

Use case technique for requirements modeling in distributed development environments

A mapping study

Viviana Alferillo, María Inés Lund

Instituto de Informática
Universidad Nacional de San Juan
San Juan, Argentina

alferillo.viviana@gmail.com; mlund@iinfo.unsj.edu.ar

Gerardo Maturro

Departamento de Ingeniería de Software
Universidad ORT Uruguay
Montevideo, Uruguay
maturro@uni.ort.edu.uy

Abstract— This paper presents the process and outcomes of a systematic literature mapping about the application of the Use Cases technique for software requirements specification in distributed software development environments. The purpose is to identify which methods, tools and methodologies are reported in literature as more frequently used and how the Use Case technique is applied when the project team is not collocated.

Keywords—Systematic literature mapping, use cases, requirements engineering, distributed software development

I. INTRODUCCION

Actualmente, el desarrollo de proyectos de software ha cruzado los límites de las fronteras, pasando de desarrollos locales a proyectos geográficamente distribuidos. Esta distribución geográfica plantea nuevos retos [1] en comunicaciones, en coordinación de las actividades, en las especificaciones de requisitos de software, en especificaciones de interfaz, en la gestión de proyectos, en los recursos humanos y cómo asegurar la comprensión entre personas que no están cara a cara, entre otros desafíos.

Cuando se trabaja en proyectos de software de desarrollo global, todas estas tareas deben realizarse de manera distribuida, involucrando actores de diferentes culturas, situados en regiones con distintas zonas horarias. Si el desarrollo de un proyecto de software, cuando todos sus integrantes se encuentran en el mismo sitio, es una tarea difícil, es posible imaginar la complejidad ahora dividiendo el equipo de trabajo en diferentes países, zonas horarias, diferencias culturales y de idiomas [2]. Estos obstáculos pueden llevar a importantes retrasos o incluso a completos fracasos, si no son considerados desde su concepción.

El desarrollo distribuido presenta una serie de ventajas, se trabaja con personal altamente calificado, existe una mayor productividad del equipo de trabajo, se organiza y aprovecha mejor el tiempo, mejora la comunicación sincrónica y asincrónica, se reducen los costos, se minimiza el desplazamiento del personal. Los miembros de un proyecto distribuido trabajan en colaboración, para lograr el éxito del Proyecto [1].

A menudo esto se hace a través de correo electrónico, internet y otras formas de rápida comunicación. La especificación de requisitos define los servicios que un sistema debe proveer y establece los límites y restricciones en las operaciones del mismo. En la actualidad, el conjunto de procesos y métodos que tienen por objetivo capturar y formalizar los requisitos de un sistema se ha denominado Ingeniería de Requisitos [3].

La Ingeniería de Requisitos es un campo muy activo dentro de la Ingeniería del software, y se dirige específicamente a ciertas actividades esenciales en el trabajo diario de las organizaciones de desarrollo de software. Se ha demostrado, mediante varios estudios experimentales [4], que la Ingeniería de Requisitos es crítica respecto del éxito o fracaso de numerosos proyectos informáticos y su mala gestión tiene una gran incidencia, probada en relación con el incremento de costos o el incumplimiento de plazos de finalización del proyecto. La Ingeniería de Requisitos, se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requisitos para un producto determinado [3].

El propósito del artículo es indagar en la utilización de la técnica de Casos de Uso en entornos de desarrollo distribuido de software para la especificación y gestión de requerimientos software.

Este trabajo fue realizado con fecha de diciembre de 2014 y forma parte de la tesis de Licenciatura del primer autor, dirigida por segundo y tercer autores [5]. Se enmarca en los proyectos: “Casos de Uso. Línea base para el proceso de desarrollo de software”. Financiado por UNSJ (Res N°. 18/14-CS) y “Red Latinoamericana de Coordinación y Cooperación para Unificar Buenas Prácticas de desarrollo de software – CoCoNet-LA”. Financiado por SPU (Res N°. 2340-SPU/2012).

El resto del trabajo está organizado de la siguiente manera. En la sección II se exponen los conceptos que enmarcan el trabajo, tales como Mapeo sistemático de literatura, Desarrollo distribuido de software, breve

descripción de la Ingeniería de Requisitos y finalmente el modelo de Casos de Uso. En la sección III se presenta el desarrollo del trabajo realizado, siguiendo los pasos del mapeo sistemático planteado. En la sección IV se interpretan los datos recolectados en el Mapeo Sistemático de Literatura. En la sección V se describen los tipos de estudios detectados y, finalmente, en la sección VI se exponen las conclusiones arribadas, consideraciones finales, trabajo a futuro y posibles ampliaciones.

II. MARCO TEÓRICO

A. Mapeo Sistemático de la Literatura

Un mapeo de la literatura, también denominado revisión de alcance, es una revisión de la literatura que suele hacerse en forma previa a una revisión completa y sistemática para determinar el alcance de la investigación existente sobre un tema particular, esto es, para evaluar y ubicar los tipos de estudios llevados a cabo hasta el presente [6]. Los estudios de mapeo pueden ser de gran importancia para los investigadores en ingeniería de software, puesto que proporcionan una visión general de la literatura en temas específicos [7] [8] a partir de preguntas que permitan resolver el cuestionamiento de investigación principal de un proyecto.

Para llevar a cabo un mapeo sistemático (MS), Petersen y otros [9] sugieren un proceso de cinco pasos para realizar el mapeo sistemático, que son los siguientes: 1) definir las preguntas de investigación, 2) realizar la búsqueda de toda la documentación existente, 3) seleccionar la documentación en base a criterios de inclusión y exclusión, 4) categorizar la documentación en base a las palabras claves y abstracts, según distintas características, y 5) extraer datos y mapear la documentación relevante seleccionada.

El proceso de mapeo sistemático seguido en la presente investigación y que se sugiere en [9], se presenta la Fig. 1.

