



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe de avance y final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	3
<b>Vigencia</b>	13/3/2018

**Unidad Ejecutora:**

Departamento de Ingeniería e Investigaciones Tecnológicas

**Código del proyecto:** C2-ING-045

**Título del proyecto de investigación:**

Adaptación Dinámica de un Proceso de Requisitos

**Programa de acreditación:**

CyTMA2 (Programa de Investigación Científica, Desarrollo y Transf. de Tecnologías e Innovaciones. UNLaM)

**Director del proyecto:**

Hadad, Graciela Dora Susana

**Co-Director del proyecto:** --

**Integrantes del equipo:**

**Docentes-investigadores:**

Ledesma, Viviana Alejandra

Mighetti, Juan Pablo

**Alumnos (de grado):**

Bedetti, Nicolás Alfredo

**Fecha de inicio:** 01/01/2017

**Fecha de finalización:** 31/12/2018

**Informe final**

**Sumario:**

1. Resumen y palabras clave .....	p. 2
2. Memoria descriptiva.....	p. 2
3. Cuerpo de anexos .....	p. 14

## 1. Resumen y palabras clave

**Resumen:** Durante varios proyectos de investigación se desarrolló una estrategia de Ingeniería de Requisitos basada en modelos en lenguaje natural, que ha sido difundida en cursos de grado y posgrado, y puesta en práctica en diversos proyectos de software de mediana a gran envergadura en la industria. Es posible mejorar esta estrategia al planificar su adaptación según las características que presenta cada proyecto de software. Estas características se manifiestan en factores observables que atañen al contexto de aplicación y al proyecto mismo. En base a estos factores se podrían tomar decisiones sobre qué artefactos de requisitos construir, qué actividades realizar y qué técnicas específicas aplicar. Es decir, se propone adaptar el proceso de requisitos a cada proyecto de software en base a factores situacionales imperantes en él, de manera de lograr una mayor productividad del proceso, procurando una mejor calidad de los requisitos. Además, ocurre frecuentemente que las condiciones iniciales de un proyecto e incluso el contexto de aplicación se alteran, pudiéndose necesitar una nueva adaptación del proceso. Es por ello, que se requiere una adaptación dinámica del proceso mediante el monitoreo de los factores que pueden afectarlo en determinados hitos. En base a la literatura y la experiencia adquirida en la aplicación de la estrategia, se han agregado varios factores situacionales al conjunto inicial, estableciéndose sus interacciones e impactos en la estrategia, junto con mecanismos de adaptación a aplicar y en qué momento del proceso reevaluar la situación. Se llevó a cabo un caso de estudio sobre la evolución de estos factores, que permitió refinar la evaluación de los mismos durante el proceso. Asimismo, se realizó un experimento controlado para comparar las situaciones más relevantes en la adaptación del proceso de requisitos. Estos trabajos permitieron apoyar el beneficio de adaptar el proceso a la situación y la necesidad de readaptarlo durante su desarrollo.

**Palabras clave:** Ingeniería de Requisitos, Ingeniería de Métodos Situacional, Modelos en Lenguaje Natural, Variabilidad de Procesos

## 2. Memoria descriptiva

### 2.1. Estado de Actividades del Proyecto

Se han cumplido con todas las actividades previstas en el cronograma, alcanzando por lo tanto el objetivo del proyecto y sus objetivos específicos. A continuación, se enumeran dichas actividades tal cual figuran en el proyecto de investigación, indicando los participantes en cada actividad:

- Estudio bibliográfico sobre Ingeniería de Requisitos. Dualde - Bedetti
- Estudio bibliográfico sobre Ingeniería de Métodos Situacional y Variabilidad de Procesos aplicable a procesos de requisitos. Ledesma, Mighetti y Hadad
- Identificación de factores situacionales, sus interacciones y el nivel de evolución de los mismos a lo largo del proceso. Ledesma, Mighetti y Hadad
- Identificación de puntos de variación en la estrategia y los factores involucrados. Ledesma y Hadad.
- Selección de casos de estudio para análisis de adaptación a características del contexto y del proyecto. Mighetti y Dualde - Bedetti
- Definición de bloques de proceso comunes y bloques de proceso para variantes alternativas de la estrategia. Ledesma, Mighetti y Bedetti
- Definición de reglas de adaptación para seleccionar los bloques de proceso adecuados en cada punto de variación. Ledesma y Hadad
- Definir las actividades, entradas y salidas del proceso de adaptación. Ledesma y Hadad
- Diseño de métricas para establecer la eficiencia y eficacia de la adaptación de la estrategia. Mighetti y Bedetti
- Diseño de las pruebas de adaptación para los casos de estudio seleccionados. Ledesma y Hadad
- Aplicación de las adaptaciones más frecuentes de la estrategia en los casos de estudio seleccionados, calculando las métricas diseñadas. Ledesma, Mighetti y Bedetti
- Supervisión de las pruebas en curso y análisis de resultados parciales. Mighetti y Hadad
- Análisis del impacto de los resultados obtenidos en función de la adaptación de la estrategia. Mighetti, Ledesma, Bedetti y Hadad
- Refinamiento de los mecanismos de adaptación en función del análisis realizado. Ledesma, Mighetti, Bedetti y Hadad

En resumen, se ha identificado un conjunto de factores situacionales con sus interacciones, incluyendo factores no contemplados inicialmente, relativos al desarrollo de software global y a la calidad esperada de los requisitos; se han determinado los puntos de variación de la estrategia de requisitos y el impacto de cada factor en cada punto de variación, identificándose aquellos factores más proclives a evolucionar; se han definido los bloques de proceso que componen la estrategia, identificando aquellos bloques comunes a cualquier alternativa de la

estrategia y aquellos variables; y se han establecido las reglas de adaptación del proceso, aplicables en cada punto de variación según los valores de los factores situacionales y el nivel de confianza asignado; estas reglas permiten seleccionar los bloques de proceso que conformarán el proceso de requisitos a seguir para un proyecto específico. Se ha definido un proceso de adaptación dinámico del propio proceso de requisitos, estableciendo las operaciones a aplicar sobre los bloques de proceso (agregado, omisión, reemplazo y parametrización), y en qué momentos y cómo readaptar el proceso de requisitos durante su avance. Estas definiciones han quedado reflejadas en la publicación de un capítulo en una enciclopedia internacional [1]. Se ha realizado un estudio de caso sobre la evolución de los factores situacionales en base a 35 proyectos de requisitos desarrollados entre 2015 y 2017. Para facilitar el análisis estadístico, el alumno Bedetti construyó una planilla de cálculo preliminar con algunas macros, la que facilitó la carga y posterior evaluación de los cambios sufridos por los factores situacionales desde el inicio hacia el final de un proceso de requisitos. La etapa inicial de este estudio de caso fue publicada en un congreso nacional [4], y los resultados finales del mismo se plasmaron en un artículo publicado en los anales de un congreso iberoamericano [6]. Posteriormente, se llevó a cabo un experimento controlado para comparar variantes del proceso de Ingeniería de Requisitos según aquellos factores situacionales más relevantes, definiéndose previamente un conjunto de métricas a aplicar. Se utilizaron dos dominios de aplicación diferentes aplicando en cada uno tres variantes de la estrategia de requisitos. Los resultados de este experimento están plasmados en la tesis de maestría de la Ing. Ledesma y se está preparando un artículo para su publicación en un congreso iberoamericano.

Algunos factores situacionales identificados en la adaptación de la estrategia de requisitos se utilizaron para establecer una familia de patrones que define las situaciones en que un proceso de ingeniería de requisitos genérico puede llevarse a cabo totalmente, parcialmente o puede no realizarse, lo que fue publicado en los anales de un congreso internacional [2]. Este trabajo fue realizado en conjunto con el investigador Dr. Julio Leite de una universidad brasileña.

Algunas adaptaciones de la estrategia de requisitos basada en lenguaje natural, referida a la construcción de escenarios futuros, se describen en detalle en el material didáctico [7], utilizado en la cátedra de Ingeniería de Requerimientos de la

UNLaM. Otros temas de investigación en Ingeniería de Requisitos fueron incluidos en el material didáctico [8], utilizado en la misma cátedra.

Por otro lado, se ha realizado un trabajo de investigación conjunta con otra universidad brasileña sobre el uso de especificaciones de requisitos en lenguaje natural en la industria del software brasileña y argentina, cuyos resultados se publicaron en [3] y fueron expuestos frente a una audiencia de investigadores y docentes latinoamericanos.

En base a estudios realizados en proyectos de investigación previos sobre la completitud de los modelos de requisitos en UNLaM y los resultados alcanzados por otros autores en el tema, es que surgió la idea de estudiar causas de omisiones en los modelos, vinculadas a aspectos cognitivos de los involucrados durante el proceso de requisitos, de manera de poder formular mecanismos de compatibilización entre la forma de abordaje del procesamiento de conocimiento por parte del ingeniero de requisitos con el estilo organizativo de la fuente de información. Las tareas iniciales encaradas en tal sentido derivaron en una publicación en los anales de un congreso nacional [5].

## **2.2. Formación de Recursos Humanos**

El Ing. Juan Pablo Mighetti finalizó su tesis de maestría “Mitigación de amenazas a requisitos en el desarrollo global de software usando LEL y Escenarios”, desarrollada en la UNLaM y dirigida por la directora del proyecto, quedando en condiciones de presentar la tesis para su posterior defensa. A partir de esta tesis, se identificaron algunos de los factores situacionales no previstos inicialmente en la adaptación del proceso de requisitos.

La Ing. Viviana Alejandra Ledesma finalizó su tesis de maestría “Estrategia de Requisitos adaptable según factores de situación”, desarrollada en la UNLaM y dirigida por la directora del presente proyecto y co-dirigida por el Ing. Jorge Horacio Doorn. La misma está en proceso de revisión final de sus directores. Esta tesis está directamente vinculada al proyecto de investigación en curso.

Por otro lado, la Ing. Ledesma recibió la categoría “V” del programa de docentes investigadores de CONEAU en Abril del 2017. La directora del proyecto recibió la

categoría “II” del programa de docentes investigadores de CONEAU en Noviembre del 2017.

Durante el transcurso del 2018, obtuvieron becas la Ing. Ledesma para la finalización de su tesis de maestría y el alumno Bedetti para realizar tareas de investigación en el presente proyecto. Los detalles de dichas becas son:

- Becario alumno de grado: Nicolás Alfredo Bedetti. Beca de Investigación Científica UNLaM 2018, con resolución rectoral N° 604/17. Beca financiada por la Universidad Nacional de La Matanza. Directora de beca: Dra. Graciela Hadad. Período: 01/01/2018 a 31/12/2018.
- Becaria graduada: Ing. Viviana Alejandra Ledesma. Programa de Becas “Formando UNLaM” financiada por la Secretaría de Políticas Universitarias. Tesis de Maestría: Estrategia de Requisitos Adaptable según Factores de Situación. Carrera: Maestría en Informática, Universidad Nacional de La Matanza. Directora de beca: Dra. Graciela Hadad. Período: 01/05/2018 a 31/12/2018.

Los integrantes del proyecto han realizado los siguientes cursos o asistido a jornadas de capacitación durante el transcurso del proyecto:

- Bedetti - Curso: Desarrollo de Aplicaciones en XAMARIN. Centro de Desarrollo e Investigaciones Tecnológicas, UNLaM. Aprobado. Duración: 120 hs. Período: 2017.
- Ledesma, Mighetti y Hadad - Taller: Sistema Guaraní: Actas Digitales. DIIT, UNLaM. Duración: 3 hs. Fecha: Mayo 2017.
- Hadad - Asistencia a congreso: IEEE ICSE 2017 (International Conference on Software Engineering), Buenos Aires. Período: 20 al 28 Mayo 2017.
- Hadad - Curso: Plagio en los Trabajos Académicos. Dictado por la Mg. Patricia Allendez Sullivan. Programa de Desarrollo de Habilidades Docentes, Universidad de Belgrano. Duración: 6 hs. en 3 encuentros. Período: 18 de Agosto al 1 de Septiembre 2017.
- Hadad - Taller: Modelado de Datos Semánticos. Dictado por la Dra. María Laura Caliusco. SAOA 2017 – Simposio Argentino de Ontologías y sus Aplicaciones, en XLIII CLEI – 46 JAIIO, UTN - Facultad Regional Córdoba. Duración: 3 hs. Fecha: 8 de Septiembre 2017.
- Hadad - Jornadas de Actualización en IoT – IEEE IoT vertical and Topical Summit - Argentina. Facultad de Ingeniería, Universidad de Buenos Aires, Ciudad de Buenos Aires. Organizador: Capítulo Argentino del IEEE Communications Society. Duración: 16 hs. Período: 25 y 26 de Octubre 2017.

- Ledesma - Jornada de Actualización sobre “Gestión de Proyectos de Investigación”. DIIT, UNLaM. Duración: 2 hs. Fecha: 26 de Abril 2018.
- Hadad - Curso: Ciencias de Datos en la Nube. Jornadas Docentes. IBM Argentina. Duración: 18 hs. Período: 23 al 25 de Julio 2018.
- Miguetti – Curso: Design Thinking aplicado al testing. Organizador: Testing Day. Duración: 4 hs. Período: agosto 2018.
- Miguetti – Curso: Testing en un mundo de Continuous Delivery. Organizador: Testing Day. Duración: 4 hs. Período: agosto 2018.
- Miguetti - Curso de Cyber seguridad. IBM Argentina. Duración: 30 hs. Período: septiembre 2018.
- Ledesma y Hadad - Jornada Formando UNLaM, en el marco del Programa de Fortalecimiento de I+D+i de la UNLaM. Organizador: Secretaría de Ciencia y Tecnología, Universidad Nacional de La Matanza, San Justo. Fecha: 30 de Octubre 2018.
- Miguetti – Curso: La neurociencia en los negocios. IBM Argentina. Duración: 16 hs. Período: octubre 2018.
- Hadad - Seminario: Visualización de la Información y minería de datos. Seminario de Actualización y Capacitación Docente. Secretaría Académica, Universidad Nacional del Oeste. Aprobado. Duración: 9 hs. Período: 18 de Octubre al 1 de Noviembre 2018.

### **2.3. Gestión de Recursos Humanos**

La composición inicial del equipo estuvo dada por: la Dra. Graciela Hadad, el Ing. Juan Pablo Mighetti, la Ing. Viviana Ledesma y la alumna de grado Vanesa Dualde.

La alumna Vanesa Dualde renunció el 5 de Junio del 2017 debido a su falta de disponibilidad horaria por iniciar una actividad laboral en el ámbito profesional. Esta baja se compensó con la incorporación a partir del 1º de Agosto del 2017 del alumno de grado Nicolás Bedetti, quien se ocupó de las tareas asignadas a la alumna renunciante.

Finalmente, el equipo quedó conformado por: la Dra. Graciela Hadad, el Ing. Juan Pablo Mighetti, la Ing. Viviana Ledesma y el alumno de grado Nicolás Bedetti.

### **2.4. Actividades de Difusión**

### 2.4.1. Publicaciones con referato

- [1] “Dynamic Situational Adaptation of a Requirements Engineering Process”, Hadad, G.D.S., Doorn, J.H., Ledesma, V.A. En: Encyclopedia of Information Science and Technology, Fourth Edition. Editorial: IGI Global, Information Science Reference, Mehdi Khosrow-Pour (ed). Hershey, PA, EEUU. Julio 2017 (2018), Category Systems and Software Engineering, capítulo 646, pp. 7422 - 7434 (7500 p.). ISBN: 978-1-5225-2255-3; EISBN: 978-1-5225-2256-0. DOI: 10.4018/978-1-5225-2255-3.
- [2] “Requirements Authorship: A Family Process Pattern”, Hadad, G.D.S., Doorn, J.H., Leite, J.C.S.P., 7th International Workshop on Requirements Patterns in conjunction with the 25th IEEE International Requirements Engineering Conference, Lisboa, Portugal, IEEE Digital Library, pp. 248-251, Septiembre 2017. Electronic ISBN: 978-1-5386-3488-2. ISBN: 978-1-5386-3489-9. DOI: 10.1109/REW.2017.32
- [3] “A intenção de uso da Linguagem Natural na especificação de requisitos: um estudo comparativo entre a Argentina e o Brasil”, Angélica T. S. Calazans, Roberto A. Paldês, Aldegol O. Paulino, Fabrício R. Freire, Jana P. Fraga, Graciela D. S. Hadad, Ari Melo Mariano, Anales de 46 JAIIO – Jornadas Argentinas de Informática, track ASSE - Simposio Argentino de Ingeniería de Software. Organizadores: UTN-FRC, SADIO y CLEI, Universidad Tecnológica Nacional, Facultad Regional Córdoba, ISSN: 2451-7593, pp. 122 - 128, Septiembre 2017.
- [4] “Evolución de los Factores Situacionales durante el Proceso de Requisitos”, Ledesma, V.A., Hadad, G.D.S., Doorn, J.H., Mighetti, J.P., Bedetti, N.A., Elizalde, M.C., Actas del WICC 2018 – XX Workshop de Investigadores en Ciencias de la Computación, Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste, Corrientes, pp. 578-582, ISBN: 978-987-3619-27-4, Abril 2018.
- [5] “Revisión con base cognitiva de un Proceso de Requisitos”, Hadad, G.D.S., Doorn, J.H., Elizalde, M.C., Actas del WICC 2018 – XX Workshop de Investigadores en Ciencias de la Computación, Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste, Corrientes, pp. 619-623, ISBN: 978-987-3619-27-4, Abril 2018.
- [6] “Estimación de los Factores Situacionales del Proceso de Requisitos”, Ledesma, V.A., Hadad, G.D.S., Doorn, J.H., Bedetti, N.A., WER 2018 - 21st Workshop on Requirements Engineering, Río de Janeiro, Brasil, Septiembre 2018. ISBN: 978-85-907171-1-9.



### **2.4.2. Publicaciones Docentes relacionadas**

- [7] “Ingeniería de Requisitos del Software Orientada al Cliente. Revisión Marzo 2018”, Hadad, G.D.S., Notas de Clase, Editorial Liga Federal Universitaria, Cátedra Ingeniería de Requerimientos, Departamento de Ingeniería e Investigaciones Tecnológicas, UNLaM, Código 1118/635-5, 2018, 194 páginas.
- [8] “Inspecciones de Modelos de Requisitos basados en Lenguaje Natural. Revisión 2017”, Hadad, G.D.S., Kaplan, G.N., Doorn, J.H., Notas de Clase, cátedra virtual de Ingeniería de Requerimientos, DIIT, UNLaM, versión actualizada Septiembre 2017, 56 páginas.

### **2.4.3. Presentaciones en reuniones científicas**

- Expositora: Hadad. Título del Artículo: A intenção de uso da Linguagem Natural na especificação de requisitos: um estudo comparativo entre a Argentina e o Brasil. Evento: XLIII CLEI - 46 JAIIO – Jornadas Argentinas de Informática, track ASSE - Simposio Argentino de Ingeniería de Software. Lugar: Universidad Tecnológica Nacional - Facultad Regional Córdoba, Ciudad de Córdoba. Fecha del evento: 4 al 8 de Septiembre 2017.
- Presentadora de pósteres y expositora en la mesa de trabajo de Ingeniería de Software: Hadad. Títulos de Artículos: “Revisión con base cognitiva de un Proceso de Requisitos” y “Evolución de los Factores Situacionales durante el Proceso de Requisitos”. Evento: WICC 2018 – XX Workshop de Investigadores en Ciencias de la Computación. Lugar: Universidad Nacional del Nordeste, Concordia. Fecha: Abril 2018.
- Expositora: Hadad. Título de Artículo: Estimación de los Factores Situacionales del Proceso de Requisitos. Evento: WER 2018 – 21st Workshop on Requirements Engineering. Lugar: Pontificia Universidade Católica de Río de Janeiro, Río de Janeiro, Brasil. Fecha del evento: 5 y 6 de Septiembre 2018.

## **2.5. Actividades de Transferencia**

### **2.5.1. Cursos de Grado y Posgrado**

Los resultados de investigaciones anteriores y de la actual se están utilizando en los siguientes cursos de grado y postgrado:

- Ingeniería de Requerimientos, UNLaM, Departamento de Ingeniería e Investigaciones Tecnológicas, carrera de grado: Ingeniería en Informática.

- Análisis de Sistemas, UNLaM, Departamento de Ingeniería e Investigaciones Tecnológicas, carrera de grado: Ingeniería en Informática.
- Ingeniería de Requerimientos, UCA, Facultad de Ingeniería y Ciencias Agrarias, carrera de posgrado: Especialización en Ingeniería de Software.
- Introducción a la Ingeniería de Requisitos, UADER, Facultad de Ciencia y Técnica, curso de posgrado.
- Ingeniería de Software V - Ingeniería de Requerimientos, UB, Facultad de Ingeniería y Tecnología Informática, carrera de grado: Licenciatura en Sistemas de Información.

### **2.5.2. Capacitaciones**

- Hadad: dictado de curso de Extensión y Transferencia: “Ingeniería de Requisitos Aplicada”. Organizador: Facultad de Ingeniería y Tecnología Informática, Universidad de Belgrano. Destinatarios: docentes UB y empresa CyS Informática S.A. Duración: 16 hs. Fecha: 10 y 17 de Diciembre de 2018.

### **2.5.3. Charlas de Divulgación**

- Hadad: disertación “¿Cómo hacer para que el cliente reciba el software que realmente necesita?” Evento: Semana de la Ciencia y de la Tecnología. Organizador: Universidad de Belgrano. Destinatarios: público en general. Fecha: 6 de Septiembre 2018.
- Ledesma: disertación “Desafíos en el desarrollo de la tesis Estrategia de Requisitos Adaptable según Factores de Situación”. Evento: Jornada Formando UNLaM, en el marco del Programa de Fortalecimiento de I+D+i de la UNLaM. Organizador: Secretaría de Ciencia y Tecnología, Universidad Nacional de La Matanza. Fecha: 30 de Octubre 2018.
- Hadad: disertación “Gestión de la calidad de un modelo léxico en el proceso de requisitos”. Evento: IV Foro de Investigadores. Organizador: Universidad de Belgrano. Destinatarios: investigadores de la UB. Fecha: 6 de Noviembre 2018.

## **2.6. Actividades de Vinculación**

A lo largo de este período se han mantenido relaciones con otras Universidades a través de la participación conjunta en publicaciones y otras actividades con:

- Instituto CEUB de Pesquisa e Desenvolvimento, Centro Universitário de Brasília (Brasília, Brasil), con quienes se publicó el artículo [3].

- Departamento de Informática, Pontificia Universidade Católica do Rio de Janeiro (Río de Janeiro, Brasil), con quienes se publicó el artículo [2] y se está escribiendo un libro sobre Ingeniería de Requisitos. En este libro también participa el Ing. Jorge H. Doorn, investigador de la UNLaM.

## **2.7. Otras Actividades Científico-Tecnológicas**

### **2.7.1. Direcciones y Tutorías de alumnos de grado y posgrado**

- Hadad: Directora de Tesis de Maestría del Ing. Juan Pablo Mighetti. Título: Mitigación de amenazas a requisitos en el desarrollo global de software usando LEL y Escenarios. Maestría en Informática, Escuela de Posgrado, Universidad Nacional de La Matanza. Inicio: Junio 2014. Para defensa de tesis.
- Hadad: Directora de Tesis de Maestría de la Ing. Viviana Ledesma. Título: Estrategia de Requisitos adaptable según factores de situación. Maestría en Informática, Escuela de Posgrado, Universidad Nacional de La Matanza. Inicio: Julio 2015. En etapa de cierre.
- Hadad: Directora de Tesis de Maestría de la Lic. Ana María Piccin. Título: Enseñanza de la programación: las concepciones de los profesores de los cursos iniciales de las carreras de grado. Maestría en Gestión de Proyectos Educativos, Universidad CAECE. Inicio: Mayo 2015. En desarrollo.
- Hadad: Co-Directora de Tesis de Doctorado de la Ing. Andrea F. Vera. Título: Modelado del registro de trazas en la Ingeniería de Requisitos. Doctorado en Ciencias Informáticas, Facultad de Informática, Universidad Nacional de La Plata. Director: Ing. Jorge H. Doorn. Inicio: Mayo 2009. En desarrollo.
- Hadad: Tutora de Trabajo Final de Especialización del Lic. Guillermo Portero. Título: Establecer las particularidades de los requisitos a especificar para desarrollar software en entornos móviles. Carrera de Especialización en Ingeniería de Software, Facultad de Ciencias Fisicomatemáticas e Ingeniería-Posgrados, Pontificia Universidad Católica Argentina. Inicio: Enero 2015. Fecha de aprobación: 10-10-2017.
- Hadad: Tutora de Trabajo Final de Especialización de la Ing. Giselle Verónica Juárez. Título: Patrones de Captura de Requisitos de Usabilidad en Aplicaciones Web. Carrera de Especialización en Ingeniería de Software, Facultad de Ingeniería y Ciencias Agrarias - Posgrado, Pontificia Universidad Católica Argentina. Inicio: Abril 2016. Enviado a evaluación en Noviembre 2018.

- Hadad: Tutora de alumno de grado para 300 horas de Trabajo Social Profesional, Facultad de Tecnología Informática, Universidad de Belgrano. Proyecto: Gestor de Inspecciones. Alumno: Tomás Damonte. Carrera: Tecnicatura en Programación de Computadoras. Período: 19/09/2017 al 28/01/2018 (6 meses).

### **2.7.2. Actividades de Evaluación**

- Hadad: Miembro del Comité de Programa con evaluación de artículos. Evento: CIBSE 2017 – XX Congreso IberoAmericano en Software Engineering, Track WER (XX Workshop on Requirements Engineering). Fecha: Mayo 2017.
- Hadad: Miembro del Panel de Expertos. Evento: Simposio Doctoral en CIBSE 2017. Lugar: Universidad Católica Argentina, Ciudad de Buenos Aires. Fecha: 23/05/2017.
- Hadad: Miembro de Comité de Programa con evaluación de artículos. Evento: 47 JAIO - Jornadas Argentinas de Informática, Simposio: CAI - 10º Congreso de AgroInformática. Fecha de evaluación: Mayo 2018. Fecha del evento: 3 al 7 de Septiembre 2018.
- Hadad: Miembro de Comité de Programa con evaluación de artículos. Evento: CIBSE 2018 – XXI Congreso IberoAmericano en Software Engineering, Track Requirements Engineering. Fecha de evaluación: Enero 2018. Fecha del evento: Abril 2018.
- Hadad: Revisora de Artículos de Revista: Journal of Software Engineering Research and Development (JSERD). ISSN: 2195-1721. Editorial: Springer Open, publicado en Mayo 2017. Fecha de evaluación: Enero a Marzo 2017.
- Hadad: Revisora de Artículos de Revista: CLEI Electronic Journal. ISSN: 0717-5000. Editorial: Centro Latinoamericano de Estudios en Informática. Fecha de evaluación: Octubre 2018.
- Hadad: Co-presidente de Comité de Programa y editora de anales. Evento: WER 2018 – 21st Workshop on Requirements Engineering. Lugar: Pontificia Universidade Católica do Rio de Janeiro, Río de Janeiro, Brasil. Fecha: 5 y 6 Septiembre 2018.
- Hadad: Jurado de tesina de grado del alumno Hugo Daniel Giardini. Título: Integración de Aplicaciones Empresariales: Arquitectura de Referencia. Carrera: Licenciatura en Informática. Escuela de Informática, Universidad Nacional del Oeste. Aprobada. Fecha: 02/03/2017.
- Hadad: Jurado en tres coloquios para cargos de profesores en la Facultad de Ingeniería y Tecnología Informática, Universidad de Belgrano. Período: Noviembre 2017.

- Hadad: Jurado en seis coloquios para cargos de profesores en la Facultad de Ingeniería y Tecnología Informática, Universidad de Belgrano. Período: Julio a Diciembre del 2018.

## 2.8. Bibliografía

- Abdullah, A., & Khan, H. (2015). FreGsd: A Framework for Global Software Requirement Engineering. *Journal of Software*, 10, 10, pp. 1189-1198.
- Aguilar, J., Garrigós, I., & Mazón, J. (2016). Requirements Engineering in the Development Process of Web Systems: A Systematic Literature Review. *Journal of Applied Sciences. Acta Polytechnica Hungarica*, 13, 3, pp. 61-80.
- Arias, M., Buccella, A., & Cechich, A. (2017). A Framework for Managing Requirements of Software Product Lines. *Proceedings of XLIII CLEI, Conferencia Latinoamericana de Informática. Córdoba, Argentina.*
- Bakhat, K., Sarwar, A., Motla, Y., & Akhta, M. (2015). A Situational Requirement Engineering Model for an Agile Process. *Bahria University Journal of Information & Communication Technology*, 8, 1, pp. 21-26.
- Delgado, A., & Calegari, D. (2017). BPMN 2.0 Based Modeling and Customization of Variants in Business Process Families. *Proceedings of XLIII CLEI, Conferencia Latinoamericana de Informática. Córdoba, Argentina.*
- Galster, M., Weyns, D., Tofan, D., Michalik, B., & Avgeriou, P. (2014). Variability in Software Systems - A Systematic Literature Review. *IEEE Transactions on Software Engineering*, IEEE, 40, 3, pp. 282-306.
- Hadad, G.D.S., & Doorn, J.H. (2013). Introducing Variability in a Client-Oriented Requirements Engineering Process. *Proceedings of ER@BR2013 - Requirements Engineering @ Brazil, ISSN: 1613-0073*, pp. 8-13. Río de Janeiro, Brasil.
- Haugen, Ø., & Øgård, O. (2014). BVR – Better Variability Results. *Proceedings of International Conference on System Analysis and Modeling: Models and Reusability (SAM 2014), Lecture Notes in Computer Science, Vol. 8769*, pp. 1-15. Springer.
- Henderson-Sellers, B., Ralyté, J., Ågerfalk, P., & Rossi, M. (2014). *Situational Method Engineering*. Springer-Verlag Berlin Heidelberg.
- Jafarinezhad, O., & Ramsin, R. (2012). Development of Situational Requirements Engineering Processes: A Process Factory Approach. *Proceedings of 36th IEEE International Conference on Computer Software and Applications*, pp. 279-288.
- Jiang, L., & Eberlein, A. (2008). A Framework for Requirements Engineering Process Development (FRERE). *Proceedings of 19th Australian Conference on Software Engineering (ASWEC 2008)*, pp. 507-516. Perth, Australia.

- Kabaale, E., & Mayoka Kituyi, G. (2015). A Theoretical Framework for Requirements Engineering and Process Improvement in Small and Medium Software Companies. *Business Process Management Journal*, 21, 1, pp.80-99.
- Khan, H., bin Mahrin, M., & bt Chuprat, S. (2014). Factors for Tailoring Requirement Engineering Process: A Review. *International Journal of Software Engineering and Technology*, 1, 1, pp. 7-18.
- Martínez-Ruiz, T., García, F., Piattini, M., & Munch, J. (2011). Modelling Software Process Variability: An Empirical Study. *IET Software*, 5, 2, pp. 172-187.
- Mighetti, J.P., & Hadad, G.D.S. (2016). A Requirements Engineering Process Adapted to Global Software Development. *CLEI Electronic Journal*, 19, 3, paper 7.
- Pacini, K., & Braga, R. (2015). An Approach for Reusing Software Process Elements based on Reusable Asset Specification: A Software Product Line Case Study: *Proceedings of the 10th International Conference on Software Engineering Advances (ICSEA 2015)*, pp. 200-206. Barcelona: España.
- Ralyté, J. (2013). Situational Method Engineering in Practice: A Case Study in a Small Enterprise. *Proceedings of 25th International Conference on Advanced Information Systems Engineering*, pp. 17-24. Valencia, España.
- Rodríguez-Lora, V., Henao-Cálad, M., & Valencia Arias, A. (2016). Taxonomías de Técnicas y Herramientas para la Ingeniería del Conocimiento: Guía para el Desarrollo de Proyectos de Conocimiento. *Ingeniare. Revista Chilena de Ingeniería*, 24,2, pp. 351-360.
- Santos, E., Castro, J., Sánchez, J., & Pastor, O. (2010). A Goal-Oriented Approach for Variability in BPMN. *Proceedings of 13th Workshop on Requirements Engineering*, pp. 17-28. Cuenca, Ecuador.

### 3. Cuerpo de anexos

**Anexo I:** Copia de certificados de participación de integrantes en eventos científicos, correspondientes a ítems 2.2 (cursos realizados), 2.4.3 (presentaciones científicas) y 2.5.3 (actividades de divulgación).

