Aprendizaje Orientado por Proyectos: Una Aplicación en los Cursos de Ingeniería de Software.

Abraham E. Dávila Ramón
Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería, Grupo de Investigación y Desarrollo en Ingeniería de Software.
Lima, Perú, Lima 32
edavila@pucp.edu.pe

Abstract

The Project Oriented Learning (POL) is a didactical technique that it is used in our software engineering courses at Informatic Engineering School. We have use POL for many years ago and its benefits are richer than only acquire knowledge. In this paper, show the educational innovation have been doing in our courses since these courses taught first time until nowadays, in particular, describe the educational model used in our software engineering courses today.
Keywords: Project Oriented Learning, Education, Software Engineering, Teaching Methods.

Resumen

El Aprendizaje Orientado por Proyectos es una técnica didáctica que se viene empleando en los cursos del área de Ingeniería de Software de la Especialidad de Ingeniería Informática desde hace varios años y cuyos beneficios van más allá de la adquisición de conocimiento. En el presente artículo, se presenta la innovación pedagógica realizada de manera continua desde su introducción hasta la fecha, y en particular, se describe el actual modelo didáctico empleado en los cursos de ingeniería de software.
Palabras claves: Aprendizaje Orientado por Proyectos, Educación, Ingeniería de Software, Modelo Didáctico.

1. INTRODUCCIÓN

En el año 1990 se crea la especialidad de Ingeniería Informática en la Facultad de Ciencias e Ingeniería (FCI) de la Pontificia Universidad Católica del Perú (PUCP) [8]. La primera vez que se dictó el curso de Ingeniería de Software y los cursos-taller complementarios (desarrollo de programas 1 y desarrollo de programas 2) se empleó un modelo didáctico diferente del resto de cursos. El modelo didáctico por un lado se basó en un equipo de estudiantes que desarrollan un proyecto de software desde la comprensión del problema hasta su total implementación; cubriendo todas las etapas usuales de un proyecto real de desarrollo de software, y que por otro lado, se complementa con las sesiones teóricas correspondientes.

La forma en que se dictó por primera vez y la forma en que actualmente se dictan todos los cursos obligatorios en el plan de estudios de la línea de ingeniería de software, ha variado de manera significativa principalmente por las experiencias docentes que se ha obtenido en todos estos años en las sucesivas innovaciones introducidas. Los proyectos que desarrollan los estudiantes, durante el semestre académico, tienen diferentes características entre curso y curso, y buscan en cada caso, emular situaciones reales de los proyectos de desarrollo de software considerando las restricciones propias del ámbito académico y buscando enfrentar a los estudiantes a diversas situaciones tal como ocurren en la vida profesional.

En el año 2000, se presentó a todos los miembros de la comunidad universitaria de la PUCP, el Plan Estratégico Institucional (PEI) 2000-2010 [9]; en él se estableció, como una meta, la introducción de metodologías que mejoren la actividad educativa en general; hasta ese momento la mayoría de los cursos se basaban en un modelo didáctico tradicional (sesiones magistrales). Tomando como marco de referencia el PEI y luego de una evaluación, por parte de la Comisión de Modernización Pedagógica (CMP) del Vicerrectorado Académico, se optó por introducir métodos activos y colaborativos [10], siendo uno interesante la técnica didáctica ABP de Aprendizaje Basado en Problemas (también conocido como BPL de sus siglas en inglés de Based Problem Learning). La corriente que se inició entre los docentes por introducir ese modelo didáctico, nos ha llevado a ahondar y descubrir que el modelo que hemos seguido desde nuestros inicios se denomina Aprendizaje Orientado por Proyectos (AOP) y que es más conocido como POL (por sus siglas en inglés de Project Oriented Learning). AOP es una técnica didáctica muy cercana al ABP y la frontera entre uno y otro es una tenue línea que a veces no se distingue [7]. Nosotros lo optamos pues sentimos que era una manera natural de aprender a administrar los proyectos de desarrollo de software.

Nuestro actual modelo didáctico se encuentra adaptado a cada uno de los cursos que comprenden la línea del área de ingeniería de software y considera las diversas variables como: número de estudiantes por equipo, tipo de proyecto de software, cantidad de entregables del proyecto y conocimientos obtenidos en los cursos.

