

INGENIERIA DEL SOFTWARE III: Una Asignatura de Ingeniería del Software Orientada a Objetos

Rebeca Cortázar, Asunción Barredo, Gloria Zaballa, José Luis Del Val

Departamento de Ingeniería del Software, ESIDE - Universidad de Deusto

Avda. Universidades, 24 48007 Bilbao (España)

Tlfn: +34-94-4139000 Fax: +34-94-4139101

E-mail: (cortazar, barredo, gloria, val)@inf.deusto.es

Resumen

En esta contribución describimos el programa de la asignatura Ingeniería del Software III, dentro del plan de estudios implantado en 1993 en la Facultad de Informática de la Universidad de Deusto. Esta asignatura es troncal y se imparte para los alumnos de 5º curso durante el primer cuatrimestre. Mediante su docencia, nuestra intención es acercar a nuestros alumnos los conceptos de la disciplina ya adquiridos en cursos anteriores, pero desde el punto de vista de la Tecnología Orientada a Objetos. Esta asignatura se ha impartido durante este curso por vez primera y en este artículo también comentamos las incidencias de esta primera edición.

Palabras Clave: Docencia, Ingeniería del Software, Orientación a Objetos, UML, Primera Experiencia.

1. Introducción

El estudio de la Ingeniería del Software es una parte fundamental en los currícula de las Ciencias de la Computación y como ha sucedido en otras áreas, la enseñanza de esta materia ha

evolucionado en paralelo al desarrollo de la Tecnología Informática.

En España, los últimos cambios sufridos en la enseñanza de esta materia han quedado reflejados en las directrices que el Ministerio de Educación y Ciencia ha fijado para la elaboración de los planes de estudio de las titulaciones universitarias de Informática. En cuanto a nuestro caso en particular, en la Facultad de Informática de la Universidad de Deusto, en 1993 comenzamos la implantación de un nuevo plan de estudios, pasando la titulación de ser una Licenciatura a ser una Ingeniería en Informática, cuya estructura era de dos ciclos: un primer ciclo de dos años y un segundo ciclo de tres. Con esta estructura y debido a que el Ministerio exigía que los créditos asignados a la Ingeniería del Software debían impartirse en el segundo ciclo, decidimos ubicarlos de la siguiente forma: dos asignaturas en cada cuatrimestre de cuarto curso, llamadas *Ingeniería del Software I y II*, y dos en el primer cuatrimestre del quinto, llamadas *Ingeniería del*

Software III y Planificación y Gestión de Proyectos.

En 1996, se decidió cambiar de nuevo la estructura de los estudios en Informática ofreciendo una titulación terminal de primer ciclo, Ingeniería Técnica en Informática de Gestión con una duración de tres años, y un segundo ciclo de dos, al final de los cuales se obtiene el título de Ingeniero en Informática. Este nuevo planteamiento curricular provocó una reorganización de los contenidos de las materias puesto que tanto en la Ingeniería Técnica, como en el segundo ciclo superior se deben impartir asignaturas de Ingeniería del Software.

Por este motivo, la planificación de la asignatura Ingeniería del Software III, que pertenece al plan del 93, se planteó como una asignatura con una vida de tres cursos académicos (97-98, 98-99, 99-00), cuyos contenidos cambiarían de situación en el plan de estudios implantado en el 96.

Para planificar el contenido de la asignatura, nos basamos obviamente en los conocimientos que nuestros alumnos habían ido adquiriendo a lo largo de los cursos anteriores. Por un lado, habían cursado las asignaturas de Programación y Bases de Datos tradicionales y en las de Ingeniería del Software, habíamos impartido los conocimientos generales sobre la disciplina y los Métodos Estructurados. Por otro lado, también habían recibido una asignatura de Programación Orientada a Objetos. Decidimos que los contenidos de Ingeniería del Software III debían ser realmente de *Ingeniería del Software Orientada a Objetos*; es decir, debíamos impartir

los conocimientos de las distintas áreas de la disciplina, pero centrándonos en esta tecnología en concreto.