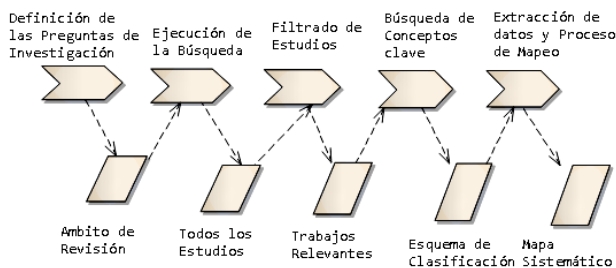


Fig. 1. Proceso de Mapeo Sistemático

B. Desarrollo Distribuido de Software

El Desarrollo Distribuido de Software (DDS) es una tendencia en crecimiento, por el auge de la globalización y fusión de empresas. Especialmente en los negocios relacionados con la producción de tecnologías de software la globalización ha producido un importante impacto, tanto en las estrategias de marketing y distribución como en la

manera en que los productos son concebidos, diseñados, contruidos, probados y entregados a los clientes [10].

El principal motivo del crecimiento del DDS es que permite disminuir los costos de desarrollo mientras se mantiene el nivel de calidad del proceso [11], y aumentar la productividad por medio de jornadas de trabajo más extensas, teniendo programadores distribuidos en sitios con amplia diferencia horaria [12]. Sin embargo, por su característica distribuida, los proyectos de DDS enfrentan varios problemas ocasionados por la distancia entre los participantes. El desarrollo distribuido de software es una fuente de nuevos retos para la Ingeniería de Requisitos [13].

C. Ingeniería de Requisitos

La Ingeniería de Requisitos (IR) cumple un papel primordial en el proceso de producción de software, ya que se enfoca en un área fundamental: la definición de lo que se debe producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes, de esta manera se pretende minimizar los problemas relacionados por la mala gestión de los requisitos en el desarrollo de sistemas.

La gestión de los requisitos es el conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y sus cambios en cualquier momento [14]. Esta actividad asegura la consistencia entre los requisitos y el sistema construido y es importante considerarla en la planificación del proyecto ya que consume tiempo y esfuerzo [15].

“La IR ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software” [3].

En el proceso de IR [16] existen cuatro actividades básicas que se tienen que llevar a cabo para completar el proceso. Las cuatro actividades son: extracción, análisis, especificación y validación. Es un conjunto estructurado de actividades, mediante las cuales se obtiene, se valida y se logra dar un mantenimiento adecuado al documento de especificación de requisitos, que es el documento final de carácter formal, que se obtiene de este proceso.

D. Casos de Uso

El modelo de Casos de Uso (CU) [17] permite la captura de potenciales requisitos de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Jacobson [18] postula que: 1) los CU describen, bajo la forma de acciones y reacciones, el comportamiento de un sistema desde el punto de vista del usuario, permiten definir los límites del sistema y las relaciones entre el sistema y el entorno. Son descripciones

de la funcionalidad del sistema independiente de la implementación, basadas en el lenguaje natural.

Jacobson [18], además de introducir los CU como elementos primarios del desarrollo del software, también diseñó un diagrama para la representación gráfica de ellos. El diagrama de casos de uso es ya también parte de Unified Modeling Language (UML) [19].

Por otro lado, Sommerville [20] propone al modelo de casos de uso como una técnica que se basa en escenarios para la obtención de requisitos.

En otras palabras, un CU es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de CU sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas [21].

Debido a esto, la identificación de los casos de uso es una de las técnicas más convenientes para la especificación de requisitos.

Por lo tanto se considera significativo y necesario determinar el estado actual de la aplicación de CU en entornos de desarrollo distribuido, para especificar los requisitos de un sistema y realizar la gestión de requisitos, analizar las tendencias, como ha evolucionado su uso y el avance a través del tiempo. Es pertinente, para determinar cómo es su uso y aplicación, que técnicas de comunicación, qué herramientas de mediación, saber cuáles son los estudios existentes sobre el uso de CU en IR en proyectos de DDS, qué tipos de estudios o qué actividades son llevados a cabo para aplicar esas técnicas, estrategias, prácticas en entornos distribuidos.

Lo anteriormente planteado lleva a la necesidad de realizar un mapeo literario, a fin de encontrar respuestas a estas inquietudes.

III. TRABAJO REALIZADO

En este capítulo se describe cada etapa realizada en el Mapeo Sistemático de la Literatura [7] [9], las preguntas que se buscan responder, las búsquedas que se realizaron, selección y clasificación de los artículos.

A. Preguntas de investigación

Las preguntas de investigación, en un MS, suelen abordar cuestiones acerca de cuáles son los investigadores que han trabajado en el tema, cuanta actividad de investigación se ha realizado, qué tipos de estudios se han llevado a cabo, y cuestiones similares [7], de modo de detectar vacíos en la investigación actual y sentar las bases para nuevas investigaciones.

El objetivo general del presente trabajo se puede resumir en comprender qué técnicas, metodologías o métodos se usan para aplicar los casos de uso en entornos distribuidos de desarrollo. De él se derivan seis preguntas

de investigación genéricas para obtener mayor conocimiento y una visión integral del tema:

1. ¿Cuáles son los estudios existentes sobre la aplicación de casos de uso en ingeniería de requisitos en entornos distribuidos de desarrollo de software?
2. ¿Qué técnicas, métodos, herramientas, etc. son mencionadas en cada estudio del punto 1?
3. ¿Qué tipo de estudios se realizaron para aplicar las técnicas, métodos, estrategias, etc. identificadas en 2?
4. ¿Qué actividad de investigación ha habido a lo largo del tiempo?
5. ¿Cuáles son los principales investigadores en el tema?
6. ¿Cuáles son las principales revistas, libros y conferencias donde se publican estudios sobre el tema?

