que es un procedimiento de captación de requisitos bastante completo – en general es el que más calidad semántica ofrece: completitud, consistencia, corrección, precisión, ambigüedad y concisión.

- Los dominios donde se ha experimentado no parecen ser un factor importante. Mayoritariamente se experimentan en sistemas que hemos llamado generales (de información y gestión), aparte de que también se mencionan sistemas embebidos, sistemas de comercio electrónico y sistemas de tiempo real, no hay una aportación clara en nuestro estudio considerando diferentes tipos de sistemas .

Como trabajos futuros sería interesante llevar a cabo replicaciones de 8 de los 10 estudios que hemos seleccionado - ya que ya existe una replicación [91] sobre un estudio [88] en la misma selección que confirma alguna información y desmiente otra. Así, a pesar de basarse en evidencias empíricas, se pueden utilizar diferentes heurísticas y esquemas de marcado, y con sujetos semejantes, para asegurar los avances en el conocimiento de construcción de CU. Se necesitan más acciones técnicas de investigación para entender los problemas del uso de las técnicas de CU en contextos específicos, donde los clientes tienen diferentes roles y necesidades que impactarían en cualquier evaluación empírica. Se requiere más investigación para evaluar las técnicas de CU en el contexto de la vida real.

Siguiendo el protocolo de investigación es importante emprender nuevos estudios en otras técnicas de especificación de requisitos para poder llevar a cabo un estudio terciario, a partir de los estudios secundarios ya realizados por ejemplo en: CU, análisis de comunicación, documentos de especificación de requisitos, etc.

Se considera recomendable actualizar en un futuro la revisión realizada para ir incluyendo los nuevos estudios primarios que vayan surgiendo. Además también, se podría extender la RS incluyendo otras técnicas de especificación, a parte de la técnica en la que nos hemos centrado que son los Casos de Uso.

Agradecimientos

Quisiera dar mis más sinceros agradecimientos a todas las personas que con su apoyo y colaboración han hecho posible la realización y ejecución de la presente Tesis de Máster, en especial a mis directores Óscar Pastor López y a Nelly Condori-Fernández.

También agradecer a mi familia y amigos la comprensión por el tiempo robado, y a mis compañeros de trabajo por el apoyo psicológico que me han transmitido cada día para seguir adelante.

A todos ellos, muchas gracias por su inestimable ayuda y colaboración.

Referencias

- [1] Booch G., Rumbaugh J., Jacobson I., The Unified Modeling Language User Guide, Addison-Wesley (1999).
- [2] A. Herrmann, M. Daneva, Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research, 16th International Requirements Engineering Conference, IEEE Computer Society, 8-12 September 2008, Barcelona, Spain, pp. 125-134.
- [3] A. M. Davis, Ó. Dieste, A. M. Hickey, N Juristo, A. Moreno, Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review, 14th International Conference on Requirements Engineering (IR 2006), IEEE Computer Society, 11-15 September 2006, Minneapolis, USA, pp. 176-185.
- [4] K. Ahmed, A Systematic Review of Software Requirements Prioritization, Master Thesis in Software Engineering, School of Engineering Blekinge Institute of Technology, Sweden, October 2006.
- [5] K. Petersen, R. Feldt, M. Shahid, M. Mattsson, Systematic Mapping Studies in Software Engineering, 12th International Conference on Evaluation and Assessment in Software Engineering (EAIS), Department of Informatics, University of Bari, Italy, June 2008.
- [6] R. Pretorius, D. Budgen, A mapping study on empirical evidence related to the models and forms used in the uml, Proceedings of the Second International Symposium on Empirical Software Engineering and Measurement, EISM, October 9-10, 2008, Kaiserslautern, Germany, pp. 342-344
- [7] T. Dyba, T. Dingsoyr, Strength of evidence in systematic reviews in software engineering, Proceedings of the Second International Symposium on Empirical

Software Engineering and Measurement, EISM, October 9-10, 2008, Kaiserslautern, Germany, pp. 178-187.

[8] J. Bailey, D. Budgen, M. Turner, B. Kitchenham, P. Brereton, S. Linkman, Evidence relating to Object-Oriented software design: A survey, First International Symposium on Empirical Software Engineering and Measurement (EISM), 2007, pp.482-484.

[9] D. I. K. Sjoberg, T. Dyba, M. Jorgensen, The Future of Empirical Methods in Software Engineering Research, International Workshop on the Future of Software Engineering, FOIS 2007, May 23-25, 2007, Minneapolis, USA, pp. 358-378.

