

USO DE REVISIONES SISTEMÁTICA COMO ESTRATEGIA DE GENERACIÓN DE CONOCIMIENTOS PARA MEJORA CONTINUA

Fernández, E., Diez, E., Malacrida, J., Britos, P., Dieste, O., García Martínez, R.

Centro de Ingeniería del Software e Ingeniería del Conocimiento. Escuela de Postgrado. ITBA
Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. UBA
Grupo de Ingeniería de Software Experimental. Facultad de Informática. UPM

{enfernan, ediez, pbritos, rgm}@itba.edu.ar

Resumen

En el presente trabajo se describe como las Revisiones Sistemáticas (una estrategia científica que apunta a generar piezas de conocimiento de alto nivel) pueden ayudar al proceso de “mejora continua” que propone CMMI a las empresas que desean certificar los más altos niveles de madurez.

Palabras claves: Software Empírico, Revisiones Sistemáticas, CMMI, Mejora continua.

1. INTRODUCCIÓN

La experimentación de software es una rama de la Ingeniería del Software (IS) que ha venido creciendo significativamente en los últimos años (por ejemplo, en [Tonella, *et al*, 2007] se identificaron 260 experimentos vinculados a técnicas de Ingeniería Inversa), cada vez son más los autores que aplican estrategias científicas en sus publicaciones para demostrar que las piezas de conocimiento generadas son realmente valiosas. Por otra parte, los modelos de madurez desarrollados para determinar en que grado los procesos de construcción de software aplicado por las distintas empresas son fiables, exigen a la IS como disciplina científica construir herramientas y técnicas fiables validadas empíricamente [Juristo y Moreno, 2001].

Esto puede lograrse de dos formas bien diferenciadas, la primera (y más antigua) mediante el desarrollo de un estudio científico de primer nivel (básicamente mediante el uso de pruebas de laboratorio o experimentos [Juristo y Moreno, 2001]) y la segunda mediante un estudio científico de segundo nivel (básicamente mediante el desarrollo de una combinación fiable de los resultados de un conjunto de experimentos previamente desarrollados [Basilli, *et al*, 1996]). Ambas estrategias tienen ventajas y desventajas, siendo la principal desventaja de la primera estrategia el alto tiempo y costo económico que involucra el desarrollo de los experimentos, los cuales muchas veces la transforman en “prohibitivas” para las compañías. Estos costos pueden reducirse considerablemente (sobre todo los vinculados a la parte económica) mediante el desarrollo de un estudio de segundo nivel.

En el presente trabajo se describe como utilizar las Revisiones Sistemáticas para obtener piezas de conocimiento fiables que ayuden a la “mejora continua” que propone CMMI [Software Engineering Institute, 1995]. En el Estado del la Cuestión (sección 2) se describen los modelos de madurez de desarrollo de software (sección 2.1) y se describe en que consisten los distintos enfoques experimentales (sección 2.2); en la definición del problema (sección 3) se identifica el problema a

tratar; en la Solución Propuesta (sección 4) se describe como aplicar las RS para obtener la mejora continua del desarrollo; en la Demostración de la solución (sección 5) se muestra el desarrollo de una Revisión Sistemática; y luego se presentan las conclusiones obtenidas (sección 6);

2. ESTADO DE LA CUESTIÓN

2.1. Modelos de Madurez

De acuerdo a los modelos SW-CMM y CMMI [Chrissis, *et al*, 2003], se puede afirmar que en una organización, la madurez de su proceso de software es el grado hasta el cual ese proceso se ha explícitamente definido, gestionado, medido, controlado y está operativo. La madurez del proceso es un indicador del potencial para el crecimiento de la capacidad (rango de resultados esperados que pueden ser alcanzados) y representa tanto la riqueza del proceso de software de una organización como la coherencia con que éste se aplica en distintos proyectos dentro de ella.