B. Búsqueda de la literatura Relevante

Las bibliotecas digitales [22] son colecciones de objetos más o menos organizadas, que sirven a una comunidad de usuarios definida, que tienen los derechos de autores presentes y gestionados y que disponen de mecanismos de preservación y conservación. Se trata de recursos informáticos documentales, a los que se accede por medio del uso de dispositivos de comunicación móvil a través de los servicios de Internet.

La búsqueda se realizó sobre las bases bibliográficas ACM, IEEEExplore, SpringerLink, ScienceDirect, y la recopilación de la documentación culminó en diciembre de 2014. La restricción a estas bases de datos bibliográficas se debe simplemente a que son las que están accesibles desde las universidades de los autores.

La identificación de la literatura relevante al tema implica definir, primeramente, las cadenas de búsqueda a utilizar en las bases bibliográficas. Para la construcción de estas cadenas se utilizaron las siguientes palabras claves:

Cadena 1: “global software development”, “global software engineering”, “distributed software development”, “distributed software engineering”, “offshore software development”, “offshore software engineering”, para considerar el entorno de desarrollo distribuido de software y todas sus acepciones.

Cadena 2: “use case”, “use case model”, “system specification” “system model”, “Unified Modeling Language”, para tener en cuenta las técnicas de modelamiento de requisitos, específicamente casos de uso.

Debido a que cada una de estas bases de datos bibliográficas propone una forma diferente de introducir las cadenas de búsqueda y los operadores lógicos OR y AND, se optó por utilizar todas las combinaciones posibles de cadenas cortas de la forma <cadena1> AND <cadena2>.

Las búsquedas se realizaron sobre los campos “Título”, “Abstract” y “Palabras claves”.

En la Tabla 1 se muestra el total de artículos que se encontraron en las respectivas búsquedas, detallando por

cada biblioteca digital, la cantidad de artículos encontrados por combinación de cadenas.

TABLA 1. CANTIDAD DE ARTÍCULOS ENCONTRADOS POR BIBLIOTECA.

Biblioteca	SDirect	Springer	IEEEEx	ACM	Total
Total Art.	28	72	117	15	232

C. Selección de los artículos pertinentes

Los 232 artículos encontrados deben ser analizados y determinar si realmente todos ellos son trabajos relacionados al tema de investigación y relevantes al objetivo que se persigue en el presente trabajo.

Los criterios de selección, también denominados criterios de inclusión/exclusión, tienen por propósito identificar, del total de artículos encontrados en las búsquedas, aquellos que proporcionan evidencias directas acerca del tema de investigación y objetivos planteados.

Los criterios definidos para este estudio son:

Criterios de inclusión: Incluir aquellos artículos que traten el tema sobre proceso de ingeniería de requisitos como así también métodos, técnicas, metodologías etc. de aplicación de casos de uso en entornos de desarrollo distribuido de software.

Criterios de exclusión: Excluir aquellos artículos donde no aparezca completa la cadena de búsqueda en el abstract, también aquellos artículos publicados en revistas o actas de conferencias no arbitradas, o en la introducción de actas o proceedings.

Se realizó la selección en base a la lectura del título, del abstract, y de las palabras claves de cada artículo resultante del proceso anterior. Se aplicaron los criterios de inclusión y exclusión, se descartaron los artículos que aparecían en más de una biblioteca digital, aquellos donde la cadena de búsqueda se encontraba en el título y al realizar la lectura del correspondiente abstract la cadena no aparecía o no correspondía al tema de investigación, también se descartaron artículos donde las cadenas de búsquedas no se encontraban como palabras claves.

De los 232 artículos, el 60% fue descartado porque al dar lectura al abstract y una mirada rápida al texto, eran trabajos que no estaban directamente vinculados al tema de investigación, solo hacían mención a las palabras incluidas en las cadenas de búsqueda, el 23% no contenían la cadena de búsqueda completa y el 8% se referían a temas de comunicación, coordinación. Sólo un 9% del total recopilado forman parte del listado final de artículos relevantes, que se presenta en la Tabla 2.

TABLA 2. CANTIDAD FINAL DE ARTÍCULOS SELECCIONADOS POR BIBLIOTECA.

Biblioteca Digital	Cantidad de Artículos	Artículos
ScienceDirect	3	[23] [24] [25]
SpringerLink	10	[26] [27] [28] [29] [30] [31] [32] [33] [34] [35]

IEEEExplore	6	[36] [37] [38] [39] [40] [41]
ACM Digital	1	[42]

D. Clasificación de los artículos

Las guías de Petersen et al. [9] sugieren la exploración de algunas partes del artículo en aquellos casos donde el abstract no se encuentra bien estructurado o es impreciso. Se decide realizar la lectura completa de cada artículo seleccionado para poder responder todas las preguntas de investigación formuladas.

La clasificación de los artículos se hizo en base a las siguientes categorías: Medios de publicación, Año de publicación, Nombres de Autores, Nombres de las revistas, libros y conferencias.

En la Tabla 3 se presenta la cantidad de artículos categorizados según el medio de publicación.

TABLA 3. CANTIDAD DE ARTÍCULOS POR MEDIOS DE PUBLICACIÓN.

Medio de publicación	Cant. Artículos	Artículos
Revista	6	[23] [24] [25] [30] [34] [35]
Sección de un libro	7	[26] [27] [28] [29] [31] [32] [33]
Conferencia	7	[36] [37] [38] [39] [40] [41] [42]

En la Tabla 4 se presenta la cantidad de artículos, ordenados cronológicamente por año desde el 2003 al 2014.

TABLA 4. CANTIDAD DE ARTÍCULOS POR AÑO DE PUBLICACIÓN.

