EXPERIENCIAS EN LA ENSEÑANZA DE INGENIERÍA DEL SOFTWARE (I.T. INFORMÁTICA DE GESTIÓN)

Miguel Angel Laguna Serrano
Departamento de Informática
E.U.P. Universidad de Valladolid
E-mail: mlaguna@dcs.eup.uva.es

La enseñanza de la Ingeniería del Software presenta problemas específicos, en particular en lo que se refiere a la extensión de las materias implicadas y al tamaño inabordable de los supuestos prácticos que sería aconsejable presentar para su estudio. Se presentan aquí algunas experiencias para tratar de solucionar esos problemas y las conclusiones extraídas.

Introducción y objetivos

En la construcción de sistemas de información se pueden distinguir dos fases genéricas, independientemente del paradigma de ingeniería elegido: la definición y el desarrollo, además del mantenimiento del sistema una vez puesto en marcha.

En la fase de definición se identifican los requisitos claves del sistema y del software. Por el contrario la fase de desarrollo está orientada al cómo, y el primer paso de esta fase corresponde al Diseño del Software. Los otros dos pasos de la fase de desarrollo corresponden a la codificación y a la prueba del software.

Los términos definidos anteriormente constituyen la materia básica objeto del estudio de la Ingeniería del Software, entendida como desarrollo de grandes sistemas de información. Siguiendo esta orientación, existen varios problemas que deben resolver quienes imparten cursos de ingeniería del software [SOM88]:

- Los estudiantes de informática carecen en los primeros años de su carrera de experiencia para percibir que los grandes sistemas informáticos no son simples versiones a escala de los problemas triviales que escriben como ejercicios. Sólo al final de la carrera, comprenden que la ingeniería del software no es lo mismo que la programación pura.

- Como el desarrollo de grandes sistemas necesita mucho tiempo, es casi imposible simular el proceso de una manera realista en el tiempo disponible para los alumnos. Este hecho hace que en muchos sitios se plantee la enseñanza de la ingeniería del software por partes y no como un todo.

La inmadurez de los estudiantes limita por fuerza el tratamiento de estos temas al último año de la carrera, aunque hay que poner énfasis desde el principio en la obtención de

programas fiables y bien documentados. Por otro lado debe tratarse de cursos eminentemente prácticos, dejando los contenidos fuertemente teóricos para otras asignaturas.

El problema que queda por resolver es la elección de trabajos prácticos apropiados, tarea difícil, dadas las limitaciones de tiempo. Esencialmente hay dos alternativas posibles:

- El proyecto elegido puede ser relativamente pequeño y bien definido, de modo que se pueda terminar durante el curso en el tiempo asignado.

- El proyecto puede ser más grande y amplio, en cuyo caso el alumno sólo puede abordarlo parcialmente.

Ninguna de las alternativas es totalmente satisfactoria. Si se escoge un sistema pequeño, las tareas de especificación son generalmente triviales y pueden parecer insignificantes, lo que no ocurre en la vida real, en la que ésta es la que fase más dificultades presenta el desarrollador, y ésto es precisamente lo que hay que mostrar al alumno.

Un enfoque de proyecto realista tiene la desventaja de que los alumnos no pueden terminarlo en un curso cargado de contenidos. Por ello nos remitimos al Proyecto Fin de Carrera como la práctica de desarrollo completo.

Como alternativa, durante el curso se puede desarrollar un proyecto práctico que haga de hilo conductor de los temas tratados, comenzando por las entrevistas realizadas en una empresa y la recopilación de los documentos empleados y de los objetivos deseados por la Gerencia, para ir cubriendo todas las etapas de análisis y especificación de un sistema informático lo más realista posible. En las fases de diseño por contra se puede tomar una parte representativa de la aplicación en lugar de intentar abordar un desarrollo completo.

Un objetivo, no secundario, es mejorar las capacidades de comunicación del alumno. Debe ponerse en práctica tanto en forma escrita como oral. El mero hecho de escribir informes técnicos es una buena manera de aprender a redactarlos, sobre todo si no se cuenta con el apremio del tiempo que supone la realización de un examen de fin de curso.