Ahora bien, una organización que ha alcanzado un alto grado de madurez (Nivel 5 – Optimizado, para los mencionados modelos) mejora continuamente sus procesos en base a un entendimiento de las causas comunes de variación inherentes a esos procesos. El foco está puesto en la mejora continua de la performance de los procesos, por medio de mejoras incrementales, innovadoras y tecnológicas.

La organización de Nivel 5, busca identificar aquellos elementos más débiles del proceso y optimizarlos en base a un análisis defecto-origen y prevención de defectos. Identifica la causa de los defectos y evita que vuelvan a ocurrir, para ello, debe analizar los defectos que fueron encontrados en el pasado y tomar acciones específicas para prevenir la ocurrencia de este tipo de defectos en el futuro.

Adicionalmente, en la organización de Nivel 5 [Software Engineering Institute], existen datos que justifican la aplicación de nuevas ideas o tecnologías a determinadas tareas críticas y la evidencia cuantitativa permite conocer el grado de eficacia alcanzado al implementarlas en proyectos piloto. Si los resultados son los esperados, la innovación es introducida en el proceso de forma ordenada y disciplinada.

En organizaciones maduras, la mejora continua es parte de su cultura. Objetivos cuantitativos de mejora de los procesos son establecidos, y revisados continuamente para reflejar cambios en objetivos de negocio y usados como criterio para gestionar la mejora de los procesos. Finalmente, los efectos de las mejoras de procesos implementadas son medidos, y evaluados contra los objetivos cuantitativos de mejora de los procesos.

2.2. Experimentación

La experimentación, en cualquier rama de la ciencia, tiene como objetivo de contrastar una o más hipótesis con la realidad para verificarlas o refutarlas [García, 2004]. En tal sentido, podemos decir que existen dos tipos de experimentos, los denominados de primer orden (son los ensayos experimentales propiamente dichos) y los denominados de segundo orden (consiste en combinar de forma objetiva un conjunto de experimentos previamente realizados). A continuación se describe en detalle en que consiste cada tipo de experimento:

- **La experimentación de primer orden** consiste en el estudio de un fenómeno, reproducido generalmente en un laboratorio, en las condiciones particulares de estudio que interesan, eliminando o introduciendo aquellas variables que puedan influir en él [Epidat, 2008]. Se entiende por variable todo aquello que pueda causar cambios en los resultados de un

experimento y se distingue entre variable independiente, dependiente y controlada. Desarrollar experimentos de este tipo permite a los ingenieros poder conocer cuales son los mejores métodos y herramientas que se deben aplicar a través de un método científico y por lo tanto objetivo. Este marco de Experimentación permite, para el caso puntual de la Ingeniería del Software, brindar información objetiva y no sólo opiniones sobre que es lo mejor a aplicar en cada etapa de un proyecto software según las circunstancias.

- **La experimentación de segundo orden** consiste en combinar el resultado de un conjunto de experimentos previamente desarrollados [Cochrane, 2008]. Para que esta combinación sea válida es necesario que se desarrolle de forma objetiva, lo cual se logra utilizando las Revisiones Sistemáticas. Una Revisión Sistemática es un procedimiento que aplica estrategias científicas para aumentar la fiabilidad del proceso de recopilación, valoración crítica y agregación de los estudios experimentales relevantes sobre un tema [Goodman, 1996]. No obstante, si bien las RS proporcionan un marco de trabajo que permite realizar la recopilación de experimentos y, en menor medida, la valoración crítica de los mismos, dentro del actual contexto de la IS, muchas veces falla a la hora de agregar los resultados. El motivo de la falla reside en que la estrategia de agregación, recomendada por la mayoría de los autores [Fernández, 2007], se basa en la aplicación del Meta-Análisis. El Meta-Análisis es un nombre colectivo que hace referencia a un conjunto de métodos estadísticos que permiten combinar resultados experimentales, siempre y cuando se verifiquen ciertas restricciones, tales como un número mínimo de experimentos, adecuadamente recopilados y homogéneos [Hedges y Olkin, 1985]. Estos problemas fueron abordados en [Fernández; 2007], donde se propone la utilización de un conjunto de métodos de agregación de forma conjunta en lugar de aplicar únicamente el método de “Diferencias Medias Ponderadas”.