Año de publicación	Cant. Artículos	Artículos
2003	1	[38]
2004	2	[28] [33]
2005	-	-
2006	1	[35]
2007	2	[25] [39]
2008	1	[29]
2009	2	[26] [31]
2010	5	[30] [34] [36] [41] [42]
2011	2	[27] [37]
2012	1	[40]
2013	2	[23] [32]
2014	1	[24]

E. Extracción de Datos y proceso de Mapeo

En la Tabla 5 se muestran los artículos incluidos en el mapeo, se indica la referencia y un pequeño resumen sobre de lo que se desarrolla en el respectivo artículo.

TABLA 5. LISTADO FINAL DE ARTÍCULOS RELEVANTES. MAPEO.

Art.	Resumen del artículo
[23]	Marco de diez heurísticas para el diseño, desarrollo y gestión de un equipo distribuido.
[24]	Método para modelar diagramas de casos de uso.
[25]	Herramienta sincrónica de modelado, mejorado con la gestión de versiones, en un ambiente de cooperación sincrónica de modelado basado en UML.
[26]	Evaluación de arquitectura de organizaciones dedicadas al desarrollo de software global.
[27]	Modelo basado en valor, que funciona en la ingeniería de software y valora cada entidad/requerimiento de acuerdo a las

	necesidades.
[28]	Para ayudar en la negociación de los casos de uso se utiliza un BBS (tablón de anuncios) durante el desarrollo de software. Cada caso de uso tiene su propio BBS. El cual es utilizado para discutir con respecto al correspondiente caso de uso.
[29]	Uso de una arquitectura de agentes diseñado para apoyar la Ingeniería de Requisitos, específicamente la Verificación y Validación de actividades (V & V), en el proceso de desarrollo distribuido.
[30]	Método que tiene como objetivo crear un entendimiento común sobre el futuro sistema por medio de "presentaciones inversas".
[31]	Herramientas y Entornos de desarrollo de colaboración que están disponibles para el apoyo de equipos distribuidos.
[32]	Método que permite el intercambio y la reutilización de conocimientos entre los equipos distribuidos a través de un repositorio de requisitos compartido.
[33]	Modelo de referencia para el GSD, el objetivo es apoyar el desarrollo de software global, actúa como una guía para el desarrollo de software realizado por equipos geográficamente dispersos y heterogéneos.
[34]	Marco de trabajo que se centra en el análisis de los factores que pueden ser problemáticos en el desarrollo de software distribuido, y sugiere un conjunto de estrategias para mejorar el proceso de obtención requisitos en tales ambientes.
[35]	Entorno para el análisis de los requisitos de lenguaje natural, con herramienta CIRCE.
[36]	Enfoque basado en la combinación de desarrollo basado en pruebas de aceptación y revisiones basadas en prueba, en equipos de desarrollo de software distribuido.
[37]	Método (think-pair-square: compartición de ideas) en el modelado de diagramas de casos de uso en el desarrollo de software distribuido.
[38]	La negociación y la facilitación en la fase de ingeniería de requisitos de desarrollo de software en organizaciones distribuidas pueden ser mejoradas por el uso de modos de comunicación asíncronos.
[39]	Método de evaluación metodológicamente sólido y rentable para los métodos de ingeniería de requisitos distribuidos, así como la infraestructura de evaluación para la realización de proyectos.
[40]	Estudio que compara diferentes técnicas en el proceso de elicitación de requisitos en ambientes distribuidos y ambientes co-ubicados.
[41]	Sistema de apoyo basado en agente para la definición de especificaciones de requisitos en el desarrollo de software distribuido.
[42]	Marco de evaluación sistemática y flexible en especificaciones de requisitos de software distribuido.

Al realizar la lectura de cada documento, se generaron nuevas categorías para clasificarlos según el tema que desarrollaban. Los temas más relevantes se relacionan a los problemas que se deben enfrentar en cuanto a la comunicación, colaboración, coordinación y cooperación, diferencias culturales y recursos humanos de los equipos de desarrollo distribuido de software. La información obtenida permitió identificar taxonómicamente los artículos y agruparlos de acuerdo a una clasificación, determinar el objetivo del mismo y poder observar cuales son las líneas de investigación que se están desarrollando en la actualidad

En la Tabla 6 se muestra el tipo de contribuciones que se identificaron en los artículos del mapeo.

TABLA 6 LISTADO DE LAS DIFERENTES CONTRIBUCIONES.

Tipo de aporte en cada artículo	Cant. Art.	Artículos
Herramientas de trabajo	3	[25] [31] [38]

colaborativo [43]		
Métodos empíricos [44]	9	[26] [28] [29] [30] [32] [36] [37] [39] [40]
Modelos conceptuales [45]	5	[24] [27] [33] [35] [41]
Marco de trabajo/framework [20]	3	[23] [34] [42]

En la tabla 7 se muestra los tipos de estudios que se identificaron en los artículos seleccionados en el mapeo.

TABLA 7. LISTADO DE LOS TIPOS DE ESTUDIOS IDENTIFICADOS EN LOS ARTÍCULOS.

Tipo de estudio [44][46] [47]	Cant. Art.	Artículos
Experimentos	10	[24] [27] [28] [31] [34] [36] [37] [38] [40][42]
Evaluaciones	5	[23] [25] [26] [32] [41]
Estudios de casos	2	[33] [35]
Estudios descriptivos	3	[29] [30] [39]

IV. INTERPRETACIÓN DE LOS RESULTADOS

A continuación se describen e interpretan los resultados obtenidos del mapeo sistemático de la literatura. La interpretación se realizará a través de cada una de las tablas obtenidas, que dan respuesta a las preguntas de investigación planteadas para este trabajo en la sección 3.1.

Pregunta 1: ¿Cuáles son los estudios existentes sobre la aplicación de casos de uso en ingeniería de requisitos en entornos distribuidos de desarrollo de software?