Un problema adicional lo constituye la interrelación entre asignaturas cercanas. Una visión completa de la ingeniería del software incluiría todas las etapas del ciclo de vida del software, desde la planificación de sistemas a la implantación y mantenimiento, pasando por el análisis, especificación, diseño, etc. Sin embargo, un aspecto fundamental del ciclo como es el diseño de programas y la prueba de los mismos constituye la columna vertebral de la asignatura de Programación II y no tiene sentido la repetición de los mismos temas. Por otra parte, el estudio del modelado de datos, sobre todo en su aspecto más formal se encuentra tratado perfectamente en la asignatura de Bases de datos y las métricas del software se enseñan en la disciplina de Calidad del Software.

En definitiva los temas tratados con detalle en nuestra asignatura son los correspondientes a las fases iniciales del ciclo de vida: planificación de sistemas, análisis de requisitos y especificación funcional y técnica del sistema. El diseño detallado, prueba y documentación de programas se incluirán en las asignaturas de Programación.

Los objetivos previstos para la asignatura vienen encuadrados en un objetivo global: producir profesionales que puedan resolver de forma sistemática y ordenada la producción de software de calidad, que responda a las necesidades y exigencias de las organizaciones. En concreto, el alumno de esta materia deberá:

- Conocer la terminología y los conceptos fundamentales propios de la asignatura.
- Conocer las técnicas y métodos utilizados en la asignatura a un nivel que le permita comparar, elegir y utilizar diferentes métodos.
- Identificar y establecer las fases y etapas que constituyen el desarrollo de un sistema de información.
- Comunicarse con el equipo de trabajo y con los usuarios finales.
- Identificar y modelar los requisitos del nuevo sistema que se pretende construir.
- Desarrollar el diseño de la aplicación considerando las características del equipamiento del que dispone para llevar a buen fin el proyecto.

Contenidos

Para conseguir estos objetivos hemos propuesto un programa diseñado para proporcionar al estudiante un cuerpo de conocimiento que incluya la cobertura de las actividades y herramientas del proceso de desarrollo de proyectos, sus aspectos y los productos que elabora, y una experiencia en el desarrollo de un pequeño proyecto, del que hablaremos más adelante.

A la hora de determinar el programa de la asignatura de Ingeniería del Software, además de los objetivos expresados, hemos tenido en cuenta el entorno en el que va a ser impartida y que incluye los elementos que a continuación se detallan y que condicionan el enfoque y contenidos de esta asignatura:

- En primer lugar se puede mencionar su orientación a la especialidad de gestión, que indica que los contenidos seguirán las técnicas y procedimientos de análisis y diseño desarrolladas para los sistemas de información empresarial.
- Hay que considerar que esta asignatura está dentro de una Ingeniería Técnica, es decir estudiantes no graduados con una formación orientada a los conceptos básicos y fundamentales y con un fuerte componente de práctica.
- Los ingenieros del software deben tener conocimientos complementarios tanto sobre la estructura y objetivos de la empresa, así como saber relacionarse de manera eficaz con diferentes niveles de usuarios y en el curso de su trabajo de análisis y diseño.

Teniendo en cuenta los elementos que acabamos de señalar, se ha pretendido elaborar un temario equilibrado que contemple las fases y elementos básicos de desarrollo de un proyecto de mecanización, que proporcione una formación de tipo practico y amplio, para que se pueda aplicar de forma inmediata y que cubra en la medida de lo posible la mayor parte de las etapas del ciclo de vida de desarrollo de un proyecto.

Las técnicas y procedimientos elegidos corresponden a los métodos de desarrollo de proyectos de gestión de más amplia difusión en España, como son MERISE [TAR86] y Análisis Estructurado de Sistemas (SSA, de origen norteamericano). El método SSA posee gran cantidad de variantes, incluyendo METRICA [MET95] que es el propuesto por el Ministerio para las Administraciones Públicas en España.

Hay que señalar que los contenidos de la asignatura no se ajustan a ningún método en concreto para no condicionar al alumno. Se ha pretendido formar en aquellas técnicas y procedimientos que son comunes a la mayoría de los métodos.

Se enfoca el desarrollo de la asignatura paralelo al de las fases principales del ciclo de vida, donde se estudia la descripción de las técnicas y las tareas a realizar para utilizarlas dependiendo de cada fase. La formación en las técnicas se realiza de modo que se aprenda cuándo, cómo, por qué y para qué se utilizan dentro del desarrollo del proyecto. Con este enfoque se consigue que la adaptación a cualquier grupo de trabajo sea lo menos costosa posible.