En consecuencia, podemos decir que el aplicar estrategias científicas para corroborar los conocimientos, como afirma Pfleeger [1999], “*permitirá ganar más entendimiento de que hace a un software bueno y como hacerlo mejor*”, transformando de esta forma al desarrollo de software en un verdadero proceso ingenieril.

3. DEFINICIÓN DEL PROBLEMA

Tomar decisiones respecto de si una técnica o metodología es mejor que otra sin el uso de estrategias científicas provee conocimientos poco fiables, ya que no puede asegurarse que las conclusiones obtenidas sean realmente válidas y repetibles en otros proyecto. Por otra parte, el desarrollo de experimentos de primer orden a gran escala (que involucre una gran cantidad de sujetos experimentales) es, en general, altamente costoso para las empresas que desarrollan software y por tanto no se desarrollan. Por último, si bien la RS podría ser una alternativa más económica para la generación de conocimientos validados científicamente, su difusión es aún escasa en el actual contexto de la IS, y muchos de los intentos de desarrollo fracasaron en la fase de agregación de resultados.

4. SOLUCIÓN PROPUESTA

Para aumentar la fiabilidad de los conocimientos utilizados por las empresas para la construcción de software, se propone la utilización de las Revisiones Sistemáticas desarrollada por [Kitchenham, 2004], en combinación con la estrategia de agregación desarrollada por [Fernández, 2007] como medio para la generación de piezas de conocimiento empíricamente validadas.

5. DEMOSTRACIÓN DE LA SOLUCIÓN

A continuación se presenta un caso testigo de como se puede generar conocimientos validados empíricamente mediante el uso de Revisiones Sistemáticas.

5.1. Definición del contexto de investigación

Las técnicas de inspecciones de software son una forma de detectar fallas en los productos software que se generan, tanto en documentos como código. Encontrar fallas en las etapas iniciales del desarrollo es importante porque el costo de corregirlas aumenta a medida que pasan las etapas [Berling y Thelin; 2004].

En un nivel general, existen dos grupos de técnicas de inspección:

- Técnicas de inspección basadas en listas de verificación (CBR): Consisten en hacer una inspección en base a un conjunto de preguntas (si/no), previamente definidas, que guían al inspector durante el proceso de búsqueda [Sabaliauskaite, *et al*, 2002].
- Técnicas de inspección basadas en escenarios (SBR): A diferencia de la anterior, le provee un escenario al inspector y describe cómo tiene que proceder y qué tiene que buscar durante la inspección [Sabaliauskaite, *et al*, 2002].

A demás de los dos enfoques mencionados anteriormente, existen otros que se consideran relevantes para el actual problema de investigación:

- Técnicas de inspección basadas en perspectivas (PBR): Desarrollada por [Basili, 1996]. Consisten en enfocar la inspección en los puntos de vista de los diferentes interesados en el software (o en una parte de él) que definen que propiedades de calidad es necesario evaluar [Biffi y Halling; 2003].
- Técnicas de inspección basadas en el uso (UBR): Propuesta por [Thelin, *et al*, 2003]. Utilizan casos de uso para guiar al inspector en la revisión. Hacen hincapié en la revisión enfocándose en el punto de vista del usuario para detectar los defectos críticos de una manera eficiente [Berling, *et al*, 2004].
- Técnicas de inspección basadas en defectos (DBR): Propuesta por [Porter, *et al*, 1995]. Fueron principalmente utilizadas en documentos de requerimientos. Estas técnicas se enfoca en detectar tipos de fallas específicos. Definen un grupo de escenarios que son procedimientos que el inspector sigue durante la revisión [Thelin, *et al*, 2003].

Es importante destacar que los últimos tres grupos de técnicas pueden agruparse en el grupo de “técnicas basadas en escenarios”, ya que, realidad, lo que aportan son diferentes estrategias para la definición de los escenarios.