Esta respuesta se encuentra, inicialmente, en los datos obtenidos en la tabla 2 (cantidad de artículos por biblioteca, obtenidos durante el estudio realizado, luego de aplicar los criterios de inclusión y exclusión), de cuyo análisis se obtiene que SpringerLink aporta el 50% de los artículos encontrados, e IEEEExplore un 30%, el 15% corresponde a ScienceDirect y el 5% restante ACM. Esto no necesariamente significa que unas bibliotecas sean mejores que las otras, puede ser consecuencia de los motores de búsqueda.

El resultado completo de todos los estudios relevantes existentes se encuentra en la tabla 5, donde se presenta cada artículo con un breve resumen y la referencia bibliográfica. Esta tabla permite partir con un conocimiento del estado del arte de la temática involucrada en el presente estudio, para avanzar en alguna línea nueva o profundizar una existente.

Pregunta 2: ¿Qué técnicas, métodos, herramientas, etc. son mencionadas en cada estudio del punto 1?

Analizando los resultados de la tabla 6 se puede obtener la respuesta a esta pregunta, los métodos y modelos resultan ser el aporte predominante en las investigaciones actuales con un total de 9 y 5 trabajos cada uno, con un porcentaje de 45% y 25% respectivamente. En segundo lugar las herramientas y marco de trabajo con 3 trabajos cada uno, con un porcentaje de aporte de un 15%. Claramente se puede determinar que la contribución más usada, en la investigación de esta temática, son los métodos empíricos.

Pregunta 3: ¿Qué tipo de estudios se realizaron para aplicar las técnicas, métodos, estrategias, etc. identificadas en 2?

De acuerdo al análisis realizado en cada artículo seleccionado, se puede observar en la tabla 7, que por medio de experimentos se obtuvieron la mayor parte de las contribuciones, con un porcentaje muy alto, el 50% respecto de los otros estudios. En segundo lugar las evaluaciones ocupan un 25% de los artículos seleccionados. En tercer lugar estudios descriptivos se identificaron con un 15% y en último lugar estudios de casos con un porcentaje de 10% del total de los artículos seleccionados.

Por lo tanto, se puede concluir que los experimentos son el medio más utilizado para realizar contribuciones en esta área de conocimiento.

Pregunta 4: ¿Qué actividad de investigación ha habido a lo largo del tiempo?

En la tabla 4 se presenta esta información. La tendencia de publicación sobre el tema es constante, entre 1 y 2 artículos por año, desde el 2003 hasta el 2014, con excepciones en los años 2005, que no se encontraron artículos relevantes a la temática y 2010 que creció a 5.

En general, se ha encontrado un número reducido de artículos publicados referidos al tema, esto puede dar un indicio que es necesario realizar más estudios sobre la técnica de CU en el DDS.

Pregunta 5: ¿Cuáles son los principales investigadores en el tema?

Analizando los investigadores de los artículos seleccionados en este estudio. Se destacan, entre todos ellos: Giuseppe Scanniello, Ugo Erra, Gregor Engels y Frank Salger, porque participaron en más de una publicación relevante, considerada en este estudio (3 publicaciones en los dos primeros y 2 en los dos segundos).

Pregunta 6: ¿Cuáles son las principales revistas, libros y conferencias donde se publican estudios sobre el tema?

En la tabla 3 se identifican los medios de publicación elegidos para la presentación de artículos. Se observa que el 30% de ellos se encuentran publicados en revista y el 70% restante, por partes iguales entre libros y conferencias. No se puede sacar ningún tipo de conclusión, no hay preferencias de medio de publicación en esta temática.

Todos los eventos aportaron un solo artículo en esta investigación, a excepción de la revista "Journal of Visual Languages & Computing", en la que se encuentran publicados 2 artículos, lo que corresponde a un 10% del total. Tampoco se puede deducir que haya algún lugar de publicación preferido, para estas temáticas.

V. DESCRIPCIÓN DE LOS TIPOS DE ESTUDIOS

De los resultados obtenidos en el MS, se considera apropiado detallar las contribuciones encontradas respecto a los tipos de estudios identificados en la tabla 7.

Experimentos: el artículo [34] presenta un experimento sobre las diferencias de idioma de las personas involucradas en el desarrollo de un sistema en forma distribuida. Esta característica resulta un aspecto negativo y se hace necesario llegar a un acuerdo de los términos que se van a emplear cuando se trabaje en la etapa de elicitación, de manera que en el futuro no haya problemas en cuanto a la terminología que se aplique.

En el artículo [31], además de identificar una serie de herramientas que se pueden utilizar para comunicación, control y coordinación en la elicitación y negociación de requisitos cuando hay distancia geográfica entre los participantes, presentan resultados de un experimento respecto a esto, usando medios de comunicación síncrona (cara a cara vs mediada por computador). Obteniendo que hay medios más apropiados para elicitación y otros para negociación de requisitos, en DDS.

En [28] [38] se proponen experimentos relacionados con la negociación de requisitos, para obtener requisitos que cumplan con lo que el cliente necesita, asignar prioridades a los requisitos, poder agruparlos para lograr que los desarrolladores implementen los requisitos de acuerdo a los plazos estipulados para entregar el sistema, de esta forma lograr un entendimiento entre clientes y desarrolladores sobre el sistema a implementar.

En [36] [42] se realizaron experimentos para mejorar la calidad de las especificaciones de requisitos, como así también, transferir conocimientos por parte del cliente al equipo de desarrollo, sin pérdida de calidad en los requisitos. Es necesario que entre los desarrolladores haya un acuerdo en los requisitos, garantizar la viabilidad de las especificaciones, asegurar la integridad en cuanto a la asignación de trabajos desde el punto de vista del desarrollador, y también garantizar que los sistemas cumplirán con los objetivos de calidad establecidos.

En [27] se realizó un experimento donde los desarrolladores asignan un cierto valor o importancia a los requisitos cuando se desarrolla software, ya que es necesario saber que requisitos serán implementados primeros que otros, de esta forma se logra un sistema desarrollado que cumpla con lo que el cliente necesita.