[10] B. Kitchenham, O. Pearl, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic Literature reviews in software engineering – A systematic literature review, Information and Software Technology, 51(1): 7-15, January 2009.

[11] Biolchini J., Mian P.G., Natali A.C., Travassos, G.H. Systematic Review in Software Engineering: Relevance and Utility, Technical Report ES67905, PESC - COPPE/UFRJ, 2005.

[12] Pete McBreen, Using Use Cases for Requirements capture, Research Report, 1999 McBreen Consulting, New York.

[13] Bashar Nuseibeh, Steve Easterbrook, Requirements Engineering: A Roadmap, In A. C. W. Finkelstein (ed) "*The Future of Software Engineering*" (companion volume to the proceedings of the 22nd International Conference on Software Engineering, ICIS'00), IEEE Computer Society Press.

[14] J. Krogstie, A. Solvberg, Information Systems Engineering: Conceptual Modeling in a quality perspective, Kompendiumforlaget, Trondheim, Norway, 2003.

- [15] A. Davis, S. Overmeyer, K. Jordan, Identifying and measuring quality in a software requirements specification, Proceedings of the First International Software Metrics Symposium, 1993.
- [16] John Krogstie, Evaluating UML Using a Generic Quality Framework, UML and the unified process, IGI Publishing, Hershey, PA, 2003.
- [17] Barbara Kitchenham, Procedures for Performing Systematic Reviews, Joint Technical Report Software Engineering Group Department of Computer Science Keele University Keele, Staffs ST5 5BG, UK, 2004.
- [18]. ANSI/IEEE Std. 729-1983: IEEE Standard Glossary of Software Engineering Terminology. New York: IEEE Inc. 1983.
- [19] Paula Mian, Tayana Conte, Ana Natali, Jorge Biolchini, Guilherme Travassos, A Systematic Review Process for Software Engineering, in EISLAW '05: 2nd Experimental Software Engineering Latin American Workshop, 2005.
- [20] Jacobson I., Booch G., Rumbaugh J., The Unified Software Development Process, Addison Wesley, 1998.
- [21] P. J. Denning. What is software quality, Communications of the ACM, 35(1): 13-15, January 1992 .
- [22] A. Cockburn, Structuring Use Cases with Goals, Journal of Object-Oriented Programming, Sep-Oct, 1997 and Nov-Dic, 1997.
- [23] B. Dobing and J. Parsons, How UML is used, Commun. ACM 49(5), 2006, pp. 109-113.
- [24] A.J.H. Simons, Use Cases Considered Harmful, Technology of Object-Oriented Languages and Systems Nancy, France, IEEE Computer Society, 1999, pp. 194-203.

- [25] John Mylopoulos, Jaelson Castro, Tropos: A Framework for Requirements-Driven Software Development, in J. Brinkkemper and A. Solvberg (eds.), *Information Systems Engineering: State of the Art and Research. Themes, Lecture Notes in Computer Science*, Springer-Verlag, June 2000, pp. 261-273.
- [26] Jaap Gordijn, Hans de Bruin, Hans Akkermans, Scenario Methods for Viewpoint Integration in e-Business Requirements Engineering, in *Proceedings of the 34th Hawaii International Conference On System Sciences*, IEEE, 2001.
- [27] Elena Navarro, Patricio Letelier, Jose Antonio Mocholi, Isidro Ramos, A Metamodeling Approach for Requirements Specification, *Journal of Computer Information Systems*, XLVI(5), pp. 67-77, Special Issue on Systems Analysis and Design, ed. Keng Siau, 2006.
- [28] Ivar Jacobson, Pan-Wei Ng, *Aspect-Oriented Software Development with Use Cases*, (*Addison-Wesley Object Technology Series*), 2004, Addison-Wesley, ISBN 0321268881.
- [29] Daniel L. Moody, The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods, in: *Proceedings of 11th European Conference on Information Systems ECIS*, 2003.
- [30] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, Software Engineering Standards Committee of the IEEE Computer Society. 1993.
- [31] IEEE Std 830-1998, An American National Standard - IEEE Guide to Software Requirements Specifications, Software Engineering Technical Committee of the IEEE Computer Society. 1984.
- [32] Sylvie Trudel, Alain Abran, Improving Quality of Functional Requirements by Measuring Their Functional Size, *Proceedings of the International Conferences on*

Software Process and Product Measurement, November 18-19, 2008, Munich, Germany.