Esta asignatura se ha impartido por vez primera en el curso 97-98, siguiendo el planteamiento que se expone a continuación.

2. Programa de la asignatura Ingeniería del Software III

Esta asignatura es troncal y tiene 6 créditos, con una carga lectiva teórica de cuatro horas de clase a la semana, durante el primer cuatrimestre de 5º curso.

2.1 Objetivos generales

Se pretende que el alumno adquiera una visión de la Ingeniería del Software Orientada a Objetos. Al finalizar el curso, los alumnos deberán poseer conocimientos acerca de los distintos métodos para llevar a cabo un proyecto orientado a objetos y en particular, deberán dominar la notación UML y el método OMT.

2.2 Objetivos específicos

- Conocer los conceptos básicos de la Orientación a Objetos.
- Conocer los principios fundamentales de la Orientación a Objetos.
- Comprender y diferenciar los distintos modelos de un dominio.

- Dado un dominio, ser capaz de elaborar un Modelo de Objetos y un Modelo Dinámico.
- Conocer las técnicas y la notación de UML.
- Ser capaz de aplicar los distintos pasos del ciclo de vida de un producto orientado a objetos.
- Conocer las técnicas y estrategias de prueba del software orientado a objetos.
- Distinguir las ventajas, inconvenientes y omisiones de cada método analizado.
- Conocer las tendencias para el desarrollo de componentes.
- Conocer y aplicar los patrones de diseño.

2.3 PROGRAMA TEORICO

- 1 Introducción
 - 1.1 ¿Qué es la Orientación a Objetos?
 - 1.2 Desarrollo Orientado a Objetos y principios fundamentales
 - 1.3 El Proceso OMT y Técnicas UML
 - 1.4 Conceptos Básicos de Arquitectura
- 2 Modelización: conceptos
- 3 Los Diagramas de Casos de Uso
- 4 Modelización de Objetos
 - 4.1 Objetos y Clases
 - 4.2 Enlaces y Asociaciones
 - 4.3 Generalización y Herencia
 - 4.4 Agregación
 - 4.5 Clases Abstractas
- 5 La Modelización Dinámica
 - 5.1 Sucesos y Estados
 - 5.2 Escenarios / Diagramas de Interacción / Diagramas de Secuencia
 - 5.3 Diagramas de Estados
 - 5.4 Concurrencia
- 6 Notación y Diagramas Adicionales de UML
 - 6.1 Diagramas de Paquetes
 - 6.2 Diagramas de Actividad
 - 6.3 Diagramas de Despliegue
- 7 La Metodología OMT de Rumbaugh
 - 7.1 Conceptualización
 - 7.2 Análisis del Dominio
 - 7.2.1 Construcción del Modelo de Objetos
 - 7.2.2 Construcción del Modelo Dinámico
 - 7.2.3 Iteración del Análisis
 - 7.3 Diseño del Sistema
 - 7.3.1 Subsistemas
 - 7.3.2 Concurrencia y asignación a Procesadores y Tareas
 - 7.3.3 Administración de Datos
 - 7.3.4 Arquitecturas comunes
 - 7.4 Diseño de Objetos
 - 7.4.1 Diseño de Algoritmos
 - 7.4.2 Diseño de la Herencia
 - 7.4.3 Diseño de Asociaciones
 - 7.4.4 Empaquetamiento
 - 7.4.5 Documentación
- 8 Diseño del Componente de Gestión de Datos
 - 8.1 Arquitectura del Componente de Gestión de Datos
 - 8.2 Correspondencia entre un Modelo de Objetos y un Modelo Relacional
 - 8.3 Patrón de Persistencia de Objetos para Bases de Datos Relacionales
- 9 Verificación y Validación del Software Orientado a Objetos
- 10 Aspectos de Gestión
 - 10.1 Ciclos de Vida
 - 10.2 Organización y Gestión del Proyecto
 - 10.3 Métricas Orientadas a Objetos
- 11 Comparación de Métodos Orientados a Objetos