En este documento, revisaremos brevemente los modelos didácticos de ABP y POL, luego se revisará el dictado de los cursos de la línea de ingeniería de software desde sus inicios y los cambios sucesivos hasta la actualidad. Finalmente se presentará una evaluación cualitativa y las conclusiones y cambios futuros a introducir en esta línea de trabajo.

2. APRENDIZAJE BASADO EN PROBLEMAS.

La técnica ABP tiene sus primeras aplicaciones en la década de los 60's en la escuela de medicina de la Universidad de Case Western Reserve en EEUU y en la Universidad de McMaster en Canada [4] planteándose situaciones de la vida real como un mecanismo integrador de los conocimientos necesarios –de diversas áreas del conocimiento- para resolver el problema.

ABP se define, según [3], [4], como un método didáctico de aprendizaje en el que grupos pequeños resuelven un problema propuesto por el docente y en cuya resolución se logra aprendizajes significativos. Los beneficios del uso del ABP se pueden resumir según [3] en los siguientes: (i) pensar críticamente y ser capaz de analizar y resolver problemas complejos de la vida real; (ii) encontrar, evaluar y utilizar las fuentes de información adecuadas; (iii) trabajar cooperando en equipos y grupos pequeños; (iv) mostrar habilidades versátiles y eficaces de comunicación, tanto verbalmente como por escrito; y (v) usar el conocimiento de los contenidos y las habilidades intelectuales adquiridas en la universidad para convertirse en permanentes estudiantes (que pueden aprender permanentemente).

3. APRENDIZAJE ORIENTADO POR PROYECTOS.

Uno de los primeros trabajos de AOP -no como una técnica didáctica propiamente- se presenta en el trabajo de Kilpatrick sobre "Desarrollo de Proyectos" en 1918 según [5] y cuya visión global abarca el proceso completo del pensamiento desde la idea inicial hasta la solución completa del problema.

AOP se define, según [5],[12],[6],[2], como un método didáctico de aprendizaje en el que los estudiantes toman una mayor responsabilidad de su propio proceso de aprendizaje aplicando en un proyecto sus habilidades y conocimientos adquiridos previamente. Sin embargo, según [11], para determinar si un proyecto es o no de AOP, se debe de cumplir: (i) los proyectos son componentes centrales y no periféricos al currículum; (ii) los proyectos se enfocan en problemas que inducen a los estudiantes a enfrentarse a los conceptos y principios básicos de una o varias disciplinas; (iii) los proyectos implican a los estudiantes en un proceso de investigación creadora; y (iv) los proyectos son dirigidos, en gran medida, por los mismos estudiantes.

Uno de los trabajos más referenciado sobre AOP es el de Blumenfeld y otros [1] en donde señalan, entre otras cosas, que los estudiantes resuelven problemas diversos y complejos para completar su proyecto al: (i) hacer y depurar preguntas; (ii) diseñar planes y/o experimentos; (iii) hacer predicciones; (iv) recolectar y analizar datos; (v) debatir ideas; (vi) establecer conclusiones; (vii) comunicar ideas; (viii) hacer nuevas preguntas; y (ix) crear artefactos. También resaltan que el proyecto es de una duración relativamente larga, centrada en un problema y que involucra conceptos de otras disciplinas y campos de estudio.

Usar AOP en un curso definitivamente introduce cambios en todos los participantes. Algunos de ellos, como lo señalan en [5],[13],[1],[11], están referidas: a la relación entre los profesores y los estudiantes, al cambio de actitud de competencia entre ellos hacia colaboración entre sus pares y de la memorización de contenidos al aprendizaje real pues lo necesitarán para resolver su problema.

4. DESCRIPCIÓN DE LOS CURSOS DEL ÁREA DE INGENIERÍA DE SOFTWARE.

Los cursos del área de Ingeniería de Software que son materia de este artículo son: Ingeniería de Software, Desarrollo de Programas 1 y Desarrollo de Programas 2. Ellos son parte de la columna vertebral en el último año y medio de estudios en nuestra especialidad y cada uno es un requisito para el siguiente.

