

Calidad en el desarrollo de Sistemas de Software

Pesado Patricia^(1,2), Bertone Rodolfo⁽¹⁾, Ramón Hugo⁽¹⁾, Pasini Ariel⁽¹⁾, Esponda Silvia⁽¹⁾, Alonso Laura⁽¹⁾

⁽¹⁾Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática – UNLP

⁽²⁾Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

(ppesado, pbertone, ramón, apasini sesponda, lalonso) @lidi.info.unlp.edu.ar

CONTEXTO

Esta línea de Investigación está definida en el Subproyecto “Calidad en el desarrollo de Sistema de Software”, dentro del Proyecto “Sistemas de Software Distribuidos. Aplicaciones en procesos industriales, E-government y E-learning” del Instituto de Investigación en Informática LIDI, presentado a la Universidad Nacional de La Plata en el marco de la convocatoria del Ministerio de Educación.

RESUMEN

El proyecto de investigación y desarrollo “Sistemas de Software Distribuidos. Aplicaciones en procesos industriales, E-government y E-learning” abarca aspectos que van desde los fundamentos del desarrollo (Técnicas de Ingeniería de requerimientos, Metodologías de Gestión y Desarrollo de Proyectos, Técnicas de Planificación, Métricas, Normas de Calidad, Web-Systems) hasta la concepción de aplicaciones específicas orientadas a los procesos industriales (sistemas de planeamiento industrial, control industrial en tiempo real), procesos de E-government (voto electrónico) y la aplicación de Tecnología Informática para Educación en ambientes distribuidos.

Es de hacer notar que este proyecto se coordina con otros dos proyectos presentados por el III-LIDI relacionados con Algoritmos Distribuidos / Paralelos y Sistemas Distribuidos / Paralelos.

En particular, el objetivo de este subproyecto es investigar y desarrollar soluciones en temas relacionados con el aseguramiento de la

calidad en cada una de las etapas del desarrollo de Sistemas de Software.

Una de las actividades propuestas es la adecuación de normas de calidad que permitan evaluar el desarrollo de sistemas en entornos de PYMES del país. En particular, se busca generar un contexto similar al definido y utilizado en otros países (México, Brasil, Colombia, España) los cuales adecuan el concepto de evaluación de calidad, adaptando las características básicas de CMM-CMMI o ISO al entorno propio del país.

Palabras claves: Ingeniería de Software – Calidad – Normas de Calidad – CMM – ISO - MOPROSOFT

1. INTRODUCCION

En la actualidad es innegable la importancia del procesamiento distribuido dentro del ámbito de la Ciencia de la Computación. Un sistema distribuido consiste en un conjunto de computadoras autónomas conectadas por una red y con soporte de software distribuido. Permite que las computadoras coordinen sus actividades y compartan los recursos de hardware, software y datos, de manera tal que el usuario percibe una única facilidad de cómputo integrada aunque esta pueda estar implementada por varias máquinas en distintas ubicaciones.[COU94]

Los conceptos de calidad que deben estar presentes en los procesos de desarrollo de software (ya sea distribuido o centralizado) han llevado a crear modelos estandarizados que permiten encuadrar a las organizaciones y, por consecuente, a sus proyectos. En líneas

generales existen dos normas ampliamente aceptadas: CMM e ISO.

El Software Engineering Institute (SEI) de la Universidad de Carnegie Mellon (USA) ha desarrollado un modelo completo que se basa en un conjunto de funciones de Ingeniería del Software que deberían estar presentes conforme las organizaciones alcanzan diferentes niveles de madurez del proceso de desarrollo de sistemas. Para determinar el estado actual de madurez, el SEI utiliza un esquema de 5 niveles. Estos niveles se obtienen como consecuencia de evaluar las respuestas obtenidas de un cuestionario basado en las características propias de la madurez de desarrollo alcanzado. El SEI ha asociado áreas claves del proceso a cada uno de los niveles de madurez, que describen las funciones de la ingeniería de software (ej: planificación del proyecto, gestión de requerimientos, etc.) que deben estar presentes para satisfacer una buena práctica para un nivel en particular. Cada área clave se describe identificando objetivos, compromisos, capacidades, actividades, métodos para supervisar la implementación, métodos para verificar la implementación.