[33] Alfred Hussein, Software Requirements Specification (ERS) Checklist, University of Calgary

[34] Karl E. Wiegers, Checklist for Requirements Specification Reviews, <http://www.tol.oulu.fi/users/ilkka.tervonen/WChecklist.pdf>

[35] Marcus Ciolkowski, Martín Soto (IEIS), Jean-Christophe Deprez, Measurement Requirements Specifications, (Specification of Goals for the QualOSS Quality Model) http://www.qualoss.org/about/Progress/deliverables/WP1_Deliverable1.2_final.pdf

[36] Phalp K., Vincent J., Cox K., Assessing the quality of use case descriptions, *Software Quality Journal*, 15(1), 69–97, March 2007.

[37] Ricardo Ramos, Jaelson Castro, Fernanda Alencar, Joao Araújo, Ana Moreira, Rosângela Penteadó, Quality Improvement for Use Case Model, in SBES '09: Proceedings of the 2009 XXIII Brazilian Symposium on Software Engineering, October 2009, IEEE Computer Society.

[38] ERS Document Guidelines, <http://dict.mizoram.gov.in/download/books/ERS%20Document%20Guidelines.pdf>

[39] E. D. Falkenberg, W. Hesse, P. Lindgreen, B. E. Nilsson, J. L. Han Oei, C. Rolland, R. K. Stamper, F. J. M. Van Assche, A. A. Verrijn-Stuart, and K. Voss. FRISCO: A Framework of Information System Concepts. Technical report, IFIP WG 8.1, December 1997.

[40] O. I. Lindland, G. Sindre, A. Solvberg, Understanding quality in conceptual modeling. *IEEE Software*, pages 42-49, April 1994.

- [41] K. Pohl, The three dimensions of requirements engineering: A framework and its applications, *Information Systems*, 19(3):243-258, April 1994.
- [42] J. A. Bubenko jr., Information modelling in the context of systems development, In *IFIP Conference*, 1980. Invited paper.
- [43] F. Dignum, T. Kemme, W. Kreuzen, R. Weigand, R. P. van de Riet, Constraint modelling using a conceptual prototyping language, *Data and Knowledge Engineering*, (2):213-254, 1987.
- [44] M. Jarke, J. A. Bubenko jr, C. Rolland, A. Sutcliffe, Y. Vassiliou. Theories underlying requirements engineering: An overview of NATURE at genesis, in *Proceedings of the IEEE International Symposium on Requirements Engineering (IR'93)*, pages 19-31, 1993.
- [45] C. H. Kung, A Temporal Framework for database specification and verification, in *Proceedings of the tenth International Conference on Very Large Data Bases (VLDB'84)*, pages 91-99, Singapore, August 27-31 1984.
- [46] R. Wieringa, Three roles of conceptual models in information system design and use, In E. Falkenberg and P. Lindgren, editors, *Information System Concepts: An In-Depth Analysis*, pages 31-51. North-Holland, 1989.
- [47] A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2001, ISBN 0201702258.
- [48] C. Rolland, C. Ben Achour, Guiding the construction of textual Use Case specifications, in *Data & Knowledge Engineering Journal*, Vol 25, N°1-2, pp.125-160, (ed. P. Chen, R.P. van de Riet), North Holland, Elsevier Science Publishers, March 1998.
- [49] Lilly S., Use Cases Pitfalls: Top 10 Problems from Real Projects Using Use Cases. *Proceedings of TOOLS USA '99*, IEEE Computer Society, april, 1999.

[50] J. Calmon, P. Gomes, A. Cruz, T. Uchoa, G. Travassos, Scientific research ontology to support systematic review in software engineering. *Advanced Engineering Informatics* 21(2): 133-151, 2007.

[51] Tore Dyba, Barbara A. Kitchenham, Magne Jorgensen, *Evidence-Based Software Engineering for Practitioners*, published by the IEEE Computer Society 22(1) 58-65, 2005.

[52] Barbara A. Kitchenham, Dag I.K. Sjoberg, O. Pearl Brereton, David Budgen, Tore Dyba, Martin Host, Dietmar Pfahl, Per Runeson, Can We Evaluate the Quality of Software Engineering Experiments?, in: 4th IEEE International Symposium on Empirical Software Engineering and Measurement (EISM'2010), September 16-17, 2010, Bolzano-Bozen, Italy, ACM, 2010.