Ingeniería de Software (IS) es el primer curso en donde se presentan metodologías de desarrollo de software y manejo de proyectos. Los estudiantes trabajan de manera paralela un proyecto de construcción de software acorde al nivel de los estudiantes durante todo el periodo lectivo. Las sesiones de teoría se llevan a cabo en aulas de clases y la supervisión del proyecto (sesiones de práctica del curso) en los laboratorios. Existe correspondencia entre lo que se dicta en clase y lo que se desarrolla en el proyecto.

Desarrollo de Programas 1 (DP1) es el primer curso de integración de conocimientos de todos los cursos previos. Se da un mayor énfasis a los temas de verificación y validación, gestión de la configuración, métricas e indicadores, entre otros. Los estudiantes trabajan un proyecto durante todo el periodo lectivo. Las sesiones teóricas se desarrollan principalmente en la primera parte del curso y durante todo el curso se supervisa y apoya el desarrollo del proyecto.

Desarrollo de Programas 2 (DP2) es el segundo curso de integración de conocimientos de la línea de Ingeniería de Software, que permite la integración de conocimientos y experiencias para el desarrollo de una solución de tecnologías de información en el dominio de los sistemas de información. Los estudiantes trabajan el proyecto durante todo el periodo lectivo. No hay sesiones teóricas, los estudiantes organizan las sesiones técnicas de acuerdo a las necesidades propias del proyecto y del equipo de desarrollo.

5. DICTADO INICIAL DE CURSOS DEL ÁREA DE INGENIERÍA DE SOFTWARE.

La primera vez que se dictó IS (1994-1), se intentó dirigir un proyecto completo entre todos los estudiantes, sin embargo debido a la falta de experiencia en este tipo de situaciones se tuvo que recortar algunas funcionalidades del sistema de información solicitado. Los estudiantes se entusiasmaron y construyeron el producto; fue una buena experiencia para ellos por que era su primer proyecto de principio a fin, es decir, desde la identificación de necesidad

del problema hasta su implantación. El contenido teórico fue revisado rápidamente y pasado a un segundo nivel de importancia por parte de los estudiantes.

La primera vez que se dictó DP1 (1994-2) se conformaron equipos pequeños para desarrollar software de acuerdo a los intereses de los propios estudiantes. La experiencia fue interesante pero la mayor dificultad era determinar una balance adecuado de la carga de trabajo entre los estudiantes.

La primera vez que se dictó DP2 (1995-1) se planteó la continuación de los trabajos inicialmente desarrollados en el curso previo; que permitió a los estudiantes completar ideas de productos interesantes y mejorar la versión anterior.

6. CAMBIOS EN LA PRÁCTICA DOCENTE EN LOS CURSOS DE INGENIERÍA DE SOFTWARE.

Los cambios en la práctica docente se fueron dando de acuerdo a las modificaciones del plan de estudios. Los cambios más significativos y en que cursos se efectuaron se presenta en la tabla 1 y tabla 2. Se han separado en dos tablas para dar énfasis a un cambio de contenidos radical que se produce en el periodo lectivo 2002-2 en los cursos IS y DP2.