**Anexo II:** Copia de artículos presentados en publicaciones periódicas, y ponencias presentadas en eventos científicos, correspondientes a referencias [1] [2] [3] [4] [5] [6]. Se adjunta un Resumen Extendido del presente Informe Final.

**Anexo III:** Conteniendo el formulario FPI-015: Rendición de gastos del proyecto de investigación acompañado de las hojas foliadas con los comprobantes de gastos.

UNLaM – DIIT – Proyecto: C2-ING-045

Informe final 01-01-2018 al 31-12-2018

## ANEXO I – COPIA DE CERTIFICADOS



# XLIII CLEI

Conferencia Latinoamericana de Informática

# 46 JAIIO

Jornadas Argentinas de Informática



Sociedad Argentina de Informática extiende el presente diploma a

**Graciela Dora Susana Hadad**

por haber participado como **Asistente** en

**ASSE 2017 - Simposio Argentino de Ingeniería de Software**

durante la XLIII Conferencia Latinoamericana de Informática y las 46° Jornadas Argentinas de Informática, realizadas del 4 al 8 de Septiembre de 2017, en la ciudad de Córdoba, Argentina.

Héctor Monteverde  
Coordinador General

Rodrigo Santos  
Coordinador General

Roberto Muñoz  
Coordinador General

Rosita Wachenchäuzer  
Coordinadora General  
Presidente de SADIO





# XLIII CLEI

Conferencia Latinoamericana de Informática

# 46 JAIIO

Jornadas Argentinas de Informática



Sociedad Argentina de Informática extiende el presente diploma a

**Graciela Dora Susana Hadad**

por haber participado como **Expositor** en

**ASSE 2017 - Simposio Argentino de Ingeniería de Software**

durante la XLIII Conferencia Latinoamericana de Informática y las 46° Jornadas Argentinas de Informática, realizadas del 4 al 8 de Septiembre de 2017, en la ciudad de Córdoba, Argentina.

Héctor Monteverde  
Coordinador General

Rodrigo Santos  
Coordinador General

Roberto Muñoz  
Coordinador General

Rosita Wachenchäuser  
Coordinadora General  
Presidente de SADIO

IEEE ICSE 2017  
Departamento de Computación, FCEN  
Universidad de Buenos Aires  
Pabellón 1, Ciudad Universitaria  
(C1248EGA) Buenos Aires, Argentina  
<http://icse2017.gatech.edu>



May 20, 2017

To Whom It May Concern:

This letter certifies that Graciela Hadad attended IEEE ICSE taking place May 20-28, 2017 in Buenos Aires, Argentina. Her registration of \$400.00 has been paid in full to SADIO – Sociedad Argentina de Informatica. Graciela has participated in the following:

CIBSE – Track WER

Article: “Inspeccion Centrada en Omisiones y Ambigüedades de un Modelo Lexico”

Doctoral Symposium – Panel

Questions regarding his registration can be directed to me at the email address below.

Sincerely,

Sebastian Uchitel General  
Chair, ICSE 2017  
[suchitel@dc.uba.ar](mailto:suchitel@dc.uba.ar)



Se certifica que

*Aziana Alejandra Bedesma*

**DNI: 22285634**

ha participado en calidad de *Expositor/a* en la *Jornada Formando UNLAM*, en el marco del Programa de Fortalecimiento de I+D+i de la Universidad Nacional de La Matanza, el día martes 30 de octubre de 2018 en el Auditorio Grande de la UNLAM.

Ana Marcela Bidina

Secretaría de Ciencia y Tecnología  
Universidad Nacional de La Matanza



Se certifica que

*Graciela Hadaad*

**DNI: 13711412**

ha participado en calidad de asistente en la *Jornada Formando UNLaM*, en el marco del Programa de Fortalecimiento de I+d+i de la Universidad Nacional de La Matanza, el día martes 30 de octubre de 2018 en el Auditorio Grande de la UNLaM.

A handwritten signature in black ink, appearing to read 'Ana Marcela Bidiña', is located in the bottom right area of the certificate.

Ana Marcela Bidiña

Secretaria de Ciencia y Tecnología  
Universidad Nacional de La Matanza



## 21st Workshop on Requirements Engineering

Rio de Janeiro, Brazil 05-06 September, 2018

### CERTIFICATE

This is to certify that **GRACIELA DORA SUSANA HADAD** has participated in the 21st Workshop on Requirements Engineering, held at PUC-Rio, Rio de Janeiro, Brasil on 5-6 de September 2018.

A handwritten signature in blue ink, appearing to read 'Julio Cesar Sampaio do Prado Leite'.

**Julio Cesar Sampaio do Prado Leite**  
General Chair

A handwritten signature in blue ink, appearing to read 'Jaelson Castro'.

**Jaelson Castro**  
General Chair



Sociedad Argentina de Informática extiende el presente diploma a

**GRACIELA DORA SUSANA HADAD**

por haber participado como **Revisor** de

**CAI 2018 - Congreso Argentino de AgroInformática**

para las 47° Jornadas Argentinas de Informática, realizadas del 3 al 7 de Septiembre de 2018,  
en la Ciudad Autónoma de Buenos Aires, Argentina.



Alejandra Garrido  
Coordinadora General de las 47 JAIIO



Claudio Zamosczyk  
Coordinador General de las 47 JAIIO



Sandra D'Agostino  
Presidente de SADIO



XX Workshop de Investigadores  
en Ciencias de la Computación



Se certifica que:

**Ledesma Viviana, Hadad Graciela, Doorn Jorge, Mighetti Juan, Bedetti Nicolás, Elizalde María Celia**

han participado en calidad de autores del artículo:

**Evolución de los Factores Situacionales durante el Proceso de Requisitos**

aceptado en el XX Workshop de Investigadores en Ciencias de la Computación, realizado en la ciudad de Corrientes, los días 26 y 27 de abril de 2018.

LIC. PATRICIA PESADO  
COORDINADORA RED UNCI



MGTER. GLADYS DAPOZO  
COMITÉ ORGANIZADOR UNNE





XX Workshop de Investigadores  
en Ciencias de la Computación



Se certifica que:

**Hadad Graciela, Doorn Jorge, Elizalde María Celia**

han participado en calidad de autores del artículo:

**Revisión con base cognitiva de un Proceso de Requisitos**

aceptado en el XX Workshop de Investigadores en Ciencias de la Computación, realizado en la ciudad de Corrientes, los días 26 y 27 de abril de 2018.

A handwritten signature in black ink, appearing to read 'Patricia Pesado'.

LIC. PATRICIA PESADO  
COORDINADORA RED UNCI



A handwritten signature in black ink, appearing to read 'Gladys Dapozo'.

MGTER. GLADYS DAPOZO  
COMITÉ ORGANIZADOR UNNE







XX Workshop de Investigadores  
en Ciencias de la Computación



Se certifica que:

**Hadad Graciela**  
**DNI 13.711.412**

Ha asistido en calidad de AUTOR al XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018), realizado en la ciudad de Corrientes, los días 26 y 27 de abril de 2018.

LIC. PATRICIA PESADO  
COORDINADORA RED UNCI



MGTER. GLADYS DAPOZO  
COMITÉ ORGANIZADOR UNNE





# Centro de Desarrollo e Investigaciones Tecnológicas (CeDIT)

---

San Justo, 3 de julio de 2017

Se deja constancia que

Nicolas Alfredo Bedetti

DNI: 37.844.342

aprobó el curso “*Desarrollo de aplicaciones en XAMARIN*”, dictado en el marco del CeDIT, con una duración total de 120 hs.



Ing. Santiago Igarza  
Coordinador Ing. Informática



Mg. Gabriel Blanco  
Vicedecano

PROGRAMA DE DESARROLLO DE HABILIDADES DOCENTES

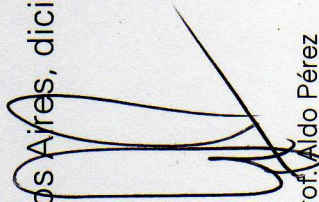
SEGUNDO CUATRIMESTRE 2017

Por cuanto *Graciela Hada*

ha aprobado el Curso *Plagio en los trabajos académicos*, dictado por la Mg. Patricia Allendez Sullivan , con una duración de 6 horas.

Por lo tanto: de acuerdo a las reglamentaciones vigentes en esta casa de estudios, se extiende el presente certificado de aprobación que así lo acredita.

Buenos Aires, diciembre de 2017



Prof. Aldo Pérez

VICEPRESIDENTE DE GESTIÓN INSTITUCIONAL

# JORNADAS DOCENTES

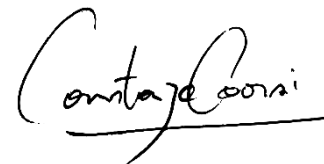
CIENCIAS DE DATOS  
EN LA NUBE

*Graciela Hadad*

Ha participado de las capacitaciones docentes organizadas por IBM Argentina durante los días 23, 24 y 25 de julio.



Stella Loiacono  
Directora de Tecnología  
IBM Argentina



Constanza Caorsi  
Responsable de Relaciones Académicas  
IBM Argentina



UNLaM – DIIT – Proyecto: C2-ING-045

Informe final 01-01-2018 al 31-12-2018

## ANEXO II – COPIA DE ARTÍCULOS

# Encyclopedia of Information Science and Technology, Fourth Edition

Mehdi Khosrow-Pour

*Information Resources Management Association, USA*

Published in the United States of America by

IGI Global  
Information Science Reference (an imprint of IGI Global)  
701 E. Chocolate Avenue  
Hershey PA, USA 17033  
Tel: 717-533-8845  
Fax: 717-533-8661  
E-mail: [cust@igi-global.com](mailto:cust@igi-global.com)  
Web site: <http://www.igi-global.com>

Copyright © 2018 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Names: Khosrow-Pour, Mehdi, 1951- editor.

Title: Encyclopedia of information science and technology / Mehdi Khosrow-Pour, editor.

Description: Fourth edition. | Hershey, PA : Information Science Reference, [2018] | Includes bibliographical references and index.

Identifiers: LCCN 2017000834 | ISBN 9781522522553 (set : hardcover) | ISBN 9781522522560 (ebook)

Subjects: LCSH: Information science--Encyclopedias. | Information technology--Encyclopedias.

Classification: LCC Z1006 .E566 2018 | DDC 020.3--dc23 LC record available at <https://lccn.loc.gov/2017000834>

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

For electronic access to this publication, please contact: [eresources@igi-global.com](mailto:eresources@igi-global.com).

# Dynamic Situational Adaptation of a Requirements Engineering Process

**Graciela Dora Susana Hadad**

*Universidad Nacional del Oeste, Argentina & Universidad de Belgrano, Argentina*

**Jorge Horacio Doorn**

*Universidad Nacional del Oeste, Argentina & Universidad Nacional de La Matanza, Argentina*

**Viviana Alejandra Ledesma**

*Universidad Nacional del Oeste, Argentina & Universidad Nacional de La Matanza, Argentina*

## INTRODUCTION

The adaptation of any process to particular situations is considered a good practice in most fields. Literature shows that this practice is quite common in Software Engineering processes, such as the methodologies Rational Method Composer (Haumer, 2005) and OPEN Process Framework (Firesmith & Henderson-Sellers, 2002). However, Requirements Engineering (RE) approaches are seldom tailored to context or project situations (Potts, 1995; Leite, Hadad, Doorn, & Kaplan, 2000; Leffingwell & Widrig, 2003; Seyff et al., 2009). Nevertheless, sometimes the elicitation activity, as part of an RE process, is performed taking into account some environmental characteristics, such as number of information sources, user geographical distribution, user time availability, user experience, among others (Maiden & Rugg, 1996; Hickey & Davis, 2003; Coulin, 2007; Carrizo, Dieste, & Juristo, 2008). Recently, some proposals have appeared to design an RE process for a specific project by selecting existent RE techniques (Lauesen, 2002; Lobo & Arthur, 2005; Alexander & Beus-Dukic, 2009).

There are activities of the requirements process that are invariant regardless of situational factors, while others should be modified, removed or replaced. Not only activities may be adapted, models produced or used in the process may be also suited for the situation (Galster, Weyns, To-

fan, Michalik & Avgeriou, 2014). This means that the RE process may be assembled like a flexible puzzle using interchangeable pieces depending on the situational factors identified.

Situational Method Engineering (SME) is advocated to build methods tailored to specific situations for the development of systems (Kumar & Welke, 1992). Following its principles, the adaptation of any software development process is based on indicators describing the situation (Khan, bin Mahrin & Chuprat, 2014). Part of the task is to compose such indicators based on observable factors, like degree of business processes reengineering, context complexity, developer expertise in the application domain, and project size, among others. Ideally, these situational factors should be taken into account before beginning the software process. However, there are factors not accurately known when initiating a software project, while other factors may change during the course of the project. Hence, a dynamic view of the adaptation of a software development process achieves a better effectiveness of the process itself. Considering that the production of requirements is the starting point of a software development, it should be necessary to pay more attention to factors impacting on the RE process.

A frequent question of practitioners is related with the obligation of performing all the process steps to reach requirements. *Is it possible to shorten the road or to follow a different one?* Under some



circumstances, there is an opportunity to reduce the RE process by deleting or simplifying activities; and sometimes different paths may be followed by choosing other techniques or even extending some activities. Project managers should make decisions depending mainly on the particular case.

Since problem domain knowledge is mostly expressed in natural language (NL), the use of an RE approach based on NL representations improves the commitment of customers and users to the project, increasing the probability of project success (Macaulay, 1993).

Therefore, in this chapter it is presented the tailoring of an RE process based on NL models, according to a particular set of situational factors. Additionally, a process for constructing this RE process, including the evaluation of such factors along software development life cycle, is proposed as an enhanced solution.

## BACKGROUND

When working on the creation of an engineering product or system, it is important to have a process. This means having a predictable set of activities, techniques, inputs and outputs helps get a high quality outcome. Hence, the way the work is performed does not depend on individual criteria, allowing repeatability of costs, times and quality, and promoting the accumulation of knowledge about the process. As a consequence, the first activity of a process to develop a product consists in defining precisely the expected outcome. When the product is a software system, this initial activity is an RE process, whose outcome is a Software Requirements Specification (SRS). The RE process is particularly different from other activities of the software development process since it is the one that most interacts with people and their environment, while other activities are mainly carried out within the development team context (Carrizo, 2009). Besides, project decisions impose constraints, tools and methods to carry out those activities. Therefore, if the requirements

process takes into account the particularities surrounding the application context and the project itself, then it will probably result both in a better SRS and in a more effective process.

Additionally, a requirements process needs appropriate and continuous communication to gain customers and users compromise. Good communication is achieved when all stakeholders use the same language. In RE, a proven way to accomplish this is by using the vocabulary of the application context (Leite, Doorn, Kaplan, Hadad, & Ridao, 2004). Communication occurs when stakeholders orally interact, and also when reports, documents and models are exhibited to customers. In this sense, NL models, such as glossaries, use cases and scenarios, stimulate stakeholders communication (Leite et al., 2004), and are the most frequently used in RE (Kaindl, 2000; Leffingwell & Widrig, 2003; Seyff et al., 2009; Antonelli, Rossi, Leite, & Oliveros, 2012).

Software development processes put into practice in real projects are often forced to be adjusted due to contingent circumstances, sometimes in a poorly controlled fashion while the project is ongoing. Hence, the evolving situation should be observed as the process goes forward in order to achieve a better tailoring. Possible adjustments to the process can be known in advance on the basis of certain characteristics, though they may change dynamically, i.e., settings are pre-planned but only implemented when an aspect of the situation changes (Rolland, 2008). In this regard, the process is defined as a set of blocks, having process blocks common to all situations and variant process blocks according to situational factors. Thus, the process is made up by assembling blocks for the particular situation (Henderson-Sellers & Ralyté, 2010). Method Engineering was specifically created to tackle this case, promoting the design, construction and adaptation of methods, techniques and tools in order to develop information systems (Brinkkemper, 1996). This discipline considers not only the creation of process blocks but also product blocks, and even blocks that assemble both process and product (Rolland, 2008; Henderson-Sellers &

Ralyté, 2010; Ralyté, 2013). Rolland (2008) not only has pointed out that the software development should begin with a definition phase of the method to be used, but also she proposes that the process must be re-adjusted throughout the life cycle process, an aspect not treated by Method Engineering. SME, as a sub-area of Method Engineering, focuses on building methods for software development, tailored to specific situations (Kumar & Welke, 1992). Thus, SME studies the factors affecting the software project and the application context (Bucher, Klesse, Kurpjuweit, & Winter, 2007). Its principles have been applied in RE to define requirements processes that are adaptable to particular circumstances by using existent modular components (Firesmith, 2004; Jafarinezhad & Ramsin, 2012; Bakhat, Sarwar, Motla, & Akhtar, 2015).

Some of the adaptation factors are related to development time constraints. Naturally, they lead to faster ways of developing software; these are steps toward agile methods (Pinheiro, 2002). A quality factor may depend on a time factor, or the other way around. However, a balance equation for both factors may be frequently achieved (Kohler & Paech, 2002; Pinheiro, 2002). This trade-off is more likely to be better faced by an RE process with the necessary tailoring (Kohler & Paech, 2002). Extremely, when time factors are a major concern, agile methods could be the best way. Though, agile practices are frequently assumed as opposite to RE practices (Cockburn, 2002; Beck, 2004), Kovitz (2002) remarks that both, a phased development and an agile one, do RE but in different styles, since requirements are always present. This vision is also shared by (Sillitti & Succi, 2005; Kohler & Paech, 2002). In this sense, an agile approach dealing with requirements may be also seen as a sort of dynamical adaptation of an RE process. As Cockburn (2002) said, making the decision on the right level of agility is the best way to succeed on a project. RE practices do not deny agility, considering that this property must be combined with quality and attending volatility without constant software fix (Pinheiro, 2002).

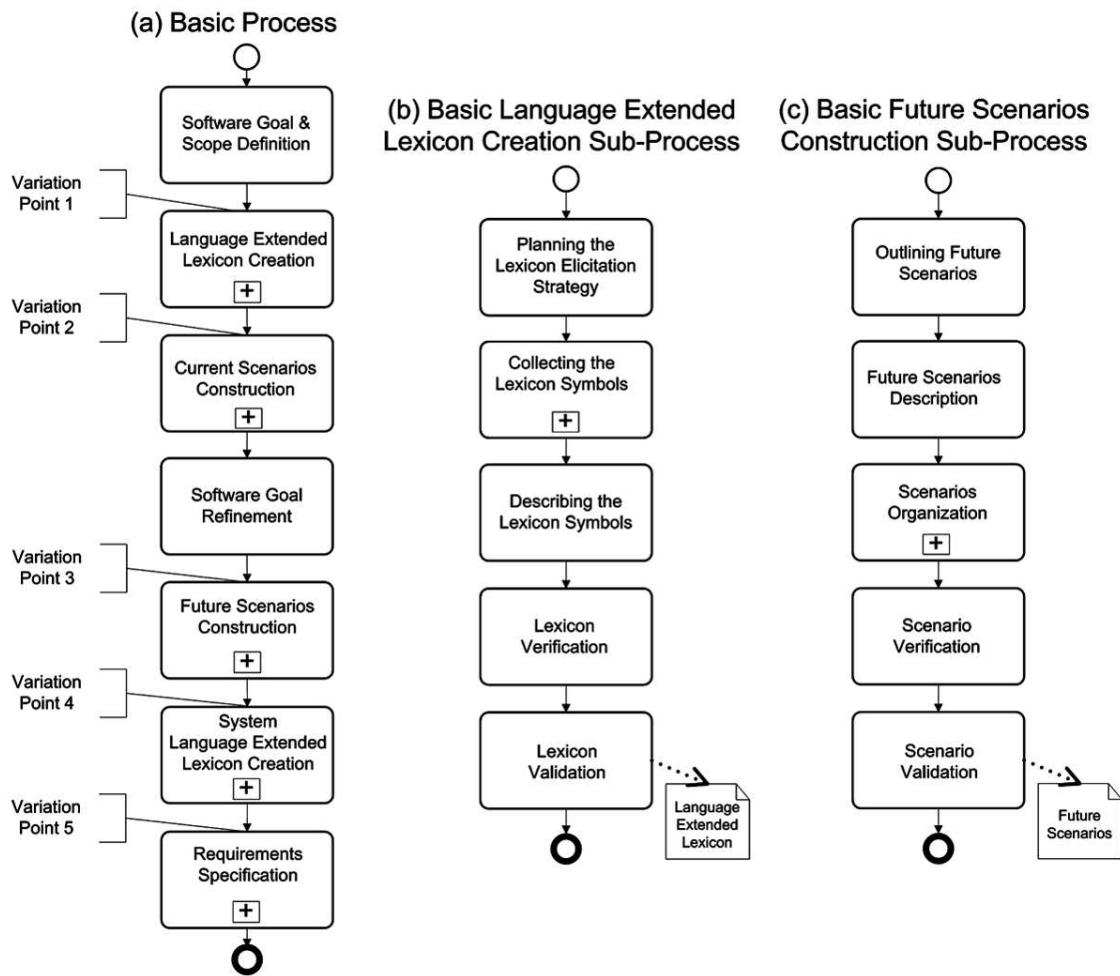
## **A REQUIREMENTS ENGINEERING PROCESS ADAPTABLE TO SITUATIONAL FACTORS**

RE process based on NL models strongly contributes to stakeholders' commitment to the project. It involves the following stages:

- *Defining the general software goal and scope.*
- *Understanding the vocabulary used in the application context, supported by the Language Extended Lexicon model (Hadad, Doorn, & Kaplan, 2009).*
- *Understanding the application context, supported by a set of current scenarios that represent the situations observed in the application context (Leite et al., 2000).*
- *Refining the general goal, by decomposing it into sub-goals.*
- *Defining the software context, by producing a set of future scenarios that represent situations envisioned in a future application context where the software system will operate (Leite et al., 2004).*
- *Defining the vocabulary used in software descriptions, by producing a Language Extended Lexicon of the System created from the previous one, adding terms used in the descriptions of future scenarios.*
- *Making explicit the requirements, by producing an SRS, after extracting requirements from future scenarios.*

This RE process has been initially developed and used in a monolithic fashion, as depicted in Figure 1(a), being independent of any special circumstance. Most main sub-processes are composed of another level of processes, some of which are atomic processes while others are decomposed, as it is shown in Figures 1(b) and 1(c). Recycles in the three diagrams were removed to emphasize the variation points. These recycles imply correcting or updating models as a consequence of a better understanding of

Figure 1. Basic Requirements Engineering Process with two main sub-processes



the problem and of the evolution of the application context.

Though Figure 1(a) shows a monolithic process as part of a cascade model, the strategy can be adapted to be used within different software process models, being this a project factor by itself (see Table 1). For instance, when following an iterative process model, the RE process is performed sequentially for the first four main sub-processes and the last three sub-processes are usually done in iterations along with a requirements change management activity throughout the software development life cycle. It should be noticed that in some cases the fourth sub-process may be included in some of the iterations. Iterations obey to a prioritization activity

within *Software Goal Refinement* sub-process. In an incremental process model, requirements are initially produced, and iterations are based on those defined requirements, obviously allowing requirements evolution. Under this process model, the entire RE process is performed as the starting point, though requirements change management occurs along development life cycle. Regarding agile practices, additionally to time constraints, another leading factor is volatility (Cockburn, 2002), which may be well coped with a change management activity. Though, it may not be the best manner when volatility is a core characteristic of the context. Therefore, an agile method might be a better proposal to satisfy customers in the meanwhile.

This basic RE process may be adapted at the variation points shown in Figure 1(a), depending on a combination of situational factors. Project managers should do it to improve the RE process and, thereby, the requirements product. Situational factors impacting on RE (see Table 1) have been selected based on experience in RE practice and literature proposals (Maiden & Rugg, 1996; Hickey & Davis, 2003; Carrizo et al., 2008; Jafarinezhad & Ramsin, 2012). These factors can be characterized by the following attributes:

- **Value:** Admissible values depend on each factor.
- **Confidence on the Value:** High, Medium, Low.

- **Origin Type:** Application Context or Project.
- **Evolution Type:** Invariant or Contingent.
- **Variation Points:** Where it impacts on.

In order to adapt the RE process, project managers assign values to factors, qualifying them with their degree of confidence on such values. It usually happens that at the very beginning some factors may be not precisely known, since right values are known after gaining a better understanding of the situation. Factors are classified according to their origin into: those related to the specific application context, and those related to the specific software project. From an evolution dimension, factors are classified into: *Invariant*, being those

*Table 1. Situational Factors impacting on RE process. Range Very High to Very Low includes five possible values.*

Context Factors	Acceptable Values	Evolution	Variation Points
Context Complexity	Very High to Very Low	Invariant	2,3
Target Customer	Tailor-made, Market-driven	Invariant	1,2,3,4
New Business	Yes, No	Invariant	1,2,3,4
Business Process Reengineering	Very High to Very Low	Contingent	3,4
Context Volatility	Very High to Very Low	Invariant	2,3
Volatility of Customer Demands	Very High to Very Low	Contingent	2,5
Users Rotation	Very High to Very Low	Invariant	2,3
Context Inconsistencies	Very High to Very Low	Invariant	2,3
Conflict of Users Interests	Very High to Very Low	Contingent	2,3
Project Factors	Acceptable Values	Evolution	Variation Points
Familiarity with the Domain	Very High to Very Low	Contingent	1,2,3,4
Size of the Project	Very High to Very Low	Contingent	2,3,5
Developers Rotation	Very High to Very Low	Invariant	2,3
Software Quality Required	Very High to Very Low	Invariant	1,2,3,4,5
Reuse existing Requirements Artifacts	Yes, No	Invariant	1,2,3,4,5
Creating Requirements Artifacts for Reuse	Yes, No	Invariant	1,2,3,4,5
Requirements Pre-Traceability	Yes, No	Invariant	1,2,3,4
Requirements Post-Traceability	Yes, No	Invariant	5
Granularity of Requirements Traceability	Individual, Group	Invariant	5
Demand to produce an SRS	Yes, No	Invariant	5
Project Time and Resource Constraints	Very High to Very Low	Contingent	1,2,3,4,5
Software Process Model	Cascade, Incremental, Iterative, Agile	Invariant	1,2,3,4,5

that do not change during the RE process; and *Contingent*, being those that may naturally evolve due to changes in the application context and/or in the project context.

Interaction among factors should be taken into account when tailoring the RE process. *Overriding interaction* indicates that a factor is discarded when another factor takes a specific value. *Limiting interaction* points out that a factor may reduce the range of acceptable values of another factor. *Incompatible interaction* implies that the values of two or more factors cannot be accepted simultaneously at the same project; thereby, the value of at least one factor has to be changed through stakeholders' negotiation to solve the inconsistency. Examples of these interactions are:

- **Overriding Interaction:** The factor Business Process Reengineering is not taken into account if the factor New Business takes the value *Yes*.

- **Limiting Interaction:** The factor Software Process Model cannot take the value *Cascade* if factor Context Volatility is *Very High*.
- **Incompatible Interaction:** The factor Software Quality Required is *Very High* while the factor Project Time and Resource Constraints is *Very Low*.

Each sub-process is defined by a sequence of process blocks, some of which include product models. At each variation point a basic process block may: i) stay as it is; ii) be deleted; iii) be replaced by a particular process block; or iv) be replaced by a process block with a partial internal variation based on parameters. Figure 2, exploiting sub-process of Figure 1(c), shows how the instance of a process should be constructed according to specific situational factors, using a repository containing all possible atomic blocks. This Figure exemplifies two process blocks that may be deleted (*Scenarios Organization* and *Sc-*

Figure 2. Defining the Future Scenarios Construction Sub-Process

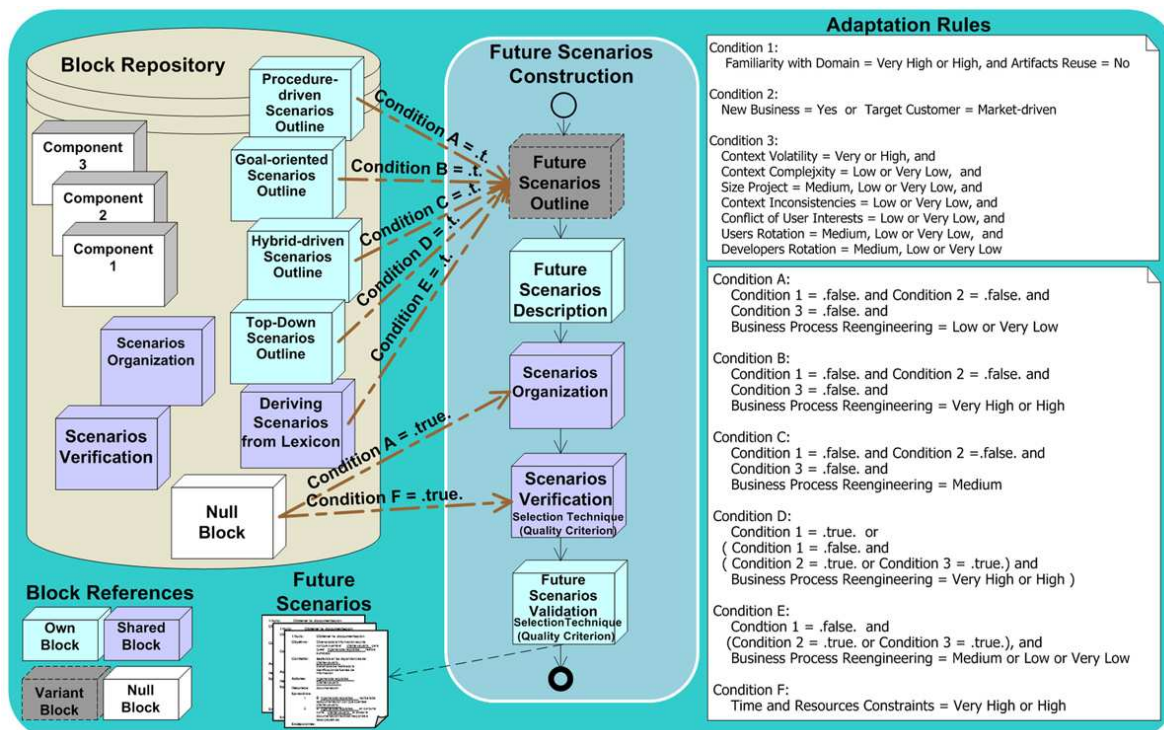


Table 2. Valid operations on atomic process blocks

BLOCK TYPE	OPERATIONS			
	Stay	Delete	Replace	Parameterize
Own	✓	✓		✓
Shared	✓	✓		✓
Variant		✓	✓	

narios Verification), a process block that should be replaced by one of five different atomic processes (Future Scenarios Outline), two process blocks that are parameterized (Scenarios Verification and Scenarios Validation), and a process block that stays independently of any factor (Future Scenarios Description).

