

Experiencias en la enseñanza de Ingeniería de Software

Lic. P. Pesado¹, Lic. M. Boracchia²

*Laboratorio de Investigación y Desarrollo en Informática³
Departamento de Informática - Facultad de Ciencias Exactas
Universidad Nacional de La Plata*

Resumen

Se discute la experiencia de los autores que han dictado el curso de Ingeniería de Software durante varios años, adecuando los contenidos conforme a la evolución y necesidades en el desarrollo de proyectos.

Se analizan las ventajas y desventajas de los distintos enfoques y la formación de los alumnos que luego de cursar Ingeniería de Software están a punto de salir al mercado laboral con su título de Analista de Computación.

Se discute el contexto de las asignaturas previas y posteriores en el caso de continuarse con la Licenciatura en Informática en la Universidad Nacional de La Plata.

Por último se mencionan los problemas típicos que se dan a los alumnos en los Trabajos Prácticos, para aplicar el modelo de proceso elegido.

Palabras clave: Ingeniería de Software. Educación.

¹ Prof. Titular con Ded. Excl. LIDI. Dpto. de Informática, Facultad de Ciencias Exactas, UNLP.

E-mail ppesado@lidi.info.unlp.edu.ar

²JTP Ded. SExcl. LIDI. Dpto. de Informática, Facultad de Ciencias Exactas, UNLP.

E-mail marcosb@lidi.info.unlp.edu.ar

³ Calle 50 y 115 Primer Piso, (1900) La Plata, Argentina, Teléfono 54-21-227707

WEB: lidi.info.unlp.edu.ar

Experiencias en la enseñanza de Ingeniería de Software

Introducción

La Ingeniería de software ha sufrido transformaciones desde su creación hasta la fecha debido a los adelantos tecnológicos, crecimiento en complejidad de los problemas a resolver, volumen de sistemas a mantener, etc. Hoy en día se reconoce a la Ingeniería de Software como una disciplina consolidada que merece tener una investigación y estudio concienzudos.

Es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelve problemas de todo tipo. Trata con áreas muy diversas como construcción de compiladores, sistemas operativos o desarrollos de Internet/intranet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, redes, etc.[PRE98][GHE91]

En la UNLP existen dos carreras, Analista en Computación (carrera de 3 años que permite una salida laboral rápida) y Licenciatura en Informática (carrera de 5 años que concluye con la presentación de una tesina de trabajo de grado).

La materia Ingeniería de Software se incluye en el tercer año de ambas carreras con duración anual. Para los que egresan con el título de Analista debe significar el aprendizaje de las metodologías y documentaciones del desarrollo de sistemas, problemas que seguramente deberán enfrentar en su vida profesional. Para los alumnos de Licenciatura debe además servir de presentación e introducción a temas más específicos desarrollados en materias optativas de 4to. y 5to. año como Ingeniería de Software de Tiempo Real, Sistemas Cliente/Servidor, Ingeniería de Software II (enfocada hacia calidad), Seminarios de Marketing y Administración Gerencial, Prototipación e Interfaces.

Aproximadamente más de un 50 % de las tesinas de Trabajo de Grado se realizan en temas relacionados con Ingeniería de Software y algunos de los resultados de estos trabajos se utilizan como material práctico en la materia.

Existe una línea de investigación en el Departamento de Informática referente a Automatización de Oficinas en cuyo temario se incluyen temas de Ingeniería de Software como Métricas y Ambientes de especificación y desarrollo que pueden ser experimentados a través de los convenios de Transferencia de Tecnología.

Además en el Departamento de Informática se ha implementado el Magister en Ingeniería de Software que permite actualizar y especializar a graduados de nuestras carreras y de carreras afines.

Análisis de los enfoques de un curso de Ingeniería de Software

Los enfoques posibles del contenido teórico de un curso de Ingeniería de Software básicamente se basan en dos aspectos:

- ◆ Metodologías de Análisis y Diseño
- ◆ Gestión de proyectos

En el primero de los casos el curso se basa en aspectos sobre todo técnicos, e implica el aprendizaje de metodologías alternativas y complementarias que permitan la modelización y simulación de sistemas de distintos tipos. Poniendo énfasis en la resolución de los problemas más que en el gerenciamiento del proceso de desarrollo.