Periodo	Curso	Cambios introducidos
1994-1	IS	Se dicta por primera vez el curso. Los estudiantes desarrollan un único proyecto trabajando en equipos.
1994-2	DP1	Se dicta por primera vez el curso. Los estudiantes en equipos pequeños desarrollan un proyecto.
1995-1	DP2	Se dicta por primera vez el curso. Los estudiantes en equipos pequeños continúan el proyecto que desarrollaron en el curso previo.
1996-1	DP1	Se introduce la lista de exigencias como herramientas para controlar el entregable final. La lista de exigencias es un catálogo de requisitos de donde los estudiantes identificaban aquellos requisitos de producto que al final tendrían que ser implementados por ellos. La lista de exigencias resultó ser practico pues los estudiantes y profesores podían negociar convenientemente el alcance del proyecto a desarrollar.
1996-2	DP1	Se introduce el uso de la evaluación de desempeño. Una herramienta para reportar el desempeño de los demás miembros del equipo; se empleó la valoración vigesimal de manera análoga a como se evalúa en el curso. La evaluación de desempeño se utilizó para identificar potenciales problemas entre los equipos y poder supervisar desde adentro el desarrollo del proyecto. Las reuniones semanales con los asistentes de docencia y los docentes complementaba la evaluación de desempeño.
1996-2	DP2	Se introduce el uso de la lista de exigencias. Los estudiantes desarrollan un proyecto en equipos pequeños, pero diferente al curso anterior y diferentes para cada equipo. Se definen una lista de diversos proyectos de donde los estudiantes elegían cual llevarían.
1997-1	DP1	Se definen trabajos con énfasis en la parte de computación (algorítmica). Además se define que al final del curso habrá una evaluación comparativa entre todas las soluciones (un ranking) que permita estimular las mejores soluciones. Se introduce una actividad de "auditoria" en el diseño. Todos los grupos rotan sus trabajos y tienen que informar si la documentación (del diseño) hasta ahora elaborada es posible de ser implementada. Los estudiantes se dan cuenta que la documentación es muy importante y exponen las deficiencias en clase.
1997-1	DP2	Se introduce la evaluación de desempeño cerrada (los estudiantes reportan evaluación de sus pares).
1998-1	DP1	Se cambia el criterio de uso en la evaluación de desempeño (solo en un horario de clases).
1999-1	DP1	Se introduce la co-evaluación y la evaluación de desempeño abierta al final de periodo. Se les pide a los estudiantes que hagan una evaluación sobre si mismo primero para que puedan fijar su nivel de referencia y luego evalúen al resto de su equipo de trabajo. Se definen cinco criterios de evaluación: conocimiento, puntualidad, motivación, responsabilidad, proactividad. La evaluación de desempeño se reportará por correo electrónico a los asistentes de docencia semanalmente.
1999-1	DP2	Se introduce la co-evaluación y la evaluación de desempeño abierta al final de periodo.

Tabla 1 Cambios en el periodo 1996-1 al 2000-1

Periodo	Curso	Cambios introducidos
2000– 2	IS	Se cambia el esquema de trabajo: se introduce el concepto de empresas de desarrollo y 3 frentes de desarrollo. Empresas con alrededor de 10 estudiantes compuesto de 3 frentes de 3 estudiantes en promedio.
2000– 2	DP1	Se formaliza el tamaño de equipo a 3 ó 4 estudiantes.
2000– 2	DP2	Se cambia el esquema de trabajo: se trabaja con una sola empresa por curso-horario que tiene la responsabilidad de desarrollar un sistemas de información tipo ERP con 6 grandes módulos. El número de estudiantes por horario es alrededor de 30 estudiantes.
2002– 2	IS	Se establece que se empleará Delphi y MSSql como herramientas de desarrollo de software.
	DP1	Se establece que el trabajo será desarrollado con Java en un esquema Cliente-servidor. Se usará XML para repositorio de información en los casos que se requieran guardar archivos.
	DP2	Se establece que se trabajará con Java en Web y usando un RDBMS de software libre (Postgresql).
2003– 1	todos	Se introduce un esquema de definición de equipos de trabajo basado en grupos parciales formados por el profesor a partir de los rendimientos académicos de los estudiantes; de modo que los equipos queden relativamente balanceados en capacidad de los alumnos.
2004– 1	IS	Se cambia la frecuencia de la evaluación de desempeño a quincenal y se cambia la valoración a percepciones de cumplimiento usando una escala de 0 a 10.

Tabla 2: Cambios en el periodo 2000-2 al 2004-1

7. LOS CURSOS DE INGENIERÍA DE SOFTWARE.

En la especialidad de informática se da énfasis a la parte algorítmica y de técnicas de programación en los primeros semestres. Luego los estudiantes toman cursos de modelamiento de sistemas de información estructurados y orientados a objetos y llevan el curso de bases de datos. Adicionalmente, en forma paralela, los estudiantes llevan el curso de sistemas operativos y otros de áreas complementarias. Los estudiantes ya tienen todas los conocimientos básicos necesarios para poder ejecutar un proyecto informático. El curso de IS brinda conocimientos teóricos sobre ingeniería de requisitos, manejo de proyectos (usando el PMBOK de PMI www.pmi.org), ciclos de vida de software, gestión de riesgos, calidad de software, estimación de software, entre otros y es el primer curso donde hacen un proyecto completo de software en equipos de trabajo desde el planteamiento hasta su implementación.