There are common elements among the various parts of the RE process. For example, Figure 2 shows two process blocks Scenarios Organization and Scenarios Verification which both are shared with another main sub-process Current Scenarios Construction. There are four types of atomic process blocks. An own block is that not shared

Figure 3. Instance of the RE Process for a specific situation

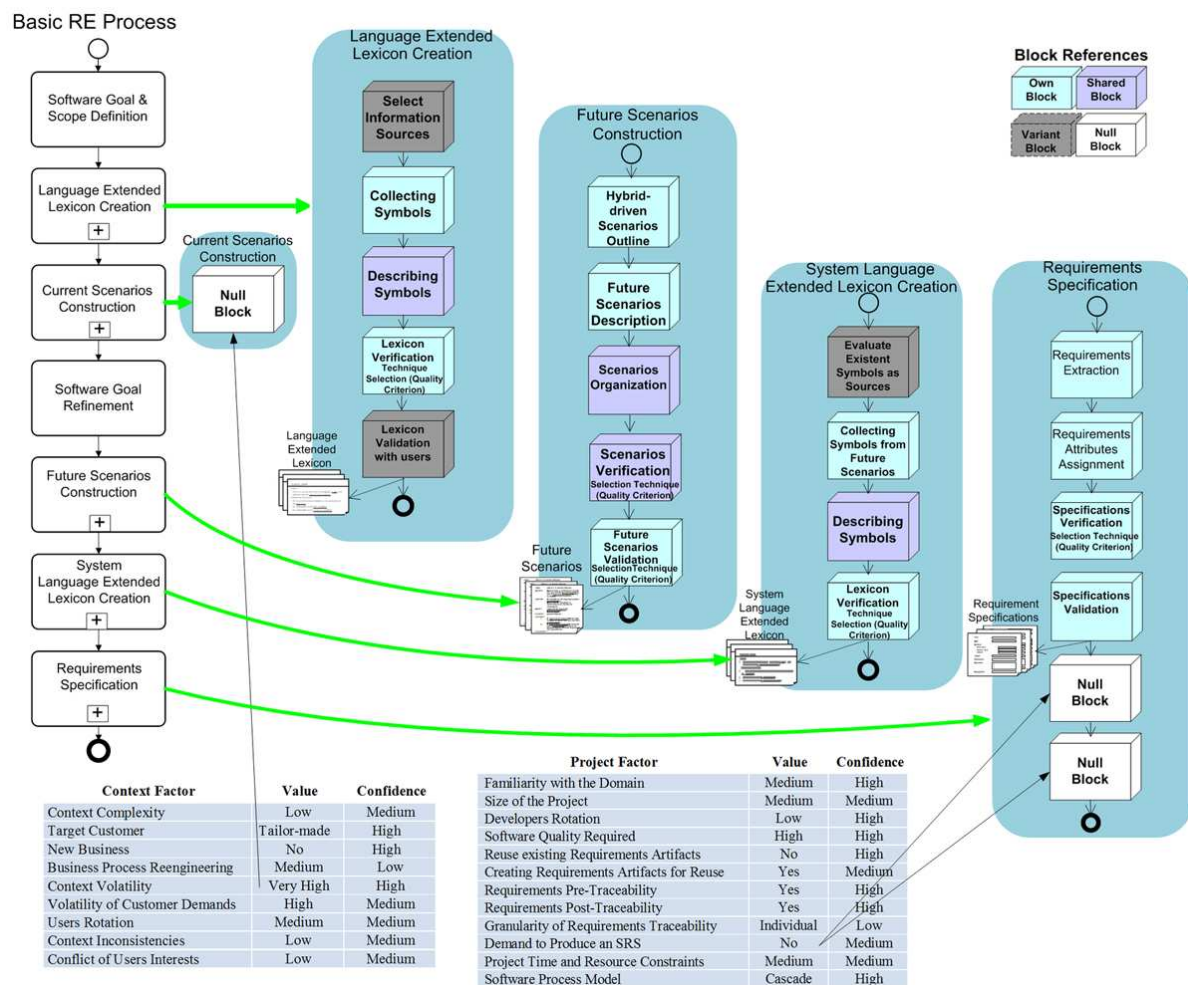
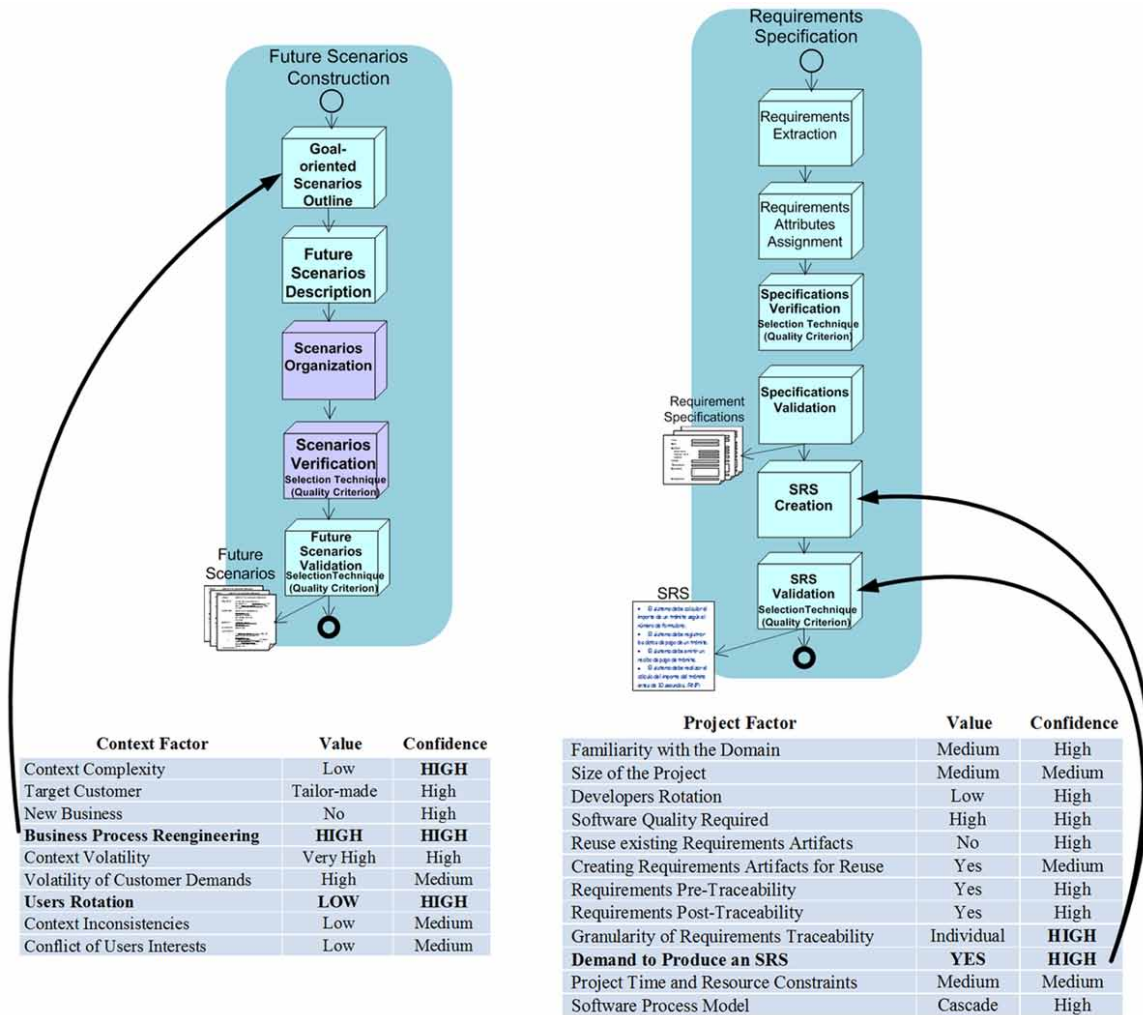


Figure 4. Evolution of the Instance of the RE Process



by any other main sub-process and is independent of situational factors. A *shared block* is common to different sub-processes of the RE process, and may depend on factors. A *variant block* belongs to a specific sub-process and depends on situational factors. A *null block* is an auxiliary block used when an entire sub-process or an atomic process must be deleted or skipped from the basic RE process. Table 2 shows operations that may be applied to each type of process block.

Some process blocks include product blocks, which contain minor variations of the NL models produced by the RE process. For example, if the lexicon model is not created, then the scenario

model does not include hyperlinks to lexicon symbols.

Figure 3 shows an instance of the RE process constructed for a specific situation, which is represented by a set of values given to factors at the beginning of the project. Those factors are sketched at the bottom of this Figure. While ongoing the project, some factors may evolve or change; this is exemplified in Figure 4 where at variation point 3 (see Figure 1), the following factors were updated: Business Process Reengineering, Users Rotation and Demand to Produce an SRS. These three factors, along with some others, gained a better confidence. The first and the third factor

have impacted on two different sub-processes, thus Figure 4 shows only sub-processes requiring a re-design.

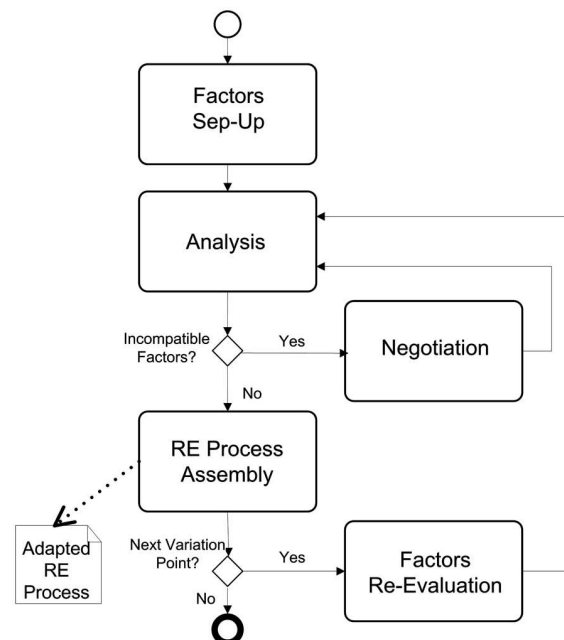
## SOLUTIONS AND RECOMMENDATIONS

The adaptation of the basic RE process involves the application of a mechanism of operations on process blocks at every variation point, using a specific combination of situational factors, following SME principles. The variation points are milestones to review the RE process and to indicate possible re-design of remaining RE activities. The process for constructing the RE process for a specific situation includes the following activities (see Figure 5):

1. **Factors Set-Up:** The situational factors are evaluated, assigning a degree of confidence to each value, and taking into account the interaction among factors. A simple form is used to collect the factors value. This form eases its filling with valid values by means of rules for factors interaction.
2. **Analysis:** The factors setting is analyzed considering mainly interactions and factors with low confidence. Project managers should take a position on every factor with low confidence: optimistic, conservative or balanced position. Under an optimistic position, factors are assigned an extreme value of its valid range depending on the meaning of the factor, while under a conservative position factors take the opposite extreme value. A balanced position involves choosing intermediate values. This re-allocation of values is done to allow assembling the process, since the original values are preserved. When incompatible interactions are observed, a negotiation must take place.
3. **Negotiation:** The purpose of this activity is to reach an agreement among stakeholders to re-assign values to incompatible factors.

4. **RE Process Assembly:** The RE process is assembled departing from the basic process, considering a subset of situational factors at each variation point (see Table 1, last column). A set of rules based on a specific combination of those involved factors defines which operation is applied over each basic process block (see an example at Figure 2, right part). Thus, this assembly activity consists in maintaining, deleting, replacing, or parameterizing process blocks.
5. **Factors Re-Evaluation:** At every variation point, those factors involved in remaining variation points must be re-evaluated. Low-confidence factors must be watched, since its level of confidence is supposed to increase due to a better perception of the situation. Contingent factors must be also watched to establish if a change in the situation has occurred. After this activity, a new factors analysis is needed in order to ensure the absence of inconsistencies among dependent factors.

Figure 5. Process for Constructing an RE Process, based on situational factors





Since requirements are the foundations for a good software product, an RE process is crucial to the project success. Project managers should take decisions about selecting the appropriate RE process for a specific situation. A way to define the situation is by its characterization through context and project factors. Therefore, project managers should define as precisely as possible those situational factors in order to construct the suitable RE process. However, they should be aware of the potential evolution of the situation and, thereby, review factors and adjust the RE process at pre-defined milestones.

## FUTURE RESEARCH DIRECTIONS

Feasible variants of the process were identified due to the experience achieved after putting into practice the basic RE process in many real cases along two decades.

Situational factors were exhaustively studied in literature, although this set is likely to be extended. Factors impacting on subsequent phases of the software development process will be studied to determine if they may influence backward on the RE process.

The decomposition of the RE process in basic process blocks, along with the set of situational factors, has allowed identifying process commonality and process variability, and hence defining the different types of process blocks needed. It is planned to evaluate the convenience of using mixed blocks, or separated process blocks from model blocks. It is also considered that more cases should be done to confirm the different branches of the adaptable RE process.

## CONCLUSION

The proposal presented in this chapter deals with variability in an RE process, driven by a dynamic adaptation based on prevailing situational factors.

Most problems have distinctive features that must be taken into account to carry out a successful requirements process. Project managers should tailor the requirements process by choosing the techniques that are most suitable for specific situations. When they are provided with guidelines on how to adapt the requirements process, they are more willingly to do it.

The process for constructing an RE process involves a set of factors that typifies situations considering both the application context and the software project context. This adaptation process establishes when the RE process should be reviewed and re-tailored based on rules and operations at each variation point. Since it is frequently observed the natural and contingent evolution of situations, it is important to have the possibility of redefining the remaining activities of the RE process once the project is in course.

The adaptation of the requirements process for several real cases has contributed to its acceptance in host organizations. It is considered that the dynamic adaptation process could be implemented with minor adjustments in other RE strategies.

## REFERENCES

- Alexander, I. F., & Beus-Dukic, L. (2009). *Discovering Requirements: How to Specify Products and Services*. Chichester, UK: John Wiley & Sons.
- Antonelli, L., Rossi, G., Leite, J. C. S. P., & Oliveros, A. (2012). Deriving requirements specifications from the application domain language captured by Language Extended Lexicon. In *Proceedings of XV Workshop on Requirements Engineering*. Buenos Aires, Argentina: Universidad Nacional de La Matanza.
- Bakhat, K. A., Sarwar, A. A., Motla, Y. H. B., & Akhtar, M. C. (2015). A Situational Requirement Engineering Model for an Agile Process. *Bahria University Journal of Information & Communication Technology*, 8(1), 21–26.

- Beck, K. (2004). *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley Professional.
- Brinkkemper, S. (1996). Method Engineering: Engineering of Information Systems Development Methods and Tools. *Information and Software Technology*, 38(4), 275–280. doi:10.1016/0950-5849(95)01059-9
- Bucher, T., Klesse, M., Kurpjuweit, S., & Winter, R. (2007). Situational Method Engineering. In *Situational method engineering: fundamentals and experiences* (pp. 33-48). Springer US. doi:10.1007/978-0-387-73947-2\_5
- Carrizo, D. (2009). *Marco para la selección de técnicas para educación de requisitos* (Doctoral Thesis). Universidad Politécnica de Madrid, Madrid, Spain.
- Carrizo, D., Dieste, O., & Juristo, N. (2008). Study of elicitation techniques adequacy. In *Proceedings XI Workshop on Requirements Engineering* (pp. 104-114). Barcelona, Spain: Universitat Politècnica de Catalunya.
- Cockburn, A. (2002). *Agile Software Development*. Boston: Addison-Wesley.
- Coulin, C. R. (2007). *A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation* (Doctoral Thesis). University of Technology, Sydney, Australia.
- Firesmith, D. (2004). Creating a project-specific requirements engineering process. *Journal of Object Technology*, 3(5), 31-44.
- Firesmith, D.G., & Henderson, B. (2002). *The OPEN Process Framework: An Introduction*. Harlow, UK: Addison-Wesley.
- Galster, M., Weyns, D., Tofan, D., Michalik, B., & Avgeriou, P. (2014). Variability in Software Systems - A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 40(3), 282–306. doi:10.1109/TSE.2013.56
- Hadad, G. D., Doorn, J. H., & Kaplan, G. N. (2009). Creating Software System Context Glossaries. In M. Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology* (2nd ed.; pp. 789–794). Hershey, PA: IGI Global. doi:10.4018/978-1-60566-026-4.ch128
- Haumer, P. (2005). *IBM Rational Method Composer: Part 1: Key concepts*. Retrieved August 27, 2015, from <http://www.ibm.com/developerworks/rational/library/dec05/haumer/index.htm>
- Henderson-Sellers, B., & Ralyté, J. (2010). Situational Method Engineering: State-of-the-Art Review. *J. UCS*, 16(3), 424–478.
- Hickey, A. M., & Davis, A. M. (2003, September). Elicitation technique selection: how do experts do it? In *Proceedings of 11th IEEE International Requirements Engineering Conference* (pp. 169-178). IEEE. doi:10.1109/ICRE.2003.1232748
- Jafarinezhad, O., & Ramsin, R. (2012, March). Towards a process factory for developing situational requirements engineering processes. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 1089-1090). ACM. doi:10.1145/2245276.2231946
- Kaindl, H. (2000). A design process based on a model combining scenarios with goals and functions. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, 30(5), 537–551. doi:10.1109/3468.867861
- Khan, H.H., bin Mahrin, M.N., & bt Chuprat, S. (2014). Factors for Tailoring Requirements Engineering Process: A Review. *International Journal of Software Engineering and Technology*, 1(1), 7–18.
- Kohler, K., & Paech, B. (2002). Requirement Documents that Win the Race: Not Overweight or Emaciated but Powerful and in Shape. In *Proceedings of International Workshop on Time-Constrained Requirements Engineering (TCRE'02)*.

Kovitz, B. L. (2002). Hidden Skills that Support Phased and Agile Requirements Engineering. In *Proceedings of International Workshop on Time-Constrained Requirements Engineering (TCRE'02)*.

Kumar, K., & Welke, R. J. (1992). Methodology Engineering: a proposal for situation-specific methodology construction. In W. W. Cotterman & J. A. Senn (Eds.), *Challenges and strategies for research in systems development* (pp. 257–269). John Wiley & Sons.

Lauesen, S. (2002). *Software Requirements: Styles and Techniques*. Addison-Wesley.

Leffingwell, D., & Widrig, D. (2003). *Managing Software Requirements: a unified approach*. Addison-Wesley Professional.

Leite, J. C. S. P., Doorn, J. H., Kaplan, G. N., Hadad, G. D., & Ridaio, M. N. (2004). Defining System Context using Scenarios. In J. C. S. P. Leite & J. H. Doorn (Eds.), *Perspectives on Software Requirements* (pp. 169–199). Springer, US. doi:10.1007/978-1-4615-0465-8\_8

Leite, J. C. S. P., Hadad, G. D., Doorn, J. H., & Kaplan, G. N. (2000). A scenario construction process. *Requirements Engineering*, 5(1), 38–61. doi:10.1007/PL00010342

Lobo, L. O., & Arthur, J. D. (2005, April). An objectives-driven process for selecting methods to support requirements engineering activities. In *Proceedings of 29th Annual IEEE/NASA Software Engineering Workshop* (pp. 118-130). IEEE. doi:10.1109/SEW.2005.18

Macaulay, L. (1993, January). Requirements capture as a cooperative activity. In *Proceedings of IEEE International Symposium on Requirements Engineering* (pp. 174-181). IEEE.

Maiden, N. A. M., & Rugg, G. (1996). ACRE: Selecting methods for requirements acquisition. *Software Engineering Journal*, 11(3), 183–192. doi:10.1049/sej.1996.0024

Pinheiro, F. (2002). Requirements Honesty. In *Proceedings of International Workshop on Time-Constrained Requirements Engineering (TCRE'02)*.

Potts, C. (1995, August). Using schematic scenarios to understand user needs. In *Proceedings of the 1st Conference on Designing interactive systems: processes, practices, methods, & techniques* (pp. 247-256). ACM. doi:10.1145/225434.225462

Ralyté, J. (2013). Situational Method Engineering in Practice: A Case Study in a Small Enterprise. In *Proceedings of CAiSE'13 Forum at the 25th International Conference on Advanced Information Systems Engineering*, pp. 17-24.

Rolland, C. (2008). Method engineering: towards methods as services. In *Making Globally Distributed Software Development a Success Story* (pp. 10–11). Springer Berlin Heidelberg. doi:10.1007/978-3-540-79588-9\_2

Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., & Ncube, C. (2009). Exploring how to use scenarios to discover requirements. *Requirements Engineering*, 14(2), 91–111. doi:10.1007/s00766-009-0077-9

Sillitti, A., & Succi, G. (2005). Requirements Engineering for Agile Methods. In A. Aurum & C. Wohlin (Eds.), *Engineering and Managing Software Requirements* (pp. 309–326). Springer-Verlag. doi:10.1007/3-540-28244-0\_14

## ADDITIONAL READING

Browne, G. J., & Ramesh, V. (2002). Improving information requirements determination: A cognitive perspective. *Information & Management*, 39(8), 625–645. doi:10.1016/S0378-7206(02)00014-9

Henderson-Sellers, B., Ralyté, J., Ågerfalk, P., & Rossi, M. (2014). *Situational Method Engineering*. Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-41467-1

Kotonya, G., & Sommerville, I. (1998). *Requirements Engineering: Process and Techniques*. John Wiley & Sons.

Lloyd, W., Rosson, M., & Arthur, J. (2002). Effectiveness of elicitation techniques in distributed requirements engineering. In *Proceedings of 10th IEEE Joint International Conference on Requirements Engineering* (pp. 311-318). doi:10.1109/ICRE.2002.1048544

Ralyté, J., Deneckère, R., & Rolland, C. (2003, January). Towards a generic model for situational method engineering. In J. Eder & M. Missikoff (Eds.), *Advanced Information Systems Engineering* (pp. 95-110). Springer-Verlag Berlin Heidelberg. doi:10.1007/3-540-45017-3\_9

Schnieders, A., & Puhmann, F. (2007). Variability modeling and product derivation in e-business process families. In *Technologies for business information systems* (pp. 63-74). Springer Netherlands. doi:10.1007/1-4020-5634-6\_6

Virtanen, P., Pekkola, S., & Paivarinta, T. (2013, January). Why SPI Initiative Failed: Contextual Factors and Changing Software Development Environment. In *Proceedings of 46th Hawaii International Conference on System Sciences (HICSS)*, (pp. 4606-4615). IEEE. doi:10.1109/HICSS.2013.609

Vlaanderen, K., Valverde, F., & Pastor, O. (2008). Improvement of a web engineering method applying situational method engineering. In *Proceedings of 10th International Conference on Enterprise Information Systems*, (3-1), (pp.147-154). Barcelona, Spain.

## **KEY TERMS AND DEFINITIONS**

**Process Block:** A well-defined unit of process that can be reused as part of any process.

**Process Commonality:** The common elements of a process that facilitate the definition of a family of processes through reuse.

**Process Variability:** The variant elements of a process, identified at variation points, which produce deviation from the standard process.

**Requirements Engineering Process:** A process to produce software requirements by means of methods, techniques and tools during elicitation, modeling, analyzing and evolution of requirements.

**Scenario:** A representation of an observed or envisioned situation in the application context.

**Situational Factor:** A characteristic of the project or the application context that may be taken into account when implementing an RE process.

**Situational Method Engineering:** A discipline that promotes the construction of methods for developing systems according to a pre-defined set of situational factors.

# Requirements Authorship: A Family Process Pattern

Graciela D. S. Hadad, Jorge H. Doorn

Escuela de Informática

Universidad Nacional del Oeste

Departamento de Ingeniería e Investigaciones Tecnológicas

Universidad Nacional de La Matanza

Buenos Aires, Argentina

Julio Cesar Sampaio do Prado Leite

Departamento de Informática

Pontificia Universidade Católica do Rio de Janeiro

Rio de Janeiro, Brasil

julio@inf.puc-rio.br

**Abstract** – Different contexts determine a different software process. This is especially true for requirements production. Different situations in a given universe of discourse will require different requirements processes. This paper proposes a family pattern that approaches the universe of discourse (context) from the viewpoint of whom are the actors demanding the software. We show five different instantiations of this pattern.

**Index Terms** – requirements patterns, requirements authorship, process patterns, family pattern

## I. INTRODUCTION

Patterns have been used in architecture for years and its acceptance in the software construction is mainly due to the work of Gamma et al. in their landmark book Design Patterns [1]. Since then, different proposals have been made towards adopting the idea of pattern, a recurring structure/solution, in different areas of software production. Requirements patterns have been an important issue, since it brings the possibility of reusing a previous defined requirement. However, there has been a tradition that software reuse is only effective when code is reused. As more and more Requirements Engineering requires a knowledge driven approach, the more the idea of reusing information at the requirements level attracts increasing attention from the research community [2], being the RePa workshop an example of this. One of such type of patterns are the so-called processes patterns [3]. Fiorini et al. [4] makes a distinction between process patterns and standards. In our case, we are dealing with a process pattern that will enable the choice or at least will bring attention to the type of process that will be needed for a given situation.

We consider that elicitation should not be centered in capturing requirements; on the contrary, elicitation should be mainly focused on capturing knowledge about the application context in order to build requirements according to the context or to an enhanced context. However, not every software development situation allows this way of producing requirements. Therefore, we define five possible situations where requirements process may or may not take place when software is needed. These situations depend on the process actors involved and the existence of an application context. Considering the aforementioned concepts in the software production, we designed patterns to better guide a requirements engineering process. The main public for these patterns are junior requirements engineers, or even senior ones with

more focused experience. The main goal is to have a clear picture of the requirements process from the point of view of whom are demanding the requirements. These patterns are presented in this paper.

## II. PATTERNS SOCIETY PROCESSES

We use the idea of “Process Patterns Society” [4] to introduce a family pattern and five individual processes that are part of the family. We also use the language proposed by Fiorini et al. [4] [5] to describe the patterns. According to Fiorini et al., a family pattern belongs to a community pattern. In our case the community pattern is requirements process, so our family is one of the families that compose this community.

The Family Process is composed of the following descriptors:

- Identification (What?) – The name of the family with keywords (tags) to help search.
- Context (Where?; When?; Who?) – Describes the location, the time and who are involved.
- Cause/Reason (Why?) – The purpose of the pattern, the “problems” it avoids.
- Analysis (How?) – Ways to verify and validate the pattern.
- Related Patterns (Who?) – Other patterns that are related to the family.
- Representation (How?) – Language used to represent the components (individual patterns).
- Solution (How?; Which?; Who?) – Describes the family components. We use Ridao’s et al. [6] idea of a decision tree as to represent the solution component for the family, since it provides a clear characterization of different possibilities.

## III. THE FAMILY PROCESS (REQUIREMENTS AUTHORSHIP)

The instantiation of the family process for the case at hand is the following one:

- **Identification** – Requirements Authorship as Anchor to Select a Requirements Process
- **Context** – In the planning phase of the requirements process, the requirements manager identifies the stakeholders that are demanding the software.
- **Cause/Reason** – Different authorship poses a problem to Requirements Engineering with respect to which process to adopt.

- **Analysis** – The decision tree helps to verify the conditions of application of each individual pattern.
- **Related Patterns** – Not known.
- **Representation** – Fiorini [4] and Ridao [6].
- **Solution** – The decision tree of Figure 1 shows the decision conditions for the application of the five processes belonging to the family. Each of them will be a different individual process pattern.

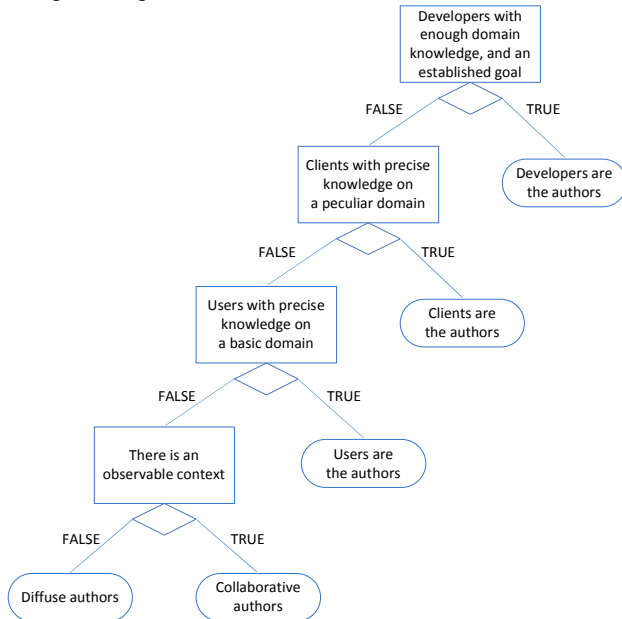


Fig. 1 – The Decision Tree<sup>1</sup>

#### IV. COMPONENTS OF THE FAMILY PROCESS

Following the pattern language proposed by [4], the individual processes, which are the components of the family process (Requirements Authorship), has the following structure: each component has a **name**, **input**, **recommendation**, **output**, and optionally may have **pre-condition**, restriction, **post-condition** and uses-pattern. We added the **Examples of use** to the pattern as to better explain where it is applied.

##### A. Developers Authorship

**Name:** Developers are the authors

**Pre-condition:** There is an observable application context. The developer organization has much knowledge and experience in a specific domain or in a client organization, needing null or minimal support from other stakeholders. A special case is when developers need a software tool for their own use, thus being clients and developers.

**Input:** System goals and clients' needs, and, optionally, detailed system plans.

<sup>1</sup> We use the term *developer* to cover different roles in a development team, such as managers, requirements engineers, designers, and programmers, among others. We mention a specific role when the distinction is appropriate. The term *client* is used for the owner of the software, who is neither a developer nor a user.

**Recommendation:** When in-house systems or bespoke systems are required for an organization, clients provide system goals and general needs. The developers transform needs into requirements by **two** main different cases:

Case 1:

**Pre-condition:** Requirements engineers build requirements from needs.

**Recommendation:** Requirements engineers model and verify the requirements. Requirements validation is not needed, or may be an informal activity. Requirements are then given to designers or programmers.

Case 2:

**Pre-condition:** Needs are communicated directly to designers or programmers.

**Recommendation:** Designers or programmers have enough knowledge to develop the solution.

**Output:** Requirements.

**Post-condition:** Each case will have different quality levels, with case 1 being the one with high quality, due to the use of Requirements Engineering knowledge, though requirements engineers impose their solution and, if needed, the necessary modifications to the current context, without negotiation, and with little elicitation and validation.

**Examples of use:** Software systems to support information systems in an organization, being developed by their own system department or by an external software factory that provides them with continuous support. Systems as part of a family product produced and maintained by a software factory. Software tools to support the development of applications in a software factory, such as a compiler, a code generator for a particular platform and a systems interface system, among others utility tools.

##### B. Clients Authorship

**Name:** Clients are the authors

**Pre-condition:** There is an observable application context. There are no users or they are merely operators in the background. The clients are experts on a specific and complex domain, which is hard to understand by other stakeholders.

**Input:** Clients' precise system goals.

**Recommendation:** Clients provide their needs as requirements. There are **three** main different cases:

Case 1:

**Pre-condition:** Requirements engineers elicit requirements.

**Recommendation:** Requirements engineers model and analyze the requirements, which are then given to designers or programmers.

Case 2:

**Pre-condition:** Needs are communicated directly to designers or programmers.

**Recommendation:** Designers or programmers interact with clients, using simple elicitation techniques (i.e. interview), without performing requirements modeling, analysis, and management activities.

Case 3:

**Pre-condition:** Only clients are involved: they have some programming skills. Requirements are difficult to communicate.

**Recommendation:** Clients will develop their own system.

**Output:** Requirements

**Post-condition:** Each case will have different quality levels, with case 1 being the one with high quality, due to the use of Requirements Engineering knowledge, though requirements engineers have no possibility to suggest improvements to the provided needs or the current context. Applying cases 2 or 3, requirements are usually not specified.

**Examples of use:** Complex systems of scientific or technological calculation in laboratories of mechanics, electronics, chemicals or bio-chemicals, automatic control systems, embedded medical systems, among others.

### C. Users Authorship

**Name:** Users are the authors

**Pre-condition:** There is an observable application context. The user acts as a client in a one person context or of very few people, or the user may be an operative agent in a big organization whose activities are not backed by the central processing system or are a gap in non-integrated systems.

**Input:** Users' precise system needs.

**Recommendation:** Users provide their needs as requirements. There are **three** main different cases:

Case 1: (unusual one)

**Pre-condition:** Requirements engineers elicit requirements.

**Recommendation:** Requirements engineers model and analyze the requirements, which are then given to designers or programmers.

Case 2:

**Pre-condition:** Needs are communicated directly to designers or programmers.

**Recommendation:** Designers or programmers interact with users, using simple elicitation techniques (i.e. interview), without performing requirements modeling, analysis, and management activities.

Case 3:

**Pre-condition:** Only users are involved: they have some basic programming skills. There are not available developers or they are not really needed.

**Recommendation:** Users will develop their own system.

**Output:** Requirements.

**Post-condition:** Each case will have different quality levels, with case 1 being the one with better quality, due to the use of Requirements Engineering knowledge, though requirements engineers have no possibility to suggest improvements to the provided needs or the current context. Applying cases 2 or 3, requirements are usually not specified.

**Examples of use:** Small application systems, such as accountability system, system for housing condominium administrators, system for musical composers, system for import-

process-export information from/to other systems. Usually these systems are built using spreadsheets or user oriented database software.

### D. Collaborative Authorship

**Name:** Collaborative authors

**Pre-condition:** There is an observable application context. Clients, users, requirements engineers and eventually experts have some knowledge about the current context to share among them.

**Input:** Clients' goals and users' needs.

**Recommendation:** Clients require the achievement of certain goals. Requirements engineers elicit information about the application context and then about users requirements. Some experts may provide specific knowledge when particularities on the application context require so. Clients and users may elaborate some proposals, but they mainly come from requirements engineers. These proposals are about the system's services and the context where the system will operate. Most proposals may produce alterations in the current context, requiring a partial or full redefinition of it. Eventually, there are cases where only an automation of the activities is expected, preserving invariant the current context. Thus, proposals are modeled, verified and negotiated among the stakeholders. Clients take decisions about the better solution helped by requirements engineers.