Se realizaron en [24] [37] experimentos respecto a la manera o forma en que participan los desarrolladores en tareas relacionadas a la elicitación de requisitos, es decir, experimentos sobre el rendimiento, la eficiencia de los desarrolladores cuando tienen que diseñar basados en el modelo de casos de uso, también el tiempo necesario para modelar esos diagramas, aplicando diferentes técnicas de elicitación de requisitos.

En [40] se exponen experimentos aplicando combinaciones de técnicas de elicitación, donde a cada combinación de técnica se le asigna un determinado valor, de esta forma tener conocimiento de cual combinación de técnica aplicar cuando se está desarrollado software de forma distribuida, es decir, para tener cierta información sobre las habilidades y la forma en que resolverán los

problemas que se puedan llegar a presentar cuando se desarrolla un sistema en estos entornos.

Evaluaciones: En [25] se llevaron a cabo evaluaciones sobre diferentes herramientas web usadas para modelar diagramas de caso de uso, teniendo en cuenta algunas características necesarias para el trabajo colaborativo en un entorno DDS (versionado, edición concurrente, identificación de cambios y autor del mismo, etc.). En [26] realizan evaluaciones sobre la arquitectura, proponiendo un método con 9 buenas prácticas que deben ser chequeadas al evaluar una arquitectura para DDS.

En [41] evalúan los conflictos que se presentan en la especificación de requisitos, en cuanto a diferencias culturales, de idioma, y proponen un sistema basado en la nube, con dos agentes asistentes, uno para la definición de requisito y otro para la traducción del lenguaje. En el artículo [23] basados en el análisis y evaluación de numerosos artículos, proponen 10 heurísticas para planificar, organizar y gestionar, cuidadosamente, proyectos de DDS.

En [32] se evalúa como mejorar el conocimiento compartido, el reuso en la IR y la colaboración en el DDS, presentan un método de IR global y una herramienta prototipo. Gestionar el conocimiento y obtener ciertos beneficios potenciales: reducir el tiempo y costo de desarrollo de proyectos de software, evitar errores y reducir la repetición del trabajo, aumentar la productividad a través de la repetición de los procesos exitosos, aumentar la calidad y tomar mejores decisiones.

Estudios descriptivos: en [30] se describe un método para evaluar especificaciones de requisitos, basados en la perspectiva del cliente, teniendo en cuenta las distancias sociales propias del DDS. El método se caracteriza por una validación estructurada e iterativa, y por medio de presentaciones inversas crea un entendimiento común, en relación al sistema a desarrollar.

Todos los actores de un sistema se encuentran distribuidos y el desarrollo del software no se puede realizar sin algún apoyo que articule las personas eficazmente, en [29] el estudio se centra en el uso de una arquitectura de agentes, diseñada para apoyar la IR, específicamente en la verificación y validación (VV) de las actividades en el proceso de DDS. Se definen metas de alto nivel para VV y hace posible establecer requisitos y asignar responsabilidades a los actores.

En [39] proponen un método para evaluar requisitos en proyectos de DDS, combinando medidas cuantitativas obtenidas de experimentos y cualitativas obtenidas por métodos observacionales.

Estudios de casos: en [33] se presenta un modelo de referencia, basado en estudios de casos. De acuerdo a los resultados de las entrevistas las principales dificultades que se presentan están relacionadas con la IR, procesos de desarrollo de software, la gestión del conocimiento, las comunicaciones, culturas e idiomas, la confianza la comunicación. El modelo de referencia (MuNNDoS) contiene las lecciones aprendidas para DDS y se representa

con dos dimensiones: la organizacional y la específica del proyecto, a fin de resolver o mitigar los problemas descriptos anteriormente.

En [35] se presenta un entorno (CIRCE) para el análisis de requisitos en lenguaje natural. Usando un paradigma transformacional, cuyo resultado es un conjunto de modelos para el documento de requisitos, que responden a los requisitos del sistema y los del proceso de escritura. Estos modelos pueden ser inspeccionados, medidos y validados, a su vez, con un conjunto de criterios. Estas características son mostradas por medio de un ejemplo

VI. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado un MS de la literatura sobre CU en el proceso de IR, en un entorno de DDS, proporcionando un marco de trabajo actualizado para facilitar la apertura del campo a nuevas investigaciones. Este estudio ha permitido, responder todas y cada una de las preguntas de investigación planteadas, a través de la información relevada, seleccionada y clasificada. Se identificaron los estudios pertinentes que existen en el área científica para, a partir de ellos, tener un conocimiento del estado del arte de la temática involucrada en el presente estudio y con esa base avanzar en alguna línea nueva o profundizar una existente.

Se pudo observar que existen muchos métodos y modelos del desarrollo de software tradicional, que han sido modificados para poder aplicarlos al DDS, una tendencia en alza actualmente.

Para la obtención de CU, en DDS, se emplean medios de comunicación sincrónicos o asincrónicos entre los miembros de los equipos de desarrollo. El uso adecuado de estos medios es fundamental para determinar fehacientemente el modelo de casos de uso.

También se observa que existen entornos que fueron transformados para permitir a los desarrolladores aplicar los casos de uso, permitiendo en el momento de desarrollar software modificarlos de forma distribuida. Herramientas colaborativas también pudieron ser identificadas, para dar apoyo a los desarrolladores en la tarea distribuida y colaborativa.

A medida que se iba realizando el trabajo, se pudo observar que existe muy poca investigación respecto a la aplicación de la técnica de casos de uso en DDS, esto se puede notar por el bajo número de estudios resultantes del MS.

Recabar información referida a un tema particular es un proceso complejo y fundamental en el contexto de un proyecto de investigación, los investigadores y las personas en general están sometidos a una gran cantidad de información. Es entonces imprescindible e importante, planificar el método de búsqueda de forma eficaz, estructurada y sistemática para localizar la bibliografía adecuada sobre el tema de estudio.