El curso de IS se dicta en el 8 nivel de estudios, el curso de DP1 se ubica en el nivel 9 y DP2 se ubica en el nivel 10. Cada uno es requisito del otro, pues consideramos que es un proceso natural de evolución y maduración del estudiante en el tema de manejo de proyectos. Las características de los cursos que actualmente se dictan en el área de ingeniería de software se presentan en las siguientes tablas resúmenes. La tabla 3, se presenta la características con un fuerte énfasis en el eje tecnológico.

Aspecto del Proyecto	Ingeniería de Software	Desarrollo de Programas 1	Desarrollo de Programas 2
Área Informática	Sistemas de Información	Ing. de computación / Ciencias de la computación.	Sistemas de información
Tipo de software	Cliente / servidor	Variado	Aplicación web
Lenguajes de Programación	Delphi, Object Pascal	Java	Java
Sistema Administrador de Bases de datos	MS-Sql Server	No usa RDBMS, sino XML, en caso requiera almacenar información	Postgresql o MySql
Arquitectura de Software	Cliente / Servidor	Elegido por los estudiantes	Tres capas
Proyecto del periodo 2002-2.	Sistema de compra, armado de paquetes y asistente de paquetes turísticos.	Software para la generación de datos de pruebas usando reglas de construcción.	Sistema para la Planificación de Recursos Empresariales (ERP).
Proyecto del periodo 2003-1.	Sistema de compra, venta y almacenamiento de una tienda de videos.	Software para la prueba de esfuerzo de aplicaciones desarrolladas en Java.	Sistema Integrado para Gobiernos Locales y Regionales.

Tabla 3: Características de los cursos de Ingeniería de Software en el eje técnico.

Las decisiones de porque cada curso cubre un tema diferente fueron acordados en reuniones de coordinación de los profesores del área. En el caso concreto de las herramientas de desarrollo empleadas, se han definido las que aparecen en el cuadro, pues en los cursos previos se ven otras herramientas y la idea es que el estudiante tenga conocimiento práctico en las que más se usan en la industria nacional. Por ejemplo el Sistema Administrador de Bases de Datos que se utiliza en el curso de Bases de Datos es Oracle, por lo que ya no se le considera en este esquema.

Aspecto del Proyecto	Ingeniería de Software	Desarrollo de Programas 1	Desarrollo de Programas 2
Total estudiantes	30	30	30
Estudiantes por empresa	10	3 a 4	30
Estudiantes por frente de trabajo	3 a 4	No aplica	6
Desarrollo de cada frente	Componente funcional.	No aplica	Componente funcional.
Coordinación interna	Sí	No	Sí
Comité de estándares	Sí	Sí	Sí

Tabla 4: Características de los cursos de Ingeniería de Software en el eje organización de equipos.

La tabla 4, presenta la información sobre los equipos de desarrollo que se conforman durante los cursos. Se emplea el concepto de empresa como principal agrupador (agrupador de mayor nivel) y el concepto de equipos de trabajo (frentes) para un siguiente nivel de agrupamiento. Por la naturaleza de cada proyecto, en algunos casos, es necesario establecer comités o coordinaciones que faciliten el tema de integración del software en desarrollo en sus distintas etapas; estas asociaciones son eventuales y en la medida de lo posible es conformado por un miembro de cada frente. El número de estudiantes por empresa y por frente es un valor referencial, y se definen en cada curso horario de acuerdo al total de inscritos.

La tabla 5, presenta la forma de participación del equipo docente en este tipo de cursos. Desde hace dos años establecimos que los asistentes de docencia tuvieran una experiencia profesional en manejo de proyectos mayor a 2 años como egresados de la universidad. Este requisito se ha ido cumpliendo paulatinamente y actualmente se cumple para todos los casos. La experiencia cuenta mucho al momento de trabajar con los estudiantes.