Y por último se han considerado elementos de formación complementaria que todo ingeniero del software necesita para poder llevar a cabo el desarrollo de sistemas: conocimiento de las estructuras de las organizaciones y de los centros de proceso de datos, nociones de sistemas de información y documentación.

Teniendo en cuenta las consideraciones de los apartados anteriores se fijan los siguientes contenidos genéricos:

TEMAS	Horas Teoría	Horas Práctic.	Horas Totales
I.- INTRODUCCIÓN Y CONCEPTOS BÁSICOS			
1. Conceptos básicos	3		3
2. Organización y dirección de proyectos software	3		3
II.- ANÁLISIS DE REQUISITOS			
3. Estudio de lo existente	5	3	8
4. Modelos de datos	6	6	12
5. Modelos de procesos (MCT)	7	5	12
6. Modelos de procesos (DFD)	5	6	11
(Problemas conjuntos)		11	12
TOTAL PRIMER CUATRIMESTRE	29	31	60
III.- DISEÑO LÓGICO Y FÍSICO			
7. Modelos externos y validación	6	3	9
8. Diseño lógico de datos y físico de datos	4	4	8
9. La Interfaz Hombre-máquina	4	2	6
10. Diseño de módulos programables	2	2	4
IV.- DESARROLLO DEL METODO			
11. Métodos Merise y METRICA 2	3		3
TOTAL SEGUNDO CUATRIMESTRE	19	11	30

El desarrollo del temario en horas se ha calculado teniendo en cuenta tanto el desarrollo teórico del tema como los ejercicios correspondientes que lo acompañan. Los doce créditos de la asignatura (seis teóricos y seis prácticos) se han repartido del siguiente modo: 9 créditos en clase (6 teóricos y 3 prácticos, correspondientes a los ejercicios) y 3 créditos de prácticas personales (proyecto) y seminarios. Los primeros nueve créditos corresponden a noventa horas de clases teóricas o de ejercicios que se reparten aproximadamente como queda reflejado en la tabla anterior.

Como se puede observar, los contenidos del primer cuatrimestre corresponden a cuatro horas semanales y los del segundo a dos horas. Esto es así porque la parte asignada a las prácticas de los alumnos se contabiliza exclusivamente a partir de febrero.

El primer cuatrimestre finaliza con un examen parcial liberatorio y tiene un peso del 50% en la asignatura. El segundo cuatrimestre y la nota de prácticas suman el otro 50%. El laboratorio, dotado de herramientas CASE, está a disposición de los alumnos desde el principio

de curso pero está pensado para el segundo cuatrimestre, cuando el estudiante ya tiene los conocimientos de análisis suficientes para desarrollar un pequeño proyecto.

En la primera parte, introductoria, se introduce el concepto de sistema y de sistema de información del que forman parte personas, software, hardware, procedimientos, datos e información. El tema 2 se inicia con una visión del Plan Informático de la empresa como guía de referencia del desarrollo de los proyectos informáticos, y se valoran las distintas posibilidades de inicio de los proyectos. Se proporciona una visión general de lo que puede ser la organización y dirección de un proyecto informático. Se estudia además el ciclo de vida de una aplicación informática, proporcionando una visión histórica de los ciclos de vida y un repaso a los ciclos de vida vigentes en la actualidad.

En la parte dedicada al Análisis de Requisitos se proporcionan las técnicas y métodos necesarios para analizar el sistema en detalle. Se detallan los métodos de recolección de datos con técnicas de entrevista, así como los soportes y métodos de análisis de los resultados de la investigación. Se estudian en detalle las técnicas de modelado de datos y tratamientos que se utilizan no solamente en el estudio del sistema actual y sus requisitos, sino que servirán para diseñar el sistema futuro. El objetivo consistirá en hacer ver la importancia de la determinación de las necesidades (requisitos) del usuario como condición necesaria para construir un sistema de calidad.

En la parte de Diseño Lógico y Físico se estudia la propuesta de varias soluciones técnicas que cumplan con las necesidades de la fase anterior. Se aborda el diseño lógico de datos y procesos así como su validación..Debido a la importancia creciente de la interfaz de usuario, se ha dedicado un tema exclusivamente a este aspecto específico del diseño. Por último, se analiza cómo completar el diseño lógico de datos y procesos con el diseño físico de los mismos, dependiendo de los condicionantes de hardware y software existentes en la organización. El objetivo es enseñar la traducción de las especificaciones lógicas a especificaciones físicas adaptadas a la arquitectura de los recursos disponibles.