En la actualidad, si bien las piezas de conocimiento generadas en los experimentos que analizan el desempeño de las técnicas de inspección de código aportan conocimientos interesante, estos estudios en general utilizan pocos sujetos experimentales [Thelin, *et al*, 2003; Denger, *et al*, 2004; Sabaliauskaite, *et al*, 2004; Lee y Bohlen, 2005], lo cual limita considerablemente a aplicación de las piezas de conocimiento generadas, ya que no pueden ser tomadas como conocimientos generales o de aplicación universal.

5.2. Aplicación de una RS

Para el desarrollo de la presente RS, se siguieron las recomendaciones de:

- [Kitchenham, 2004] para el armado del protocolo de Revisión,
- [Griman, 2007] para el armado de la estrategia de búsqueda de documentos, y
- [Fernández, 2007] para la agregación de los resultados de los experimentos.

5.2.1. Definición de la pregunta de investigación

Dado que la variante de la técnica basada en escenario mas difundida es la “basada en perspectiva”, la pregunta de investigación va a estar centrada en determinar si:

¿Son mejores las técnicas basadas en listas de verificación o las basadas en perspectivas?

Para poder evaluar objetivamente la pregunta de investigación, se analizará el desempeño de las técnicas en base a los resultados obtenidos para la variable respuesta: **Efectividad**, la cual se estima como la cantidad de errores detectados sobre la cantidad de errores total que existía en el código o documento.

5.2.2. Resultado de la búsqueda

Para identificar los distintos estudios experimentales se desarrolló un proceso de búsqueda, basado en las recomendaciones hechas en [Griman, 2007], sobre la base bibliográfica de [Scopus, 2008]. A continuación, en la tabla 1, se describe la lista de estudios seleccionados luego del proceso de búsqueda, en el cual se identificaron estudios experimentales (experimentos y Quasi-Experimentos) que analicen técnicas de lectura de código:

Id	Referencia
1	An Experimental Comparison of Checklist-Based Reading and Perspective-Based Reading for UML Design Document Inspection [Sabaliauskaite, et al, 2002]
2	Investigating the Accuracy of Defect Estimation Models for Individuals and Teams Based on Inspection Data [Biffl, et al, 2003]
3	Investigating the Active Guidance Factor in Reading Techniques for Defect Detection [Denger, et al, 2004]
4	Comparing Code Reading Techniques Applied to Object-oriented Software Frameworks with regard to Effectiveness and Defect Detection Rate [Abdelnabi, et al, 2004]
5	An Experimental Comparison of Usage-Based and Checklist-Based Reading [Thelin, et al, 2003]
6	Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection Teams [Biffl y Halling; 2003]
7	The Development and Evaluation of Three Diverse Techniques for Object-Oriented Code Inspection [Dunsmore, et al, 2003]
8	A Case Study of Reading Techniques in a Software Company [Berling y Thelin, 2004]
9	A Case Study of Reading Techniques in a Software Company [Berling y Thelin; 2004]
10	Assessing the Impact of Active Guidance for Defect Detection: A Replicated Experiment [Lanubile, et al, 2004]
11	Investigating the Effect of Expert Ranking of Use Cases for Design Inspection [Winkler, et al, 2004]

Tabla 1: Presentación de los estudios seleccionados

A continuación, en la tabla 2, se describan los datos más significativos desde el punto de vista de la agregación de resultados que presentan los estudios de la tabla 1.

Id	Tratamientos	Variables respuesta	Variables estadísticas
1	CBR, PBR	Efectividad	Medias, Sujetos y Desvío Estándar
2	CBR, SBR	Efectividad	Medias, Sujetos y Desvío Estándar
3	CBR, PBR	Efectividad	Medias, Sujetos
4	CBR, SBR	Cantidad de Errores	Medias y Sujetos
5	CBR, UBR	Efectividad	Medias, Sujetos y Desvío Estándar
6	CBR, SBR	Efectividad	Medias y Sujetos
7	CBR, UBR	Realiza una lectura sobre casos de uso.	Medias, Sujetos y Desvío Estándar
8	CBR, UBR	Efectividad	Medias y Sujetos
9	CBR, PBR	Efectividad	Medias, Sujetos
10	CBR, PBR	Efectividad	Medias, Sujetos y Desvío Estándar
11	CBR, UBR	Efectividad	Medias, Sujetos y Desvío Estándar