Aspecto del Proyecto	Ingeniería de Software	Desarrollo de Programas 1	Desarrollo de Programas 2
Nivel de participación de los asistentes de docencia en el proyecto.	Alta, guían –sin hacerles la solución- a los equipos de trabajo en sus decisiones. Influyen en las decisiones. Actúan como asesores del equipo.	Media, cuestionan las decisiones y ofrecen varias alternativas, pero se insiste que es el equipo quien asume la responsabilidad de la decisión. Actúan como asesores.	Baja, cuestionan todas las decisiones funcionales y técnicas. Actúan como auditores.
Nivel de participación de los docentes en el proyecto.	Supervisa a los equipos, a los asistentes de docencia y negocia los alcances del proyecto. Revisa y acuerda el plan de trabajo durante 3 diferentes momentos del proyecto.	Supervisa y negocia el alcance del proyecto al inicio y a la mitad del periodo. Introduce cambios en el escenario de desarrollo para que los equipos reaccionen ante las eventualidades.	Supervisa continuamente el desarrollo del proyecto y el alcance del mismo. La fecha de entrega es definida por la empresa.

Tabla 5. Características de los cursos de Ingeniería de Software en el eje de los docentes.

Los requisitos establecidos para poder llevar el curso de Ingeniería de Software, que es el primero de la línea, son los siguientes: (i) Sistemas de información 2 (modelado y diseño orientado a objetos con el Lenguaje Unificado de Modelado UML), (ii) Lenguaje de Programación 2 (Programación orientada a objetos), (iii) Bases de datos (diseño de bases de datos) y (iv) cursos de administración. Todo estos conocimientos previos permite que el curso se oriente más a la parte de gestión del proyecto de software y a los temas en procesos, metodologías, requisitos y calidad.

Los cursos definidos y manejados de la forma ya descrita, cumplen con los criterios establecidos por Thomas[11] para ser considerados del tipo AOP y el resumen se presenta en la tabla 6 y se aprecia las características señaladas por Blumenfeld [1] en la tabla 7. Los proyectos tienen una duración promedio de 15 semanas y demandan el conocimiento del dominio del problema.

Criterios para determinar si es AOP según Thomas	Ing.Sw	DP1	DP2
El proyecto es un componente central y no periférico	X	XX	XX
Los problemas inducen a enfrentamiento a los conceptos y principios ...	X	XX	X
El proyecto implica un proceso de investigación creadora.	X	X	X
El proyecto es dirigido principalmente por los alumnos.	X	XX	XXX

Tabla 6. Cumplimiento de criterios para ser considerados cursos AOP

La tabla 6, muestra los niveles de cumplimiento de cada criterio. La diferencia en el grado de cumplimiento se debe a que en IngSw existe más influencia del equipo docente sobre varios aspectos como la planificación maestra, la definición del proceso seguido y la supervisión más cercana de las decisiones. En cambio en el otro extremo está DP2, donde los alumnos toman el control total de todo el proyecto y el único dato de referencia es la fecha final del proyecto, todo lo demás lo hacen ellos. Con respecto al criterio de los problemas inducen al enfrentamiento con los conceptos, se nota más en DP1 porque el problema usualmente así lo amerita, al ser un problema más computacional las herramientas de modelado de sistemas de información no les resulta del todo adecuadas y ellos deben resolver dicha situación.

Conducta de alumnos observados en el caso de AOP según Blumenfeld.	Ing.Sw	DP1	DP2
Hacer y depurar preguntas	X	X	X
Diseñar planes y/o experimentos	X	XX	XXX
Hacer predicciones /estimaciones	X	X	XX
Recolectar y analizar datos	X	X	X
Debatir ideas	XX	X	XXX
Establecer conclusiones	X	XX	XX
Comunicar ideas	X	XX	XX
Hacer nuevas preguntas	X	X	X
Crear artefactos.	X	XX	XXX

Tabla 7. Conductas observadas en los cursos de AOP.

Las conductas observadas y resumidas en la tabla 7 están en correspondencia con las características propias de los cursos. En DP2 se aprecia con mayor énfasis las conductas observadas en la aplicación de AOP, pues los estudiantes tienen un control mayor del proyecto en si.

8. EL PROCESO DE SOFTWARE.

Luego de trabajar con diversos alternativas de ciclo de vida en los cursos, se optó por que en el curso de Ingeniería de Software se use un ciclo de vida de proyecto basada en dos iteraciones en la parte de construcción. En la figura 1 se muestra el ciclo de vida establecido por los profesores para dicho curso. En los cursos de DP1 y DP2 los estudiantes tienen la responsabilidad de definir el ciclo de vida de sus proyectos.