En la segunda alternativa, el objetivo está en la planificación, seguimiento y control de los proyectos de desarrollo de sistemas. En esta visión se incluyen muchos más aspectos de administración.

En cuanto a la parte práctica, el primer enfoque permite elegir un temario de trabajos prácticos en el que se incluyan problemas a resolver de diversos tipos de sistemas, lo que permitirá seleccionar distintas técnicas de modelización según el problema abordado. El segundo enfoque necesariamente debe recurrir a la planificación de un sistema medianamente complejo que permite aplicar técnicas de seguimiento y control, tomar decisiones de continuar/abandonar, hacer análisis de riesgo, utilizar herramientas automatizadas o semi-automatizadas, etc.

Por la formación histórica de los Informáticos, la tendencia más natural es volcarse al primer enfoque, tratando de conocer la mayor cantidad de metodologías que permitan resolver el problema técnico, pero la evolución del proceso de desarrollo de software hace necesario e impostergable el conocimiento de técnicas de gerenciamiento, para producir desarrollos de calidad en los aspectos del proceso y del producto final.[ART93]

Esta evolución que se ha vivido como una necesidad del mercado, ha influido en la temática de la materia, incluyendo en el curso que se brinda en la actualidad aspectos de ambos enfoques, lo que logra una formación más integral, para que el egresado pueda hacer frente a cualquiera de los roles que pueden caerle en suerte en su desarrollo profesional.

Ejes temáticos del curso actual

Tratando de cumplir con los requerimientos del mercado actual, el curso de Ingeniería de Software actualmente se estructura con un primer semestre basado en la presentación de Metodologías de Análisis y Diseño y un segundo semestre centrado en el Gerenciamiento de Proyectos. [PES98]

Los siguientes puntos resumen el eje temático del primer semestre:

- ◆ Actividades Genéricas del Ciclo de Vida
- ◆ Metodologías de Análisis y Diseño Estructurado
- ◆ Metodologías Orientadas a Objetos
- ◆ Técnicas de Especificación de Requerimientos.
- ◆ Herramientas de prototipación
- ◆ Herramientas CASE

La idea en este semestre es que el alumno conozca distintas estrategias y las aplique a los largo de los trabajos prácticos basados en sistemas chicos que permiten realizar varias prácticas. [PAI88, DAV93 ,MCC92 ,JAC95 ,HAT88] Respecto de las metodologías para orientación a objetos, la presentación que se hace en este curso se complementa con una materia específica que los alumnos cursan de Programación orientada a objetos en el primer semestre de 3er. año.

En el segundo semestre se presentan los siguientes aspectos:

- ◆ Modelo de Proceso: Lineal y Evolutivo.
- ◆ Planificación de proyectos.
- ◆ Análisis de Riesgo.
- ◆ Estimaciones.
- ◆ Línea Base y Métricas.
- ◆ Calidad

En este segundo semestre los alumnos se dividen en equipos, a cada comisión se le asigna el desarrollo de un sistema de complejidad media. La cantidad de alumnos promedio de un curso de Ingeniería es de 150 alumnos, los auxiliares ofician como los clientes/usuarios que solicitan el desarrollo y los alumnos utilizan las clases prácticas como entrevistas en las que definen los requerimientos e iteran en la definición del sistema. El objetivo de este semestre es enfrentar al alumno con un desarrollo desde el inicio hasta la implementación, incluyendo en la entrega toda la documentación de soporte. [PUT92, JON96, JON94, KAN95]

Otro concepto que se ha incluido en el temario es el de Calidad: Aseguramiento de Calidad, Calidad Total, Modelo de Madurez de las Organizaciones de Desarrollo de Software, Normas ISO 9000. [CAR95, KEH96]

Herramientas utilizadas para la práctica

El enorme problema que existe es el costo de las herramientas para hacer Ingeniería de Software y las plataformas donde estas herramientas corren. En la actualidad se utilizan varios productos que se han conseguido como donación y fundamentalmente ambientes desarrollados por los Laboratorios de Investigación del Departamento y por los alumnos a través de la Tesina de la Licenciatura en Informática.