[53] Boehm B.W., Sullivan K.J., Software economics: a roadmap, in *Proceedings of International Conference on Software Engineering – Future of Software Engineering Track*. 319–343. June 2000.

[54] Gerald Kotonya, Ian Sommerville, Steve Hall, Towards A Classification Model for Component-Based Software Engineering Research, in *Proceedings of the 29th IEEE Euromicro Conference*, Antalya, Turkey, pp 43-52, 1-6 September 2003.

[55] Betty H. C. Cheng, Joanne M. Atlee, Research Directions in Requirements Engineering, 2007 *Future of Software Engineering*, p.285-303, May 23-25, 2007

[56] Kitchenham, B.A., Dyba, T. and Jorgensen, M. 2004 Evidence-based Software Engineering. *Proceedings of the 26th International Conference on Software Engineering, (ICIS'04)*, IEEE Computer Society, Washington DC, USA, pp 273-281.

[57] Barbara Kitchenham, What's up with software metrics? – A preliminary mapping study, *The Journal of Systems & Software* (January 2010), 83 (1), pg. 37-51

[58] Dyba T., Dingsoyr, T. Empirical studies of agile software development: A systematic review, *IST*, 50, pp 833-859, 2008b.

[59] Tao Yue, Lionel C. Briand, Yvan Labiche, A Systematic Review of transformation Approaches between User Requirements and Analysis Models, pages 1-25, Springer London, 2010

[60] Mikael Svahnberg, Tony Gorschek, Robert Feldt, Richard Torkar, Saad Bin Saleem, Muhammad Usman Shafique, A systematic review on strategic release planning models, *Information and Software Technology*, v.52 n.3, p.237-248, March, 2010

[61] Mellado D., Blanco C., Sanchez L.E., Fernández-Medina E., A systematic review of security requirements engineering, *Computers Standards & Interfaces* v32, 153-165, 2009.

[62] M. B. P. Budgen D., Turner and B. Kitchenham, Using mapping studies in software engineering, in 20th Annual Meeting of the Psychology of Programming Interest Group (PPIG), pages 195--204, Lancaster, United Kingdom, Sept. 2008.

[63] Condori-Fernandez Nelly, Daneva Maya, Sikkel Klaas, Wieringa Roel, Dieste Oscar, Pastor Oscar, (2009) A Systematic Mapping Study on Empirical Evaluation of Software Requirements Specifications Techniques in: Third International Symposium on Empirical Software Engineering and Measurement, 15-16 Oct 2009, Buena Vista, US. pp. 503-505. IEEE Computer Society Press. ISBN 978-1-4244-4841-8.

[64] Racheva Z., Daneva M., Sikkel K., Value Creation by Agile Projects: Methodology or Mystery?, in: 10th International Conference on Product-FoCued Software Process Improvement (FROFES 2009), 15-17 Jun 2009, Oulu, Finland. pp. 141-155. Lecture Notes in Business Information Processing 32. Springer Verlag. ISBN 978-3-642-02151-0.

[65] Guilherme H. Travassos, Oscar Dieste, Martin Höst, Oscar Pastor, Investigating Results Variations In Synthesis Accomplished By Independent Research Groups Using A Common Research Protocol In Software Engineering.

[66] Charters Stuart, Budgen David, Turner Mark, Kitchenham Barbara, Brereton Pearl & Linkman, Stephen (2009), Objectivity in Research: Challenges from the Evidence-Based Paradigm, 20th Australian Software Engineering Conference, ASWEC 2009, IEEE Computer Society Press.

[67] Alan M. Davis, Ann M. Hickey, Óscar Dieste Tubío, Natalia Juristo Juzgado, Ana María Moreno, A Quantitative Assessment of Requirements Engineering Publications - 1963-2006, REFSQ 2007: 129-143

[68] O. Dieste and N. Juristo, Systematic Review and Aggregation of Empirical Studies on Elicitation Techniques, IEEE Transactions on Software Engineering, vol. 2008

[80] Bernardez B., Genero M., Duran A., Toro M., A controlled experiment for evaluating a metric-based reading technique for requirements inspection, in 10th International Software Metrics Symposium (METRICS 2004), IEEE Computer Society, ISBN 0-7695-2129-0, pág. 257-275.