**Output:** Requirements. A future application context, partially or totally re-defined if needed.

**Post-condition:** A Requirements Engineering process was performed carrying out elicitation, modeling, analysis, and requirements management activities, producing high quality requirements, integrated with the context and aligned with clients' goals.

**Examples of use:** Software for information systems.

### E. Diffuse Authorship

**Name:** Diffuse authors

**Pre-condition:** Non-observable application context. Each stakeholder has not enough knowledge about the problem (Clients only have general ideas. Users are quite far from giving any useful information about behaviors or characteristics of the future context and the system. Requirements engineers alone have also no precise knowledge about what must be done. Experts may have innovative and/or specific technical issues.).

**Input:** Clients' imprecise system goals or ideas.

**Recommendation:** Clients provide proposals in terms of some general goals or ideas about their intentions. Experts on different areas provide proposals on possible and partial solutions about the system and its context of use. Requirements engineers may sketch some technological proposals, probably helped by designers. Partial solutions are studied and discussed by stakeholders. Requirements engineers integrate partial solutions. Experts do a feasibility analysis on these proposals from their technical domain. Requirements engineers do a feasibility analysis on these proposals from their computational domain. Clients take decisions about the better solution helped by experts and requirements engineers.

**Output:** Requirements. An application context defined.

**Post-condition:** A Requirements Engineering process was performed with intensive elicitation, several modeled proposals, analysis, and negotiation activities, producing high quality requirements integrated with the defined context.

**Examples of use:** Complex and/or innovative systems, commonly in System Engineering and Computer Engineering, such as automatic mechanization factory, complex logistic warehousing, urban buses supervising system, water distribution systems among land irrigation, city needs and hydroelectric generation

## V. CONCLUSIONS

The use of patterns is a necessary resource to the modeling of complex processes elements. A pattern highlights the most important elements and defines a language to approach similar cases.

The patterns family presented synthesizes a wide knowledge available in the literature and our experience on different types of situations, i.e., pattern A [7], pattern B [8] [9], pattern C [10], pattern D [11] [12] and pattern E [13] [14]. Our contribution relies on the idea of applying the process family concept to the perspective of requirements authorship, that is, what type of demanding actor will define the requirements. The structure we provide should help requirements engineers, mainly junior ones, to plan their processes according to the overall situation they may face. Of course, that other types may be proposed, but by focusing on clients, users and developers, we are making a coarse abstraction that is common in the literature.

Sultan and Miranskyy [15] propose a pattern for requirements elicitation concerning four stakeholder viewpoints. The pattern is based on six questions to elicit requirements in an organizational context, mainly asking who the actors are, what they need, why they need them, which are the options, and when, where and how needs should be satisfied. The pattern is also useful in requirements analysis; however, it is neither applicable on other types of context nor to elicit detailed context knowledge.

There is much in literature about developing situational Requirements Engineering processes [16] [17], and also about standard Requirements Engineering process pattern combined with processes blocks [18]; however, we present here a higher level of abstraction about when full, partial or no Requirements Engineering could be applied and the level of requirements quality it may achieve.

Understanding the requirements authorship issues is not as simple as it may appear at a first glance. There are some difficult situations where the role of the requirements engineers is not clear. Patterns B and C clearly show (case 1) that even in the context of an apparently well-defined set of requirements their modeling and analysis would make a big difference. Pattern E reinforces the idea that requirements production should be done concurrently with the overall definition of complex systems contexts. Pattern A emphasizes the fact that even in the situation of very context-experienced developers the Requirements Engineering improves the process quality.

## ACKNOWLEDGMENT

Leite acknowledges the support of CNPq and Faperj (Cientista do Nosso Estado).

## REFERENCES

- [1] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Design, Addison-Wesley, 1995.
- [2] Supakkul, S., Hill, T., Chung, L., Tun, T.T, Leite, J.C.S.P., An NFR Pattern Approach to Dealing with NFRs, Requirements Engineering IEEE International Conference on, 179-188, 2010.
- [3] Ambler, S. W., Process Patterns – Building Large-Scale Systems Using Object Technology, Cambridge University Press/ SIGS Books, 1998.
- [4] Fiorini, S.T., Leite, J.C.S.P., Lucena, C.J.P., Process Reuse Architecture, Advanced Information Systems Engineering International Conference on, LNCS 2068, 284–298, 2001.
- [5] Fiorini, S.T., Leite, J.C.S.P., Arquitetura para Reutilização de Processos (Process Reuse Architecture), PhD Thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2001.
- [6] Ridao, M., Doorn, J.H., Leite, J.C.S.P., Domain independent regularities in scenarios, Requirements Engineering Fifth IEEE International Symposium on, 120-127, 2001.
- [7] Schmid, K., Strategically Defining and Exploiting Product Portfolios with a Product Line Approach, Perspectives on Software Requirements, Kluwer, 223-251, 2004.
- [8] Sempau, J., Sanchez Reyes, A., Salvat F., Oulad Ben Tahar H., Jiang, S. B., Fernandez Varea, J. M., Monte Carlo simulation of electron beams from an Accelerator head using PENELOPE, Physics in medicine and biology, 46(4), 1163-1186, 2001.
- [9] Zave, P., Requirements for Evolving Systems: A Telecommunications Perspective, Requirements Engineering Fifth IEEE International Symposium on, 2-9, 2001.
- [10] Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G., Mehandjiev, N., Meta-design: a manifesto for end-user development, Communications of the ACM, 47(9), 33-37, 2004.
- [11] Konaté, J., Sahraoui, A. E. K., Kolfschoten, G. L., Collaborative Requirements Elicitation: A Process-Centred Approach, Group Decision and Negotiation, Springer, 23(4), 847-877, 2014.
- [12] Sutcliffe, A., Collaborative requirements engineering: bridging the gulfs between worlds, Intentional Perspectives on Information Systems Engineering, Springer, 355-376, 2010.
- [13] Kauppinen, M., Savolainen, J., Mannisto, T., Requirements Engineering as a Driver for Innovations, Requirements Engineering IEEE International Conference on, 15-20, 2007.
- [14] Souza, R.A.C., Cysneiros Filho, G.A.A., Batista, G.H.C., A Heuristic Approach for Supporting Innovation in Requirements Engineering, 19<sup>th</sup> Workshop on Requirements Engineering, 2015.
- [15] Sultan, M., Miranskyy, A., IEEE Fifth International Workshop on Requirements Patterns, 1-8, 2015.
- [16] Jafarinezhad, O., Ramsin, R., Towards a process factory for developing situational requirements engineering processes, 27th ACM Symposium on Applied Computing, 1089-1090, 2012.
- [17] Khan, H.H., bin Mahrin, M.N., bt Chuprat, S., Factors for Tailoring Requirements Engineering Process: A Review, International Journal of Software Engineering and Technology, 1(1), 7–18, 2014.
- [18] Li Jiang, A Framework for the Requirements Engineering Process Development, PhD Thesis, University of Calgary, 2005.



# A intenção de uso da Linguagem Natural na especificação de requisitos:

um estudo comparativo entre a Argentina e o Brasil

Angélica T. S. Calazans<sup>1</sup>, Roberto A. Paldês<sup>1</sup>, Aldegol O. Paulino<sup>1</sup>, Fabrício R. Freire<sup>1</sup>, Jana P. Fraga<sup>1</sup>, Graciela D. S. Hadad<sup>2,3</sup>, Ari Melo Mariano<sup>4</sup>

<sup>1</sup> Centro Universitário de Brasília, Instituto CEUB de Pesquisa e Desenvolvimento, Brasília, Brasil  
{angelica.calazans, roberto.paldes}@uniceub.br

<sup>2</sup> Universidad Nacional del Oeste, Escuela de Informática, Buenos Aires, Argentina

<sup>3</sup> Universidad Nacional de La Matanza, DIIT, Buenos Aires, Argentina  
ghadad@uno.edu.ar

<sup>4</sup> Universidade de Brasília, EPR/PPCA, Brasília, Brasil  
arimariano@unb.br

**Resumo** — O objetivo deste trabalho é realizar um estudo comparativo sobre os fatores de influência do uso da linguagem natural na especificação de requisitos na Argentina e o Brasil. A metodologia adotada é um estudo descritivo, de abordagem quantitativa por meio de equações estruturais. O modelo teórico proposto baseia-se na Teoria Unificada de Aceitação e Uso da Tecnologia. Foi elaborado um questionário com 17 perguntas para 45 analistas de requisitos da Argentina e 50 analistas de requisitos do Brasil. O instrumento de pesquisa foi validado e pode ser reutilizado. Identificou-se que os motivos que levam os analistas a utilizar a linguagem natural na especificação é a expectativa de um rendimento elevado e a percepção que o uso da linguagem natural leva a uma maior adesão às necessidades do cliente. O estudo destaca outras expectativas podem ser confirmadas com fatores de influência para a Argentina, mas não para o Brasil.

**Palavras Chave**— Engenharia de Requisitos, Especificação de Requisitos, Linguagem Natural, Modelo de percepção.

## I. INTRODUÇÃO

A necessidade de se obter um software com requisitos completos, consistentes, relevantes e que possam atender às necessidades do cliente [1] deixa clara a importância da Engenharia de Requisitos no processo de desenvolvimento de sistemas de software. As atividades de extração e manutenção dos requisitos, partes integrantes de um processo de Engenharia de Requisitos, criam um vínculo entre os stakeholders cliente e equipe de desenvolvimento onde as visões distintas convergem para uma linguagem comum.

Ao mesmo tempo, é primordial uma especificação de requisitos clara, o mais completa possível e sem ambiguidades para geração de documentos consistentes e de fácil entendimento por todas as partes envolvidas na construção de uma solução de software. Essas características vêm ao

encontro à proposta de utilização de Linguagem Natural (LN), pois ela possibilita a estruturação das ideias que dão base a um artefato sólido, de onde se desdobrarão as demais atividades de desenvolvimento do software.

A LN é amplamente utilizada na redação de requisitos dada a sua capacidade de descrever aquilo que é essencial, ou seja, escrever os requisitos de usuário em uma linguagem que os não-especialistas possam compreender [2]. Apesar das desvantagens de sua utilização [3] [4] [5], e até mesmo do desconhecimento de profissionais da área a respeito de condições facilitadoras do seu uso de forma mais efetivo, a LN tem a sua importância reconhecida [6].

Para avaliar o impacto da escrita em LN nos dias atuais é necessário obter a percepção das pessoas que fazem uso desse recurso, os engenheiros ou analistas de requisitos e analistas de sistemas com essa atribuição. Modelos de percepção que envolvem tecnologia têm sido estudados para diferentes áreas [7] [8] [9] através da escala de medida UTAUT (*Unified Theory of Acceptance and Use of Technology* / Teoria Unificada de Aceitação e Uso da Tecnologia). Porém os estudos sobre UTAUT estão aplicados a Tecnologia de uma maneira geral, sendo necessário adaptar este estudo ao da LN. Entender os motivos da sua adoção ou não na engenharia de requisitos é indispensável a melhoria da técnica, assim como da relação das soluções trazidas pelos engenheiros e os usuários envolvidos nas diferentes etapas do processo.

No Brasil, a escala de medida UTAUT já foi utilizada para avaliar o impacto da escrita em LN nos dias atuais [10] e seu uso em diferentes contextos culturais permite uma aproximação acerca do comportamento de ambos os países em relação a LN. Em um mundo onde a interação entre profissionais ocorre de modo global, entretanto, há a necessidade de uma análise mais ampla, envolvendo

profissionais de culturas diversas, onde os aspectos semelhantes e divergentes nos possíveis stakeholders de países distintos sejam compreendidos. Isso possibilita o questionamento o quão diferente podem ser as percepções de profissionais da Argentina e do Brasil que atuam na área de requisitos a respeito da utilização da LN na construção de uma especificação de requisitos de software.

Considerando o exposto, a presente pesquisa objetiva responder as seguintes questões: quais são as percepções dos profissionais dos países analisados acerca da LN na especificação dos requisitos de um produto de software? Essas percepções variam, considerando as visões dos profissionais da Argentina e do Brasil? Para responder a essas questões, o objetivo geral desse trabalho é analisar e comparar a influência das expectativas de desempenho e esforço, bem como condições facilitadoras, com relação a utilização da LN na especificação de requisitos de dois países. É interessante ressaltar que na pesquisa não foram analisadas técnicas específicas de representação em linguagem natural.

No intuito de estabelecer essa comparação, este trabalho foi elaborado aplicando os mesmos critérios já utilizados no estudo realizado no Brasil [10]. Para isso, na seção 2 são descritos, de forma breve, os conceitos relacionados a LN, na seção 3 a metodologia da pesquisa. Na seção 4 os resultados obtidos de cada país, sua análise e comparação. Finalmente, na seção 5 são realizadas as considerações finais e conclusões.

## II. OS REQUISITOS E A LINGUAGEM NATURAL

Os requisitos podem ser definidos como as condições do que o software deve cumprir: os serviços que ele deve oferecer e as restrições ao seu funcionamento [3]. O processo de Engenharia de Requisitos é composto de várias etapas, entre elas descobrir (elicitar), analisar, documentar os requisitos e verificar as funções e restrições definidas [11].

A elicitação de requisitos engloba a identificação, captura e integração de requisitos decorrentes da comunicação entre um grupo de analistas e as partes interessadas, gerando descrições textuais e/ou gráficas que refletem os conceitos mais relevantes do domínio para o desenvolvimento de uma aplicação [12].

Para facilitar essa comunicação, os requisitos são geralmente escritos e mantidos em LN, que é uma forma de comunicação compartilhada por todos os atores do processo, ainda que a flexibilidade da LN traga também riscos de ambiguidade e desentendimento.

No contexto da presente pesquisa, considera-se a Linguagem Natural o uso de frases bem estruturadas para descrever os requisitos de um sistema. Uma LN é elaborada naturalmente pelo ser humano, sem rigores de preparação [13]. É o idioma normalmente utilizado na comunicação em uma comunidade. O padrão IEEE 29148 [14] aponta diversas propriedades para que a especificação de requisitos de software obtenha um bom nível de qualidade entre elas a isenção de ambiguidade e o uso da LN para a descrição de requisitos.

Existem vários motivos para que os requisitos sejam escritos em LN [15] e permaneçam nessa linguagem: é uma

forma de comunicação primária compartilhada por todos os atores do processo de desenvolvimento de software; os requisitos são especificados considerando diferentes níveis de abstração; a relação custo benefício, a necessidade de reação às mudanças de mercado, a rapidez necessária ao desenvolvimento do produto de software, são favoráveis à LN; oferece melhor apoio para a gestão de requisitos errados, incompletos ou parcialmente definidos; pode capturar propriedades externas dos requisitos, como as reais intenções dos usuários.

Além disso, representações em linguagem natural, segundo [16], tendem a ser mais fáceis de entender para todos os stakeholders sendo, portanto, mais acessíveis. Para [17], os requisitos são especificados, na maior parte da indústria, utilizando uma linguagem natural não estruturada. Segundo esse autor, é mais fácil para todos os interessados, mesmo aqueles que carecem de formação técnica, entender a linguagem natural do que outras notações não textuais.

Já [18], em um estudo experimental realizado na Argentina sobre práticas de Engenharia de Requisitos no desenvolvimento de aplicações web em 25 empresas de software, analisaram, entre outros aspectos, o tipo de representação dos requisitos. Os resultados indicaram que 52% da amostra utilizam linguagem natural e 64% utilizam casos de uso, sendo que algumas empresas utilizam mais de uma representação.

Por outro lado, os requisitos escritos em LN podem ser imprecisos, incompletos e ambíguos. A informalidade pode ajudar na discussão entre as partes interessadas no início do projeto, mas pode levar à confusão, inconsistência, falta de automação, ambiguidade e aos erros [19] [20]. Outro desafio na especificação dos requisitos de software, utilizando a LN, é o grande volume de documentos gerados, notadamente em grandes projetos. A medida que cresce a documentação da especificação, aumentam as dificuldades com a interpretação, a consistência e a manutenção [5].

Para reduzir a ambiguidade, uma das soluções foi o desenvolvimento de um Léxico Ampliado da Linguagem [21] tornando mais preciso o vocabulário utilizado na especificação de requisitos. Segundo [19], técnicas de Processamento de Linguagem Natural de têm sido propostas para melhorar a especificação dos requisitos utilizando o modo semi-automático ou totalmente automático, mas até agora não têm sido amplamente implementadas. Alguns pesquisadores afirmam que o processamento de linguagem não é maduro o suficiente para ser aplicado em engenharia de requisitos, mas várias propostas têm sido apresentadas com resultados promissores.

Autores com [22] investigaram o estado atual da automação do processo de elicitação de requisitos, em linguagem natural, considerando as pesquisas existentes que identificam ferramentas automatizadas e semi-automatizadas a partir de documentos de elicitação de requisitos. Realizaram uma revisão sistemática considerando 36 artigos e classificaram esses trabalhos em grupos de ferramentas de total e parcial automatização, com relação a vários domínios, entre eles a elicitação de requisitos. Segundo esses autores, a Linguagem

Natural com sua ambiguidade e inconsistência inerentes, força os engenheiros a verificar os requisitos manualmente e, por vezes, os resultados do automatismo são o retrabalho. Isto inclui corrigir erros, estender ou limitar as passagens de texto identificado ou simplesmente adicionando requisitos que foram desconsiderados. Consequentemente, o automatismo geralmente é complementado com funcionalidades manuais de suporte a elicitação.

Reforçando essa afirmação, em [23] se propõem a adaptar os avanços recentes em processamento de linguagem natural e tecnologias semânticas para gerar modelos UML, teste de casos, e mesmo código fonte. Ao final sugerem a necessidade de evolução de classes de ferramentas, estudo ontológico e outros aspectos para que isso seja possível, apesar de alguns resultados promissores de seu trabalho. Citam também a necessidade de interferência humana para resolver problemas de ambiguidades.

Considerando o contexto de elicitação de requisitos não funcionais, autores como [24], citam que a detecção precoce de requisitos não-funcionais é crucial na avaliação da arquitetura. Esses autores propõem uma abordagem de categorização textual semi-supervisionado para a identificação automática e classificação dos requisitos não-funcionais. Utilizam um método para detecção e classificação e técnicas de aprendizado semi-supervisionado. Essa abordagem semi-supervisionada resultou em taxas de acerto acima de 70%, usando coleções padrões de documentos.

Ou seja, utilizando ou não modelos automatizados, semi-automatizados ou manuais, as organizações continuam a utilizar a linguagem natural em parte das atividades na engenharia de software, especificamente na especificação de requisitos. Identificar e compreender a percepção dos usuários com relação a essa utilização pode vir a contribuir para esse processo no futuro. A seguir detalha-se a metodologia da pesquisa.

### III. METODOLOGIA DA PESQUISA

O método adotado neste estudo foi o descritivo, de abordagem quantitativa por meio de equações estruturais. Equações estruturais é um termo que não designa uma técnica estatística específica, mas a uma série de técnicas e procedimentos utilizados para validar modelos teóricos que definem relações causais, hipotéticas entre variáveis [25], explicitando através de parâmetros o efeito da interação entre elas.

O modelo teórico proposto baseia-se na UTAUT. Esse modelo foi elaborado para explicar a aceitação de uma tecnologia por um indivíduo ou grupo [26]. Por ser comprovado e bem aceito em pesquisas científicas [27], ele foi escolhido para servir de base para esse trabalho.

Para isso elabora-se uma série de hipóteses a partir do modelo UTAUT e usa-se as respostas de um questionário para confirmá-las ou rejeitá-las através de fórmulas matemáticas.

Por meio da base teórica e segundo o modelo, existem quatro fatores independentes que poderão influenciar a aceitação de uma tecnologia: Expectativa de Desempenho (PE), Expectativa de Esforço (EE), Influência Social e

Condições Facilitadoras (FC), e dois fatores dependentes destes: Intenção de Uso (BI) e Comportamento de Uso (IC), conforme se vê na Fig. 1.

Seguindo esses fatores e para responder aos questionamentos da pesquisa, foram elaboradas hipóteses adaptadas para a aceitação e uso da Linguagem Natural.

- H1: A expectativa de um rendimento elevado leva os analistas a usar a Linguagem Natural na Especificação de Requisitos.
- H2: A expectativa de um baixo esforço leva os analistas a usar a Linguagem Natural na Especificação de Requisitos.
- H3: O uso de Linguagem Natural na Especificação de Requisitos leva a uma maior adesão às necessidades do cliente.
- H4: A organização onde o analista trabalha estimula o uso da Linguagem Natural na Especificação de Requisitos.

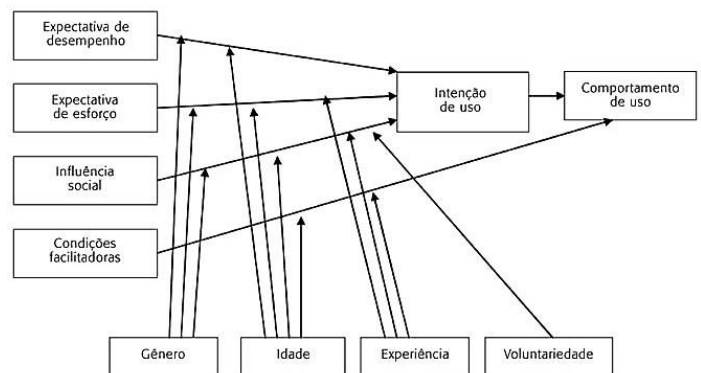


Fig. 1. O modelo UTAUT original [28]

O modelo UTAUT foi adaptado às hipóteses, obtendo-se o resultado apresentado na Fig. 2. A Influência Social foi descartada devido a que o uso da LN não é obrigatório [27].

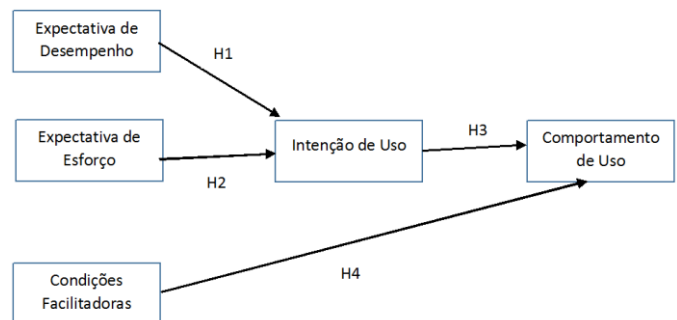


Fig. 2. Modelo UTAUT adaptado

O modelo é formado por variáveis que se apresentam em diversas ocasiões com aspectos subjetivos, necessitando de indicadores objetivos para realizar uma aproximação ao seu conceito. Deste modo, a técnica apropria-se de desenhos experimentais ou quase experimentais por meio de questionários ou bases de dados [27]. Assim, foi utilizado um questionário como input dos dados. O instrumento foi

adaptado do próprio modelo de UTAUT, com 17 questões relacionadas à expectativa de desempenho, expectativa de esforço, condições facilitadoras, intenção de uso e comportamento no uso da LN, objetivando identificar a aceitação e uso da LN nas Especificações de Requisitos pelos utilizadores dessa tecnologia (Quadro 1).

O questionário foi disponibilizado via *Google Docs* e enviado a profissionais que atuam com análise de requisitos em empresas do ramo de tecnologia (indústria de software) na Argentina e no Brasil [10]. Os participantes acessaram uma URL indicada e responderam a questões relacionadas indicando o grau de concordância com as afirmações expostas (uso da escala de Likert de 7 pontos): 1 para discordo inteiramente, 2 para discordo em grande parte, 3 para discordo parcialmente, 4 para neutro, 5 para concordo parcialmente, 6 para concordo em grande parte e 7 para concordo inteiramente.

QUADRO 1  
FATORES E INDICADORES DO MODELO UTAUT ADAPTADO À TECNOLOGIA LN NAS ESPECIFICAÇÕES DE REQUISITOS

Expectativa de Desempenho
PE1 - A linguagem natural é útil na especificação de requisitos.
PE2 - Utilizar a linguagem natural capacita-me a executar a especificação de requisitos mais rapidamente.
PE3 - Utilizar a linguagem natural aumenta minha produtividade na especificação de requisitos.
PE4 - Se eu uso a linguagem natural eu aumento minhas chances de ganhar uma promoção.
Expectativa de Esforço
EE1 - Minha interação com a linguagem natural é clara e compreensiva.
EE2 - É fácil tornar-se experiente em usar linguagem natural para especificar requisitos.
EE3 - É fácil usar a linguagem natural para especificar requisitos.
EE4V - É fácil aprender a usar a linguagem natural.
Condições Facilitadoras
FC1 - Eu tenho os recursos necessários para usar a linguagem natural na especificação de requisitos.
FC2 - Eu tenho o conhecimento necessário para usar a linguagem natural na especificação de requisitos.
FC3 - A linguagem natural não é compatível com outros métodos que eu uso para especificar requisitos.
FC4 - Uma pessoa específica (ou grupo) está disponível para prover-me assistência nas dificuldades com a linguagem natural.
Intenção de Uso da Linguagem Natural
BII - Eu pretendo continuar usando a linguagem natural para especificar requisitos no futuro.
BI2 - Eu sempre tento usar a linguagem natural na especificação de requisitos.
BI3 - Eu planejo continuar usando a linguagem natural frequentemente na especificação de requisitos.
Comportamento no uso da Linguagem Natural
IC1 - Eu já fiz uso da linguagem natural na especificação de requisitos.
IC2 - Eu uso a linguagem natural na especificação de requisitos com muita frequência.

#### IV. RESULTADOS E ANÁLISE

Os dados obtidos pelo questionário disponibilizado via *Google Docs* na Argentina, referem-se ao período de outubro e dezembro de 2016 e no Brasil [10], ao período de maio a outubro de 2015. Foram respondidos 45 questionários na Argentina e 50 no Brasil. A amostra na Argentina foi constituída em 69% por homens e 31% por mulheres, todos com idade média de 34 anos e 10 meses, e com uma

experiência em requisitos média de 7 anos e 6 meses. No Brasil, identificou-se que 66% dos analistas respondentes eram homens e 34% mulheres, todos com idade média de 37 anos e 3 meses. A amostra apresentava uma média de experiência com requisitos de 7 anos e 3 meses.

##### A. Argentina

O modelo proposto foi submetido a testes com a finalidade de dotá-lo de confiabilidade e validade no contexto Argentino. Após a coleta de dados, estes foram consolidados em uma planilha Excel. Foram usados os de confiabilidade simples, confiabilidade composta, variância explicada e validade discriminante, utilizando-se o software SmartPLS (*Smart Partial Least Square*) 2.0. O software é uma ferramenta para modelagem de equações estruturais [28] e para a análise de experimentos que exijam múltiplas variáveis ou mesmo relacionar influência direta e indireta de diversos fatores. Os modelos de equações estruturais utilizam modelos estatísticos de múltiplas variáveis para avaliar o grau de predição de variáveis independentes sobre as variáveis dependentes. Porém inicialmente é necessário saber se os itens referentes a cada variável estão correlacionados entre si (Tabela I).

TABELA I  
CONFIABILIDADE DE ITEM ANTES E DEPOIS DA DEPURAÇÃO - ARGENTINA

Variáveis	Itens	Confiabilidade de Item	Confiabilidade de Item depurado
Expectativa de Desempenho	PE1	0,56	<del>0,56</del>
	PE2	0,48	0,48
	PE3	0,90	0,91
	PE4	0,64	0,72
Expectativa de Esforço	EE1	0,84	0,89
	EE2	0,86	0,81
	EE3	0,65	<del>0,65</del>
	EE4	0,45	<del>0,45</del>
Condições Facilitadoras	FC1	0,99	1,00
	FC2	0,99	1,00
	FC3	-0,28	<del>-0,28</del>
	FC4	0,22	<del>0,22</del>
Intenção de Uso da Linguagem Natural	BI1	0,74	0,73
	BI2	0,88	0,88
	BI3	0,93	0,93
Utilidade na Especificação de Requisito	IC1	0,96	0,96
	IC2	0,97	0,97

Para assegurar a confiabilidade de item, os valores de correlação devem ser iguais ou superiores a 0,707 [28]. Pode-se observar que os indicadores PE1, PE2, EE3, EE4, FC3 e FC4 não foram confiáveis, assim foram depurados do modelo. A cada retirada de item os valores são recalculados.

Uma vez realizada a confiabilidade dos itens, os próximos testes a serem realizados são confiabilidade composta e os

testes de validade como variância média extraída AVE (*Average Variance Extracted*) e Validade Discriminante.

Para assegurar um valor aceitável para confiabilidade composta, é necessário que os valores sejam iguais ou superiores a 0,7 [28]. A aprovação deste teste garante que o grupo de itens de cada variável seja suficiente para explica-las. O primeiro teste de validade é o AVE, responsável por explicar o grau de conversão dos itens para sua respectiva variável e não para outra, pois um indicador que não possua ao menos 50% de diferença em relação aos indicadores de outras variáveis, pode ocasionar em um problema de validade. Assim, para valores de AVE satisfatórios, espera-se valores maiores ou iguais a 0,5 ou 50% [28]. Na Tabela II encontram-se os valores do modelo Argentino.

TABELA II  
CONFIABILIDADE COMPOSTA E AVE - ARGENTINA

Variáveis	Confiabilidade composta	AVE
Condições Facilitadoras	1,00	1,00
Expectativa de Desempenho	0,80	0,68
Expectativa de Esforço	0,85	0,74
Intenção de Uso da Linguagem Natural	0,89	0,73
Utilidade na Especificação de Requisitos	0,96	0,93

O último teste de validade é a Validade Discriminante que assegura que cada variável tenha independência em relação às demais. Para assegurar sua autonomia no modelo se deve garantir que a raiz quadrada do AVE seja superior as forças das correlações (Tabela III).

TABELA III  
VALIDADE DISCRIMINANTE – ARGENTINA

	Condições Facilitadoras	Expectativa de Desempenho	Expectativa de Esforço	Uso da Linguagem Natural
Condições Facilitadoras	<b>1</b>			
Expectativa de Desempenho	0,324569	<b>0,8216337</b>		
Expectativa de Esforço	0,468128	0,472066	<b>0,858419</b>	
Intenção de Uso da Linguagem Natural	0,498549	0,580291	0,485742	<b>0,851697</b>
Utilidade na Especificação de Requisitos	0,660265	0,299634	0,438285	0,589262

As raízes quadradas aparecem (Tabela III) em negrito e seu valor deve ser superior à força das correlações das demais variáveis que estão abaixo de cada raiz quadrada em negrito. Pode-se observar que as raízes quadradas de AVE foram superiores às cargas fatoriais dos itens, assim pode-se dizer que o modelo é válido e confiável, podendo ser usado em

pesquisas futuras que adotem o mesmo instrumento.

Os resultados permitem a validação do modelo proposto, permitindo avaliar as hipóteses propostas (Fig. 3).

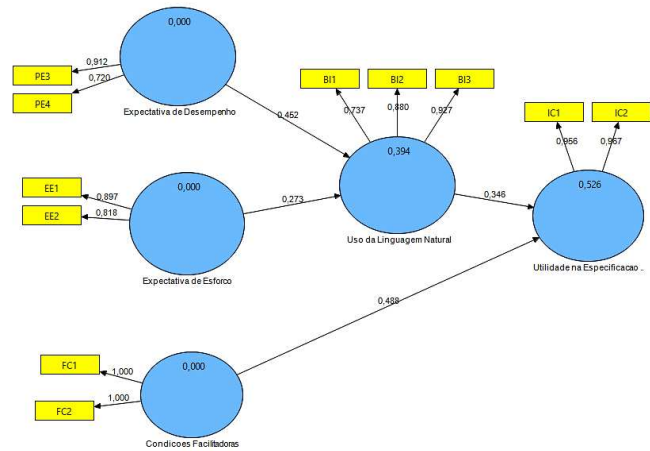


Fig. 3. Modelo de pesquisa depurado

Uma vez atestada a confiabilidade e validade do modelo, segue-se para a mensuração e resultados na análise. A valoração do modelo é realizada através da Análise do R2 e do Beta. O R2 explica em que grau a variável dependente é predita pela independente. E o Beta explica a força de influência de cada variável independente sobre e dependente.

Pode-se observar na Tabela IV que a Intenção de Uso da Linguagem Natural é explicada pela Expectativa de Esforço e pela Expectativa de Desempenho em um 39,4% (R2 = 0,394), enquanto a Utilidade na Especificação de requisitos foi predito pelas Condições Facilitadoras e da Intenção de Uso da Linguagem Natural, que é o problema central deste estudo, em 52,6% (R2 = 0,526).