9. EVALUACIÓN EN LOS CURSOS DE INGENIERÍA DE SOFTWARE.

Antes de hablar sobre la evaluación a los estudiantes, es importante señalar, que los proyectos se desarrollan en aproximadamente 15 semanas de trabajo y tomando en cuenta la carga académica de los estudiantes, es correcto pensar que el producto final dista funcionalmente de un producto real. Cada curso, impone ciertas restricciones en la parte funcional al momento de la implementación final o establece un alcance de acuerdo a las negociaciones que se desarrollan entre profesores y equipos de trabajo.

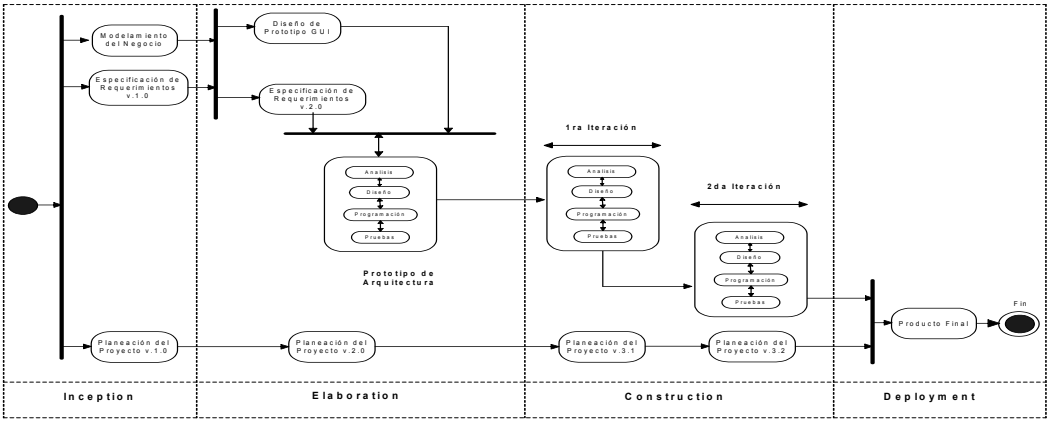

Figura 1. Ciclo de vida seguido para el curso de ingeniería de software.

La evaluación en cada uno de los cursos dependen de diversos componentes. Para cursos bajo AOP, la evaluación final depende fuertemente del producto desarrollado. Una práctica empleada es que el producto debe funcionar de manera correcta al final del periodo lectivo, si ello no ocurre entonces el estudiante no es aprobado. Si el producto funciona correctamente, entonces el estudiante, el trabajo y la gestión del proyecto son evaluados por los asistentes de docencia, el profesor del cursos y por los propios estudiantes. La evaluación se realiza durante todo el proyecto conforme se van entregando los avances, pero la calificación final queda supeditada al correcto funcionamiento del software de acuerdo a las funcionalidades establecidas.

10. CONCLUSIONES Y TRABAJO FUTURO.

Los cursos de Ingeniería de Software, Desarrollo de Programas 1 y Desarrollo de Programas 2 son ejemplos concretos de la aplicación del modelo de didáctico de Aprendizaje Orientado por Proyectos y que cumple con lo afirmado por Thomas[11] y Blumenfeld [1]. Aunque inicialmente no se partió del conocimiento de los modelos didácticos, la experiencia docente permitió derivarlos a la situación actual y que ofrece mayores beneficios según diversos autores [2],[3] y [4]. La experiencia obtenida nos está permitiendo introducir mejoras a los cursos previos.

Algunas mediciones que se están haciendo en el uso de ABP están arrojando resultados alentadores sobre los beneficios en su aplicación. De manera análoga, apoyados en una encuesta abierta –todavía- realizada sobre un grupo de egresados, comentan que el manejo de las diversas situaciones que se enfrentan en los cursos suelen ser similares a los que viven en las empresas actualmente y que les ha servido el haberlo vivido previamente para tomar decisiones.

La manera de evaluar a los alumnos ha tenido necesariamente que evolucionar desde los modelos tradicionales hacia modelos más convenientes. Un caso interesante se dan por el lado de la AutoEvaluación y la Co-Evaluación de los equipos de estudiantes sobre el producto desarrollado y sobre la gestión del desarrollo, en ella los estudiantes asumen una posición crítica sobre su trabajo y la de sus compañeros. Sobre la parte de evaluación de desempeño de los equipos de trabajo se está desarrollando una herramienta informática que apoye esta actividad.