1.Herramientas de Administración de Proyectos (Ej.: Herramientas para Planificación Temporal)

2.Herramientas para especificación y modelización. (Ej.: Ambiente para modelización a través de Redes de Petri. [ABA94])

3.Herramientas para simulación. (Ej.: Ambiente para simulación de trayectorias)

4.Ambientes CASE que soportan variedad de metodologías (Ej.: CASE para metodologías estructuradas, para objetos, etc.)

5.Ambientes CASE con extensiones para Tiempo Real (Ej.: Ambiente ISTRAC [MAR95]).

6.Ambientes para evaluación de software a través de métricas.[PES96]

7.Derivadores de Código (Ej.: Herramientas para derivación de código ADA a partir de una especificación)

8.Ambientes para testeos y debugging

9. Herramientas para especificación de requerimientos (Ej.: Técnica JAD [DEG97,AUG91])

10. Herramientas para generación de Interfaces.

11. Ambientes para evaluación de organizaciones (Ej.: a través de la metodología CMM [BOR96]).

12.Trabajos en lápiz y papel.

Problemas planteados a los alumnos

A lo largo de los años se han ensayado 2 técnicas:

Definir un tema base y desprender Sistemas relacionados, lo cual tiene el atractivo que la etapa de Ingeniería de Sistemas puede ser compartida por los distintos grupos.

Definir temas distintos que barran áreas diferentes de desarrollo.

Como ejemplo del primer tipo de técnica hemos planteado sistemas más complejos, como por ejemplo automatizar el departamento de alumnos de la facultad, en cuyo caso hay toda una etapa inicial de definiciones compartida por todos los grupos (representados por sus líderes de proyecto) y una etapa posterior en que se distribuyen los distintos subsistemas entre los grupos para que apliquen las metodologías. Una ventaja de esta aproximación es que el proyecto puede tener mayor envergadura y la aplicación de las herramientas y metodologías se aproxima más a la realidad. Una desventaja es que la responsabilidad del éxito o fracaso del proyecto queda algo diluida entre los grupos y aparecen muchas zonas grises de difícil arbitraje. Una dificultad de esta aproximación es encontrar el sistema adecuado, ya que debería ser medianamente conocido, para que la etapa de definiciones no consuma la mayor parte del tiempo asignado al proyecto.

Como ejemplos del segundo tipo de técnica, hemos elegidos sistemas más bien pequeños que puedan ser implementados por un grupo de 3 a 5 alumnos, sobre algún tema de actualidad, algún sistema comercial o algún sistema de uso práctico cotidiano, para cumplir con la ley de la intensidad del aprendizaje al estimular el interés de los alumnos. A lo largo de los años hemos planteado sistemas de administración de alguna promoción (Serviclub de YPF, El Gran DT de Clarín), sistemas de agenda impositiva, sistemas de administración de algún comercio (videoclub, Farmacia, Supermercado) con algún detalle de procesamiento distribuido para agregarle algo de dificultad, etc. En algunos de estos casos hacíamos incapié en las interfaces, en otros, la complejidad estaba en el procesamiento.

En ambas técnicas los grupos de alumnos deben llegar hasta la implementación de los sistemas. La elección del lenguaje de programación la realiza cada grupo, debiendo justificar el uso de ese lenguaje de programación particular. En algunos casos el lenguaje elegido no se ajusta armoniosamente a la metodología de análisis y diseño estructurado, en cuyo caso los alumnos deben agregar a la documentación clásica, un escrito con una discusión de los problemas encontrados y las soluciones propuestas y adoptadas para lograr un ajuste lo más homogéneo posible.

Líneas de trabajo actuales

A través del proyecto de Investigación de "Automatización de oficinas. Aplicaciones Industriales" existente en el LIDI, se estudian ambientes de especificación basados en distintas técnicas.

La línea de trabajos de grado dirigidos apuntan a crear herramientas automatizadas que asistan a las distintas actividades del proceso de desarrollo.