Tabla 2: Descripción de los estudios seleccionados

5.2.1. Resultados obtenidos mediante el uso de técnicas de agregación paramétricas

Dado que la calidad de los reportes de los estudios identificados, no permite la aplicación de un proceso de agregación estándar [Kitchenham, 2004], se procedió a aplicar un proceso de agregación de más de un nivel de evidencia [Fernández, 2007]. Dentro del cual se han podido generar evidencia de nivel I (son las de mayor nivel de fiabilidad ya que se obtienen mediante técnicas estadísticas del tipo paramétricas) y II (son de un nivel de fiabilidad medio, ya que se obtienen mediante técnicas estadísticas del tipo no paramétricas, las cuales tienen un error medio cuando indican que no existen diferencias significativas entre los tratamientos). A continuación se detallan los resultados obtenidos (Para la aplicación de los métodos de agregación, la técnica basada en listas de verificación ha sido tomada como el tratamiento experimental y la técnica basada en perspectivas ha sido tomada como el tratamiento de control):

Evidencias de nivel I

Resultados obtenidos mediante el método Diferencias Medias Ponderadas:

Estudio (Peso)	d	L _l	L _u
1 (79 %)	0,077	-0,456	0,610
9 (20 %)	0,057	-0,991	1,105
Global	0,073	-0,402	0,548

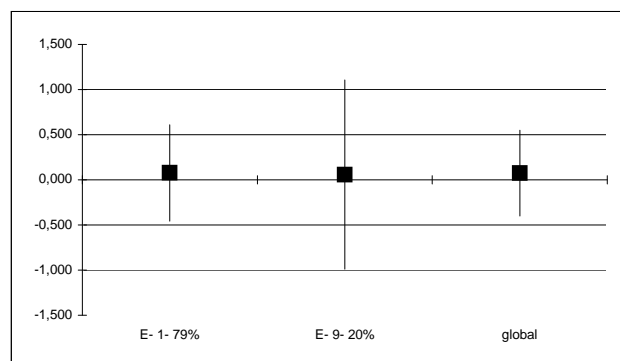


Figura 1: Resultados Obtenidos mediante la aplicación del método DMP

Resultados obtenidos mediante el método Response Ratio – Paramétrico

Estudio (Peso)	RR	L_l	L_u
1 (99 %)	1,016	0,920	1,122
9 (1 %)	1,147	0,108	12,149
Global	1,016	0,920	1,122

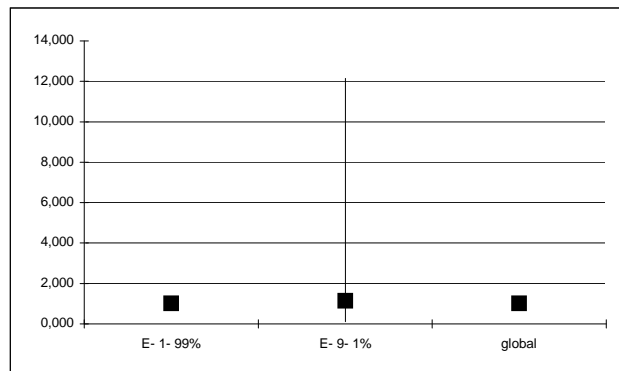


Figura 2: Resultados Obtenidos mediante la aplicación del método RR - Paramétrico

Evidencias de nivel II

Resultados obtenidos mediante el método Response Ratio – no Paramétrico

Id	RR	L_l	L_u
1	1,016	0,596	1,733
3-a	0,769	0,201	2,937
3-b	0,603	0,165	2,201
3-c	1,456	0,342	6,205
8	0,900	0,154	5,265
9	1,147	0,395	3,329
Global	0,980	0,659	1,458

Nota: El artículo 3 presenta 3 estudios experimentales.