Como trabajo a futuro sería posible ampliar las bibliotecas consultadas hacia otras que no se consideraron en este trabajo, para abarcar toda la comunidad científica, revisar las cadenas de búsqueda y verificar si existen otras acepciones para los términos elegidos. Además analizar más profundamente los trabajos involucrados en este MS, a fin de recabar otra información y generar nuevas clasificaciones.

También se pretende realizar una revisión sistemática de la literatura [48] debido al bajo número de publicaciones encontradas, esta metodología tiene una fase de revisión de trabajos mucho más rigurosa, permite establecer el estado de evidencia a través de la exhaustiva extracción de datos cuantitativos y meta-análisis, y por tanto responder a preguntas de investigación mucho más específicas.

REFERENCIAS

- [1] W. Kobitzsch, D. Rombach, y R. L. Feldmann, «Outsourcing in India [software development]», *IEEE Softw.*, vol. 18, n.º 2, pp. 78-86, mar. 2001.
- [2] S. Krishna, S. Sahay, y G. Walsham, «Managing Cross-cultural Issues in Global Software Outsourcing», *Commun ACM*, vol. 47, n.º 4, pp. 62-66, abr. 2004.
- [3] R. Pressman, «Ingeniería del software: Un enfoque práctico, 7ma Edición», *FreeLibros - Tu Biblioteca Virtual*. [En línea]. Disponible en: <http://www.freelibros.org/ingenieria/ingenieria-del-software-un-enfoque-practico-7ma-edicion-roger-s-pressman.html>. [Accedido: 30-abr-2015].
- [4] L. Westfall, «Las fallas en la Ingeniería de Requisitos», *Rev. Ing. USBMed*, vol. 2, n.º 2, pp. 40-47, 2011.
- [5] V. Alferillo, «Casos de uso en el proceso de ingeniería de requisitos para el desarrollo distribuido de software. Mapeo Sistemático de la Literatura», tesina de grado, Universidad Nacional de San Juan, San Juan, Argentina, 2015.
- [6] «Wiley: Systematic Reviews in the Social Sciences: A Practical Guide - Mark Petticrew, Helen Roberts». [En línea]. Disponible en: <http://www.wiley.com/WileyCDA/WileyTitle/productCd-1405121106.html>. [Accedido: 30-abr-2015].
- [7] B. A. Kitchenham, D. Budgen, y O. Pearl Brereton, «Using mapping studies as the basis for further research – A participant-observer case study», *Inf. Softw. Technol.*, vol. 53, n.º 6, pp. 638-651, jun. 2011.
- [8] G. Maturro y J. Saavedra, «Factores que Inciden en la Mejora de Procesos Software. Un mapeo sistemático de la literatura», en *Proceedings 15th Ibero-American Conference on Software Engineering, CibSE 2012*, Buenos Aires, Argentina, 2012, pp. 84-97.
- [9] K. Petersen, R. Feldt, S. Mujtaba, y M. Mattsson, «Systematic Mapping Studies in Software Engineering», en *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, Swinton, UK, UK, 2008, pp. 68-77.
- [10] J. D. Herbsleb y D. Moitra, «Global software development», *IEEE Softw.*, vol. 18, n.º 2, pp. 16-20, mar. 2001.
- [11] J. Audy, R. Evaristo, y M. B. Watson-Manheim, «Distributed Analysis: The Last Frontier?», en *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 1 - Volume 1*, Washington, DC, USA, 2004, p. 10010.2-.
- [12] C. Ebert y P. De Neve, «Surviving global software development», *IEEE Softw.*, vol. 18, n.º 2, pp. 62-69, mar. 2001.
- [13] K. Fryer y M. Gothe, «Global software development and delivery: Trends and challenges», *IBM Developer Works*, 15-ene-2008. [En línea]. Disponible en: https://www.ibm.com/developerworks/rational/library/edge/08/jan08/fryer_gothe/. [Accedido: 30-abr-2015].
- [14] D. Leffingwell y D. Widrig, *Managing software requirements: a use case approach*. Boston: Addison-Wesley, 2003.
- [15] A. Duran Toro, «Un entorno metodológico de ingeniería de requisitos para sistemas de información», De Sevilla, España, 2006.
- [16] Lizka Johany Herrera J., «Ingeniería De Requerimientos Ingeniería De Software», 2001. [En línea]. Disponible en: <http://www.monografias.com/trabajos6/resof/resof2.shtml>. [Accedido: 30-abr-2015].
- [17] I. Jacobson, «Object-oriented Development in an Industrial Environment», en *Conference Proceedings on Object-oriented Programming Systems, Languages and Applications*, New York, NY, USA, 1987, pp. 183-191.
- [18] I. Jacobson, *Object Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley Professional, 1992.
- [19] «UML - Unified Modeling Language - OMG», <http://www.uml.org/>. [En línea]. Disponible en: <http://www.uml.org/>.
- [20] I. Sommerville y M. I. A. Galipienso, *Ingeniería del software*. Pearson Educación, 2005.
- [21] G. Booch, J. Rumbaugh, y I. Jacobson, *The Unified Modeling Language User Guide*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1999.
- [22] «Biblioteca digital», *Wikipedia, la enciclopedia libre*. 30-abr-2015.
- [23] M. A. Babar y C. Lescher, «Global Software Engineering: Identifying Challenges is Important and Providing Solutions is Even Better», *Inf Softw Technol*, vol. 56, n.º 1, pp. 1-5, ene. 2014.
- [24] Giuseppe Scanniello y Ugo Erra, «Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments», *J. Vis. Lang. Comput.*
- [25] A. De Lucia, F. Fasano, G. Scanniello, y G. Tortora, «Enhancing collaborative synchronous UML modelling with fine-grained versioning of software artefacts», *J. Vis. Lang. Comput.*, vol. 18, n.º 5, pp. 492-503, oct. 2007.
- [26] F. Salger, «Software Architecture Evaluation in Global Software Development Projects», en *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, R. Meersman, P. Herrero, y T. Dillon, Eds. Springer Berlin Heidelberg, 2009, pp. 391-400.
- [27] M. Ramzan, A. Batool, N. Minhas, Z. U. Qayyum, y M. A. Jaffar, «Automated Requirements Elicitation for Global Software Development (GSD) Environment», en *Software Engineering, Business Continuity, and Education*, T. Kim, H. Adeli, H. Kim, H. Kang, K. J. Kim, A. Kiumi, y B.-H. Kang, Eds. Springer Berlin Heidelberg, 2011, pp. 180-189.
- [28] M. Shimakage y A. Hazeyama, «A Requirement Elicitation Method in Collaborative Software Development Community», en *Product Focused Software Process Improvement*, F. Bomarius y H. Iida, Eds. Springer Berlin Heidelberg, 2004, pp. 509-522.
- [29] M. Sayão, A. H. Filho, y H. A. do Prado, «Requirements Engineering for Distributed Development Using Software Agents», en *Advances in Conceptual Modeling – Challenges and Opportunities*, I.-Y. Song, M. Piattini, Y.-P. P. Chen, S. Hartmann, F. Grandi, J. Trujillo, A. L. Opdahl, F. Ferri, P. Grifoni, M. C. Caschera, C. Rolland, C. Woo, C. Salinesi, E. Zimányi, C. Claramunt, F. Frasinca, G.-J. Houben, y P. Thiran, Eds. Springer Berlin Heidelberg, 2008, pp. 272-281.
- [30] D. M. Wiener y D.-I. R. Stephan, «Reverse Presentations», *Bus. Inf. Syst. Eng.*, vol. 2, n.º 3, pp. 141-153, abr. 2010.
- [31] F. Lanubile, «Collaboration in Distributed Software Development», en *Software Engineering*, A. D. Lucia y F. Ferrucci, Eds. Springer Berlin Heidelberg, 2009, pp. 174-193.
- [32] J. M. C. de Gea, J. Nicolás, J. L. F. Alemán, A. Toval, A. Vizcaino, y C. Ebert, «Reusing Requirements in Global Software Engineering», en *Managing Requirements Knowledge*, W. Maalej y A. K. Thurimella, Eds. Springer Berlin Heidelberg, 2013, pp. 171-197.
- [33] R. Prikladnicki, J. L. N. Audy, y R. Evaristo, «A Reference Model for Global Software Development: Findings from a Case Study», en *International Conference on Global Software Engineering, 2006. ICGSE '06*, 2006, pp. 18-28.
- [34] Gabriela N. Aranda, Aurora Vizcaino, y Mario Piattini, «A framework to improve communication during the requirements