Como colofón, se presentan en forma de seminario los detalles de los dos métodos que han servido como base para desarrollar la asignatura (Merise y METRICA), refiriéndonos fundamentalmente a la documentación que ha de generarse, los hitos de control, etc.

Métodos de enseñanza

La enseñanza se ha dividido en dos grandes bloques: 90 horas en el aula y 30 horas correspondientes a trabajo personal del alumno. Por otro lado, el plan de estudios incluye la asignatura como 6 créditos prácticos y 6 teóricos. Esta doble división nos lleva a plantear dos tipos de clases: la exposición teórica, si bien acompañada siempre de ejemplos clarificadores, y la clase de resolución de supuestos prácticos.

De las clases magistrales se ha escrito y debatido mucho a lo largo del tiempo pero es claro que constituyen una herramienta imprescindible para la transmisión de conocimientos conceptuales e ideas básicas de toda disciplina. El uso de transparencias puede hacer más amena la presentación de ciertos temas, pero sin dejar de lado el uso de la pizarra.

Pero para alcanzar los objetivos que nos hemos propuesto no son suficientes esas clases magistrales. Especialmente en aquellos temas que presentan técnicas determinadas, es necesario resolver variados problemas que afiancen los conocimientos que se desean transmitir.

Es conveniente incluir ejercicios de carácter sencillo y sistemático que permitan repetir, sobre casos particulares, técnicas y razonamientos invariantes de la asignatura documentados suficientemente durante el desarrollo de la exposición teórica. Con su resolución el alumno podrá afianzar sus conocimientos. Por otro lado se incluyen ejercicios conjuntos, tan reales como permita la capacidad y nivel de formación de los alumnos y cuya resolución implique una utilización simultánea de todos los conceptos aprendidos (de ahí las 11 horas dedicadas a resolver supuestos prácticos de modelado al final del primer cuatrimestre).

El papel de las prácticas

Las prácticas de laboratorio juegan un papel indispensable en la formación del alumno. Las prácticas proporcionan la posibilidad de obtener una experiencia directa del material básico de la asignatura; el alumno puede enfrentarse a problemas que exigen trabajar con las técnicas y métodos adecuados a la resolución de cada problema práctico planteado.

En el tema de la Ingeniería del Software, las prácticas en el laboratorio no son estrictamente necesarias. El desarrollo de especificaciones funcionales o la creación de modelos de datos, por ejemplo, se pueden abordar sin mayores problemas con lápiz y papel. Sin embargo, creemos conveniente la utilización de un laboratorio de Ingeniería del Software dotado con las máquinas y el software necesario para soportar las practicas con herramientas automatizadas de construcción de software cuyo uso está cada vez más extendido en la industria. Concretamente se dispone desde este curso de diez licencias de EasyCase para que el alumno desarrolle su proyecto.

Con el fin de evaluar la capacidad de utilización profesional de los conocimientos teóricos, esto es, los campos de las habilidades y capacidad de comunicación se propone a los alumnos la realización de un proyecto, guiado bajo enseñanza tutorizada.

La elaboración del trabajo se efectuará en grupos de dos o tres alumnos. El objeto y circunstancias determinantes de la situación a mecanizar es de libre elección, concretándose en los primeros meses lectivos y requiere la aprobación previa del profesor para asegurar que por su envergadura y definición pueda concluirse durante el curso.

Los objetivos de este trabajo son:

- Poner en práctica y reforzar los conocimientos sobre procedimientos y herramientas que se imparten a lo largo del curso.
- Mostrar al estudiante alguno de los problemas que se presentan en el desarrollo de proyectos de mecanización.
- Desarrollar la capacidad de comunicación del estudiante y la de trabajo en equipo
- Aprender el uso de una herramienta CASE para el análisis y la validación.

La estructura del proyecto se adaptara a cada una de las partes del Análisis y el Diseño desarrolladas en el programa práctico de la asignatura. Ya se ha comentado en la parte de introducción que parece conveniente pedir a los alumnos un análisis completo y su validacion datos/tratamientos y el diseño detallado de algunas partes seleccionadas de la aplicación elegida. Se pretende que el alumno practique las técnicas de diseño de la interfaz, redacción de cuadernos

de carga, etc. pero sin llegar a abrumarlo puesto que en el proyecto fin de carrera ya tendrá la oportunidad y la necesidad de completar un proyecto hasta el final, incluyendo la programación y prueba del sistema.