TABELA IV  
VALORAÇÃO DO MODELO – ARGENTINA

Hipóteses	Beta	T Statistics	R2
H1-Expectativa de Desempenho → Intenção de Uso de Linguagem Natural	0,452	4,85	0,394
H2-Expectativa de Esforço → Intenção de Uso de Linguagem Natural	0,273	2,88	
H3-Intenção de Uso da Linguagem Natural → Utilidade na Especificação de Requisitos	0,346	3,66	0,526
H4-Condições Facilitadoras → Utilidade na Especificação de Requisitos	0,488	4,64	

Para que seja aceitável a predição deve ser de ao menos 10%, sendo reveladora acima de 30% [27]. Neste caso pode-se se dizer que o estudo tem uma predição reveladora.

Analisando os Betas, pode-se perceber (Tabela IV) a expectativa de desempenho (0,452) possui uma maior influência na Intenção de Uso da linguagem Natural do que a Expectativa de Esforço (0,273), e a Utilidade na Especificação de Requisitos é influenciada com mais força pelas Condições

facilitadoras (0,488) do que pela Utilidade na Especificação de Requisitos (0,346). Os Betas revelam o primeiro teste de hipótese e para que as hipóteses sejam válidas devem apresentar um valor igual ou superior a 0,2 [27].

Um segundo teste de hipóteses é o *T-Student*. Pode-se perceber que o teste *T-Student* (Tabela IV), são superiores a 1,96 [28], confirmando todas as hipóteses do modelo argentino como verdadeiras.

Assim pode-se perceber que a Utilidade na especificação de requisitos apenas é mais efetiva se existem condições facilitadoras para tal, que se pode traduzir desde apoio da alta direção da empresa, ou mesmo colaboração da equipe envolvida. Outro fator a ser levado em consideração é a expectativa do desempenho que impacta fortemente a intenção de uso da Linguagem Natural, ou seja, será usada se percebe uma melhoria potencial do desempenho. Sem embargo, todos os fatores se mostraram importantes, porém o maior grau de importância das condições facilitadoras e da expectativa de desempenho norteiam quais fatores devem ser trabalhados como prioridade. Assim todas as hipóteses foram aceitas como verdadeiras.

### B. Brasil

No ano anterior, um estudo foi realizado por parte da equipe, tendo como alvo o Brasil. A seguir apresentam-se sucintamente os resultados do Brasil [10]. A Tabela V apresenta o modelo mensurado elaborado para este país. Os critérios de confiabilidade e validade foram realizados e obtiveram os resultados desejados para o modelo brasileiro.

Pode-se observar na Tabela V que a LN é explicada pela Expectativa de Esforço e pela Expectativa de Desempenho em 14,7%, enquanto o Comportamento de Uso da LN na Especificação de Requisitos, problema central do estudo brasileiro, foi predita em 63,4% pelas variáveis Intenção de Uso da Linguagem Natural e Condições Facilitadoras. Para que seja aceitável a predição deve ser de ao menos 10%, sendo reveladora acima de 30% [28]. Neste caso pode-se se dizer que o estudo tem uma predição reveladora.

TABELA V  
VALORAÇÃO DO MODELO – BRASIL

Hipóteses	Beta	T Statistics	R2
H1-Expectativa de Desempenho → Intenção de Uso de Linguagem Natural	0,296	0,60	0,147
H2-Expectativa de Esforço → Intenção de Uso de Linguagem Natural	0,138	2,38	
H3-Intenção de Uso da Linguagem Natural → Utilidade na Especificação de Requisitos	0,776	0,92	0,634
H4-Condições Facilitadoras → Utilidade na Especificação de Requisitos	0,062	9,08	

Embora os valores de predição sejam satisfatórios, apenas duas hipóteses foram validadas parcialmente (H1 e H3), apresentando betas superiores a 0,2, porém falhando no teste *T-Student*.

### C. Comparação entre a Argentina e o Brasil

A seguir apresentam-se os graus de influência de cada variável independente em sua dependente. Esta análise é desvelada através dos valores de Beta. A literatura explica que para que as relações que se apresentam como hipóteses devem ter valor igual ou superior a 0,2 para serem significativas e apenas as relações significantes asseguram hipóteses verdadeiras [28].

Assim, para os dados argentinos, o grau de influência de cada variável nas predições (Beta) obtidas foram respectivamente de 0,468, de 0,254, de 0,346 e 0,488, podem dar confiabilidade à comprovação das hipóteses H1, H2, H3 e H4. Com relação aos dados brasileiros, o grau de influência de cada variável obtida foi respectivamente de 0,307, de 0,121, de 0,776 e de 0,062. Assim, para os dados brasileiros somente se pode dar confiabilidade à comprovação das hipóteses H1 e H3.

Os resultados dos dois países permitem constatar o reconhecimento por parte dos analistas argentinos e brasileiros que o uso da LN leva a maior adesão às necessidades dos clientes (H3). Usando LN é provável que a comunicação entre as partes se torne mais transparente, aumentando a qualidade dos requisitos. Isso ratifica os achados citados em outras pesquisas [15].

Os resultados permitem também identificar que há uma forte concordância dos analistas argentinos e brasileiros de que a Expectativa de Desempenho elevado conduz à Intenção de Uso de Linguagem Natural na especificação de requisitos (H1). Ou seja, com a melhor troca de informações entre as partes e, assim, evitando ruídos na comunicação é possível levantar os mesmos requisitos em menos tempo [19].

Com índices menores de concordância, mais ainda significativos, os dados argentinos permitem identificar que: a Expectativa de um baixo esforço leva os analistas a usar a Linguagem Natural na Especificação de Requisitos (H2); e que as organizações onde os analistas trabalham estimulam o uso da LN na especificação de requisitos (H4). Esses índices não são significativos nos dados brasileiros. Isso possibilita inferir que, talvez a especificação formal de modelos para os requisitos, nas organizações brasileiras, pode estar sendo preferida para retratar os mesmos conceitos em um nível mais alto de abstração, facilitando a visualização do problema e da sua solução, a rastreabilidade entre os diversos modelos e a aproximação com a fase de desenvolvimento do software [15].

Pode-se também mensurar o percentual de influência destas variáveis dos resultados argentinos e brasileiros multiplicando a o Beta da Variável Latente por suas Correlações (Tabela VI).

Nesta análise, com relação aos dados argentinos, verifica-se que as Condições Facilitadoras se destacaram, colaborando em 32,2% na predição da Intenção de Uso da Linguagem Natural, assim como a Expectativa de Desempenho influencia em 27,2% na Intenção de Uso da Linguagem Natural. No caso brasileiro, a Expectativa de Desempenho também influencia em 11,4% na intenção de uso da linguagem natural. Nos dados brasileiros a Utilidade na especificação é a variável que mais influencia (61,6%) a intenção de uso da linguagem natural.

TABELA VI  
PERCENTUAL DE INFLUÊNCIA DAS VARIÁVEIS ARGENTINA VS BRASIL

Variáveis	Argentina				Brasil			
	BI	IC	Correlação das variáveis latentes	% de Explicação	BI	IC	Correlação das variáveis latentes	% de Explicação
FC		0,488	0,6603	32,22%		0,062	0,283	1,8%
EE	0,273		0,4857	13,26%	0,121		0,280	3,4%
PE	0,452		0,5803	26,23%	0,307		0,370	12,4%
BI		0,346	0,5889	20,38%		0,776	0,794	61,6%

## V. CONCLUSÃO

O objetivo deste trabalho foi realizar um estudo comparativo sobre os fatores de influência para a intenção de uso da LN na especificação de requisitos considerando a percepção de técnicos de dois países: Argentina e o Brasil. A metodologia adotada foi um estudo descritivo, de abordagem quantitativa por meio de equações estruturais.

O modelo teórico proposto baseia-se na UTAUT. Foi elaborado um questionário, validado com  $\alpha=0,817$ , com 17 perguntas para 45 analistas de requisitos da Argentina e 50 analistas de requisitos do Brasil que trabalham na indústria de software.

O instrumento de pesquisa foi validado e pode continuar a ser usado.

Identificou-se que os motivos que levam os analistas dos dois países a utilizar a LN na especificação é a expectativa de um rendimento elevado (H1) e a percepção que o uso da LN leva a uma maior adesão às necessidades do cliente (H3). Dessa forma as Hipóteses 1 e 3 foram comprovadas para ambos os países. São elas:

- H1: A expectativa de um rendimento elevado leva os analistas a usar a Linguagem Natural na Especificação de Requisitos.
- H3: O uso de Linguagem Natural na Especificação de Requisitos leva a uma maior adesão às necessidades do cliente.
- As outras expectativas foram confirmadas com fatores de influência para um dos países participantes da pesquisa (Argentina). Ou seja, para a Argentina as Hipóteses 2 e 4 também foram confirmadas.

Destaca-se que essas hipóteses não foram confirmadas para o Brasil. O que permite inferir que, no Brasil, apesar dos analistas de requisitos reconhecerem a importância do uso da LN. Por outro lado, os mesmos profissionais não identificam condições facilitadoras para sua utilização mais efetiva nas organizações.

Essa constatação permite, portanto, verificar que, a despeito do embasamento técnico e conceitual fornecido pelas pesquisas nos últimos anos, é preciso a realização de investigações científicas voltadas o apoio organizacional sobre a utilização da LN na especificação dos requisitos. Estudos futuros poderiam apontar quais são as razões para esse

descompasso entre a visão técnica e a visão gerencial.

Sugere-se ainda, a expansão da pesquisa para outros países da América Latina de modo a se obter uma melhor compreensão desse mercado, conforme forem sendo comprovados ou ratificados os achados atuais.

## REFERÊNCIAS

- [1] E. J.R. de Castro, A.T.S. Calazans, R. A. Paldes, F. de A. Guimaraes, *Engenharia de Requisitos Um enfoque prático na construção de software orientado ao negócio*, Florianópolis: Bookess, 2014.
- [2] J. A. Dorigan, and R. M. de Barros, "A model of requirements engineering process for standardization and quality increase", in *Proc. of ADIS International Conference Applied Computing*, 2012, pp.343-347.
- [3] I. Sommerville, *Engenharia de software*, 9.ed., Addison-Wesley, 2011.
- [4] R.. G. Bustos, "Procesamiento de Lenguaje Natural en Ingeniería de Requisitos: Contribuciones Potenciales y Desafíos de Investigación", in *Proc. of CibSE 2015: Conferencia Iberoamericana de Software Engineering*, Lima, 2015, pp.1-3.
- [5] P. S. Lopes, "Uma taxonomia da pesquisa na área de Engenharia de Requisitos", Tese de Mestrado, Universidade de São Paulo, 2002.
- [6] E. Kroth, and G. Dessbesell, "Emprego de técnicas de representação do conhecimento como forma de apoio à engenharia de requisitos", in *Proc. of XVIII Congreso Argentino de Ciencias de la Computación*, 2012.
- [7] P. Ramírez-Correa, "Uso de internet móvil en Chile: explorando los antecedentes de su aceptación a nivel individual", *Ingeniare. Revista chilena de ingeniería*, vol. 22, no. 4, pp. 560-566, 2014.
- [8] J. Farias, P. Lins, P. Albuquerque, "A propensão de usuários à adoção de tecnologias: um estudo com usuários e não usuários do Progama Nota Legal no Distrito Federal", in *Proc. of XI Simpósio Brasileiro de Sistemas de Informação*, Goiânia, 2015.
- [9] M. Visentini, F. Chagas, D. Bobsin, "Análise Bibliométrica das Pesquisas sobre redes sociais virtuais publicadas em âmbito nacional", in *Proc. of SEPE-Seminário de Ensino, Pesquisa e Extensão da UFFS*, 2015.
- [10] R.A. Paldês, A.T.S. Calazans, A.M. Mariano, E.J.R.de Castro, "A utilização da linguagem natural na especificação de requisitos: um estudo por meio das equações estruturais", in *Proc. of XIX Ibero-American Conference on Software Engineering*, 2016, pp. 365-378.
- [11] SWEBOOK. *The Guide to the Software Engineering Body of Knowledge*, Los Alamitos, California: IEEE Computer Society, 2014.
- [12] B. Manrique Losada, and C. Zapata Jaramillo, "Transformación de lenguaje natural a controlado en la educación de requisitos: una síntesis conceptual basada en esquemas preconceptuales", *Revista Facultad de Ingeniería de Antioquia*, no. 70, pp.132-145, Jan/Mar 2014.
- [13] V. M. A. Lima, "Terminologia, comunicação e representação documentária", Tese de Mestrado, Universidade de São Paulo, 1998.
- [14] IEEE 29148-2011, IEEE Systems and software engineering - Life cycle processes - Requirements engineering. IEEE, Nova York, 2011.
- [15] M. Sayão, "Verificação e validação de requisitos: Processamento da Linguagem natural e agentes", Tese de Doutorado, Pontificia Universidade Católica do Rio de Janeiro, 2007.
- [16] A. Sutcliffe, and P. Sawyer, "Requirements Elicitation: Towards the Unknown Unknowns", in *Proc. of Research Track IEEE, RE 2013*, Rio de Janeiro, 2013.
- [17] J. M. C. Geal, J. Nicolas, J. L. F. Aleman, A. Toval, S. Ouhbi, and A. Idri, "Co-located and distributed natural-language requirements specification: traditional versus reuse-based techniques", *Journal of software: evolution and process*, vol. 28, no. 3, pp.205-227, 2016.
- [18] A. Oliveros, F. J. Danyans, and M. L. Mastropietro, "Práticas de Ingeniería de Requerimientos en el desarrollo de aplicaciones Web", in *Proc. of CIBSE 2014: track Workshop em Engenharia de Requisitos*, 2014.
- [19] S. Ghosh, D. Elenius, W. Li, P. Lincoln, N. Shankar, and W. Steiner, "ARSENAL: Automatic Requirements Specification Extraction from Natural Language", Cornell University Library, Jul., p.p.1-32, 2014.
- [20] H. Yang, , A. Roeck, V. Gervasi, A. Willis., and B. Nuseibeh, "Analysing Anaphoric Ambiguity in Natural Language Requirements", *Requirements Eng*, vol. 16, no. 3, pp. 163-189, May 2011.
- [21] P. Engiel, J. Pivatelli, P. Moura, R.. Portugal, and J. C. S. P. Leite, "Um processo colaborativo para a construção de léxicos: o caso da divulgação de transparência", in *Proc. of CibSE 2015: track XVIII Workshop on Requirements Engineering*, Lima, Perú, 2015.

- [22] H. Meth, and A. Maedche, “The State of the Art in Automated Requirements Elicitation”, *Information and Software Technology*, October 2013.
- [23] W. F. Tichy, and S. J. Koerner, “Text to software: developing tools to close the gaps in software engineering”, in *Proc. of the FSE/SDP Workshop on Future of Software Engineering Research (FoSER’10)*, 2010, pp. 379–384.
- [24] A. Casamayor, D. Godoy, and M. Campo, “Identification of non-functional requirements in textual specifications: a semi-supervised learning approach”, *Information and Software Technology*, vol. 52, no. 4, pp. 436–445, 2010.
- [25] J. Marôco, “Análise de Equações Estruturais: Fundamentos teóricos, software & aplicações”, *ReportNumber*, Pêro Pinheiro, 2010.
- [26] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, “User acceptance of information technology: Toward a unified view”, *MIS Quarterly*, vol. 27, no. 3, pp. 425–478, 2003.
- [27] P. E. Ramírez, A. M. Mariano, and E. A. Salazar, “Propuesta Metodológica para aplicar modelos de ecuaciones estructurales con PLS: El caso del uso de las bases de datos científicas en estudiantes universitarios”, *Revista ADMpg Gestão Estratégica*, vol. 7, no. 2, artículo 15, 2014.
- [28] W.W. Chin, “The partial least squares approach for structural equation modelling, en Methodology for Business and Management”, *Modern Methods for Business Research*, p.p. 295-336, Hillsdale: Lawrence Erlbaum Associates, 1998.



## Evolución de los Factores Situacionales durante el Proceso de Requisitos

Viviana A. Ledesma<sup>1</sup>, Graciela D.S. Hadad<sup>1,2</sup>, Jorge H. Doorn<sup>1,2</sup>, Juan P. Mighetti<sup>1</sup>,  
Nicolás A. Bedetti<sup>1</sup>, María Celia Elizalde<sup>2</sup>

<sup>1</sup>DIIT, Universidad Nacional de la Matanza

<sup>2</sup>Escuela de Informática, Universidad Nacional del Oeste

{vledesma, ghadad}@unlam.edu.ar, jdoorn@exa.unicen.edu.ar, {jmighetti, nbedetti}@unlam.edu.ar,  
melizalde@uno.edu.ar

### RESUMEN

La literatura presenta una diversidad de procesos de software, que dependen en gran medida del tipo de software a construir y de la envergadura del mismo. La selección del proceso más adecuado y la adaptación a un contexto específico suele ser a su vez fuertemente dependiente de la experiencia de los desarrolladores. Sin embargo, condicionar el análisis de los factores situacionales de esta manera es una restricción innecesaria. Por otra parte, muchos de estos factores impactan en el proceso de software desde sus etapas iniciales. Es por ello que es beneficioso realizar una adaptación lo más temprana posible, en función de factores observables en el contexto. Es decir, considerar los factores situacionales desde el proceso de requisitos mismo. A partir de varios casos reales donde se aplicó un proceso de requisitos, se inició la evaluación de dichos factores, observándose que algunos de ellos son difíciles de establecer con precisión en los primeros pasos del proceso o que por su naturaleza cambian a lo largo del proceso. Conocer qué factores son más proclives a evolucionar por mejora de la estimación inicial o por cambios genuinos permitirá una mejor adaptación del proceso de requisitos.

**Palabras clave:** Ingeniería de Requisitos, Proceso de Requisitos, Ingeniería de Métodos Situacional, Variabilidad de Procesos.

### CONTEXTO

La línea de investigación que se presenta es parte de los proyectos de investigación “Adaptación Dinámica de un Proceso de

Requisitos” del programa CYTMA2 de la Universidad Nacional de La Matanza (UNLaM) y “Tratamiento de los Factores Situacionales y la Completitud en la Ingeniería de Requisitos” de la Universidad Nacional del Oeste (UNO).

### 1. INTRODUCCIÓN

Implementar una gestión adecuada de los procesos permite a las organizaciones no solo reducir esfuerzos, sino además obtener ventajas competitivas frente a otras organizaciones. Esto no resulta sencillo especialmente cuando se trabaja en ambientes extremadamente dinámicos, los cuales conllevan un alto nivel de variabilidad. Es por ello que los constantes cambios que surgen en dichos entornos crean la necesidad de adaptar continuamente estos procesos. Adaptarse y soportar estos cambios resulta clave para el éxito de las organizaciones [1].

El análisis de variabilidad de procesos involucra establecer qué partes varían (puntos de variación), las causas de esa variación, las alternativas posibles de resolución (flujos variantes) y el mecanismo de resolución de la variabilidad (aplicando operaciones de adición, eliminación, reemplazo o parametrización de partes del proceso) [2, 3].

Es así que la variabilidad puede interpretarse como planear o anticiparse al cambio, en lugar de improvisar cada vez que surge una necesidad o una situación no prevista [3].

Existe un gran número de trabajos de investigación relacionados al análisis de la variabilidad, que ofrecen numerosas propuestas para dominios específicos. Un

campo de estudio es la representación de la variabilidad en los Modelos de Proceso de Negocio [2, 4]. Bajo este enfoque el análisis de la variabilidad consiste en la definición de caminos alternativos de ejecución en un flujo de trabajo, donde es posible indicar qué partes del proceso de negocio permanecen abiertos al cambio o no totalmente definidas, soportando diferentes versiones del mismo proceso, dependiendo del uso proyectado o el contexto de ejecución [5]. En las Líneas de Producto Software también se aplica la gestión de variabilidad [6]. Una técnica que la pone en práctica es la aproximación Base-Variación-Resolución, que propone la construcción de un modelo base, un modelo de variación y un modelo de resolución, para definir de forma explícita las partes del modelo que son fijas, las que pueden cambiar, y las condiciones que harán que éstas cambien [7]. Basándose en la visión de Líneas de Producto Software, se da sustento al concepto de Líneas de Proceso Software [2, 8], facilitando el ajuste sistemático del proceso, incluyendo la reutilización y evolución de forma planificada.

Por lo anterior, se hace evidente que la variabilidad es un factor clave que aplica a muchos aspectos del desarrollo de software en general [9]. En ese sentido, se han propuesto distintas técnicas para modelar la variabilidad que van desde diagramas de actividad de UML [10], hasta otros más específicos tales como la notación para el Modelado de Procesos de Negocio [11], la Especificación de Recursos Reutilizables [12] que define de manera estándar la información asociada a un activo o recurso, incluyendo las guías necesarias para facilitar la manipulación y reutilización del mismo, o el Metamodelo de Ingeniería de Procesos de Software (SPEM) [13] que especifica un lenguaje para modelar familias de procesos de software.

Desde otra visión del modelado de procesos, la Ingeniería de Métodos se enfoca en el diseño de métodos mediante la composición de fragmentos o bloques de proceso existentes, con el fin de mejorar la productividad de los mismos [14]. A partir de esta disciplina, surgió la Ingeniería de

Métodos Situacional (IMS), enfocada en el diseño de métodos de desarrollo de software adaptados a situaciones particulares [15, 16]. La situación se describe a través de la estimación de un conjunto de factores, en base a los cuales se determina la adaptación a realizar del proceso de software [17]. Estos factores muestran tanto peculiaridades del contexto de aplicación como del proyecto de software [18]. Por lo tanto, un proceso se construye para una situación dada a partir de la combinación de bloques de proceso predefinidos y en función de los factores situacionales identificados, donde existen bloques comunes a cualquier situación y otros variantes [16].

La IMS no solo considera el desarrollo de componentes de proceso sino también componentes de producto, e incluso componentes que ensamblan tanto proceso como producto (bloques mixtos) [19]. La ventaja que conlleva el enfoque de uso separado de bloques de proceso y bloques de producto sobre los bloques mixtos, es que el primero brinda una mayor flexibilidad en el proceso de adaptación, ya que varios bloques de proceso podrían asociarse a los mismos bloques de producto [15, 19].

En general, las técnicas que aplican la IMS coinciden en los siguientes pasos para la construcción del nuevo método:

- Caracterizar el proyecto, a partir de definir la situación.
- Seleccionar los fragmentos existentes que conformarán el método.
- Ensamblar los fragmentos de modo tal que se obtenga el método para ese proyecto específico.

En el proceso de IMS propuesto en [16], el ingeniero de métodos selecciona los bloques del proceso base y construye un proceso específico, en función de los factores situacionales y las guías de construcción existentes.

Han surgido algunas propuestas desde la IMS que abordan el diseño y mejora de procesos de Ingeniería de Requisitos (IR), adaptándolos a las características particulares de un proyecto. Algunas propuestas están orientadas a definir un proceso de IR

utilizando frameworks, ya sea que existan en el mercado o propuestos por los autores, partiendo de una base de conocimiento sobre actividades y técnicas de IR, que sirve de apoyo para obtener un proceso acorde a las necesidades del proyecto [20, 21]. Varias propuestas se enfocan exclusivamente en la selección de las técnicas de elicitación que mejor se adapten al contexto situacional [22, 23]. Otros autores presentan modelos para adaptar procesos de IR para proyectos de determinada naturaleza o alcance, tales como desarrollo de software global o métodos ágiles, tomando algunos principios de la IMS para flexibilizar el proceso, incluyendo otros atributos contextuales [24, 25]. En otros trabajos [26, 27] se han aplicado algunos principios de la IMS para definir procesos de IR adaptables, mediante el uso de componentes modulares existentes, incluso combinando nociones de Variabilidad de Procesos y de Líneas de Producto Software.

En estos trabajos, se aplican en mayor o menor medida algunos conceptos de la IMS, proponiendo factores situacionales que son evaluados para definir el contexto, donde algunos enfoques se basan en la reutilización de componentes para conformar el nuevo proceso de IR. Sin embargo, en general no presentan propuestas de adaptación dinámica del proceso. Una aproximación a ello es propuesto en [28], donde se enfatiza que el conocimiento acerca del dominio del problema y del dominio del proyecto se va adquiriendo a medida que avanza el proceso de IR, con lo cual en cada iteración se puede mejorar la actividad de elicitación y de esta forma se mejoraría el producto final. En este sentido, Rolland [19] menciona que las características que describen una situación pueden cambiar a lo largo del proceso, pudiendo ser necesario readaptar el proceso inicialmente construido para esa situación.

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

A lo largo de dos décadas en varios proyectos de investigación, se ha desarrollado un proceso de IR basado en modelos en lenguaje natural [29], que ha sido difundido

en cursos de grado y posgrado, y puesto en práctica en diversos proyectos de software de mediana a gran envergadura en la industria. Aún cuando este proceso puede considerarse suficientemente maduro, todavía requiere mejoras en algunos aspectos. Una mejora apunta a la flexibilización y optimización del proceso, por lo que se propone mejorar la aplicación de la IMS incorporando nociones de análisis de variabilidad de procesos, y utilizando el proceso existente y las estrategias ya desarrolladas para su adaptación estática, como base para encarar su adaptación dinámica. Esto involucra tomar decisiones referidas a qué artefactos de requisitos construir, qué actividades del proceso realizar y qué técnicas específicas aplicar en cada actividad, de manera de lograr una mayor productividad del proceso, y procurando una mejor calidad de los requisitos del software, considerando la posibilidad que los factores situacionales hayan evolucionado o hayan sido estimados incorrectamente. Entender las decisiones adoptadas en la aplicación de una variante específica del proceso de IR permitirá establecer mejoras en adaptaciones posteriores.

Manejar alternativas en el proceso de IR implica la existencia de bloques de proceso comunes independientes de la adaptación, bloques de proceso alternativos y otros bloques de proceso con alguna variabilidad interna menor. Se asume que la variabilidad en los procesos también podrá influir en los productos, pudiendo entonces tenerse modelos alternativos (por ejemplo, diferentes estructuras para el documento de especificación de requisitos) o familia de modelos (por ejemplo, un conjunto de escenarios vinculado a un glosario del universo de discurso, o vinculado a un glosario que describe términos del sistema de software, o sin vínculos a ningún glosario).

Asimismo, ocurre con frecuencia que las condiciones iniciales de un proyecto e incluso el contexto de aplicación se alteran, pudiéndose necesitar una nueva adaptación del proceso [19]. Es por ello que se propone una adaptación continua del proceso mediante

el monitoreo de los factores que pueden afectarlo en determinados hitos, proveyendo guías precisas para su reajuste.

### 3. RESULTADOS OBTENIDOS/ESPERADOS

A continuación se enumeran los resultados ya alcanzados: i) se ha identificado un conjunto amplio de 25 factores situacionales con sus interacciones y tipo de evolución, incluyendo factores no contemplados inicialmente relativos al desarrollo de software global y a la calidad esperada de los requisitos; ii) se han determinado los puntos de variación del proceso de requisitos y los factores que impactan en cada punto; iii) se han definido preliminarmente los bloques de proceso necesarios para componer el proceso; y iv) se han establecido las reglas de adaptación del proceso en cada punto de variación, las que permiten seleccionar los bloques que conformarán el proceso para la situación específica. Con base en estas definiciones, se diseñó una versión preliminar de los mecanismos de adaptación dinámica del proceso de requisitos, donde para cada proyecto de software se debe estimar un valor para cada factor situacional, el grado de certeza esperado en su estimación “a priori” y la posibilidad de reevaluarlo en etapas intermedias del proceso de requisitos. Estas definiciones han quedado parcialmente reflejadas en [30].

Se ha iniciado un estudio de la evolución de los factores situacionales durante la aplicación del proceso de IR, disponiéndose de 35 casos de estudio desarrollados entre 2015 y 2017. Se está desarrollando una herramienta de apoyo para la carga y posterior análisis de los cambios sufridos por los factores situacionales al inicio y al final del proceso de requisitos, combinado con el grado de confiabilidad asumido para el factor. En una etapa muy temprana del análisis, se ha notado que ciertos factores, como complejidad del contexto, grado de reingeniería del proceso de negocio, conflicto de intereses de usuarios y envergadura del proyecto, deben ser observados con especial atención durante el proceso de IR pues suelen

cambiar, mientras que factores como novedad del contexto y nivel de rotación del equipo de desarrollo suelen ser percibidos desde el inicio con mayor precisión.

En las próximas etapas de investigación, se profundizará el análisis de la evolución que sufren los factores situacionales a lo largo del proceso de requisitos desde su estimación inicial a su valoración final. De esta manera se intentará identificar el impacto que valoraciones dudosas en ciertos factores pueden tener en el proceso de requisitos adaptado. Se propone establecer alguna heurística que permita alcanzar un valor más preciso desde el inicio de la planeación de dicho proceso.

Por otro lado, se llevará a cabo la puesta a prueba de la adaptación del proceso de IR en aquellas situaciones consideradas más usuales (dada la diversidad de situaciones posibles) con el fin de comprobar el nivel de impacto alcanzado en la productividad del proceso.

### 4. FORMACIÓN DE RECURSOS HUMANOS

En el proyecto de UNLaM participan tres investigadores y un alumno becario. En el proyecto de UNO participan en este tema otros dos investigadores, uno en formación.

La línea de investigación presentada aquí es parte directa de la tesis de maestría “Estrategia de Requisitos adaptable según factores de situación” que está desarrollando la Ing. Ledesma en UNLaM. El Ing. Mighetti finalizó su tesis de maestría “Mitigación de amenazas a requisitos en el desarrollo global de software usando LEL y Escenarios” en UNLaM, faltando presentarla para su defensa; esta tesis ha permitido identificar factores situacionales no considerados inicialmente.