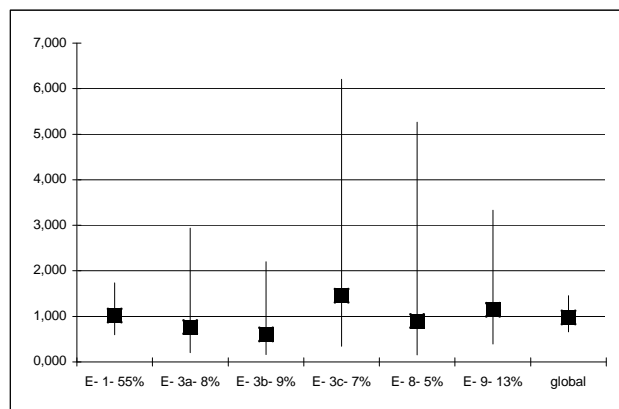


Figura 3: Resultados Obtenidos mediante la aplicación del método RR – No Paramétrico

5.2.2. Evaluación de los resultados

En el presente proceso de agregación se consiguió obtener evidencia de dos niveles diferentes:

- De nivel I, las mas fiables en cuanto a la precisión de los métodos aplicados y calidad de los estudios incluidos, y
- De nivel II, las cuales, son tan fiables como las de nivel I cuando indican que las diferencias entre los tratamientos son significativas, pero, menos fiables cuando indican lo contrario (poseen un error de nivel II mayor).

Resultados obtenidos

- El proceso de agregación incluye 6 estudios experimentales que en conjunto incluyen a 102 sujetos experimentales.
- Los resultados obtenidos para el nivel de Evidencia I (basada en 2 estudios) no indican diferencias apreciables a favor de ninguna de las dos técnicas.

- Los resultados obtenidos para el nivel de Evidencia II (basada en 6 estudios) no indican diferencias apreciables a favor de ninguna de las dos técnicas. Coincidiendo con los resultados del nivel I.
- A nivel de estudios, el experimento 3b muestra una fuerte ventaja de la técnica CBR respecto de PBR, mientras que el estudio 3c, muestra un resultado antagónico al anterior.
- El experimento 1 utiliza una cantidad de sujetos experimentales mucho mayor que el resto de los experimentos. Esto se ve reflejado en el peso que el mismo tiene en los resultados finales.

Como corolario de lo expuesto, podemos decir que los resultados de los distintos niveles de evidencia son concordantes entre sí, y expresa que no existen diferencias significativas en el desempeño de ambas técnicas.

6. CONCLUSIONES

En el presente trabajo se ha podido demostrar que:

- Las Revisiones Sistemáticas son una alternativa económica para la generación de nuevas piezas de conocimientos.
- Los conocimientos obtenidos mediante las RS pueden ser aplicables a la “mejora continua” del proceso de construcción de software.
- Como toda disciplina científica la validación empírica de los conocimientos es fundamental para la evolución de la misma.

7. FUTURAS LÍNEAS DE INVESTIGACIÓN

Si bien las piezas de conocimiento generados mediante el uso de RS son de nivel general y, por tanto, aplicables a cualquier organización, sería importante desarrollar una estrategia para analizar la aplicación de las piezas de conocimiento obtenidas en una organización puntual.

8. REFERENCIAS

- Abdelnabi Z., Cantone G., Ciolkowski M., Rombach D.; 2004; *Comparing Code Reading Techniques Applied to Object-oriented Software Frameworks with regard to Effectiveness and Defect Detection Rate*; Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04)
- Basili, V. R., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sörumgård, S., Zelkowitz, M.; 1996; *The empirical investigation of perspective-based reading*, International Journal on Empirical Software Engineering, Vol. 1, No. 2; pp. 133–164.
- Berling T. y Thelin T.; 2004; *A Case Study of Reading Techniques in a Software Company*; Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04)