La presentación del trabajo se hace actualmente por escrito, recomendando su adaptación al esquema de documentación de METRICA y el uso de la herramienta CASE mencionada

Actividades complementarias

Los Seminarios específicos son un instrumento que nos permite abordar la presentación de un tema muy determinado en pocos días. En nuestro caso en dos aspectos concretos: presentación de un método determinado (METRICA 2, por ejemplo) y de manejo de herramientas CASE disponibles en el laboratorio. En este caso se utiliza directamente la proyección en pantalla de la salida de un ordenador personal para enseñar el manejo de EasyCase de forma cómoda y simultánea para todos los alumnos.

Por último las tutorías, individuales o en grupos de dos o tres estudiantes se plantean con el objetivo de ayudar al alumno a discutir y esclarecer dificultades que surgen en algunas de las actividades docentes antes mencionadas. En particular, la mayor incidencia de las tutorías se refiere a problemas que encuentran los alumnos en el desarrollo de su proyecto práctico, que hace que se desborden las horas de prácticas oficialmente asignadas.

Entre las técnicas disponibles para la evaluación de este tipo de los conocimientos, se considera como más adecuada la realización de una prueba escrita del tipo supuesto práctico. Para la superación de los objetivos de la asignatura de Ingeniería del Software, se precisa la calificación de apto tanto en el proyecto como en la prueba escrita. Sin embargo, la calificación final no se obtendrá por media aritmética del resultado de las dos pruebas, sino que al proyecto se le atribuye un 25% y al supuesto el 75%, algo menor de la relación entre los créditos teóricos y prácticos. Entre las razones que motivan esta reducción se encuentran que la contribución de cada miembro del grupo a la realización del proyecto nunca puede determinarse con exactitud, y que su realización a "libro abierto", no asegura el conocimiento ágil que se debe poseer sobre las distintas técnicas y herramientas.

Conclusiones

Teniendo en cuenta las lecciones aprendidas en la aplicación de estas ideas, hemos extraído varias conclusiones.

La realización de un trabajo práctico en el que los alumnos no se enfrentan a un problema de libro sino a auténticos usuarios es fundamental para que capten la esencia de la materia y creemos estar en la línea acertada.

Por otro lado, dada la carga docente de los nuevos planes de estudio es necesario profundizar mucho más en la coordinación entre las distintas asignaturas implicadas en el desarrollo del software. Aunque los contenidos teóricos están razonablemente coordinados, queremos dar un paso más y optimizar el tiempo que el alumno pasa en los laboratorios de la Escuela. En este sentido, el primer paso que se propone consiste en unificar las prácticas de Ingeniería del Software y Bases de Datos (contemplando como opcional la asignatura de Calidad del Software): el mismo trabajo práctico de Ingeniería debe ser la base del trabajo de diseño e implantación de una base de datos en una máquina y con un gestor concreto. El

hecho de que las asignaturas se cursen simultáneamente y se recomiende su matriculación conjunta debe facilitar las cosas.

Se debe introducir una planificación más estricta de entregas parciales del proyecto, para obligar a los alumnos a un trabajo más continuado y formativo.

En la medida que lo permita la carga docente, se deben incluir presentaciones orales y defensas de los proyectos realizados, para mejorar las capacidades de comunicación verbal de los alumnos y no limitarse a la redacción escrita.

Se debe completar el laboratorio de Ingeniería del Software con otras herramientas CASE, en concreto generadores de código y otras herramientas de diseño de bajo nivel ("lower case") que completen la visión que recibe el alumno sobre el tema.

Por último, en el tema fundamental de contenidos, se manifiesta de manera cada vez más clara la necesidad de incorporar el modelado y diseño orientados a objetos en el curriculum del alumno de Informática. Nuestra propuesta pasa por incorporar estos conocimientos el próximo año, coincidiendo con la implantación definitiva del nuevo plan de estudios que supone un incremento en el número de créditos de las asignaturas de Programación e Ingeniería del Software.

Bibliografía

[SOM88] Sommerville I., "Ingeniería del Software", Addison-Wesley Iberoamericana, 1988.

[TAR86] Tardieu, H. y otros, "La Methode Merise" (2 tomos)", Les Editions d'Organization.1986.

[MET95] Ministerio para las Administraciones Públicas, "METRICA Versión 2.1, Metodología de Planificación y Desarrollo de Sistemas de Información", Tecnos, 1995.