### 5. BIBLIOGRAFÍA

- [1] Wheelen, T.L., Hunger, J.D. (2012) Strategic Management and Business Policy: Toward Global Sustainability, 13° ed., Pearson.
- [2] Schnieders, A., Puhlmann, F. (2006) Variability Mechanisms in E-Business Process Families. Intl Conference on Business Information Systems, pp. 583-601.
- [3] Galster, M., Weyns, D., Tofan, D., Michalik, B., Avgeriou, P. (2014) Variability in Software Systems - A Systematic Literature Review. IEEE

- Transactions on Software Engineering, 40, 3, pp.282-306.
- [4] Groefsema, H., Bulanov, P., Aiello, M. (2011) Declarative Enhancement Framework for Business Processes. Intl Conference on Service-Oriented Computing, pp.495-504. New York: Springer.
- [5] Santos, E., Castro, J., Sánchez, J., Pastor, O. (2010) A Goal-Oriented Approach for Variability in BPMN. 13th Workshop on Requirements Engineering, pp. 17-28.
- [6] Pol'la, M., Buccella, A., Cechich, A., Arias, M. (2014) Un modelo de metadatos para la gestión de la variabilidad en líneas de productos de software. 43 JAIIO, Buenos Aires.
- [7] Haugen, Ø., Øgård, O. (2014) BVR – Better Variability Results. Intl Conference on System Analysis and Modeling: Models and Reusability. SAM 2014, Lecture Notes in Computer Science, vol. 8769, pp. 1-15. Springer.
- [8] Simidchieva, B., Clarke, L., Osterweil, L. (2007) Representing Process Variation with a Process Family. Software Process Dynamics and Agility. ICSP 2007. Lecture Notes in Computer Science, vol. 4470, pp. 109-120. Springer, Berlín.
- [9] Chen, L., Babar, M. (2010) Variability Management in Software Product Lines: An Investigation of Contemporary Industrial Challenges. 14th Intl Conference on Software Product Lines: going beyond, pp. 166-180. Springer-Verlag Berlin.
- [10] Razavian, M., Khosravi, R. (2008) Modeling Variability in Business Process Models Using UML. 5th Intl Conference on Information Technology: New Generations. EEUU: IEEE.
- [11] Delgado, A., Calegari, D. (2017) BPMN 2.0 Based Modeling and Customization of Variants in Business Process Families. XLIII CLEI, Córdoba.
- [12] Pacini, K., Braga, R. (2015) An Approach for Reusing Software Process Elements based on Reusable Asset Specification: A Software Product Line Case Study. 10th Intl Conference on Software Engineering Advances, pp. 200-206. Barcelona.
- [13] Martínez-Ruiz, T., García, F., Piattini, M., Munch, J. (2011) Modelling Software Process Variability: An Empirical Study. IET Software, 5,2, pp.172-187.
- [14] Brinkkemper, S. (1996) Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology, 38, 4, pp. 275-280.
- [15] Henderson-Sellers, B., Ralyté, J. (2010) Situational Method Engineering: State-of-the-Art Review. Journal of Universal Computer Science, 16, 3, pp. 424-478.
- [16] Henderson-Sellers, B., Ralyté, J., Agerfalk, P., Rossi, M. (2014) Situational Method Engineering, Springer-Verlag Berlin Heidelberg.
- [17] Khan, H.H., bin Mahrin, M., bt Chuprat, S. (2014) Factors for Tailoring Requirement Engineering Process: A Review. Intl Journal of Software Engineering and Technology, 1,1, pp.7-18.
- [18] Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R. (2007) Situational Method Engineering. Situational method engineering: fundamentals and experiences, Springer US, pp. 33-48.
- [19] Rolland, C. (2008) Method engineering: towards methods as services. Making Globally Distributed Software Development a Success Story, Springer Berlin, pp. 10-11.
- [20] Zowghi, D., Firesmith, D., Henderson-Sellers, B. (2005) Using the OPEN Process Framework to Produce a Situation-Specific Requirements Engineering Method. 1st Intl Workshop on Situational Requirements Engineering Processes, pp.59-74, Paris.
- [21] Jiang, L., Eberlein, A. (2008) A Framework for Requirements Engineering Process Development (FRERE). 19th Australian Conference on Software Engineering.
- [22] Hickey, A., Davis, A. (2003) Elicitation technique selection: how do experts do it? 11th IEEE Intl Requirements Engineering Conference, pp.169-178.
- [23] Carrizo, D., Dieste, O., Juristo, N. (2008) Study of elicitation techniques adequacy. 11th Workshop on Requirements Engineering, pp. 104-114.
- [24] Bakhat, K.A., Sarwar, A.A., Motla, Y.H.B., Akhtar, M.C. (2015) A Situational Requirement Engineering Model for an Agile Process. Bahria University Journal of Information & Communication Technology, 8, 1, pp. 21-26.
- [25] Khan, H., bin Mahrin, M., Mali, M. (2016) Situational Requirement Engineering Framework for Global Software Development: Formulation and Design. Bahria University Journal of Information & Communication Technologies, 9, 1, pp. 74-84.
- [26] Coulin, C., Zowghi, D., Sahraoui, A. (2006) A Situational Method Engineering Approach to Requirements Elicitation Workshops in the Software Development Process. Software Process: Improvement and Practice, 11, 5, pp. 451-464.
- [27] Jafarinezhad, O., Ramsin, R. (2012) Development of Situational Requirements Engineering Processes: A Process Factory Approach. 36th IEEE Intl Conference on Computer Software and Applications, pp.279-288.
- [28] Hickey, A.M., Davis, A.M. (2004) A Unified Model of Requirements Elicitation. Journal of Management Information Systems, 20, 4, pp.65-84.
- [29] Leite, J., Doorn, J., Kaplan, G., Hadad, G., Ridao, M. (2004) Defining System Context using Scenarios. Perspectives on Software Requirements, Kluwer Academic Publishers, cap.8, pp.169-199.
- [30] Hadad, G.D.S., Doorn, J.H., Ledesma, V.A. (2018) Dynamic Situational Adaptation of a Requirements Engineering Process. Encyclopedia of Information Science and Technology, 4º ed., IGI Global, EEUU, cap. 646, pp. 7422-7434.

## Revisión con base cognitiva de un Proceso de Requisitos

Graciela D.S. Hadad<sup>1,2</sup>, Jorge H. Doorn<sup>1,2</sup>, María C. Elizalde<sup>1</sup>

<sup>1</sup> Escuela de Informática, Universidad Nacional del Oeste

<sup>2</sup> DIIT, Universidad Nacional de la Matanza

ghadad@uno.edu.ar, jdoorn@exa.unicen.edu.ar, melizalde@uno.edu.ar

### RESUMEN

En todas las etapas del desarrollo de software, están presentes variadas tareas cognitivas, inmersas en el proceso mismo. Estas tareas cognitivas han sido poco estudiadas, aún cuando afectan el desempeño de las personas intervinientes en el proyecto. Si bien todas las actividades del desarrollo de un sistema de software involucran tareas cognitivas, las correspondientes a las primeras etapas tienen una mayor presencia de las mismas. Es decir, durante el proceso de requisitos, la colaboración, la comunicación y la comprensión del contexto donde se desempeñará el sistema son partes inherentes del mismo. Es por ello que se considera que la psicología cognitiva, y particularmente la ergonomía cognitiva, pueden ayudar a lograr una mejor comprensión de las actividades humanas que se llevan a cabo en esta etapa del desarrollo de software. En el presente proyecto se planifica estudiar, revisar, evaluar y eventualmente proponer modificaciones a las heurísticas de un proceso específico de requisitos, focalizando en aspectos cognitivos de los involucrados, de manera tal de mejorar el desarrollo de las actividades del proceso y, por ende, mejorar la calidad del producto final.

**Palabras clave:** Ingeniería de Requisitos, Proceso de Requisitos, Ergonomía Cognitiva, Elicitación, Lenguaje Natural.

### CONTEXTO

La línea de investigación que se presenta es parte de los proyectos de investigación “Tratamiento de los Factores Situacionales y la Completitud en la Ingeniería de Requisitos” de la Universidad Nacional del Oeste (UNO) y “Reflexividad como herramienta en la

Ingeniería de Requisitos” de la Universidad Nacional de La Matanza (UNLaM).

### 1. INTRODUCCIÓN

En muy diversas actividades de la Ingeniería de Software, en general, y de la Ingeniería de Requisitos, en particular, es altamente recomendable utilizar heurísticas, especialmente en aquellas relacionadas con modelos construidos manualmente o en forma parcialmente automatizada. Frecuentemente estas heurísticas son adaptadas, no siempre en forma planificada, a circunstancias particulares del contexto de cada proyecto de desarrollo, tanto por razones relacionadas con el proyecto como por características de los usuarios de las mismas [1]. Sin embargo, rara vez se las cuestiona de raíz, procurando concentrar la atención en el sustento conceptual de las mismas.

Una consecuencia inmediata de lo anterior es preguntar cómo es que han sido construidas esas heurísticas y, por supuesto, la pregunta más general acerca de cómo debería ser construida cualquier heurística.

La respuesta a la segunda pregunta tiene la apariencia de ser sumamente simple y justamente esa apariencia es el origen de su dificultad intrínseca. En virtud del alto contenido de información de la mayoría de los modelos utilizados en las diferentes actividades de la Ingeniería de Requisitos, resulta ser que las heurísticas deberían guiar al ser humano en el desarrollo de las actividades cognitivas involucradas.

La mera introducción de la palabra *cognitiva* hace evidente lo erróneo de considerar simple la construcción de una heurística en ésta y en muchas otras áreas de la Ingeniería de Software.

Abordar la planificación y ordenamiento

de cualquier actividad intelectual, aún aquellas con componentes rutinarios, es difícil ya que se deben considerar las fortalezas y debilidades del ser humano en relación con dichas actividades.

El estudio del desempeño del ser humano en entornos deportivos, laborales y situaciones de todo tipo tiene larga data, aunque en la mayoría de los casos orientados a problemáticas de actividades motrices y uso de herramientas. Estos estudios han dado lugar a la disciplina denominada *ergonomía* [2, 3], también conocida como *factores humanos*.

Posteriormente, los estudios relacionados con los factores humanos han ido adquiriendo un perfil cada vez más relacionado con aspectos sensoriales y cognitivos [4], habiéndose centrado principalmente en las interacciones con sistemas complejos [5]. La introducción de estos enfoques en los sistemas de procesamiento de datos se produjo en el área de visualización y diseño de interfaces [4, 6]. Aún hoy en día, la mayoría de estos estudios siguen estando en las cercanías de aspectos sensoriales y no verdaderamente cognitivos.

Todo lo anterior hace evidente que este es un terreno prácticamente inexplorado cuyas características son simultáneamente muy promisorias y desafiantes. Las promesas residen en el hecho que al poner en el centro del problema las peculiaridades de la cognición humana se avizoran importantes mejoras en la forma de abordar la elicitación de conocimiento y en su posterior elaboración y registro, mientras que los desafíos residen en que los aportes de las ciencias cognitivas tienen aún pocos resultados de importancia pragmática.

Como consecuencia, el presente proyecto de investigación tiene esencialmente características exploratorias [7, 8]. En otras palabras, se trata de elaborar hipótesis que permitan guiar futuros proyectos en los que las mismas sean confirmadas, mejoradas o refutadas. Usualmente se suele considerar que la definición de las heurísticas es una tarea sencilla y casi irrelevante. Definitivamente este no es el caso. En diversos proyectos

previos, se ha comprobado lo irreductible que son algunos de las debilidades conocidas en el proceso de requisitos, especialmente aquellas vinculadas con los problemas de completitud [9, 10, 11, 12].

Las cuatro fuentes básicas de hipótesis [7] son: i) marcos conceptuales o teóricos, ii) referencias bibliográficas, iii) investigaciones previas, y iv) intuiciones, sospechas y experiencias de la vida diaria. Se estima que en el presente proyecto se deberá recurrir a todas ellas.

La necesidad de tener en cuenta las limitaciones del ser humano en cuanto a su capacidad de comprensión y análisis de problemas de toda naturaleza ha estado presente en la informática desde etapas muy tempranas y sigue estando. Posiblemente el ejemplo más relevante en este sentido sea la fuerte recomendación de guiarse por enfoques top-down en diversas áreas y en utilizar el método de los refinamientos sucesivos en particular [13]. Pese a lo valioso de estas contribuciones es relevante destacar que en el artículo original de Niklaus Wirth no hay mención alguna a la necesidad de concentrar la atención del programador en unos pocos aspectos en cada momento del desarrollo. Tampoco se indica que se esté enfrentando la creación de un programa para atender una necesidad perfectamente comprendida y especificada.

Al considerar el traslado de estos enfoques al área de la Ingeniería de Requisitos, en su muy valiosa contribución [14], Michael Jackson enfáticamente aconseja no usar enfoques top-down en virtud que hacen necesario fragmentar el sistema a estudiar en el peor momento posible, que es cuando menos se lo conoce. Es decir, en el momento en que existe mayor probabilidad de cometer errores por la falta de comprensión acabada del problema. Nuevamente se encuentra aquí un sustento, algo más explícito, de atender las necesidades cognitivas de las personas intervinientes.

Otros ejemplos similares son el uso de abstracción mediante lenguajes de modelado del dominio [15] y la descomposición mediante diagramas de flujos de datos [16].

Se pueden agregar otros muchos y muy variados ejemplos, en los que se pueden observar innovaciones relacionadas con aspectos cognitivos. Sin embargo, en casi todos los casos, las cuestiones cognitivas tienen un rol de telón de fondo. Es decir, son poco analizadas y su influencia sobre las innovaciones introducidas no ha sido efectivamente probada.

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Una somera revisión de la literatura en lo que se refiere a las heurísticas que soportan fuertemente la gran mayoría de los métodos de la Ingeniería de Software muestra que las mismas están básicamente orientadas a describir el contenido de los modelos que a ayudar a elaborar conceptualmente ese contenido.

Si se analiza el Proceso Unificado en su flujo de trabajo Requisitos, éste propone la captura de requisitos centrándose en el modelado del dominio, del negocio y de los casos de uso, con heurísticas que atienden puramente a construirlos [17]. Por ejemplo, el modelo de dominio se describe mediante diagramas de clase UML cumpliendo con: “las clases del dominio aparecen como i) objetos del negocio que representan cosas que se manipulan en el negocio, ii) objetos del mundo real y conceptos que el sistema usará, y iii) sucesos que ocurrirán o han ocurrido en el entorno de trabajo” [17, pág.113]. Para la descripción de casos de uso, el Proceso Unificado menciona: “i) Debe definir el estado inicial como precondition, ii) Debe definir los posibles estados finales, iii) La interacción del sistema con los actores y qué cambios producen”, entre otros puntos [17, pág. 149]. En general, se observa que el Proceso Unificado no propone ninguna ayuda acerca de cómo seleccionar elementos del dominio, del negocio o del sistema, tales como clases, actores e interacciones relevantes, o como descartar lo superfluo.

En general, la literatura referida al Desarrollo Dirigido por Modelos (MDD) [18] considera que los dos primeros modelos CIM (Computational Independent Model) y PIM

(Platform Independent Model) en la cadena de transformación no pueden ser obtenidos en forma automatizada a partir de reglas de construcción (caso de CIM) ni de reglas de derivación del modelo previo (caso de PIM), sino que en ambos casos deben obtenerse a través de heurísticas, las que atienden aspectos netamente relacionados con el meta-modelo que los soporta sin guiar en cómo obtener el contenido. Es decir, en este paradigma aún no se han podido establecer reglas que automaticen la construcción de estos modelos ni que colaboren con la determinación de qué tipo de información del contexto de aplicación es necesario capturar, siendo considerada hoy en día una actividad “artesanal” [15]. Como declara Booch et al. [19] en su manifiesto de MDD solo se automatiza la transformación de modelos que no dependan de la “ingenuidad humana”.

Numerosos autores reportan que incrementa la probabilidad de éxito de un proyecto de software el estudio del dominio del problema y luego que los requisitos de ese sistema se plasmen en modelos creados en Lenguaje Natural [20, 21]. Las mejoras registradas en la calidad de los requisitos tienen su fundamento en que estas representaciones, tales como glosarios, casos de uso y escenarios, promueven la comunicación entre todos los involucrados en un proyecto de desarrollo de software y facilitan la validación de los requisitos elaborados. Una revisión hecha por Rolland et al. [22] muestra que, de doce enfoques propuestos en la literatura en el ámbito de la Ingeniería de Requisitos, todos ellos usan una notación de texto para describir escenarios, que en algunos casos se combinan con otros medios, tales como gráficos o imágenes. En un estudio posterior sobre la práctica en Ingeniería de Requisitos [23], se concluyó que el 51% de las organizaciones (sobre un total de 194) usa representaciones informales (por ejemplo, el lenguaje natural) y el 27% semi-formales, quedando los modelos formales relegados a un uso de apenas el 7% de las organizaciones.

En proyectos de investigación previos, se ha desarrollado una estrategia de IR basada en



modelos en lenguaje natural [24]. Aún cuando esta estrategia puede considerarse suficientemente madura, es deseable que la misma pueda ser mejorada en varios aspectos. Este proyecto está enmarcado en esa estrategia.

La estrategia procura alcanzar una profunda comprensión del dominio de la aplicación para definir una solución óptima a través de un sistema de software. Es decir, la estrategia presenta dos grandes etapas bien distinguibles: una de aprendizaje y la otra de definición. Cuando hay un conocimiento previo del dominio de la aplicación la primera fase se convierte en un proceso confirmatorio.

Los modelos que se utilizan en esta estrategia [24] son:

- un modelo léxico, LEL (Léxico Extendido del Lenguaje), el cual describe el vocabulario utilizado en el dominio de la aplicación,
- un modelo organizacional que describe los procesos actuales del negocio utilizando, denominado Escenarios Actuales,
- un modelo organizacional que describe los procesos del negocio proyectados en base al sistema de software a desarrollar, denominado Escenarios Futuros.

La mayoría de las heurísticas de esta estrategia de IR adolecen de las mismas debilidades que se han mencionado para el Proceso Unificado y para MDD, en el sentido que están fuertemente orientadas al contenido de los modelos que se construyen, con pocas pautas sobre cómo capturar y analizar la información a ser registrada en ellos.

### 3. RESULTADOS OBTENIDOS/ESPERADOS

En experiencias preliminares [25, 26] se han aplicado nociones de ergonomía cognitiva a la construcción del modelo Léxico Extendido del Lenguaje, poniendo especial énfasis en el uso de una heurística radicalmente diferente de la usual y orientada a facilitar la percepción del significado de los términos por parte del ingeniero de requisitos. Los resultados de este estudio fueron notoriamente prometedores, ya que se logró

una cantidad de términos notoriamente superior a la que se había obtenido anteriormente.

En la Tabla 1 se resumen los datos recogidos en una de estas experiencias [27]. Las omisiones indicadas en la misma están calculadas comparando el LEL obtenido con una versión anterior del mismo LEL creado por otros autores, habiendo sido ambas experiencias controladas en el sentido que ambos trabajos fueron realizados con el mismo material y con absoluto desconocimiento del otro trabajo y de la intención de hacer una comparación posterior.

**Tabla 1.** Omisiones en la versión de un LEL respecto a otra versión creada aplicando nociones cognitivas

Tipo de Término	Omisiones
Sujetos	17 %
Objetos	27 %
Verbos	62 %
Estados	44 %
Total	39 %

Es notable el déficit en la detección de los verbos con las heurísticas preexistentes. También se destaca que siempre se tuvo la suposición, poco fundada, que la mayor debilidad de la actividad de construcción del LEL residía en la identificación apropiada de los estados, hecho que parece no ser cierto.

Estos resultados son meramente indicativos, por dos razones: i) no tienen significación estadística, y ii) no se han establecido hipótesis que vinculen las visiones cognitivas con la heurística utilizada.

Tal como se mencionó en la introducción, desde el punto de vista epistemológico, el presente proyecto se puede catalogar como una investigación esencialmente exploratoria, en la que se aspira a: i) detectar las principales dificultades cognitivas que enfrenta el ingeniero de requisitos al elicitar y modelar el conocimiento que se adquiere del universo del discurso y al concebir el conjunto de servicios que deberá prestar el sistema de software a ser desarrollado, y ii) proponer nuevas heurísticas que no sólo describan los modelos a ser confeccionados sino que efectivamente contribuyan a facilitar la elaboración de la información que se registre en los mismos.

Obviamente, se espera que estos resultados puedan luego ser contrastados en investigaciones confirmatorias.

#### 4. FORMACIÓN DE RECURSOS HUMANOS

En el proyecto de la UNLaM participan tres investigadores, dos de ellos en formación. En el proyecto de la UNO participan en este tema otros dos investigadores, uno de ellos en formación.

La línea de investigación presentada aquí es parte directa de las tesis de maestría de la Ing. Renata Guatelli y de las tesis de doctorado de la Mg. Gladys Kaplan y la Ing. Andrea Vera, y colabora con la tesis de doctorado de la Lic. María Pepe.

#### 5. BIBLIOGRAFÍA

- [1] Hadad, G.D.S., Doorn, J.H., Ledesma, V.A. (2018) Dynamic Situational Adaptation of a Requirements Engineering Process, en: Encyclopedia of Information Science and Technology, Fourth Edition, IGI Global, cap. 646, pp. 7422-7434.
- [2] Murrell, K. (1965) Ergonomics Man in His Working Environment, Springer, pp 103-124.
- [3] Yong Zhou, Y., (1974) Applied Ergonomics Handbook, Elsevier, pp. 1-7.
- [4] van der Veer, G. (2008) Cognitive Ergonomics in Interface Design - Discussion of a Moving Science, Journal of Universal Computer Science, Vol. 1, N° 16, pp. 2614-2629.
- [5] Gersh, J.R., McKneely, J.A., Remington, R.W. (2005) Cognitive engineering: Understanding human interaction with complex systems, Johns Hopkins APL Technical Digest, Vol. 26, N° 4, pp. 377-382.
- [6] Kylie, M., Gomes, K.M., Riggs, S.L. (2017) Analyzing Visual Search Techniques using Eye Tracking for a Computerized Provider Order Entry (CPOE) Task, Human Factors and Ergonomics Society Annual Meeting, pp. 691-695.
- [7] Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, M.P. (2010) Metodología de la Investigación, 5ta. ed., Mc Graw Hill.
- [8] Popper, K. (1963) Conjectures and Refutations: The Growth of Scientific Knowledge, Routledge.
- [9] Doorn, J.H., Ridao, M.N. (2003) Completitud de Glosarios: Un estudio experimental, VI Workshop on Requirements Engineering, pp. 317-328.
- [10] Ridao, M.N., Doorn, J.H. (2006) Estimación de Completitud en Modelos de Requisitos Basados en Lenguaje Natural, XI Workshop on Requirements Engineering, pp. 146-157.
- [11] Doorn, J.H., Ridao, M.N. (2008) Completeness Concerns in Requirement Engineering, en: Encyclopedia of Information Science and Technology, Second Edition, IGI Global.
- [12] Hadad, G.D., Litvak, C. S., Doorn, J.H., Ridao, M. (2015) Dealing with Completeness in Requirements Engineering, en: Encyclopedia of Information Science and Technology, Third Edition, IGI Global.
- [13] Wirth, N. (1971) Program Development by Stepwise Refinement. Communications of the ACM, Vol. 14, N° 4, pp. 221-227.
- [14] Jackson, M., A. (1995) Software Requirements & Specifications: a lexicon of practice, principles and prejudices, Addison Wesley.
- [15] Pons, C., Giandini, R., Pérez, G., (2010) Desarrollo de Software Dirigido por Modelos: Conceptos teóricos y su aplicación práctica, Editorial de la Universidad Nacional de La Plata / McGraw-Hill Educación.
- [16] Jourdon, E. (1989) Modern Structured Analysis, Englewood Cliffs, NJ, Yourdon Press/Prentice Hall.
- [17] Jacobson, I., Booch, G., Rumbaugh, J., (2000) El Proceso Unificado de Desarrollo de Software, Pearson Educación, Madrid.
- [18] Karow, M., Gehlert, A. (2006) On the Transition from Computation Independent to Platform Independent Models, 12th American Conference on Information Systems, Mexico.
- [19] Booch, G., Brown, A., Iyengar, S., Rumbaugh, J., Selic, B. (2004) An MDA Manifesto, en Frankel, D. and Parodi J. (eds) The MDA Journal: Model Driven Architecture Straight from the Masters.
- [20] Ryan, K. (1993) The Role of Natural Language in Requirements Engineering, IEEE International Symposium on Requirements Engineering, San Diego, pp. 240-242.
- [21] Loucopoulos, P., Karakostas, V. (1995) System Requirements Engineering, McGraw-Hill, Londres.
- [22] Rolland, C., Souveyet, C, Ben Achour, C. (1989) Guiding Goal Modeling Using Scenarios, IEEE TSE, Vol. 24, N° 12, pp. 1055-1071.
- [23] Neill, C.J., Laplante, P.A. (2003) Requirements Engineering: The State of the Practice, IEEE Software, Vol.20, N° 6, pp.40-45.
- [24] Leite, J., Doorn, J., Kaplan, G., Hadad, G., Ridao, M. (2004) Defining System Context using Scenarios. Perspectives on Software Requirements, Kluwer Academic Publishers, cap.8, pp.169-199.
- [25] Litvak, C.S., Hadad, G.D.S., Doorn J.H. (2017) Nominalizations in Requirements Engineering Natural Language Models, en: Encyclopedia of Information Science and Technology, Fourth Edition, IGI Global.
- [26] Litvak, C.S., Hadad, G.D.S., Doorn, J.H. (2016) Procesamiento de Lenguaje Natural para Estudiar Completitud de Requisitos. Workshop de Investigadores en Ciencias de la Computación, pp. 498-502.
- [27] Litvak, C.S., Doorn J.H. (2016) Aporte del Procesamiento de Lenguaje Natural a la Ingeniería de Requisitos, 4to Congreso Nacional de Ingeniería Informática / Sistemas de Información, pp.463-467.

# Estimación de los Factores Situacionales del Proceso de Requisitos

Viviana A. Ledesma<sup>1,2</sup>, Graciela D. Hadad<sup>1,2</sup>, Jorge H. Doorn<sup>1,2</sup> y Nicolás A. Bedetti<sup>2</sup>

<sup>1</sup>Escuela de Informática, Universidad Nacional del Oeste, Argentina

<sup>2</sup>DIIT, Universidad Nacional de La Matanza, Argentina

vledesma@unlam.edu.ar; ghadad@uno.edu.ar;

jdoorn@exa.unicen.edu.ar; nbedetti@unlam.edu.ar

**Resumen.** Para obtener un producto software de buena calidad es vital aplicar un proceso apropiado de requisitos. En ese sentido, la literatura ofrece procesos de requisitos bien definidos de muy diverso alcance. Sin embargo, la experiencia indica que gran parte de los proyectos reales tienen características que hacen conveniente efectuar cambios en los procesos. Para ello, es necesario disponer de un método claro que guíe cómo adaptar el proceso en cada situación. Estas situaciones se pueden caracterizar a través de un conjunto de factores observables. Usualmente es difícil establecer con precisión el valor de estos factores al inicio del proceso de requisitos. También debe considerarse la posible evolución de ciertos factores por cambios en el contexto. En otras palabras, a veces no alcanza con solo adaptar un proceso antes de su puesta en ejecución, sino que este puede requerir readaptaciones durante su desarrollo, ya sea por precisarse factores con valoraciones iniciales dudosas o incorrectas, o por ocurrir cambios en el contexto que implican cambios en algunos factores. Finalmente, es necesario estimar cuán viable resulta establecer los factores con una confiabilidad tal que su aplicación para adaptar el proceso resulte aceptable. Por tal motivo, se ha estudiado la evolución de 21 factores durante la aplicación de un proceso de requisitos en 35 proyectos de la industria. Esto ha permitido corroborar la evolución de las situaciones en las que está inmerso un proceso de requisitos y, asimismo, identificar factores a los cuales debe prestarse mayor atención para una adecuada adaptación dinámica del proceso.

**Palabras Clave:** Proceso de Requisitos, Factores Situacionales, Variabilidad de Procesos, Ingeniería de Métodos Situacional.

## 1 Introducción

La literatura muestra que las técnicas de elicitación de conocimiento utilizadas en el proceso de Ingeniería de Requisitos (IR) se eligen a menudo en función de las características particulares del contexto de aplicación, principalmente para aquellas fuentes de información que involucran seres humanos [1]. Sin embargo, los factores situacionales que caracterizan esas particularidades, rara vez se toman en cuenta en otras actividades del proceso de requisitos. Sería conveniente que las peculiaridades, que

podrían obstaculizar o favorecer un proceso de IR, sean identificadas al comienzo del proyecto a fin de permitir su adaptación. Sin embargo, algunos factores pueden evolucionar a lo largo del ciclo de desarrollo del software lo que hace necesaria la readaptación del proceso en posteriores hitos estratégicos.

El objetivo del presente trabajo es estudiar la percepción de los ingenieros de requisitos respecto de la situación imperante en el proceso de IR, a través de la valoración del conjunto de factores que caracterizan dicha situación y la verosimilitud o estabilidad de dichas valoraciones a lo largo del proceso. Debe para ello tenerse en cuenta que si bien las evoluciones a lo largo del proyecto son generalmente impredecibles, las apreciaciones iniciales dudosas o incorrectas son potencialmente reducibles. Es entonces relevante la calidad con que es posible realizar la estimación de los factores situacionales, y en consecuencia la calidad de la adaptación del proceso de requisitos a su contexto.

En la sección siguiente se exponen conceptos sobre la variabilidad de procesos y enfoques cuyo conocimiento es necesario para crear o adaptar procesos de software, y específicamente procesos de requisitos, en función de los factores que caracterizan esa variabilidad. En la sección 3 se describe un conjunto de factores situacionales, sus interacciones y su tipo de evolución. En la sección 4 se presenta el estudio de caso llevado a cabo para ponderar la calidad de estimación de los factores situacionales y en la sección 5 se analizan los resultados obtenidos. Finalmente, se exponen conclusiones y futuros trabajos.

## 2 Adaptabilidad de Procesos

Uno de los desafíos que enfrentan las organizaciones tiene que ver con optimizar la gestión de sus procesos en entornos altamente dinámicos, por lo que disponer de procesos que sean flexibles y puedan ajustarse para soportar posibles cambios en ese entorno, resulta clave para su éxito [2]. La gestión de la variabilidad de procesos permite hacer frente a esta problemática, siendo aplicable en distintos áreas [3, 4, 5].

La variabilidad es la habilidad de un proceso o un producto de ser extendido, personalizado o configurado para ser utilizado bajo un determinado contexto [6]. Por ende, la variabilidad implica proyectar o adelantarse a los cambios, en vez de improvisar al surgir una necesidad o situación no prevista [7]. La gestión de la variabilidad involucra definir:

- los puntos de variación en el proceso, que corresponden a los momentos u oportunidades donde los elementos del proceso pueden cambiar;
- los posibles flujos variantes, que son las partes del proceso que representan la variabilidad y que serán implementados en los puntos de variación; y
- el mecanismo de variabilidad a usar, que puede ser: parametrización, adición, omisión, reemplazo de un elemento simple y variabilidad de tipo de dato.

Desde otra perspectiva, la Ingeniería de Métodos Situacional (IMS) tiene por objetivo construir métodos específicos que se ajusten a las situaciones particulares de su contexto, de modo flexible, adaptable, ágil y repetible; enfocándose en el diseño, construcción y adaptación de métodos, técnicas y herramientas de soporte para el desarrollo de sistemas de software [8, 9, 10].

Bajo esta disciplina, se entiende como *situación* a la combinación de circunstancias en un momento dado y en un contexto específico, que afecta la forma de trabajo y los productos a generar. Una situación debe poder describirse a partir de un conjunto de factores, que sirven de base para determinar la adaptación a realizar sobre el proceso de software [11]. Estos factores muestran particularidades del contexto de aplicación y de las características propias del proyecto de software [10].

Entonces, la forma más eficaz de adaptar un proceso consiste en fragmentar el mismo en bloques más pequeños, los que deben tener un objetivo propio, cumplible mediante una o más actividades, las que se realizan aplicando una o más técnicas incluidas en ese bloque y que pueden generar uno o más artefactos [8]. Algunos bloques son comunes a todas las situaciones, mientras que otros dependen de ciertos factores de situación. Por lo tanto, el proceso se conforma con la composición de los bloques apropiados para una situación particular [12].

Contrastando el enfoque de la IMS con el de variabilidad de procesos, se observa que ambos favorecen la adaptabilidad en base a características particulares y la reutilización. Sin embargo, la IMS presenta una visión más general, dirigida a la construcción de un proceso para una situación determinada a partir de bloques que existen o pueden ser creados, en cambio, la variabilidad parte de un proceso base ya existente y se avoca a establecer sus variaciones posibles o caminos alternativos en función de condiciones pre-definidas que pueden darse al aplicar dicho proceso.

En lo que respecta a la Ingeniería de Requisitos, se han generado varias propuestas orientadas a la creación y mejora de sus procesos, acorde a particularidades propias de cada proyecto. Estos trabajos se pueden catalogar en:

- A) *Orientados a definir un proceso de IR utilizando frameworks*. Se pone a disposición una base de conocimiento sobre actividades y técnicas de IR, que sirve de guía para obtener un proceso de IR acorde a las necesidades del proyecto [13, 14].
- B) *Enfoques para seleccionar técnicas de elicitación que mejor se adaptan al contexto situacional*. Estos trabajos solo contemplan la actividad de elicitación en el proceso de IR [1, 15, 16].
- C) *Modelos que permiten adaptar el proceso de IR para una clase específica de proyecto*. Se trata de propuestas para la adaptación situacional para proyectos de determinada naturaleza o alcance, por ejemplo, desarrollo global de software o métodos ágiles [17, 18].
- D) *Guías que facilitan la creación de un proceso de IR adaptado al contexto de aplicación*. Aportan mecanismos para la creación de un proceso de IR ajustado a las características situacionales del entorno que, a diferencia del grupo C, pueden aplicarse en proyectos de diferente naturaleza [19, 20].

En todos estos trabajos, se aplican en mayor o menor medida algunas nociones de la IMS, y se proponen algunos factores situacionales que son evaluados con el objetivo de definir el contexto. Particularmente, en los grupos A y D se han encontrado enfoques basados en el uso de distintos componentes para conformar el nuevo proceso de IR. Por su parte, solo en el grupo D existen propuestas que combinan los conceptos de la IMS con algunos principios de variabilidad de procesos.

El trabajo presentado por [15] (grupo B) plantea un enfoque parcialmente dinámico, destacando que el conocimiento acerca del dominio del problema y del dominio del proyecto se va adquiriendo a medida que avanza el proceso de IR, con lo cual en

cada iteración se pueden seleccionar técnicas de elicitación más apropiadas. Además, mencionan que las características situacionales del contexto del problema suelen ser estáticas mientras que las relacionadas con el proyecto pueden cambiar, aunque no presentan evidencia al respecto. En trabajos posteriores donde presentan una ontología con cerca de 50 características situacionales [16], no abordan el tema de la evolución de dichos factores, ni de la calidad con que estiman esos factores. En ese sentido, Rolland [9] también ha señalado la existencia de factores contingentes que cambian durante el ciclo de vida del proyecto, lo que debería acarrear una adaptación dinámica de los procesos.