[81] Kamsties E., Von Knethen A., Reussner R., A controlled experiment to evaluate how styles affect the understandability of requirements specifications, Information and software technology, ISSN 0950-5849, Vol. 45, N° 14, 2003, pags. 955-965.

[82] Pow-Sang J.A., Nakasone A., Imbert R., Moreno A.M., An approach to determine software requirement construction sequences based on use cases, Proceedings of the 2008 Advanced Software Engineering and Its Applications, Pages: 17-22, Year of Publication: 2008, ISBN:978-0-7695-3432-9, IEEE Computer Society.

[83] Briand L.C., Labiche Y., Di Penta M., Yan-Bondoc H., An experimental investigation of formality in UML-based development, *IEEE Transactions on Software Engineering* Volume 31, Issue 10 (October 2005), Pages: 833 – 849, ISSN:0098-5589, IEEE Press.

[84] Laporti V., Borges M.R.S., Braganholo V., Athena: A collaborative approach to requirements elicitation, *Computers in Industry*, Volume 60, Issue 6 (August 2009), ISSN:0166-3615, Elsevier Science Publishers B. V.

[85] Lange C.F.J., Chaudron M.R.V., Effects of defects in UML models - An experimental investigation, in *Proceedings of the 28th international conference on Software engineering*, Shanghai (China), ISSSION: Research papers: development with UML, Pages: 401 – 411, 2006, ISBN:1-59593-375-1, ACM.

[86] España S., Condori-Fernandez N., Gonzalez A., Pastor O., Evaluating the completeness and granularity of functional requirements specifications: A controlled experiment, in *Proceedings of the 2009 17th IEEE International Requirements Engineering Conference*, Pages: 161-170, ISBN ~ ISSN:1090-705X , 978-0-7695-3761-0, IEEE Computer Society.

[87] Laitenberger O., Atkinson C., Schlich M., El Emam K., Experimental comparison of reading techniques for defect detection in UML design documents, *Journal of Systems and Software* Volume 53 , Issue 2 (August 2000), Pages: 183 – 204, ISSN:0164-1212, Elsevier Science Inc.

[88] Ben Achour C., Rolland C., Maiden N.A.M., Souveyet C., Guiding use case authoring: results of an empirical study, in *Proceedings of 4th IEEE International Symposium on Requirements Engineering*, 7-11 June 1999 Limerick , Ireland IEEE Comput. Soc, 1999, pp.36-43.

[89] Hogganvik I., Stolen K., On the comprehension of security risk scenarios, in *Proceedings of the 13th International Workshop on Program Comprehension*, p.115-124, May 15-16, 2005.

[90] Moisiadis Frank, Prioritizing scenario evolution, Proceedings of the 4th International Conference on Requirements Engineering (ICIR'00), Page: 85, ISBN:0-7695-0565-1, IEEE Computer Society.

[91] Cox K., Phalp K., Replicating the CREWS use case authoring guidelines experiment, Empirical Software Engineering Journal, 5(3), pp.245-268.

[92] Udomchaiporn A., Prompoon N., Kanongchaiyos P., Software requirements retrieval using use case terms and structure similarity computation, IEEE Proceedings of 13th Asia Pacific Software Engineering Conference (APISC'06), Bangalore, India, December 6-8, 2006.

[93] Islam S., Omasreiter H., Systematic use case interviews for specification of automotive systems, in Proceedings of the 12th Asia-Pacific Software Engineering Conference, Pages: 17 - 24, 2005, ISBN ~ ISSN:1530-1362 , 0-7695-2465-6, IEEE Computer Society.

[94] Carmine Gravino, Giuseppe Scanniello, Genoveffa Tortora, An empirical investigation on dynamic modeling, in Proceedings of 11th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Toulouse, France, September 28 - October 3, 2008, Springer Verlag LNCS Press, pp. 615-629.

[95] Bente Anda, Dag Sjoberg, Magne Jorgensen, Quality and understandability of use case models, in: ECOOP 2001 - Object-Oriented Programming, 15th European Conference, ed. by J. Lindskov Knudsen, pp. 402-428, Budapest, Hungary, Springer-Verlag. Lecture Notes in Computer Science (ISBN: 3-540-42206-4), 2001.

[96] Cox Karl, Phalp Keith, Shepperd Martin, Comparing Use Case Writing Guidelines, in: Achour-Salinesi, C., Opdahl, A., Pohl, K., Rossi, M. (Eds.), Proceedings of IRFSQ'2001 - 7th International Workshop on Requirements

Engineering: Foundations for Software Quality, Essener Informatik Beitrage, Interlaken, Switzerland, June 4-5, p. 101-112.