- 11.2 Booch
- 11.3 HOOD
- 11.4 Diseño por Responsabilidad de Wirfs-Brock
- 12 Componentes
- 13 Patrones de Diseño

2.4 Programa Práctico

A lo largo del cuatrimestre los alumnos deberán realizar un trabajo práctico obligatorio de Análisis y Diseño Orientado a Objetos. Para ello constituirán grupos de cuatro alumnos. El seguimiento del trabajo se realizará mediante entrevistas en las que será obligatoria la presencia de todos los componentes del grupo. El trabajo práctico es requisito indispensable para superar la asignatura.

Es obligatorio el uso de una Herramienta CASE, pudiendo elegir entre WithClass 97, Paradigm Plus o Rational Rose.

Como resultado del trabajo práctico, se entregarán dos documentos, debidamente encuadernados: el documento de especificación de la fase de Análisis y el documento de especificación de la fase de Diseño. Dichos documentos deberán tener la composición que se define a continuación.

Documento de Especificación de Requisitos del Software:

- 1 Índice
- 2 Introducción
 - 2.1 Objetivos a cumplir: propósito y ámbito.
 - 2.2 Descripción general
- 3 Catálogo de Requisitos
 - 3.1 Diagramas de Casos de Uso
 - 3.1.1 Descripción de los Actores
 - 3.1.2 Descripción de los Casos de Uso
- 4 Modelo de Objetos
 - 4.1 Diagrama de Clases
- 5 Modelo Dinámico
 - 5.1 Diagramas de Interacción
 - 5.2 Diagramas de Estados (sólo clases con comportamiento dinámico importante)
- 6 Apéndices

Documento de Especificación del Diseño:

- 1 Índice
- 2 Introducción
 - 2.1 Objetivos del sistema
 - 2.2 Principales funciones del software
 - 2.3 Restricciones y limitaciones del diseño
- 3 Estructura de la Arquitectura del Sistema
 - 3.1 Diagrama de Paquetes
 - 3.2 Diagrama de Despliegue
- 4 Diseño del Modelo de Objetos
 - 4.1 Diagrama de Clases detallado y refinado
- 5 Diseño de Datos
 - 5.1 Modelo de Datos de la Base de Datos Relacional
 - 5.2 Diseño del Componente de Gestión de Datos
- 6 Diseño de la Interfaz
 - 6.1 Prototipo de la interfaz.
- 7 Diseño de la Interacción con otros sistemas
- 8 Apéndices

2.5 Organización

Los contenidos teóricos se impartirán en aula, debiendo desarrollarse tres casos que muestren los pasos del proceso de desarrollo, la notación UML y la división en capas arquitectónicas. Estos casos son el análisis y diseño de una Red Bancaria de Cajeros Automáticos, de una Terminal Punto de Venta y de un sistema de Recolección de Datos Meteorológicos.

En cuanto a la organización y seguimiento del trabajo práctico, esto se realizará a través de cuatro entrevistas obligatorias. El objetivo de cada entrevista será el siguiente:

- 1º Presentación, planteamiento y definición de la práctica; delimitación de la envergadura, si procede. Comienzo aproximado: 27 de Octubre.
- 2º Entrega de documentación parcial y revisión de la fase de inicial de Análisis del Dominio. (Casos de uso y Modelo de Objetos inicial). Comienzo aproximado: 10 de Noviembre.
- 3º Entrega de documentación de análisis y revisión de la fase de Especificación de Requisitos del Software. (Modelo de Objetos refinado y Modelo Dinámico). Comienzo aproximado: primeros de Diciembre.
- 4º Entrega final de la documentación y revisión de la fase de Diseño. (Arquitectura del Sistema, Diseño de Datos, Diseño de la Interfaz e Interacción con otros sistemas). Comienzo: Enero.