El CMMI o "Modelo de Capacidad y Madurez - Integración" es una ampliación de CMM creada con el objetivo de realizar algunas mejoras respecto al SW-CMM (e integrarlo con el SE-CMM y el IPD-CMM, que pasaron a ser considerados como "obsoletos") [CMU/SEI02]. El CMMI incluye cuatro disciplinas: (1) Software, (2) Ingeniería de sistemas, (3) Desarrollo integrado de procesos y productos y (4) Gestión de proveedores. A su vez incorpora una nueva representación la que permite evaluar el nivel en cada área independientemente [CHR05]. Otra diferencia importante comparada con CMM radica en el hecho que se establece mayor énfasis en el uso continuo de métricas, insistiendo en la necesidad de la trazabilidad para la construcción del software [LAN05]. Se genera, además, mayor detalle de las áreas de proceso relativas a la ingeniería.

Por otro lado, los estándares de calidad ISO 9000 describen los elementos de garantía de calidad en términos genéricos que pueden aplicarse a cualquier organización. Para identificarse con uno de los modelos de sistema de garantía de calidad de ISO 9000 las

operaciones de la organización son examinadas minuciosamente para evaluar el ajuste correspondiente. Luego de ese registro, la organización recibe un certificado avalado por los auditores que debe ser revalidado periódicamente apuntando a un ajuste continuado a la norma.

La mejora continua en los procesos y productos de software es lo que se conoce como calidad total, que en gran parte es conseguida a través de la aplicación y evaluación de las métricas, y la realimentación del proceso productivo con estos resultados.

En particular, cualquier de las dos líneas de trabajo en calidad, ISO o CMM, presentan un conjunto de características que en un entorno de PYMES de Argentina resultan complejas de evaluar. Esta complejidad puede verse, básicamente, en dos aspectos: (1) el costo económico asociado con el proceso de SQA, en particular con la certificación y (2) con el entorno de trabajo del país que hace que algunas características contempladas en ISO o CMM resulten de difícil cumplimiento.

En algunos países iberoamericanos se está intentando abordar este problema, aunque de forma aislada, con algunas iniciativas dignas de destacar como el modelo "MoProSoft" de México (OKT03), el modelo "MR mps" de Brasil (WEB04), o el modelo SIMEP-SW de Colombia (HUR03), incluso la metodología Métrica v.3 propugnada por el MAP en España también pretende conseguir la mejora de los procesos y productos software.

Esta línea de trabajo tiene como propuesta establecer una línea base para el control de la calidad del software (de proceso y de producto) enfocada, primordialmente, a la utilización por la industria PYME de producción de software en nuestro país, objetivo en el que se trabaja con otras Universidades del país.

2. LINEAS DE INVESTIGACION Y DESARROLLO

- Análisis, estudio y discusión de normas clásicas de calidad : línea ISO y CMM - CMMI.

- Análisis , estudio y discusión de la norma ISO 15504 (Spice).
- Estudio de las características más importantes para evaluación de calidad que permitan certificar a PYMES
- Análisis de normas de calidad para PYMES existentes ("MoProSoft", "MPS.BR", modelo SIMEP-SW, metodología Métrica v.3)
- Estudio de métodos para evaluar calidad de requerimientos.
- Planeación de calidad trabajando con métodos ágiles.