- Biffi S. y Halling M.; 2003; *Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection Teams*; IEEE Transactions on Software Engineering, Vol. 29, No. 5, May 2003; pp. 385-397
- Biffi S., Halling M., Köszei S.; 2003; *Investigating the Accuracy of Defect Estimation Models for Individuals and Teams Based on Inspection Data*; Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03)
- Chrissis M., Konrad M. y Shrum S.; 2003; CMMI: Guidelines for process integration and product improvement, Addison-Wesley.
- Cochrane; 2008; *Curso Avanzado de Revisiones Sistemáticas*; www.cochrane.es/?q=es/node/198
- Denger C., Ciolkowsky M., Lanubile F.; 2004; *Investigating the active guidance factor in reading techniques for defect detection*; Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04)
- Dunsmore A., Roper M., Wood M.; 2003; *The Development and Evaluation of Three Diverse Techniques for Object-Oriented Code Inspection*; IEEE Transactions On Software Engineering, Vol. 29, No. 8, August 2003
- Epidat, 2008; programa de libre distribución desarrollado por instituciones públicas y dirigido a epidemiólogos y otros profesionales de la salud; www.paho.org/spanish/sha/epidat.htm
- Fernández, E.; 2007; Aggregation process with multiple evidence level for experimental studies in software engineering; 2^{do} International Doctoral Symposium on Empirical Software Engineering (IDoESE); 75-81
- García, R.; 2004; Inferencia Estadística y Diseño de Experimentos; eudeba; Buenos Aires Argentina
- Goodman C.; 1996; *Literature Searching and Evidence Interpretation for Assessing Health Care Practices*; SBU; Stockholm.
- Grimán Padua; 2007; Proposal of a review process of empirical studies in software engineering; Internacional Doctoral Symposium on Empirical Software Engineering (IDoESE). 25-32
- Hedges, L.; Olkin, I.; 1985; *Statistical methods for meta-analysis*. Academic Press.
- Kitchenham, B. A.; 2004; *Procedures for performing systematic reviews*. Keele University; TR/SE-0401. Keele University Technical Report.
- Juristo N y Moreno A.; 2004; *Basics of software engineering experimentation*; Boston: Kluwer Academic Publisher.
- Lajeunesse, M & Forbes, M.; 2003; *Variable reporting and quantitative reviews: a comparison of three meta-analytical techniques*. Ecology Letters, 6: 448-454.
- Lanubile F., Mallardo T., Calefato F., Denger C., Ciolkowski M.; 2004; *Assessing the Impact of Active Guidance for Defect Detection: A Replicated Experiment*; Proceedings of the 10th International Symposium on Software Metrics (METRICS'04)
- Lee K. y Boehm B.; 2005; *Empirical Results from an Experiment on Value-Based Review (VBR) Processes*; Proceedings of the 2005 International Symposium on Empirical Software Engineering (ISESE'05)
- Porter A., Votta L., Basili V.; 1995; *Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment*; IEEE Transactions on Software Engineering, Vol. 21, No. 6, June 1995; pp. 563-575

Sabaliauskaite G., Matsukawa F., Kusumoto F., Inoue K.; 2002; *An Experimental Comparison of Checklist-Based Reading and Perspective-Based Reading for UML Design Document Inspection*; Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02)

Scopus; 2008; www.scopus.com

Software Engineering Institute; 1995; *The Capability Maturity Model: Guidelines for Improving the Software Process*; Addison-Wesley, Reading, Pa.

Thelin T., Runeson M., Wohlin C.; 2003; *An Experimental Comparison of Usage-Based and Checklist-Based Reading*; IEEE Transactions on Software Engineering, Vol. 29, No. 8, August 2003; pp. 687-704

Tonella P., Torchiano M., Du Bois B., Systä T.; 2007; *Empirical studies in reverse engineering: state of the art and future trends*; Empir Software Eng (2007) 12:551–571.

Winkler D., Halling M., Biffel S.; 2004; *Investigating the Effect of Expert Ranking of Use Cases for Design Inspection*; Proceedings of the 30th EUROMICRO Conference (EUROMICRO'04)