En la gran mayoría de los estudios relacionados con la IMS no se abordan dos aspectos esenciales relacionados con: la adaptación incorrecta del proceso, y la probabilidad de evolución o de estimación incorrecta de los factores que impulsan esa adaptación. La primera cuestión puede ocurrir bajo 3 variantes: i) la omisión inapropiada de hacer una actividad que hubiera sido necesaria, ii) el reemplazo incorrecto de una actividad por otra, o iii) la inserción innecesaria de una actividad. Es claro que el último este aspecto es fuertemente dependiente del segundo, por lo que un estudio de la evolución y estimación de los factores situacionales puede revelar cuestiones no tan fácilmente visibles, que terminan en adaptaciones fallidas.

El trabajo de investigación en curso, cuya parte referida a la evolución y estimación de factores se presenta en este artículo, se enmarca en el grupo D, pues tiene por finalidad establecer mecanismos claros para adaptar un proceso de IR de acuerdo a las características particulares del contexto de aplicación y del proyecto, y ajustarlo en función de esa evolución, considerando que la misma puede provenir no solo de un cambio real en la situación sino también de una mejor comprensión de esa situación. Para ello, el proceso de adaptación propuesto realiza reevaluaciones de los factores en ciertos hitos del proceso de IR, establecidos como puntos de variación, guiando un reajuste del proceso de IR [21].

### 3 Factores Situacionales en un Proceso de Requisitos

Con base en la propuesta de la IMS, se han estudiado diversos factores considerados en la literatura [1, 20] para distintas actividades de la IR. Como resultado de ese estudio [22] y de la experiencia adquirida a lo largo de más de dos décadas en un proceso de IR basado en modelos en lenguaje natural [23], se identificaron 21 factores situacionales (ver Tabla 1). Debido a que es frecuente que las condiciones iniciales de un proyecto e incluso el contexto de aplicación sean mal percibidas o se alteren durante el proceso, es necesario prestar atención al tipo de evolución que dichos factores sufren. Desde esta dimensión pueden clasificarse en (ver tercera columna en Tabla 1):

- *Invariante*: aquel factor que, por su propia naturaleza, no tiene cambios durante el proceso de IR.
- *Contingente*: aquel factor que puede evolucionar naturalmente debido a cambios en el contexto de aplicación y/o en el contexto del proyecto.

Otro aspecto a considerar es el nivel de confianza de los ingenieros de requisitos respecto al valor asignado a cada factor situacional. Desde esta perspectiva, los valores de los factores se tipifican en:

- *Seguro*, cuando se considera un nivel alto de certeza al valor asignado al factor, lo que indicaría en principio que sólo podrían ocurrir cambios en aquellos catalogados como contingentes y sólo debido a cuestiones relacionadas con el contexto.
- *Razonablemente seguro*, cuando se considera que es baja la probabilidad de haberse cometido un error en su estimación.
- *Dudoso*, cuando no es posible establecer el valor del factor con precisión, sino recién cuando se logre una mejor comprensión de la situación.

**Tabla 1.** Factores Situacionales bajo estudio

Factor Situacional del Contexto de Aplicación	Valores Admisibles	Evolución
Complejidad del Contexto	Muy Alta/Alta/Media/Baja/Muy Baja	Invariante
Tipo de Cliente	Cliente Específico/Mercado Potencial	Invariante
Novedad del Contexto	Sí/No	Invariante
Reingeniería en el Proceso de Negocio	Muy Alta/Alta/Media/Baja/Muy Baja	Contingente
Volatilidad del Contexto	Muy Alta/Alta/Media/Baja/Muy Baja	Invariante
Volatilidad de Requerimientos del Cliente	Muy Alta/Alta/Media/Baja/Muy Baja	Contingente
Inconsistencias en el Contexto	Muy Alta/Alta/Media/Baja/Muy Baja	Invariante
Conflicto de Intereses de Usuarios	Muy Alta/Alta/Media/Baja/Muy Baja	Contingente
Rotación de Usuarios	Muy Alta/Alta/Media/Baja/Muy Baja	Invariante
Factor Situacional del Proyecto de Software	Valores Admisibles	Evolución
Conocimiento Previo del Dominio	Muy Alta/Alta/Media/Baja/Muy Baja	Contingente
Envergadura del Proyecto	Muy Alta/Alta/Media/Baja/Muy Baja	Contingente
Rotación del Equipo Desarrollador	Muy Alta/Alta/Media/Baja/Muy Baja	Invariante
Calidad Exigida del Software	Muy Alta/Alta/Media/Baja/Muy Baja	Invariante
Reuso de Artefactos de Requisitos Existentes	Sí/No	Invariante
Creación de Artefactos de Requisitos para Reuso	Sí/No	Invariante
Pre-Rastreabilidad de Requisitos	Sí/No	Invariante
Post-Rastreabilidad de Requisitos	Sí/No	Invariante
Granularidad de Rastreabilidad de Requisitos	Individual/Grupal	Invariante
Exigencia de Producir el Documento ERS	Sí/No	Invariante
Restricciones de Tiempo y Recursos del Proyecto	Muy Alta/Alta/Media/Baja/Muy Baja	Contingente
Modelo de Proceso de Software	Cascada/Incremental/Iterativo/Ágil	Invariante

Entonces, se debe evaluar cada factor y estimar su valor (ver Tabla 1), otorgándole un nivel de confianza. Esto es necesario ya que, en algunos casos, los factores pueden no conocerse de manera precisa. Aquellos factores cuyo nivel de confianza es *Dudoso* requieren ser observados a medida que el proceso avanza, con el objeto de establecer su valor con mayor precisión y, de ser necesario, re-adaptar el proceso de IR.

Por otra parte, algunos factores tienen influencia sobre otros. Si se considera que A y B son distintos factores situacionales, puede existir alguna de estas interacciones:

- *Interacción neutra*: indica que los factores A y B son totalmente independientes entre sí, no existe influencia entre ambos.
- *Interacción invalidante*: indica que si el factor A toma cierto valor no se debe tener en cuenta el factor B. Ejemplo: Si *Novedad del Contexto* es “Sí”, no se consideran los factores *Reingeniería en el Proceso de Negocio*, *Inconsistencias en el Contexto*, *Conflicto de Intereses de Usuarios* y *Rotación de Usuarios*.
- *Interacción limitante*: señala que si el factor A toma cierto valor entonces se restringen los valores que puede tomar el factor B. Ejemplo: Si *Volatilidad del contexto* es “Alta” o “Muy Alta”, *Modelo de Proceso de Software* no puede tomar el valor “Cascada”.

- *Interacción incompatible*: indica que A y B no pueden tomar a la vez ciertos valores, por lo que se requiere corregir la incompatibilidad mediante acuerdo entre los involucrados para modificar el valor de uno o todos los factores afectados. Ejemplo: Se debe negociar si *Restricciones de Tiempo y Recursos para el Proyecto* es “Alto” o “Muy Alto” y *Calidad Exigida del Software* es “Alta” o “Muy Alta”.

#### 4 Descripción del Estudio de Caso sobre Factores Situacionales

Son muy difundidos los peligros del estudio de caso como método para la comprensión de un problema [24] en virtud de sus características de ser intuitivo, primitivo y difícil de gestionar. Aunque parte de estos peligros no son tales, sino que provienen de la usual confusión entre los tipos de evidencias manipuladas (datos cualitativos o cuantitativos), los mecanismos de recolección de información y la estrategia de investigación (estudio de caso). Claramente, el estudio de caso puede combinarse con cualquier forma de evidencia y con cualquier mecanismo de recolección de información [25]. La diferencia entre un estudio de un caso y un experimento es que en el primero se examina un fenómeno en su contexto real, especialmente cuando dicho fenómeno no puede separarse fácilmente de su contexto o cuando esta interrelación es de interés, mientras que en el segundo se separa deliberadamente el fenómeno de su contexto.

La presencia combinada del fenómeno y su contexto genera la presencia de tantas variables que hace inviable el diseño de experimentos clásicos. Es por ello que en el presente trabajo se llevó a cabo el estudio de caso.

Este consistió en evaluar los factores situacionales usados para definir cómo ajustar el proceso de IR en una situación específica, en el contexto de personas que están adquiriendo conocimiento de tres dominios en forma concurrente; estos dominios son: i) el proceso de requisitos, ii) la estimación de los factores situacionales, y iii) un proceso de negocios de la vida real en el que se planifica incorporar un sistema de software. Este caso, aun con la gran imprecisión en la definición de sus límites, permite estimar la viabilidad de la aplicación de la IMS en problemas de la vida real y obtener información acerca de la validez de las suposiciones realizadas en forma preliminar acerca de la variabilidad de los factores situacionales a lo largo del proceso de IR.

La cantidad de factores situacionales, 21, y la cantidad de clases cualitativas posibles (ver Tabla 1) hacen que la cantidad de combinaciones, 1680, sea imposible de cubrir, aun con un ejemplar por combinación. Además, como los procesos del negocio utilizados fueron tomados estrictamente de la realidad, no resulta siquiera posible tener un ejemplar para cada posible valor de cada uno de los factores situacionales. Esto se muestra en la Tabla 2 en la que el factor situacional *Complejidad del Contexto* no tiene un solo ejemplar para los valores “Muy Alta” y “Muy Baja”. Pese a todas las limitaciones indicadas, el presente estudio trasciende la característica de exploratorio, siendo en todos sus aspectos, descriptivo y en mucho de ellos explicativo.

Entonces, se realizó el estudio sobre 35 proyectos para evaluar la evolución de los factores situacionales y el nivel de confianza asignado a cada valor en contextos de organizaciones reales para la producción de requisitos de software.

El estudio se llevó a cabo siguiendo el mismo proceso de requisitos [23] en todos



los proyectos, sin ser adaptado a la situación concreta de cada proyecto. Los proyectos se originaron en diferentes organizaciones del Área Metropolitana de Buenos Aires. Cada proyecto fue desarrollado por un equipo diferente, conformado por 4 a 6 alumnos de 4º año de la carrera de Ingeniería en Informática. Cada proyecto se completó durante un cuatrimestre (16 semanas) del curso de Ingeniería de Requisitos, entre 2015 y 2017. Los proyectos se desarrollaron estudiando el contexto de aplicación en cada organización y planteando una solución a través de un Documento de Especificación de Requisitos bajo el estándar IEEE Std 29148:2011. Como parte del proceso se construyeron tres tipos de modelos en lenguaje natural [23], aplicando técnicas de elicitación tradicionales, negociación, inspecciones, validaciones informales, versionado de modelos y rastreabilidad mediante matrices.

**Tabla 2.** Datos recolectados de tres Factores Situacionales

Proyecto	CUATRIMESTRE	AÑO	Complejidad del contexto				Volatilidad del contexto				Volatilidad de los requerimientos del cliente			
			VALORES		CONFIANZA		VALORES		CONFIANZA		VALORES		CONFIANZA	
			Inicial	Final	Inicial	Final	Inicial	Final	Inicial	Final	Inicial	Final	Inicial	Final
1	2	2015	Baja	Baja	Seguro	Seguro	Muy Baja	Muy Baja	Seguro	Seguro	Baja	Muy Baja	Seguro	Seguro
2	2	2015	Media	Baja	Seguro	Seguro	Baja	Baja	Seguro	Rz Seguro	Baja	Baja	Seguro	Dudoso
3	2	2015	Media	Media	Seguro	Seguro	Baja	Baja	Seguro	Seguro	Baja	Media	Seguro	Rz Seguro
4	2	2015	Baja	Baja	Seguro	Seguro	Baja	Muy Baja	Rz Seguro	Seguro	Baja	Baja	Rz Seguro	Seguro
5	2	2015	Media	Media	Rz Seguro	Seguro	Media	Muy Alta	Rz Seguro	Rz Seguro	Alta	Muy Alta	Rz Seguro	Seguro
6	2	2015	Media	Media	Rz Seguro	Seguro	Baja	Baja	Rz Seguro	Seguro	Baja	Baja	Rz Seguro	Seguro
7	2	2015	Baja	Media	Rz Seguro	Rz Seguro	Baja	Baja	Dudoso	Rz Seguro	Baja	Baja	Rz Seguro	Rz Seguro
8	2	2015	Baja	Baja	Seguro	Rz Seguro	Baja	Muy Baja	Seguro	Seguro	Media	Muy Baja	Rz Seguro	Rz Seguro
9	2	2015	Baja	Baja	Rz Seguro	Seguro	Media	Baja	Rz Seguro	Seguro	Media	Media	Dudoso	Seguro
10	2	2015	Media	Baja	Seguro	Seguro	Muy Baja	Baja	Seguro	Seguro	Muy Baja	Baja	Seguro	Seguro
11	2	2015	Baja	Baja	Rz Seguro	Seguro	Baja	Muy Baja	Seguro	Seguro	Media	Baja	Rz Seguro	Seguro
12	2	2015	Media	Media	Seguro	Seguro	Alta	Media	Rz Seguro	Rz Seguro	Baja	Baja	Rz Seguro	Rz Seguro
13	1	2016	Media	Baja	Rz Seguro	Seguro	Baja	Alta	Seguro	Seguro	Media	Alta	Seguro	Seguro
14	1	2016	Alta	Media	Seguro	Seguro	Baja	Baja	Rz Seguro	Seguro	Baja	Baja	Seguro	Seguro
15	1	2016	Alta	Media	Seguro	Seguro	Alta	Alta	Seguro	Rz Seguro	Alta	Baja	Rz Seguro	Seguro
16	1	2016	Media	Media	Rz Seguro	Seguro	Baja	Baja	Rz Seguro	Seguro	Baja	Media	Rz Seguro	Seguro
17	1	2016	Media	Media	Dudoso	Seguro	Muy Baja	Baja	Seguro	Seguro	Muy Baja	Baja	Seguro	Seguro
18	2	2016	Media	Media	Rz Seguro	Rz Seguro	Baja	Baja	Seguro	Rz Seguro	Media	Media	Seguro	Dudoso
19	2	2016	Media	Media	Rz Seguro	Seguro	Media	Baja	Rz Seguro	Rz Seguro	Alta	Baja	Rz Seguro	Seguro
20	2	2016	Media	Media	Seguro	Seguro	Muy Baja	Baja	Seguro	Rz Seguro	Baja	Baja	Rz Seguro	Seguro
21	2	2016	Media	Media	Rz Seguro	Seguro	Muy Baja	Baja	Seguro	Seguro	Baja	Baja	Seguro	Seguro
22	2	2016	Media	Baja	Rz Seguro	Seguro	Baja	Muy Baja	Seguro	Seguro	Baja	Baja	Seguro	Seguro
23	2	2016	Media	Media	Seguro	Rz Seguro	Baja	Baja	Rz Seguro	Seguro	Baja	Baja	Seguro	Seguro
24	2	2016	Baja	Media	Rz Seguro	Seguro	Baja	Muy Baja	Seguro	Seguro	Baja	Baja	Rz Seguro	Rz Seguro
25	1	2017	Media	Baja	Rz Seguro	Seguro	Muy Baja	Baja	Seguro	Seguro	Muy Baja	Baja	Seguro	Seguro
26	1	2017	Media	Media	Dudoso	Rz Seguro	Baja	Alta	Seguro	Dudoso	Media	Baja	Dudoso	Seguro
27	1	2017	Media	Media	Rz Seguro	Rz Seguro	Baja	Baja	Rz Seguro	Seguro	Baja	Baja	Dudoso	Rz Seguro
28	1	2017	Media	Media	Seguro	Rz Seguro	Baja	Baja	Rz Seguro	Rz Seguro	Media	Baja	Rz Seguro	Rz Seguro
29	1	2017	Media	Media	Dudoso	Rz Seguro	Baja	Media	Dudoso	Rz Seguro	Baja	Alta	Dudoso	Rz Seguro
30	2	2017	Media	Baja	Rz Seguro	Seguro	Muy Baja	Baja	Seguro	Rz Seguro	Baja	Baja	Rz Seguro	Rz Seguro
31	2	2017	Media	Media	Rz Seguro	Seguro	Muy Baja	Baja	Seguro	Rz Seguro	Baja	Baja	Seguro	Rz Seguro
32	2	2017	Alta	Media	Rz Seguro	Seguro	Baja	Muy Baja	Seguro	Seguro	Baja	Baja	Rz Seguro	Seguro
33	2	2017	Baja	Media	Rz Seguro	Seguro	Baja	Baja	Rz Seguro	Seguro	Baja	Baja	Seguro	Seguro
34	2	2017	Media	Baja	Seguro	Seguro	Muy Baja	Muy Baja	Seguro	Seguro	Baja	Muy Baja	Rz Seguro	Seguro
35	2	2017	Media	Media	Rz Seguro	Seguro	Muy Baja	Muy Baja	Seguro	Rz Seguro	Baja	Media	Seguro	Seguro

La recolección de datos se realizó en dos momentos distintos, donde cada equipo completó en cada instancia un formulario (ver Fig. 1), asignando valores a cada factor y calificando el valor asignado con un nivel de confianza. El formulario incluye los valores admisibles para cada factor y su reglas de interacción invalidante y limitante (de corresponder), anexo con una descripción de cada factor. La primera recolección tuvo lugar luego de que cada equipo realizó una o dos entrevistas abiertas para obtener una visión global del macrosistema en la organización correspondiente. Previamente, los equipos habían recibido instrucciones generales del proceso de IR a llevar a cabo. A medida que los equipos avanzaban en el proceso se fueron dando

explicaciones detalladas de cada etapa del mismo. La otra recolección de datos se realizó al finalizar el proceso de IR, donde nuevamente se completó el mismo formulario, sin tener acceso a los valores iniciales asignados. Cabe mencionar que los equipos no tuvieron conocimiento sobre el propósito de la toma de datos solicitada. La Tabla 2 muestra una planilla parcial con datos crudos de 3 factores situacionales obtenidos en las dos tomas de datos realizadas en los 35 proyectos.

FACTORES SITUACIONALES				FACTORES DEL PROYECTO			NIVEL DE CONFIANZA				
Proyecto: _____		Fecha: ___/___/___									
Responsable: _____											
FACTORES DEL CONTEXTO			NIVEL DE CONFIANZA								
			Seguro	Razonablemente Seguro	Dudoso						
1	Complejidad del contexto	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	10	Conocimiento previo del dominio	Muy alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Tipo de Cliente	Mercado Potencial <input type="checkbox"/> Cliente Específico <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11	Envergadura del proyecto	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Novedad del contexto	si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	12	Rotación del equipo desarrollador	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Reingeniería en el proceso de negocio (Completar sólo si Novedad del contexto es "No" y Tipo de cliente es "Cliente Específico")	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	13	Calidad exigida en el software	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Volatilidad del contexto	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	14	Reuso de artefactos de requisitos existentes	si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Volatilidad de los requerimientos del cliente	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	15	Creación de artefactos de requisitos para reuso	si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	Inconsistencias en el contexto (Completar sólo si Novedad del contexto es "No")	Muy alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16	Pre-Rastreabilidad de los requisitos	si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	Conflicto de intereses de usuarios (Completar sólo si Novedad del contexto es "No" y Tipo de cliente es "Cliente Específico")	Muy alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	17	Post-Rastreabilidad de los requisitos	si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	Rotación de usuarios (Completar sólo si Novedad del contexto es "No" y Tipo de cliente es "Cliente Específico")	Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	18	Granularidad de Rastreabilidad de los requisitos (Completar sólo si Pre-Rastreabilidad es "Si" o Post-Rastreabilidad es "Si")	Individual <input type="checkbox"/> Grupales <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
						19	Exigencia de producir el documento ERS	si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
						20	Restricciones de tiempo y recursos para el proyecto	Muy alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
						21	Modelo de proceso de software	Cascada <input type="checkbox"/> Incremental <input type="checkbox"/> Iterativo <input type="checkbox"/> Ágil <input type="checkbox"/> Otro <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 1. Formulario de Recolección de Datos de Factores Situacionales (Fuente: [21])

## 5 Análisis de Resultados

En un pre-proceso de los datos adquiridos, para cada factor se descartaron aquellos proyectos con datos incompletos en el valor inicial, el valor final, el nivel de confianza inicial o el nivel de confianza final. Esto se puede apreciar en la Tabla 2 con aquellos datos tachados, donde por ejemplo, para el factor *Volatilidad de los Requerimientos de los Clientes* se descartaron 2 proyectos. Se considera que deberá realizarse un estudio que permita analizar una posible explicación a los proyectos con omisiones en la información. Obviamente, se han considerado especialmente aquellas situaciones en las que debido a dependencias entre factores, carecía de sentido asignarle valor a alguno de ellos. Esto se evidencia en la Tabla 3, donde los factores con un alto porcentaje de proyectos inválidos (*Reingeniería en el Proceso de Negocio*, *Inconsistencias en el Contexto*, *Conflicto de Intereses de Usuarios*, *Rotación de Usuarios* y *Granularidad de Rastreabilidad de Requisitos*) son claramente aquellos factores dependientes de otros por relaciones del tipo invalidantes.

Se utilizaron técnicas de la estadística descriptiva, para el análisis de la valoración de factores, habiéndose obtenido los totales por cada valor posible inicial y final, la cantidad de valores que cambiaron y la cantidad de valores que cambiaron a valores no próximos. Esto último significa que la valoración "Muy Alta" o "Alta" dada a un factor es considerada en el mismo rango de valores y, por lo tanto, no se los considera como un cambio relevante; lo mismo ocurre entre "Baja" y "Muy Baja". Esto no aplica a valores binarios (ejemplo: *Tipo de Cliente*) o valores nominales no ordinales

(ejemplo: *Modelo de Proceso de Software*).

En base a los valores finales de los factores, estos se pueden resumir en los siguientes aspectos que caracterizaran los proyectos:

- El 66% de los proyectos fueron de complejidad media y el resto de complejidad baja a muy baja.
- El 49% de los proyectos eran de mediana envergadura y el otro 49% de pequeña envergadura, con un solo proyecto catalogado de gran envergadura.
- El 82% tuvieron una volatilidad del contexto entre baja y muy baja, solo 4 proyectos consideraron volatilidad alta o muy alta.
- El 76% de los proyectos eran con baja/muy baja inconsistencia del contexto.
- El 81% percibió una baja/muy baja rotación de usuarios.
- El 76% de los proyectos tuvieron bajo/muy bajo conocimiento previo del dominio relativo a la organización, se presentó un solo proyecto con alto conocimiento.
- El 63% indicó baja/muy baja rotación del equipo y los restantes un nivel medio.
- El 53% percibió un nivel alto/muy alto en la calidad necesaria del software y un 44% calidad media, donde un solo proyecto calificó como baja la calidad exigida.
- El 64% aplicaron baja/muy baja reingeniería en el proceso de negocio, con un solo proyecto que aplicó alta reingeniería; el resto aplicó una reingeniería media.
- El 76% informó una baja/muy baja volatilidad en los requerimientos de los clientes, con un solo proyecto con alta volatilidad.
- El 89% de los proyectos tuvieron un nivel bajo/muy bajo de conflicto de intereses, sin ningún proyecto con alta conflictividad.
- El 62% consideró que no se requería crear artefactos de requisitos para reuso, aunque el restante 38% estimó que sí.
- El 50% consideró que las restricciones impuestas de tiempo (16 semanas) y recursos (cantidad de miembros del equipo) eran medias, mientras que de los restantes proyectos la mitad consideró que eran limitaciones altas y la otra mitad bajas.

Cada proyecto tenía un cliente específico dado que se desarrollaba para una organización real. Sin embargo, 4 proyectos inicialmente evaluaron el factor *Tipo de Cliente* incorrectamente, y posteriormente 3 de ellos lo rectificaron. Aunque todos los proyectos eran para contextos existentes, hubo un equipo que percibió incorrectamente que el desarrollo era para un contexto nuevo y lo calificó además como seguro.

Aunque en todos los proyectos se elaboró el Documento de Especificación de Requisitos, en 6 proyectos (18%) no percibieron tal exigencia. Ningún proyecto hizo reuso de artefactos dado que no se disponía de un repositorio de artefactos de requisitos; sin embargo, el 20% de los proyectos calificó incorrectamente a este factor.

Los proyectos debían manejar *Pre-rastreabilidad* entre los artefactos construidos durante el proceso. El 15% no percibió la realización de esta actividad como tal, aun cuando fue realizada. En la preparación no se dieron indicaciones sobre la *Post-rastreabilidad* dado que era una tarea no involucrada en el proceso de IR, por lo que los resultados de este factor se descartaron para el análisis. Respecto a la *Granularidad de Rastreabilidad de los Requisitos*, esta se llevó a cabo en todos los proyectos en forma individual por requisito. Debe tenerse en cuenta que este factor depende de los otros dos factores relacionados con rastreabilidad (interacción invalidante), por lo que

hubo solo 13 proyectos con valores iniciales y finales y 14 proyectos sin valor inicial por su dependencia de los otros dos factores, constituyéndose efectivamente en 27 proyectos considerados válidos, de los cuales el 48% percibió un manejo de la rastreabilidad individual.

Respecto al factor *Modelo de Proceso de Software*, el tema no fue considerado, pues solo se llevó a cabo el proceso de IR hasta obtener el conjunto más completo posible de requisitos, habiendo iteraciones entre las etapas, ya sea por correcciones o por mejor comprensión del macrosistema. Por tanto, se descartó el análisis de este factor, aunque por los resultados se observa que no fue bien comprendido.

Por otro lado, se hizo una evaluación del nivel de confianza con el que se calificó a los valores asignados a cada factor. Para ello, se calcularon el total de proyectos con apreciaciones seguras, razonablemente seguras y dudosas para los valores iniciales y finales de cada factor, junto con la cantidad de factores que mantuvieron su calificación de confianza (discriminando entre los que se mantuvieron en seguros o razonablemente seguros y los que se mantuvieron dudosos), los que mejoraron (por ejemplo: de dudoso a razonablemente seguro) y los que empeoraron (discriminando si empeoraron a razonablemente seguro o a dudoso). Estos datos se presentan en la Tabla 3, donde se observa que 2 factores invariantes tuvieron la mayor cantidad de proyectos que mantuvieron su nivel de confianza en seguro o razonablemente seguro: *Tipo de Cliente* y *Novedad del Contexto*, mientras que los 3 factores que en mayor medida mejoraron el nivel de confianza también son invariantes: *Complejidad del Contexto*, *Creación de Artefactos para Reuso* y *Pre-Rastreabilidad de Requisitos*.

**Tabla 3.** Resultados de Nivel de Confianza por Factor, siendo S: Seguro, RS: Razonablemente Seguro y D: Dudoso.

Factores Situacionales	Proyectos Incompletos	% Proyectos Incompletos	Total Proyectos Válidos	Nivel de Confianza										
				Inicial			Final			Mejoró	Se mantuvo en		Empeoró a ...	
				S	RS	D	S	RS	D		S/RS	D	RS	D
Complejidad del contexto	0	0%	35	13	19	3	27	8	0	54%	37%	0%	9%	0%
Tipo de cliente	0	0%	35	26	7	2	29	6	0	17%	74%	0%	9%	0%
Novedad del contexto	0	0%	35	28	7	0	27	8	0	11%	74%	0%	14%	0%
Reingeniería en el proceso de negocio	7	20%	28	11	17	0	20	8	0	43%	46%	0%	11%	0%
Volatilidad del contexto	1	3%	34	21	11	2	20	13	1	26%	50%	0%	21%	3%
Volatilidad de los requerimientos del cliente	2	6%	33	15	14	4	23	8	2	42%	45%	0%	6%	6%
Inconsistencias en el contexto	6	17%	29	11	13	5	18	8	3	45%	41%	0%	3%	10%
Conflicto de intereses de usuarios	8	23%	27	16	9	2	23	3	1	33%	59%	0%	4%	4%
Rotación de usuarios	9	26%	26	18	6	2	19	6	1	23%	58%	0%	15%	4%
Conocimiento previo del dominio	1	3%	34	24	10	0	26	6	2	24%	53%	0%	18%	6%
Envergadura del proyecto	0	0%	35	13	19	3	22	10	3	40%	37%	0%	14%	9%
Rotación del equipo desarrollador	0	0%	35	18	10	7	19	13	3	34%	37%	6%	20%	3%
Calidad exigida en el software	1	3%	34	14	16	4	22	10	2	35%	53%	6%	6%	0%
Reuso de artefactos de requisitos existentes	0	0%	35	21	6	8	28	7	0	37%	51%	0%	11%	0%
Creación de artefactos de requisitos para reuso	1	3%	34	13	16	5	23	9	2	50%	32%	0%	12%	6%
Pre-Rastreabilidad de los requisitos	1	3%	34	10	15	9	20	12	2	50%	32%	6%	12%	0%
Post-Rastreabilidad de los requisitos	1	3%	34	12	13	9	16	14	4	41%	32%	9%	15%	3%
Granularidad de rastreabilidad de requisitos	22	63%	13	3	6	4	6	6	1	46%	38%	8%	8%	0%
Exigencia de producir el documento ERS	2	6%	33	18	10	5	28	4	1	42%	52%	0%	3%	3%
Restricciones de tiempo y recursos del proyecto	1	3%	34	13	16	5	17	16	1	35%	44%	3%	18%	0%
Modelo de proceso de software	1	3%	34	11	19	4	22	11	1	44%	44%	3%	9%	0%

Asimismo, se estudiaron los cambios de nivel de confianza sobre aquellos valores que sufrieron cambios relevantes (ver Tabla 4). En otras palabras, por factor se calcularon las cantidades de proyectos cuya valoración inicial fue percibida dudosa, razonablemente segura o segura, para los proyectos con cambios a valores no próximos.

**Tabla 4.** Resultados de la Evolución de Valores y de Nivel de Confianza

Factores Situacionales	Total de Proyectos Válidos	% Nivel de Confianza modificado	% Dudosos al inicio	% Dudosos al final	Análisis sobre Proyectos con Cambios Relevantes				
					Proyectos con Cambios Relevantes	% Valores con Cambios Relevantes	% Dudosos al inicio	% Seguros al inicio	% Raz. Seguros al inicio
Complejidad del contexto	35	63%	9%	0%	13	37%	0%	38%	62%
Tipo de cliente	35	26%	6%	0%	3	9%	67%	0%	33%
Novedad del contexto	35	26%	0%	0%	4	11%	0%	50%	50%
Reingeniería en el proceso de negocio	28	54%	0%	0%	15	54%	0%	33%	67%
Volatilidad del contexto	34	50%	6%	3%	6	18%	17%	33%	50%
Volatilidad de los requerimientos del cliente	33	55%	12%	6%	11	33%	18%	27%	55%
Inconsistencias en el contexto	29	59%	17%	10%	8	28%	13%	50%	38%
Conflicto de intereses de usuarios	27	41%	7%	4%	4	15%	25%	25%	50%
Rotación de usuarios	26	42%	8%	4%	7	27%	14%	71%	14%
Conocimiento previo del dominio	34	47%	0%	6%	12	35%	0%	67%	33%
Envergadura del proyecto	35	63%	9%	9%	16	46%	6%	38%	56%
Rotación del equipo desarrollador	35	57%	20%	9%	14	40%	14%	64%	21%
Calidad exigida en el software	34	41%	12%	6%	18	53%	17%	28%	56%
Reuso de artefactos de requisitos existentes	35	49%	23%	0%	12	34%	33%	50%	17%
Creación de artefactos de requisitos para reuso	34	68%	15%	6%	14	41%	21%	43%	36%
Pre-Rastreabilidad de los requisitos	34	62%	26%	6%	15	44%	20%	33%	47%
Post-Rastreabilidad de los requisitos	34	59%	26%	12%	12	35%	17%	33%	50%
Granularidad de rastreabilidad de requisitos	13	54%	31%	8%	3	23%	33%	33%	33%
Exigencia de producir el documento ERS	33	48%	15%	3%	10	30%	20%	60%	20%
Restricciones de tiempo y recursos del proyecto	34	53%	15%	3%	13	38%	8%	46%	46%
Modelo de proceso de software	34	53%	12%	3%	13	38%	8%	23%	69%

Hubo 13 factores situacionales con más de un 50% de proyectos que modificaron su nivel de confianza (ver columna “% Nivel de Confianza Modificado” en Tabla 4), siendo esto independiente del tipo de evolución del factor, pues 6 de esos factores eran invariantes y para los que se presumía una mejor apreciación inicial. Esto evidencia la dificultad en establecer valores confiables al inicio del proceso de IR. Por otro lado, hubo algunos factores, en una escasa cantidad de proyectos, cuya valoración final fue dudosa, los que en su mayoría tenían números altos en cantidad de cambios en el nivel de confianza, tales como: *Inconsistencias en el Contexto* y *Envergadura del Proyecto*.