Aplicar AOP a estos cursos resulta muy conveniente pues se trata de cursos integradores de conocimientos ubicados en los últimos semestres de la carrera. Sin embargo demanda un tiempo mayor por parte del equipo docente quienes se apoyan –como resulta obvio- en todas las bondades que ofrecen hoy las tecnologías de la información. El equipo docente (profesores y asistentes de docentes) trabajan relativamente más que en los cursos tradicionales (según lo que se observa), sin embargo el trabajo es más estimulante porque todos aprenden durante la ejecución del proyecto. Con apoyo del Centro del Magisterio Universitario (MAGIS-PUCP) se realizarán algunas mediciones sobre el uso de está técnica didáctica.

Agradecimiento

Deseo expresar mi agradecimiento a Silvana Díaz que preparó el diagrama de actividades para un trabajo interno de la especialidad, a Carla Basurto por las sugerencias y a Luis Bretel por animarme a escribir este artículo y por revisarlo.

Referencias

[1] Blumemfeld, P., Soloway, E., Marx, R., Krajcik, J., Guzdial, M., Palincsar A., Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. Educational Psychologist, 26(3&4), 369-398. Lawrence Erlbaum Associates, Inc, 1991.

[2] Bucciarelli, L. Project Oriented Learning as Part of Currículo Development. 3rd NTVA Industrial Ecology Seminar and Workshop. 1998. http://www.indecol.ntnu.no/indecolwebnew/events/conferences/ntva/3rd_ntva/3rd_ntva.htm

[3] Duch , B., Groh, S., Allen, D., ¿Por qué el Aprendizaje Basado en Problemas? Un Estudio de Casos del Cambio Institucional en la Educación de Pregrado. El poder del Aprendizaje basado en Problemas. Editado por Duch , B., Groh, S., Allen, D. (en proceso de publicación 2004).

[4] ITESM, El Aprendizaje Basado en Problemas como Técnica Didáctica. Dirección de Investigación y Desarrollo Educativo. Material de Curso. Vicerrectoria Académica, Instituto Tecnológico y de Estudios Superiores de Monterrey. 2003.

[5] ITESM, El método de proyectos como técnica didáctica. Dirección de Investigación y Desarrollo Educativo. Material de Curso. Vicerrectoria Académica, Instituto Tecnológico y de Estudios Superiores de Monterrey. 2003.

[6] Meyer, V, Project Oriented Learning (POL) as a Comunication Tool of Enviromental Sciences in the Community of Sohanguve – A Case Study. 7th International Conference on Public Communication of Science and Technology. 2002

[7] Oakey, J. Project-Based and Problem-Based: The same or different?. http://pblmm.k12.ca.us/PBLGuide/PBL& PBL .htm [13/02/2004 08:30:51 p.m.]

[8] Peralta, O. La Sección Ingeniería Informática. *Ingeniería en la PUCP*. Año 1. No 1, pp 37-39. 1999

[9] PUCP. Plan Estratégico Institucional. http://www.pucp.edu.pe/dape/ . Dirección Académica de Planeamiento y Evaluación. [16/04/2004 06:13:45 p.m.]

[10] PUCP. Sobre la Comisión de Modernización Pedagógica. http://www.pucp.edu.pe/cmp/ . Comisión de Modernización Pedagógica. [18/04/2004 02:32:29 p.m.]

[11] Thomas, J., A Review of Research on Project-Based Learning, http://www.autodesk.com/foundation. March 2000.

[12] Vélez de C, A. Aprendizaje Basado en Proyectos Colaborativos en la Educación Superior. Brasilia: IV Congreso RIBIE.1998. http://phoenix.sce.fct.unl.pt/ribie/cong_1998/trabalhos/190m.pdf

[13] Zahava, S., Polak S., An Organizer for Project-Based Learning and Instruction in Computer Science. Proceeding of the 4th annual SIGCSE/SIGCUE ITICSE Conference on Innovationa and Technology in Computer Science Education. 1999. pp. 88-90.