2.6 Evaluación

Se realizará un examen escrito formado por dos partes: una teórica de cuestiones breves y un ejercicio de modelización completo. La evaluación del trabajo práctico será individual para cada alumno y valorada por el profesor de prácticas durante las cuatro entrevistas. La nota final estará formada en un 20% por la nota del trabajo práctico y un 80% la nota del examen, siendo la proporción de este un 20% la teoría y un 60% el ejercicio de modelización.

2.7 Bibliografía Básica

Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorensen, W. 1996. *Modelado y Diseño Orientados a Objetos*. Metodología OMT. Prentice-Hall, ISBN: 013-240698-5.

Rumbaugh, J; Booch, G.; Jacobson, I. 1997. Unified Modeling Language. Notation Guide Versión 1.1. Documentation Set en WWW: <http://www.rational.com>.

2.8 Bibliografía Complementaria

Jacobson, I.; Christenson, M.; Jonsson, P.; Övergaard, G. 1996. *Object Oriented Software Engineering. A use case driven approach*. Addison-Wesley, Revisión de 1996, ISBN: 0-201-54435-0.

Coad, P.; North, D.; Mayfield, M. 1995. *Object Models. Strategies, Patterns & Applications*. Prentice-Hall, ISBN: 0-13-108614-6.

Coad, P.; Mayfield, M.; 1997. *Java Design. Building better Apps & Applets*. Prentice-Hall, ISBN: 0-13-271149-4.

Binder, R.V. 1994. 'Design for Testability in Object Oriented Systems'. *Communications of the ACM*, Vol 37, Nº 9, Septiembre 1994, pg: 87-101.

McGregor, J.D.; Korson, T.D. 1994. 'Integrated Object Oriented Testing and Development Process.' *Communications of the ACM*, Vol 37, Nº 9, Septiembre 1994, pg: 59-77.

Rumbaugh, J. 1995. OMT: The Development Process. *Journal of Object Oriented Programming*, Vol. 8, Nº 2, Mayo 1995, pg: 8-16.

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. 1995. *Design Patterns. Elements of Reusable Object Oriented Software*. Addison-Wesley, ISBN: 0-201-63361-2.

Yourdon, E.; Whitehead, K.; Thomann, J.; Opper, K.; Nevermann, P. 1995. *Mainstream Objects. An Analysis and Design Approach for Business*. Prentice-Hall, ISBN: 0-13-209156-9.

Henderson-Sellers, B. 1996. *Object Oriented Metrics. Measures of Complexity*. Prentice-Hall, ISBN:0-13-239872-9.

3. La primera edición de la asignatura

Se matricularon un total de 265 alumnos divididos en tres grupos de aproximadamente 90 personas

para las clases teóricas, y unos 70 grupos de prácticas.

Al finalizar esta primera edición, llegamos a la conclusión de que nuestro programa teórico había resultado algo ambicioso, considerando que contábamos únicamente con unas 48 horas lectivas. La modelización resultó una actividad difícil para nuestros alumnos (que acabaron el cuatrimestre con la impresión que era la asignatura más complicada que habían cursado). Por ello, tuvimos que hacer hincapié en la primera parte del programa y no llegamos a impartir los temas 9, parte del 10, 11, 12 y parte del 13. Considerando que no habíamos detectado problemas de asimilación en las asignaturas anteriores del área, nos extrañaron los comentarios de nuestros alumnos. Un análisis un poco más profundo de la situación nos llevó a la conclusión de que el problema radicaba en el hecho de intentar enseñar métodos de análisis y diseño, sin una sólida base de programación. Así, nuestros alumnos habían asimilado perfectamente los conceptos relacionados con la Metodología Estructurada, fundamentados en sus conocimientos de programación; sin embargo, su contacto con la Programación Orientada a Objetos había sido escasa, y demostró ser insuficiente. Estos alumnos comenzaron su carrera en 1993 y son los primeros en cursar este plan de estudios; los que comenzaron sus carreras en años posteriores, han tenido oportunidad de asistir a seminarios sobre Lenguajes y Herramientas Orientadas a Objetos y por ello, el perfil de los alumnos que recibirán la asignatura en el 98-99 es distinto, con lo que esperamos poder terminar el programa propuesto.