3. RESULTADOS OBTENIDOS/ ESPERADOS

- Avanzar en la capacitación continua de los miembros de la línea de investigación.
- Redactar la Línea Base que permita evaluar calidad en la construcción del software siguiendo un estilo de trabajo propuesto por la norma MoProSoft.
- Evaluar la línea base obtenida bajo diferentes entornos de producción
 - Adaptar una Norma de Calidad para aplicación en PYMES del país, sobre la base de las existentes.
 - Realizar experiencias de evaluación de productos de software distribuidos basados en WEB
 - Proponer metodologías para desarrollar sistemas con técnicas como XP garantizando umbrales mínimos de calidad.
 - El III-LIDI participa del Proyecto “Construcción de un Modelo de Proceso para Certificación de Calidad en la Industria del Software y su aplicación en Pequeñas y Medianas Empresas de Desarrollo” subsidiado por la CIC y en colaboración con la UNLM.
 - El III-LIDI participa del Proyecto CYTED de cooperación internacional “COMPETISOFT: Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica”. En este proyecto se trabaja con Universidades del país (UNComahue, UNSur, UNLaMatanza) y del exterior (Universidad Nacional Autónoma de México, Universidad Nova de Lisboa – Portugal, Universidad de Campinas –

Brasil, Universidad Internacional SEK-ICAPI – Ecuador, Universidad Nacional de Colombia, Universidad Eafit – Colombia, Universidad de La Habana – Cuba, Universidad de Asunción - Costa Rica, Universidad de los Andes - Venezuela, Universidad Técnica Federico Santamaría – Chile, Universidad de la República – Uruguay, Universidad Católica del Maule – Chile, Universidad de Castilla-La Mancha - España)

- Se encuentra en desarrollo una herramienta WEB para relevamiento de capacidades de empresas de desarrollo de software, basado en la norma Mexicana MoProSoft.

4. FORMACION DE RECURSOS HUMANOS

Se encuentran en este momento en desarrollo 3 Tesinas de Grado de Licenciatura y 2 Tesis de Magister, en temas relacionados con el Proyecto.

5. BIBLIOGRAFIA

- [ACM] Colección de Communications of the ACM
- [ACM] Colección de ACM SIGSOFT (Special Interest Group on Software Engineering)
- [Ahe03] Dennis M. Ahern, et al. CMMI Distilled: A Practical Introduction to Integrated Process Improvement, Second Edition. Software Engineering Inst. Carnegie Mellon Univ. 2003
- [Bus05] M. Bush, D. Dunaway. CMMI® Assessments: Motivating Positive Change. Addison-Wesley. 2005
- [Chr05] Mary Beth Chrissis, et al. CMMI : Guidelines for Process Integration and Product Improvement. Software Engineering Inst. Carnegie Mellon Univ.