[97] Phalp K.T., Vincent J., Cox K., Improving the quality of use case descriptions: empirical assessment of writing guidelines, 14th Intl Software Quality Management Conference, Southampton, 10-12 April, 2006, The British Computer Society.

[98] Mustapha Tawbi, Camille Ben Achour, Fernando Vélez, Guiding the process of requirements elicitation through scenario analysis: results of an empirical study, in 10th International Workshop on Database and Expert Systems Applications, 1-3 Proceedings (<http://computer.org/cspress>). ISBN: 0-7695-0281-4.

[99] Irit Hadar, Tsvi Kuflik, Anna Perini, An Empirical Study of Requirements Model Understanding: Use Case vs. Tropos Models, in Proceedings of the 25th Symposium On Applied Computing, ACM SAC 2010, Switzerland, March 2010.

Lista de Tablas

Tabla 1. "Mapping studies" en Ingeniería de Software	29
Tabla 2. Atributos de calidad recomendados por el estándar 830	35
Tabla 3. Atributos de calidad propuestos por A. Davis	36
Tabla 4. Clasificación de los atributos de calidad	41
Tabla 5. Componentes de la revisión del problema	45
Tabla 6. Combinación para la obtención de la cadena de búsqueda.....	48
Tabla 7. Proceso de selección de estudio	50
Tabla 8. Información a extraer	51
Tabla 9. Distribución de la literatura encontrada.....	55
Tabla 10. Estudios seleccionados.....	59
Tabla 11. Información a extraer.....	62
Tabla 12. Lista de dominios	64
Tabla 13. Información descriptiva y técnica por estudio seleccionado.....	65
Tabla 14. Distribución de dimensiones y atributos de calidad junto con los estudios correspondientes.....	69
Tabla 15. Distribución de atributos de calidad detectados y estudios seleccionados.....	72
Tabla 16. Distribución de formatos de descripción con los estudios primarios correspondientes.....	75
Tabla 17. Distribución de dominios con los estudios primarios correspondientes.....	76
Tabla 18. Distribución de estudios en relación a la dimensión de calidad y el formato de descripción.....	78
Tabla 19. Distribución de estudios en relación a la dimensión de calidad y al dominio	78
Tabla 20. Lista de estudios seleccionados de Casos de Uso.....	80
Tabla 21. Distribución de dimensiones de calidad junto con sus atributos correspondientes y los estudios de Casos de Uso afectados.....	81

Tabla 22. Lista de estudios de Casos de Uso con los atributos de calidad correspondientes.....	82
Tabla 23. Distribución de los formatos de descripción con los estudios de Casos de Uso correspondientes.....	84
Tabla 24. Distribución de dominios con los estudios de Casos de Uso	85
Tabla 25. Distribución de los estudios de Casos de Uso en relación de las dimensiones de calidad y los formatos de descripción	86
Tabla 26. Distribución de estudios de Casos de Uso en relación a la dimensión de calidad y los dominios	86
Tabla 27. Puntuaciones de evaluación para los estudios de Casos de Uso..	88
Tabla 28. Relevancia de los estudios de Casos de Uso.....	89
Tabla 29. Puntuaciones por pregunta de evaluación en los estudios de Casos de Uso	90

Lista de Figuras

Figura 1. Proceso de Revisión Sistemática	23
Figura 2. Modelo de marco de calidad de Lindland et al	38
Figura 3. Marco de calidad de Krogstie para el Modelado Conceptual (versión inicial).....	39
Figura 4. Marco de calidad de Krogstie para el Modelado Conceptual (versión 1.4).....	40
Figura 5. Distribución de la literatura incluida.....	56
Figura 6. Relación de estudios por dimensión de calidad	70
Figura 7. Relación de atributos de calidad con estudios seleccionados	74
Figura 8. Relación de formatos de descripción con estudios seleccionados .	75
Figura 9. Relación de dominios con los estudios seleccionados	77
Figura 10. Relación de dimensión de calidad con los estudios de Casos de Uso.....	82
Figura 11. Relación de atributos de calidad con los estudios de Casos de Uso	83
Figura 12. Relación de los formatos de descripción con los estudios de Casos de Uso	84
Figura 13. Relación de los dominios con los estudios de Casos de Uso.....	85