- elicitation process in GSD projects», *Requir. Eng.*, vol. 15, n.º 4, pp. 397-417, nov. 2010.
- [35] V. Ambriola y V. Gervasi, «On the Systematic Analysis of Natural Language Requirements with CIRCE», *Autom. Softw. Eng.*, vol. 13, n.º 1, pp. 107-167, ene. 2006.
- [36] F. Salger y G. Engels, «Knowledge transfer in global software development: leveraging acceptance test case specifications», en *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2*, New York, NY, USA, 2010, pp. 211–214.
- [37] U. Erra y G. Scanniello, «Assessing think-pair-square in distributed modeling of use case diagrams», en *2011 First International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2011, pp. 77-84.
- [38] C. L. Campbell y B. Van De Walle, «Asynchronous requirements engineering: enhancing distributed software development», en *International Conference on Information Technology: Research and Education, 2003. Proceedings. ITRE2003*, 2003, pp. 133-136.
- [39] M. Geisser, T. Hildenbrand, F. Rothlauf, y C. Atkinson, «An Evaluation Method for Requirements Engineering Approaches in Distributed Software Development Projects», en *International Conference on Software Engineering Advances, 2007. ICSEA 2007*, 2007, pp. 39-39.
- [40] S. Zapata, E. Torres, G. Sevilla, L. Aballay, y M. Reus, «Effectiveness of traditional software requirement elicitation techniques applied in distributed software development scenarios», en *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, 2012, pp. 1-7.
- [41] Y. Kaeri, Y. Manabe, C. Moulin, K. Sugawara, y J.-P. A. Barthes, «A Cloud-Based Support System for Offshore Software Development», en *2010 International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, 2010, pp. 315-319.
- [42] F. Salger, G. Engels, y A. Hofmann, «Assessments in global software development: a tailorable framework for industrial projects», en *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2*, New York, NY, USA, 2010, pp. 29–38.
- [43] J. Whitehead, «Collaboration in Software Engineering: A Roadmap», en *Future of Software Engineering*, Los Alamitos, CA, USA, 2007, vol. 0, pp. 214-225.
- [44] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, y A. Wesslén, *Experimentation in Software Engineering: An Introduction*, 1st ed., vol. 6. Springer, 1999.
- [45] M. Gea, F. L. Gutierrez, J. L. Garrido, y J. Cañas, «Teorías y Modelos Conceptuales para un diseño basado en grupos», en *IV Congreso Internacional de Interacción Persona- Ordenador*, Vigo, España, 2003.
- [46] N. Juristo y A. M. Moreno, *Basics of software engineering experimentation*, 1.ª ed. Springer, 2001.
- [47] M. V. Zelkowitz y D. Wallace, «Experimental validation in software engineering», *Inf. Softw. Technol.*, vol. 39, n.º 11, pp. 735-743, 1997.
- [48] B. Kitchenham y S. Charters, «Guidelines for performing Systematic Literature Reviews in Software Engineering», 2007.