- [CMU/SEI02]** CMU/SEI.CMMI SM for Systems Engineering and Software Engineering (CMMI-SE/SW, V1.1).CMU/SEI.2002. CMU/SEI.Capability Maturity Model® Integration (CMMISM). Version 1.1..
- [Gal05]** Galin, D.; Avrahami, M.;“Do SQA programs work - CMM works. a meta analysis”, Proceedings. IEEE International Conference on 22-23 Feb. 2005 Page(s):95 - 100.Software - Science, Technology and Engineering.2005
- [Gal03]** Daniel Galin.Software Quality Assurance: From Theory to Implementation.Prentice Hall.2003
- [God04]** Nina S. Godbole .Software Quality Assurance: Principles And Practice. Alpha science. 2004
- [Guo04]** Guoping Jiang; Yingwu Chen.“Coordinate metrics and process model to manage software project risk”, Proceedings. 2004 IEEE International Volume 2, 18-21 Oct. 2004 Page(s): 865 - 869 Vol.2.Engineering Management Conference. 2004
- [IEEE]** Colecciones de Transaction on Software Engineering
- [IEEE]** Colecciones de Computer
- [ISO95]** ISO/IEC .12207: 1995 Information Technology – Software life cycle processes.ISO/IEC.1995
- [ISO04a]** ISO/IEC .15504-1: 2004 Information Technology – Process assessment – Part 1: Concepts and vocabulary.ISO/IEC_2004
- [ISO03]** ISO/IEC. 15504-2: 2003 Information Technology – Process assessment – Part 2: Performing an assessment.ISO/IEC.2003
- [ISO04b]** ISO/IEC. 15504-3: 2004 Information Technology – Process assessment – Part 3: Guidance on performing an assessment.ISO(IEC.2004
- [ISO/IEC04c]** ISO/IEC. 15504-4: 2004 Information Technology – Process assessment – Part 3: Guidance on use for process improvement and process capability determination.ISO/IEC.2004
- [ISO/IEC09]** ISO/IEC. 15504-5: 1999 Information Technology – Software Process assessment – Part 5: An assessment model and indicator guidance.ISO/IEC.1999
- [Jon94]** Caper Jones .Assessment and Control of Software Risk.Yourdon Press.1994
- [Kan03]** Stephen Kan. .Metrics and Models in Software Quality Engineering (2nd Edition).Addison Wesley. 2003
- [Ken00]** Beck Kent. Una aplicación de la programación extrema. Aceptar el cambio.Addison Wesley. 2000
- [Kot98]** G. Kotonya and I. Sommerville,.Requirements Engineering: Processes and Techniques, .Wiley.1998
- [Kus99]** Kusters, R.J.; van Solingen, R.; Trienekens, J.J.M.“Strategies for the identification and specification of embedded software quality”. STEP '99. Proceedings 30 Aug.-2 Sept. 1999 Page(s):33 - 39.Software Technology and Engineering Practice, .1999
- [Lan05]** Susan K. Land.Jumpstart CMM/CMMI Software Process Improvements: Using IEEE Software Engineering Standards.Wiley-IEEE Computer Society Press.2005
- [Lap02]** Laplante, P.A.; Neill, C.J.; Jacobs, C.; “Software requirements practices: some real data”, Proceedings 27th Annual NASA Goddard/IEEE 5-6 Dec. 2002 Page(s):121 - 128.Software Engineering Workshop, .2002
- [Lef00]** Leffingwell, D., Widrig, D.. Managing Software Requirements. A Unified Approach. Addison-Wesley.2000
- [Lou95]** Loucopoulos, P; Karakosas, V..Systems Requeriments Engineering. .McGraw Hill. Book Company.1995
- [Ple02]** Pleeger.Ingeniería de Software: Teoría y Práctica.Prentice-Hall.2002
- [PMI04]** Project Management Institute. A Guide to The Project Management Body Of Knowledge (PMBOK Guides).Project Management Institute.2004
- [Pre02]** R. Pressman.Ingeniería de Software: Un Enfoque Práctico.McGraw-Hill.2002
- [Ram04]** Hugo Ramón, Raúl Champredonde, Mariano Merino, Patricia Pesado, Armando De Giusti. Administración informática de colecciones de datos heterogéneos. Aplicación a museos de Ciencias Naturales. X Congreso Argentino de Ciencias de la Computación. CACIC 2004. Argentina.
- [Som05]** Sommerville Ian. .Requeriments Engineering, A good practice guide. .John Wiley.2005
- [WWW1]** <http://www.sei.cmu.edu/cmmi/...>
- [WWW2]** <http://www.iso.org..>
- [WWW3]** <http://www.ieee.net..>
- [WWW4]** <http://www.acm.org..>
- [WWW5]** www.systemsguild.com..
- [WWW6]** <http://www.tickit.org..> Teacher Institute for Curriculum Knowledge about Integration of Technology (TickIT) Modelo de Procesos para la Industria de Software MOPROSOFT.
- [Wyc03]** R. Wysoki.Effective Project Management: Traditional, Adaptive, Extreme, .Wiley .2003
- [You01]** Young, R..Effective Requirements Practices.Addison-Wesley. 2001