En cuanto a cambios relevantes en la valoración de los factores (ver columnas en color en Tabla 4), hubo 2 factores donde más del 50% de los proyectos cambiaron su valor inicial: *Reingeniería en el Proceso de Negocio* y *Calidad Exigida del Software*, siendo el primero un factor contingente y el segundo invariante. Para el primer factor, esto podría implicar la necesidad de prestarle una mayor atención al momento de valorarlo, mientras que para el segundo factor es posible que su significado fuese ambiguo o no fuera clara la manera de lograr un conocimiento sobre el mismo. Otros 2 factores invariantes tuvieron también porcentajes altos de cambio, donde la *Envergadura del Proyecto* puede asociarse a la misma problemática del factor *Reingeniería en el Proceso de Negocio*, y la *Pre-Rastreabilidad de Requisitos* puede tener la misma dificultad de valoración que *Calidad Exigida del Software*. En estos 4 factores el nivel de confianza dado inicialmente a los valores que luego cambiaron fue “Razonablemente Seguro” en un porcentaje muy elevado.

Además, se corrobora que los factores con mayor porcentaje de nivel de confianza inicial “Seguro” respecto a cambios relevantes en sus valores, tuvieron efectivamente baja cantidad de cambios relevantes (ver en Tabla 4), tal es el caso de *Rotación de*

**Análisis del Rigor de la Aplicación del Estudio de Caso.** El fenómeno bajo estudio consistió en un conjunto de factores situacionales y la calidad con que se los estima en el transcurso de un proceso de requisitos desarrollado en organizaciones reales, con el propósito de describir y en lo posible explicar esas calidades y evoluciones, para poder luego refinar una estrategia de adaptación dinámica del proceso de IR. Es decir, se estudió el fenómeno en su contexto natural. Este estudio se basó en múltiples fuentes de evidencia: 35 proyectos, cuyos datos se analizaron transversalmente para cada factor, logrando una razonable confiabilidad. La descripción presentada de los pasos seguidos y bajo qué condiciones se realizó la recolección de datos permitirá la realización de repeticiones, contribuyendo a la confiabilidad experimental. Respecto a la validez de construcción, los datos recolectados se han mostrado parcialmente en la Tabla 2, basado en la toma de datos a través de formularios estandarizados para todos los proyectos. Respecto a la validez interna, el análisis de los datos se presentan en las Tablas 3 y 4, donde se describe la evolución de los valores de los factores y del nivel de confianza de los mismos en porcentajes de proyectos considerados. Los equipos fueron homogéneos, recibiendo el mismo nivel de capacitación para llevar a cabo su proyecto. Respecto a la validez externa, se planeó el conjunto de factores a estudiar en base a trabajos propios [22] y de otros autores en IMS y Variabilidad de Procesos. Además, debe considerarse que se puso especial cuidado en la elección de los proyectos involucrados. Este estudio de caso ha permitido tener una mejor comprensión de la evolución de los factores y su interrelación, y de los riesgos de cometer errores en su estimación y qué factores son más sensibles a esos riesgos. Esto permitirá realizar mejoras a las guías de adaptación del proceso de IR en dos aspectos: en cuanto a la heurística de evaluación y análisis de los factores situacionales en cada punto de variación y en cuanto a la descripción específica de cada factor.

## 6 Conclusiones y Futuros Trabajos

La adaptación situacional de los procesos depende fuertemente de la calidad de los factores situacionales, pues la eliminación, reemplazo o adición de una actividad es función directa de los valores que se le asignan a dichos factores. Esa asignación es en realidad una estimación que debe ser lo más aproximada posible a la situación efectivamente imperante. Por otro lado, dependiendo de la naturaleza del proceso y de la adaptación que se trate, la eliminación, reemplazo o adición innecesaria de una actividad puede tener una importancia diferente, sin embargo, es claro que en condiciones similares, la primera de ellas es mucho más perniciosa que las otras dos. Además, esta adaptación puede verse comprometida cuando estos factores evolucionan y ello no es percibido oportunamente. Una valoración imprecisa es perniciosa y la evolución de la situación puede agravar la calidad de esa valoración. A pesar de la variedad de estudios relacionados con la IMS, estos aspectos no son tratados con la profundidad necesaria, lo que resulta en una carencia grave.

A partir de este estudio de caso se ha podido observar una alta variabilidad en la

estimación de los factores situacionales. Se han identificado factores que sufrieron cambios relevantes en más del 40% de los proyectos. Además, la gran mayoría de los factores (exceptuando solo 2) tuvo arriba del 40% de los proyectos con cambios en el nivel de confianza de la estimación del factor. Estos resultados no pueden generalizarse; solo describen cuestiones no evidentes, aunque acotadas a un conjunto de proyectos de pequeña a mediana envergadura, de una complejidad baja a media y desarrollados por equipos con poca experiencia en IR.

En resumen, este estudio permitió establecer con evidencia observable aquellos factores situacionales que cambian más frecuentemente o son mal estimados en el proceso específico de IR puesto en práctica. Como consecuencia, se les debe prestar mayor atención antes y durante la definición y re-adaptación del proceso, teniendo en consideración que la valoración de algunos factores además impacta en la valoración de otros y, por ende, puede desencadenar otro tipo de ajustes en el proceso.

Del análisis de los resultados del estudio de caso presentado, se observa la necesidad de disponer de un proceso de requisitos que se ajuste dinámicamente durante su desarrollo. Gran parte de estas adaptaciones a lo largo del proceso se deben principalmente a cambios por apreciaciones imprecisas de la situación. Se ha podido detectar que las valoraciones inciertas o erróneas de estos factores se deben en gran parte a la falta de comprensión del contexto y, en menor medida, al desconocimiento de algunas actividades de la IR o dificultad de interpretación de las definiciones de algunos factores. Respecto a la primera causa, se propone definir una heurística que ayude a una valoración temprana menos dudosa y/o que mida las consecuencias de una adaptación imprecisa y permita redefinir transitoriamente esos valores iniciales para una adaptación del proceso que mitigue futuros cambios. Para una mejor comprensión del significado de los factores situacionales, se mejorará la definición de los mismos, tratando de reducir ambigüedades y ofreciendo precisiones sobre sus posibles valores.

## Referencias

1. Carrizo, D., Dieste, O., Juristo, N.: Study of elicitation techniques adequacy. En: 11th Workshop on Requirements Engineering, pp. 104-114. Barcelona, España (2008).
2. Wheelen, T.L., Hunger, J.D.: Strategic Management and Business Policy: Toward Global Sustainability, 13<sup>o</sup> ed., Pearson (2012).
3. Cetina, C., Giner, P., Fons, J., Pelechano, V.: Autonomic Computing Through Reuse of Variability Models at Runtime: The Case of Smart Homes. *Computer*, IEEE Computer Society, 42(10), 37-43 (2009).
4. Simidchieva, B., Clarke, L., Osterweil, L.: Representing Process Variation with a Process Family. *Software Process Dynamics and Agility. ICSP 2007. Lecture Notes in Computer Science*, vol. 4470, pp. 109-120. Springer, Berlín (2007).
5. Santos, E., Castro, J., Sánchez, J., Pastor, O.: A Goal-Oriented Approach for Variability in BPMN. 13th Workshop on Requirements Engineering, pp.17-28. Cuenca, Ecuador (2010).
6. Hallerbach, A., Bauer, T., Reichert, M.: Managing Process Variants in the Process Lifecycle. 10th Intl Conference on Enterprise Information Systems, pp.154-161. España (2008).
7. Galster, M., Weyns, D., Tofan, D., Michalik, B., Avgeriou, P.: Variability in Software Systems - A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 40(3), 282-306 (2014).

8. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. *Information and Software Technology*, 38(4), 275-280 (1996).
9. Rolland, C.: Method Engineering: Towards Methods as Services. Making Globally Distributed Software Development a Success Story. En: *Intl Conference on Software Process (ICSP 2008)*, pp.10–11. Leipzig, Alemania (2008).
10. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: *Situational Method Engineering. Situational method engineering: fundamentals and experiences*, Springer US, pp. 33-48 (2007).
11. Khan, H.H., bin Mahrin, M., bt Chuprat, S.: Factors for Tailoring Requirement Engineering Process: A Review. *Intl Journal of Software Engineering and Technology*, 1(1), 7-18 (2014).
12. Henderson-Sellers, B., Ralyté, J.: *Situational Method Engineering: State-of-the-Art Review*. *Journal of Universal Computer Science*, 16(3), 424-478 (2010).
13. Zowghi, D., Firesmith, D., Henderson-Sellers, B.: Using the OPEN Process Framework to Produce a Situation-Specific Requirements Engineering Method. En: *1st Intl Workshop on Situational Requirements Engineering Processes*, pp.59-74. París, Francia (2005).
14. Jiang, L., Eberlein, A.: A Framework for Requirements Engineering Process Development (FRERE). *19th Australian Conference on Software Engineering*. Perth, Australia (2008).
15. Hickey, A.M., Davis, A.M.: A Unified Model of Requirements Elicitation. *Journal of Management Information Systems*, 20(4), 65-84 (2004).
16. Hickey A.M., Davis A.M.: An Ontological Approach to Requirements Elicitation Technique Selection. En: Sharman R., Kishore R., Ramesh R. (eds) *Ontologies. Integrated Series in Information Systems*, vol. 14. Springer, Boston, MA (2007).
17. Bakhat, K.A., Sarwar, A.A., Motla, Y.H.B., Akhtar, M.C.: A Situational Requirement Engineering Model for an Agile Process. *Bahria University Journal of Information & Communication Technology*, 8(1), 21-26 (2015).
18. Khan, H., bin Mahrin, M., Mali, M.: *Situational Requirement Engineering Framework for Global Software Development: Formulation and Design*. *Bahria University Journal of Information & Communication Technologies*, 9(1), 74-84 (2016).
19. Coulin, C., Zowghi, D., Sahraoui, A.: A Situational Method Engineering Approach to Requirements Elicitation Workshops in the Software Development Process. *Software Process: Improvement and Practice*, 11(5), 451-464 (2006).
20. Jafarinezhad, O., Ramsin, R.: Development of Situational Requirements Engineering Processes: A Process Factory Approach. En: *36th IEEE Intl Conference on Computer Software and Applications*, pp. 279-288 (2012).
21. Ledesma, V.A., Hadad, G.D.S., Doorn, J.H.: Adaptación Dinámica de un Proceso de Requisitos Orientado al Cliente. En: *4to Congreso Nacional de Ingeniería Informática / Sistemas de Información*. Salta, Argentina. ISSN: 2347-0372 (2016).
22. Hadad, G.D.S., Ledesma, V.A., Doorn, J.H.: Proceso de Requisitos Adaptable a Factores Situacionales. En: *XVI Workshop de Investigadores en Ciencias de la Computación*, pp. 448-452. Ushuaia, Argentina (2014).
23. Leite, J.C.S.P., Doorn, J.H., Kaplan, G.N., Hadad, G.D.S., Ridao, M.: *Defining System Context using Scenarios. Perspectives on Software Requirements*, Kluwer Academic Publishers, cap.8, pp.169-199 (2004).
24. Kyburz-Graber, R.: Does case-study methodology lack rigour? The need for quality criteria for sound case-study research, as illustrated by a recent case in secondary and higher education. *Journal of Environmental Education Research*, 10(1), 53-65 (2004).
25. Yin, R. K.: The case study crisis: some answers. *Administrative Science Quarterly*, 26(1), 58-65 (1981).



# Título del Proyecto: Adaptación Dinámica de un Proceso de Requisitos

## Resumen Extendido correspondiente al Informe Final

**Director:** Graciela Dora Susana Hadad (ghadad@unlam.edu.ar)

**Co-Director:** Juan Pablo Mighetti (jmighetti@unlam.edu.ar)

**Integrantes:**

**Investigador**

Viviana Alejandra Ledesma

**Becario**

Nicolás Alfredo Bedetti

**Resumen:** Para obtener software de buena calidad es vital realizar las actividades apropiadas del proceso de requisitos, por tal motivo, la literatura presenta variedad de procesos específicos y bien definidos. Por otro lado, la experiencia indica que cada proyecto tiene características particulares, las que pueden ser definidas a partir de factores situacionales observables, relacionados con el contexto de aplicación y con el proyecto en sí mismo. Para un tratamiento adecuado de estas características, es recomendable efectuar adaptaciones en el proceso, siendo por ende necesario disponer de un método claro que guíe las decisiones sobre cómo ajustar el proceso y los modelos a construir. Al inicio del proceso de requisitos no es sencillo percibir las particularidades que cada proyecto presenta. A medida que avanza el proceso, suele ocurrir que algunos factores evolucionan y otras se perciben con mayor precisión. Mediante un estudio de caso se pudo analizar claramente la evolución y estimación de estos factores situacionales. En el presente proyecto de investigación se ha definido un proceso de adaptación dinámica de un proceso de requisitos en hitos preestablecidos y en base a un conjunto de factores, considerando la evolución de estos y sus interacciones.

**Palabras clave:** Ingeniería de Requisitos, Proceso de Requisitos, Ingeniería de Métodos Situacional, Variabilidad de Procesos

**Abstract:** To obtain good quality software it is essential to carry out the appropriate activities of the requirements process. Therefore, the literature presents a variety of specific and well-defined processes. On the other hand, experience indicates that each project has its particular characteristics, which can be defined from observable situational factors related to the application context and to the project itself. To give an adequate treatment of these characteristics is advisable to make adaptations in the process, being therefore necessary to have a clear method to guide the decisions on how to adjust the process and the models to build. At the beginning of the requirements process it is not easy to perceive the particularities that each project presents. As the process progresses, it usually happens that some factors evolve and others are perceived with greater precision. Through a case study, the evolution and estimation of these situational factors could be clearly analyzed. The present research project has allowed to define a dynamic adaptation process of a requirements process at pre-established milestones and based on a set of situational factors, considering their evolution and their interactions.

**Keywords:** Requirements Engineering, Requirements Process, Situational Method Engineering, Process Variability

**Introducción**

Cuando se trabaja en el desarrollo de un producto es importante contar con un proceso bien definido, lo cual implica disponer de una serie de actividades predefinidas, con técnicas a aplicar, entradas y salidas, que en su conjunto ayuden a conseguir un resultado de alta calidad. La primera actividad de estos procesos consiste en definir el resultado esperado. Cuando el producto es un sistema de software, la actividad inicial es un proceso de Ingeniería de Requisitos (IR) cuyo resultado es la especificación de requisitos.

En algunas circunstancias los gerentes de proyecto deben considerar la posibilidad de acortar el proceso de IR mediante la simplificación u omisión de algunas actividades, mientras que en otras pueden surgir diferentes alternativas, tales como la elección de ciertas técnicas de elicitación, modelado o validación, inclusive puede ser oportuno agregar algunas otras actividades. Sería deseable que estas decisiones sobre las elecciones, refinamientos o ajustes al proceso puedan tomarse en una forma consciente y planificada para cada proyecto particular [HEN10].

Se propone definir mecanismos para la adaptación de una estrategia de IR basada en modelos escritos en lenguaje natural [HAD08], de acuerdo con la situación bajo la que se desarrollará el proyecto de software. Esta propuesta se apoya en dos ejes conceptuales: la Variabilidad de Procesos y la Ingeniería de Métodos Situacional (IMS). El primero se refiere a la habilidad de un proceso de ser extendido, personalizado o configurado para ser reutilizado bajo un determinado contexto, que involucra identificar qué puede variar, porqué y cuáles son las variantes posibles [SCH06] [GAL14]. El otro eje, la IMS se dedica esencialmente a la construcción de métodos adaptados a situaciones concretas para el desarrollo de software [BRI96]. Desde este enfoque, la adaptación de un proceso de software se basa en diversos factores que en conjunto definen una situación [KHA14].

La adaptación propuesta del proceso de IR involucra un aspecto poco mencionado en la literatura que es considerar la evolución de las características particulares del contexto de aplicación y del proyecto, y ajustarlo en función de la evolución de la situación, considerando que la misma puede provenir no solo de un cambio real en la situación sino también de una mejor comprensión de esa situación. Para ello, el proceso de adaptación propuesto realiza reevaluaciones de los factores en ciertos hitos del proceso de IR, establecidos como puntos de variación, guiando un reajuste del proceso de IR.

## **Problemática a Resolver**

Uno de los desafíos a los que se enfrentan las organizaciones tiene que ver con mejorar la gestión de sus procesos, disponer de procesos que sean flexibles, que puedan ajustarse para soportar posibles cambios resulta clave para su éxito [WHE12]. Es por ello que la adaptación de cualquier proceso a situaciones específicas se considera una buena práctica en la mayoría de las disciplinas.

Se observa en la literatura un gran número de trabajos de investigación relacionados a la gestión de la variabilidad, con propuestas para dominios específicos, como por ejemplo en computación autónoma, líneas de producto software, líneas de proceso y modelado de procesos de negocio [SCH06] [GAL14]. Los distintos trabajos que estudian este paradigma coinciden en que el análisis de la variabilidad implica definir:

- los puntos de variación en el proceso, es decir, dónde el proceso puede cambiar;
- los posibles flujos variantes, que son las partes del proceso que representan la variabilidad e implementan los puntos de variación; y

- el mecanismo de variabilidad a usar para la resolución, que puede ser: parametrización, adición, omisión, reemplazo y variabilidad de tipo de dato.

Por consiguiente, adaptar sistemáticamente un proceso comprende identificar adecuadamente, por un lado, las posibles alternativas de variación del proceso y, por otro, los aspectos del contexto que determinan la selección apropiada de estas variaciones. La variabilidad implica proyectar o adelantarse a los cambios, en vez de improvisar cada vez que surge una necesidad o una situación no habitual [GAL14]. En consecuencia, es evidente que la variabilidad puede aplicarse a la mayoría, sino a todos, los aspectos del desarrollo de software. Sin embargo, bajo este paradigma no se considera una variabilidad dinámica, sino que la parte variable queda definida de antemano.

Desde otro enfoque, la IMS se centra en el diseño, construcción y adaptación de métodos, técnicas y herramientas de soporte para el desarrollo de sistemas de software [BRI96]. Una situación debe comprenderse como la combinación de circunstancias en un momento dado y en un contexto específico que puede definirse a partir de diversos factores, tanto internos como externos, que afectan el comportamiento y la eficiencia organizacional, y la forma de trabajo y los artefactos a producir. Por lo tanto, cada situación es una combinación del contexto de aplicación y de las características propias del proyecto de software. Entonces, un proceso se compone de fragmentos o bloques más pequeños, que se caracterizan por tener un objetivo, el cual puede ser alcanzado por una o más actividades que se realizan aplicando una o más técnicas incluidas en ese fragmento y que pueden generar uno o más artefactos [BRI96]. Algunos de los bloques son comunes a todas las situaciones mientras que otros son variantes de acuerdo a distintos factores de situación. Por lo tanto, el proceso se conforma con la unión de ciertos bloques para una situación particular [HEN10]. Cabe mencionar que en la gran mayoría de los estudios relacionados con la IMS no se aborda un aspecto esencial relacionado con la probabilidad de evolución o de estimación incorrecta de los factores que definen esa situación, es decir, consideran a la misma de manera estática para formular el proceso.

Existen ciertas semejanzas al comparar el enfoque de la IMS con el de Variabilidad de Procesos, como por ejemplo, ambos favorecen la adaptabilidad en base a características particulares y la reutilización. Sin embargo, se observa que la IMS presenta una visión más general, dirigida a la construcción de un método para una situación determinada, a partir de fragmentos o bloques que existen o pueden ser creados, en cambio, la variabilidad parte de un proceso base ya existente y se avoca a establecer sus variaciones posibles o caminos alternativos según ciertas condiciones que pueden darse al aplicar el proceso.

Existen trabajos que aplican los principios de la IMS en la IR [JAF12] [KHA14] [BAK15] [KHA16], aunque sin considerar la evolución de la situación en que está inmerso el proceso, impidiendo por ende una adaptación dinámica del mismo.

### **Metodología del trabajo desarrollado**

Se realizó una revisión bibliográfica, enfocada principalmente en los temas de Ingeniería de Métodos Situacional y Variabilidad de Procesos. En base a esta revisión se analizó un conjunto amplio de factores que pueden influenciar un proceso de requisitos, seleccionándose aquellos con un impacto efectivo sobre la estrategia de IR bajo estudio.

Se llevó a cabo un estudio de caso sobre la evolución y estimación de los factores

situacionales seleccionados, en base a 35 proyectos de requisitos desarrollados entre 2015 y 2017. En este caso de estudio se analizaron los cambios sufridos por los factores desde el inicio hacia el final del proceso de requisitos. Esto permitió identificar el impacto que valoraciones dudosas en ciertos factores pueden incidir negativamente en el proceso de requisitos adaptado, y se definió una heurística para alcanzar un valor más preciso desde el inicio de la adaptación del proceso.

En base a los estudios bibliográficos, la experiencia en la aplicación de la estrategia de IR y de proyectos de investigación previos, y los resultados del caso de estudio, se definieron distintos tipos de interacciones entre los factores situacionales, reglas de adaptación de la estrategia y, finalmente, el proceso de adaptación dinámica de la estrategia, a través de reevaluaciones de los factores en distintos hitos.

Posteriormente, se llevó a cabo un experimento controlado para comparar variantes del proceso de IR según aquellos factores situacionales más relevantes, definiéndose previamente un conjunto de métricas a aplicar. Se utilizaron dos dominios de aplicación diferentes desarrollando en cada uno las tres variantes consideradas más relevantes en la estrategia de requisitos. Este trabajo permitió confirmar la necesidad de adaptación del proceso para lograr resultados más efectivos.

## **Resultados Obtenidos**

Como resultado del proyecto de investigación se alcanzaron los siguientes resultados:

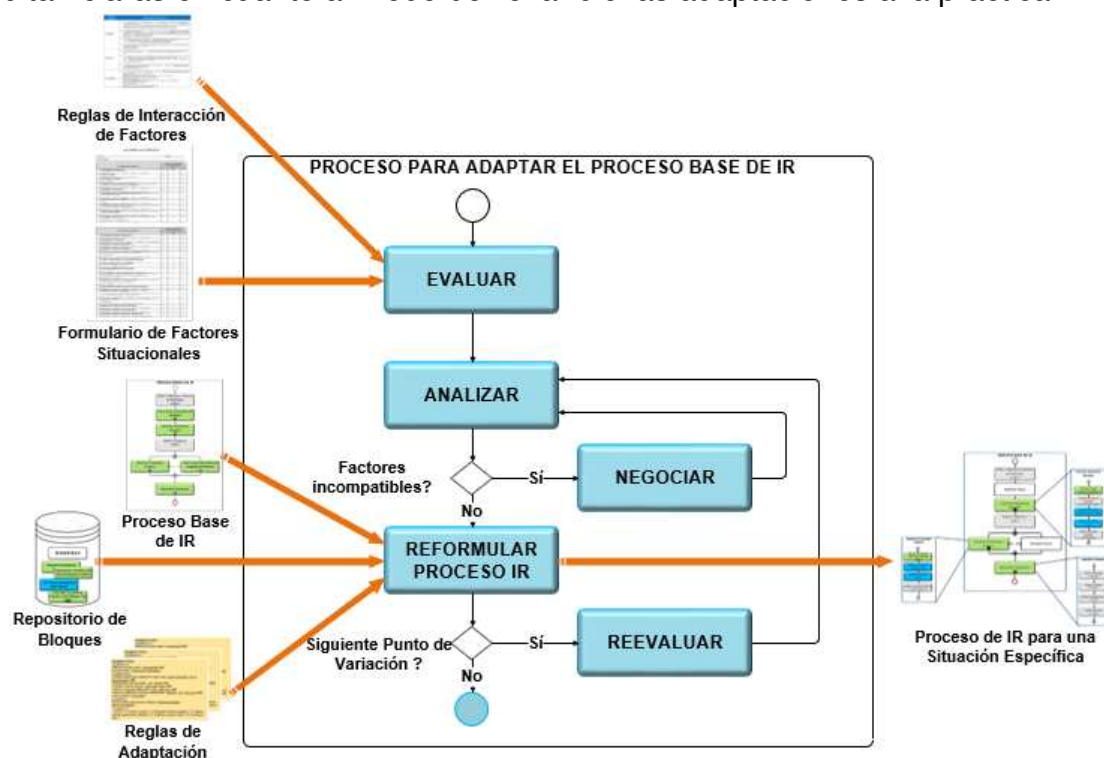
- La fragmentación de la estrategia básica de requisitos en bloques de proceso, y de sus posibles variantes en bloques de proceso alternativos.
- La definición de cada bloque de proceso con la descripción de la actividad involucrada, sus entradas y salidas, y recomendaciones de técnicas a utilizar.
- La definición de un conjunto de factores situacionales con sus valores admisibles, sus interacciones y el tipo de evolución posible.
- La definición de los puntos de variación del proceso de requisitos y los factores que impactan en cada punto.
- La definición de reglas de adaptación que indican qué bloques de proceso utilizar para formular o reformular el proceso de requisitos en cada punto de variación.
- La definición de un proceso dinámico de adaptación del proceso de IR, que aplica las reglas de adaptación en los puntos de variación y reevalúa los factores situacionales en cada punto, según el tipo de evolución posible del factor y la ponderación del nivel de confianza asignado al valor del factor (ver Figura 1).

Los productos enumerados se reflejan en las publicaciones en [1] [2] [4] [6]. Las definiciones detalladas están volcadas en la tesis de maestría de la Ing. Ledesma, desarrollada en UNLaM. A partir de la tesis de maestría del Ing. Mighetti, finalizada en el transcurso del proyecto en UNLaM, se identificaron factores no previstos inicialmente en la adaptación del proceso. Además, se han realizado otras publicaciones relacionadas [3] [5].

## **Conclusiones**

Proporcionar guías sobre cómo adaptar el proceso de IR, facilita planear su adaptación antes de iniciarlo, en lugar de realizar ajustes a medida que ocurren contingencias previsibles. Existen varias propuestas orientadas a la definición y a la mejora del proceso de IR, ajustándolo a las características particulares de un proyecto. Sin embargo, estas

aproximaciones no son tratadas con la profundidad necesaria y en la mayoría de los casos no resultan claras en cuanto al modo de llevar dichas adaptaciones a la práctica.



**Figura 1.** Proceso de adaptación dinámica del proceso de requisitos

La propuesta elaborada en este proyecto aporta una mejora a la estrategia de IR basada en modelos en lenguaje natural, a partir de un refinamiento del proceso de IR. La mejora consiste en la adaptación sistemática y dinámica de la estrategia de acuerdo con la evolución del contexto situacional en que se aplica. El proceso se descompuso en bloques de proceso, favoreciendo el reuso y brindando mayor flexibilidad a la adaptación del proceso en sí mismo. Asimismo, la adaptación puede impactar en los productos del proceso, de manera tal, que algunos modelos también son ajustables.

## Publicaciones y/o Transferencias Realizadas

Se realizaron las siguientes publicaciones con referato:

- [1] "Dynamic Situational Adaptation of a Requirements Engineering Process", Hadad, G.D.S., Doorn, J.H., Ledesma, V.A. En: Encyclopedia of Information Science and Technology, 4th Edition, IGI Global, Information Science Reference, Mehdi Khosrow-Pour (ed). Hershey, EEUU. Julio 2017 (2018), pp.7422-7434. ISBN: 978-1-5225-2255-3.
- [2] "Requirements Authorship: A Family Process Pattern", Hadad, G.D.S., Doorn, J.H., Leite, J.C.S.P., 7th International Workshop on Requirements Patterns in conjunction with the 25th IEEE International Requirements Engineering Conference, Lisboa, Portugal, IEEE Digital Library, pp.248-251, Septiembre 2017. ISBN: 978-1-5386-3489-9.
- [3] "A intenção de uso da Linguagem Natural na especificação de requisitos: um estudo comparativo entre a Argentina e o Brasil", Calazans, A.T.S., Paldês, R.A., Paulino, A.O., Freire, F.R., Fraga, J.P., Hadad, G.D.S., Melo Mariano, A., 46 JAIIO - Jornadas Argentinas de Informática, track ASSE - Simposio Argentino de Ingeniería de Software, Córdoba, pp.122-128, Septiembre 2017. ISSN: 2451-7593.
- [4] "Evolución de los Factores Situacionales durante el Proceso de Requisitos", Ledesma, V.A., Hadad, G.D.S., Doorn, J.H., Mighetti, J.P., Bedetti, N.A., Elizalde, M.C., WICC

2018 – XX Workshop de Investigadores en Ciencias de la Computación, Corrientes, pp. 578-582, Abril 2018. ISBN: 978-987-3619-27-4.

- [5] “Revisión con base cognitiva de un Proceso de Requisitos”, Hadad, G.D.S., Doorn, J.H., Elizalde, M.C., WICC 2018 – XX Workshop de Investigadores en Ciencias de la Computación, Corrientes, pp. 619-623, Abril 2018. ISBN: 978-987-3619-27-4.
- [6] “Estimación de los Factores Situacionales del Proceso de Requisitos”, Ledesma, V.A., Hadad, G.D.S., Doorn, J.H., Bedetti, N.A., WER 2018 - 21st Workshop on Requirements Engineering, Río de Janeiro, Brasil, Septiembre 2018. ISBN: 978-85-907171-1-9.

Se realizó la exposición de [3] y [6] en JAIIO y WER respectivamente. Se presentaron los pósteres [4] y [5] en WICC, exponiendo en la mesa de trabajo de Ingeniería de Software.

Los resultados de investigaciones anteriores y de la actual se están utilizando en cursos de posgrado en UCA y UADER, y en cursos de grado en UNLaM y UB.

Por otro lado, a lo largo de este período se han mantenido relaciones con otras universidades mediante la participación conjunta en publicaciones y otras actividades:

- Instituto CEUB de Pesquisa e Desenvolvimento, Centro Universitário de Brasília (Brasil), con quienes se publicó el artículo [3].
- Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro (Brasil), con quien se publicó [2], y se está escribiendo un libro sobre Ingeniería de Requisitos, donde también participa el Ing. J.H. Doorn, investigador de la UNLaM.

## **Bibliografía Utilizada**

- [BAK15] Bakhat, K.A., Sarwar, A.A., Motla, Y.H.B., Akhtar, M.C., “A Situational Requirement Engineering Model for an Agile Process”, Bahria University Journal of Information & Communication Technology, 8, 1, 2015, pp. 21-26.
- [BRI96] Brinkkemper, S., “Method Engineering: Engineering of Information Systems Development Methods and Tools”, Information & Software Technology, 38, 4, 1996, pp. 275-280.
- [GAL14] Galster, M., Weyns, D., Tofan, D., Michalik, B., Avgeriou, P., “Variability in Software Systems - A Systematic Literature Review”, IEEE Transactions on Software Engineering, 40, 3, 2014, pp. 282-306.
- [HAD08] Hadad, G.D.S., “Uso de Escenarios en la Derivación de Software”, Tesis Doctoral, Facultad de Ciencias Exactas, Universidad Nacional de La Plata, 2008.
- [HEN10] Henderson-Sellers, B., Ralyté, J., “Situational Method Engineering: State-of-the-Art Review”, Journal of Universal Computer Science, 16, 3, 2010, pp. 424-478.
- [JAF12] Jafarinezhad, O., Ramsin, R., “Development of Situational Requirements Engineering Processes: A Process Factory Approach”, 36th IEEE International Conference on Computer Software and Applications, 2012, pp. 279-288.
- [KHA14] Khan, H.H., bin Mahrin, M.N., bt Chuprat, S., “Factors for Tailoring Requirement Engineering Process: A Review”, International Journal of Software Engineering and Technology, 1, 1, 2014, pp. 7-18.
- [KHA16] Khan, H., bin Mahrin, M., Mali, M., “Situational Requirement Engineering Framework for Global Software Development: Formulation and Design”, Bahria University Journal of Information & Communication Technologies, 9,1, 2016, pp.74-84.
- [SCH06] Schnieders, A., Puhmann, F., “Variability Mechanisms in E-Business Process Families”, Intl Conference on Business Information Systems, 2006, pp. 583-601.
- [WHE12] Wheelen, T., y Hunger, D., “Strategic Management and Business Policy. Toward Global Sustainability”, 13° edición, Pearson, 2012.