En cuanto a las prácticas, éstas transcurrieron sin incidencias y sólo un grupo abandonó antes de terminar. La dirección y seguimiento de las prácticas resultó una actividad muy absorbente a la que se dedicaron tres profesores. Como hemos mencionado anteriormente, se inscribieron aproximadamente unos 70 grupos, mayoritariamente de cuatro personas. Como nuestros alumnos tienen que realizar un proyecto fin de carrera durante el último curso, algunos decidieron aprovechar esfuerzos y comenzar la especificación de sus proyectos de manera que les sirviera como trabajo práctico de la asignatura. Esto nos llevó a admitir grupos formados por dos y tres personas e incluso trabajos individuales.

Para el seguimiento y control de las prácticas organizamos un esquema de cuatro entrevistas formales obligatorias, reservando un espacio de tiempo de media hora para cada grupo. Espaciamos las entrevistas dejando aproximadamente quince días entre ellas, periodo en el que los grupos preparaban la documentación a entregar y evaluar. Así, los tres profesores involucrados habíamos planeado que nuestro papel iba a consistir en hacer de 'usuario insatisfecho', planteando dudas, inquiriendo sobre decisiones de modelización, poniendo pegos, añadiendo requisitos, etc. De esta manera, en base a las respuestas de cada uno de nuestros alumnos, podíamos realizar una evaluación individualizada. Además, para garantizar mejor esta evaluación individual, pedimos a cada componente del grupo que se responsabilizara de un subsistema o parte del proyecto, de modo que se convirtiera en un 'experto' dispuesto a contestar a cualquier

pregunta sobre dicha parte. (Una de las cuestiones que nos planteamos a la hora de decidir el número de componentes de los grupos fue cómo valorar el nivel de trabajo individual que cada miembro iba a realizar, considerando que era relativamente fácil que uno o dos componentes se escamotearan del trabajo; esta división de tareas y el hecho de tener que contestar a cuestiones directas planteadas por el profesor resultó un éxito; todos participaron activamente en la realización de trabajo práctico. De hecho, estamos utilizando esta estrategia en todas las prácticas de las asignaturas del área).

Como las entrevistas formales estaban planteadas como hitos de entrega de documentación, publicamos unas horas de atención a las prácticas, ajenas a las entrevistas, para la resolución de dudas y problemas. Hay que mencionar que la mayor parte de los grupos superaron con éxito los requisitos planteados en las entrevistas, entregando documentación que apenas necesitó ser corregida, gracias a que al menos visitaron a sus profesores de prácticas una media de dos veces antes de la entrevista formal. Con la mayor parte de los grupos, hicimos el papel de 'usuario insatisfecho' durante las visitas informales de dudas en las que corrigieron sus modelos una y otra vez. Por supuesto, también hubo grupos que se presentaron únicamente en las convocatorias obligatorias y nos atreveríamos a decir que todos ellos vieron sus documentaciones rechazadas, debiendo rehacerlas para la siguiente entrevista.

4. Conclusiones

Durante este próximo curso 98-99, esperamos cubrir el programa completo de la asignatura, ya que contaremos en primer lugar con alumnos que han tenido una experiencia mayor en Programación Orientada a Objetos y por lo tanto, es posible que podamos explicar de un modo más sucinto la primera parte de la materia. Así, esperamos cubrir el plan teórico completo y si hubiera alguna hora lectiva adicional, nos proponemos introducir Conceptos de Interoperabilidad. En relación a la coordinación de las prácticas, es probable que el esquema siga siendo el mismo, puesto que nos ha dado muy buenos resultados.