La ampliación del ambiente de Métricas para procesar Software codificado en distintos lenguajes de programación (Ej.: orientados a Eventos o a Objetos), es motivo de análisis actualmente.

Es necesario poseer medios de estimación para sistemas con características propias (Ej.: sistemas de Tiempo Real)

Está previsto el desarrollo de un Ambiente de Simulación de trayectorias con restricciones implementado sobre robots manipuladores y con desplazamiento con los que se ha equipado el LIDI recientemente.

Conclusiones

La amplitud de las cuestiones relacionadas con Ingeniería de Software han marcado las pautas para variar a lo largo de los años el temario contenido en la materia. El resultado ha sido una reducción en los temas asociados con metodologías para analizar y diseñar un sistema, con el objetivo de aumentar la preparación de los alumnos en los temas de Administración de Proyectos.

Las innovaciones en las metodologías de análisis de requerimientos, hacen necesaria mantener la discusión abierta sobre el contenido técnico del primer semestre y por otro lado las estrategias y ambientes para seguimiento de proyectos se imponen día a día en las organizaciones. A nuestro entender, los esfuerzos que se realicen en pro de conocer y utilizar herramientas de gerenciamiento de proyectos para contener y controlar desarrollos, ayudan a obtener calidad que permite realizar un mantenimiento efectivo de los sistemas.

Bibliografía

[ABA94] M. J. Abasolo, M. Cantarella, M. Naiouf, P. Pesado, A. de Giusti, "Especificación y derivación de sistemas de tiempo real con Redes de Petri extendidas", Proceedings I International Congress of Information Engineering, 1994

[ART93] L. Arthur, "Improving Software Quality", Wiley, 1993

[AUG91] J. August, "Joint Application Design", Yourdon Press, 1991

[BOR96] M. Boracchia, L. Rodríguez, P. Pesado, A. De Giusti, "Un prototipo para Evaluación de Organizaciones de Desarrollo de Software", Proceedings 2do. Congreso Argentino de Ciencias de la Computación, 1996

[CAR95] Carnegie Mellon University, "The Capability Maturity Model", Addison Wesley, 1995

[DAV93] A. Davis, "Software requirements", Prentice Hall, 1993

- [DEG97] A. De Giusti, M. Balda, A. Vicenzi, R. Bertone, "JAD CASE: Tasks and documents specification for requirements engineering process", Proceedings XXIII Conferencia Latinoamericana de Informática, 1997
- [GHE91] C. Ghezzi, M. Jazayeri, D. Mandrioli, "Fundamentals of Software Engineering", Prentice Hall, 1991
- [HAT88] D. Hatley, I. Pirbhai, "Strategies for Real-Time System Specification", Dorset House, 1988
- [JAC95] M. Jackson, "Software Requirements and Specifications", Addison Wesley, 1995
- [JON94] C. Jones, "Assessment and control of software risks", Yourdon Press, 1994
- [JON96] C. Jones, "Applied Software Measurement", Mc Graw Hill, 1996
- [KAN95] S. Kan, "Metrics and Models in Software Quality Engineering", Addison Wesley, 1995
- [KEH96] R. Kehoe, A. Jarvis, "ISO9000-3", Springer, 1996
- [MAR95] M. Mariangelli, J. Spognardi, A. De Giusti, P. Pesado, "Diseño e implementación de un ambiente CASE para sistemas de tiempo real", Proceedings 1er. Congreso Argentino de Ciencias de la Computación, 1995
- [MCC92] C. McClure, "CASE, la automatización del software", Ra-Ma, 1992
- [PAI88] M. Page-Jones, "Structured Systems Structured", Yourdon Press, 1988
- [PES96] P. Pesado, C. Borroni, J. Zorzano, A. De Giusti, "Un ambiente para evaluación de sistemas de software mediante métricas clásicas", Proceedings 2do. Congreso Argentino de Ciencias de la Computación, 1996
- [PES98] P. Pesado "Programa de la asignatura Ingeniería de Software". Licenciatura en Informática. UNLP. 1998
- [PUT92] L. Putnam, "Measures for excellence", Yourdon Press, 1992
- [PRE98] R. Pressman, "Ingeniería de software", Mc Graw Hill, 1998