

Un modelo ágil de inserción de alumnos de ingeniería de software en la industria. El caso del curso CC61A “Proyecto de Software”

Agustín Villena Moya
avillena@dcc.uchile.cl

María Cecilia Bastarrica
cecilia@dcc.uchile.cl

Departamento de Ciencias de la Computación (DCC)
Facultad de Ciencias Físicas y Matemáticas (FCFM)
Universidad de Chile
Blanco Encalada 2120

Resumen

Desde el año 1998 se ha implementado en el DCC una experiencia en donde los alumnos desarrollan en un semestre un proyecto de software para resolver necesidades de organizaciones del mundo real, emulando un modelo de *outsourcing*. En un inicio se empleó el ciclo de vida de cascada, con resultados no siempre satisfactorios, teniendo muchos proyectos que no alcanzaban sus objetivos. Esto llevó a intentar diversas mejoras hasta que el año 2005 se implementa un enfoque ágil en la gestión de los proyectos, logrando mejoras sustantivas en la efectividad y repetibilidad de resultados de los proyectos. En este modelo se organiza el desarrollo en 3 ciclos de un mes cada uno, al final de cada cual se genera un producto funcional que es complementado en los ciclos siguientes. La planificación es adaptativa y puede ser ajustada en todo momento del proyecto. Se utilizan herramientas simples para la planificación y gestión de riesgos que permiten aprender y tomar decisiones correctivas oportunamente. La estructura del equipo de desarrollo es horizontal, debiendo los alumnos asumir roles multifacéticos en el desarrollo, y se enfatiza la habilidad de interpretar las necesidades de negocio de los clientes. En la actualidad se apunta a dar mayor fortaleza organizacional a esta experiencia para difundir esta experiencia y proyectarla hacia todo el mercado nacional de modo de servir no sólo como curso formativo para los alumnos, sino también como una forma de posicionar a nuestros egresados dentro del mercado chileno.

1. INTRODUCCIÓN

Las estadísticas de la industria de software en EEUU indican que en el 2006 sólo un 35% de los proyectos de software lograban el éxito [7], entendido éste último como la generación de un producto con funcionalidades suficientes, dentro de los plazos y costos definidos en un comienzo*. Lo anterior lleva a plantear que existe un gran trabajo por hacer en definir y aplicar buenas prácticas que mejoren el éxito y repetibilidad de buenos resultados en los proyectos de software. El Departamento de Ciencias de la Computación de la Universidad de Chile (DCC) asume en este sentido un desafío en su misión de liderar la investigación y adopción de las mejores prácticas para la industria informática nacional. Adicionalmente, el DCC tiene la necesidad de aportar a sus alumnos una experiencia donde ellos se enfrenten a

* No existen datos similares sobre la realidad de la industria chilena

problemas reales que ameriten un desarrollo de software en un contexto real y así adquirir destrezas importantes para su vida profesional. Ambos objetivos son los que se busca lograr en el curso denominado “CC61A – Proyecto de Software”, último curso obligatorio de la carrera de Ingeniería Civil en Computación.

En el año 1998 CC61A asume el modelo de emular una empresa de *outsourcing*, aportando equipos de entre 5 y 7 alumnos a organizaciones externas para resolver problemas reales. En ese entonces se usó el tradicional modelo de cascada, donde el desarrollo está guiado fuertemente por la generación de artefactos de documentación de análisis, diseño, pruebas, etc. El 2005 se cambió el enfoque hacia uno basado en metodologías ágiles, en particular recogiendo las prácticas de *Extreme Programming* correspondientes al ámbito de la gestión de proyectos, por sobre las prácticas de programación de código. Los proyectos asumieron el enfoque iterativo incremental, estructurándose en 3 ciclos de 4 semanas cada uno, debiendo generar incrementos de valor para el cliente a través de productos funcionales de software. La organización de los equipos pasó a ser horizontal con roles multifacéticos para cada integrante, y liderados por un coordinador de equipo. El trabajo pasó de ser a distancia a realizarse en las instalaciones del cliente, y se puso énfasis en el crear un sólo equipo junto al cliente con el objetivo de obtener buenos resultados. Desde entonces se ha logrado una mejora sustantiva en los resultados y en la repetibilidad de éstos, logrando una tasa de puesta en producción de los productos de software generados de un 94,7% hasta la fecha. Los clientes se han mostrado satisfechos con los resultados y muchos de ellos han participado nuevamente en la experiencia, lo cual también ha contribuido a que exista un número creciente de proyectos concursando cada semestre. Por último, y no menos importante, un número significativo de alumnos ha sido contratado por las empresas donde desarrollaron su proyecto de CC61A.

En la sección 2 de este artículo presentamos el modelo del curso CC61A, partiendo por una descripción global de éste y una reseña histórica de la evolución de este modelo. Luego se describen las fases en las que se estructura el calendario del curso, el modelo evaluativo utilizado para calificar el rendimiento de los alumnos, y los criterios de selección utilizados para escoger las organizaciones clientes. Finalmente se exponen las bases conceptuales sobre las que se basa el modelo actual del curso, en particular cómo se aplican las prácticas ágiles en él. En las conclusiones, presentamos los resultados logrados por el modelo actual del curso, y los desafíos pendientes para el DCC con respecto a esta experiencia.

2. EL MODELO DEL CURSO “CC61A – PROYECTO DE SOFTWARE”

CC61A es un curso obligatorio para la especialidad de Ingeniería Civil en Computación en la Universidad de Chile, que los alumnos realizan justo al final de la carrera, y está pensado como un ensayo controlado del mundo profesional. El modelo general de la experiencia es similar al de sub-contratación (*outsourcing*), donde la organización cliente interactúa con una contraparte proveedora de servicios de desarrollo la cual tiene una organización interna pre-definida y que no puede ser intervenida por el cliente. El proceso de definición de requisitos y de control de avance se realiza a través de medios e instancias formales. Ambas partes están representadas respectivamente por una Contraparte (Cliente) de la empresa cliente y un Coordinador de Proyecto del equipo de desarrollo. Este equipo es conformado por entre 5 y 7 alumnos del último año de la carrera de Ingeniería Civil en Computación, con dedicación efectiva de 16 horas semanales cada uno, para un proyecto concreto de desarrollo de 12

semanas de duración. Cada equipo es apoyado internamente por un ingeniero en computación con varios años de experiencia en proyectos de desarrollo de software quien asume el rol de tutor, controlando el avance del proyecto y aportando experiencia, pero sin intervenir directamente en el desarrollo. Los tutores están liderados por el profesor de cátedra quién asume un rol de gestión del curso, coordinando la selección de proyectos y relación con los clientes, custodiando el buen cumplimiento de las reglas por parte de los clientes y correcta aplicación de la metodología por los alumnos.

2.1. Evolución del curso

Hasta el año 1997, el curso CC61A consistía principalmente en una práctica profesional que requería de un trabajo de media jornada en una empresa durante un semestre. Esta modalidad no difería conceptualmente en nada de las otras prácticas profesionales también obligatorias dentro de la carrera. En 1998 se inicia en el DCC un ambicioso programa denominado “*Enseñanza y Transferencia Tecnológica en Ingeniería de Software*”, al alero del cual se reformuló el curso hacia un modelo de proyectos donde los alumnos, trabajando en equipos, realizaban proyectos para organizaciones externas por lo general a distancia pero con la supervisión del profesor del curso, en un intento de darle un carácter formativo guiando a los alumnos en buenas prácticas de ejercicio profesional. La metodología de desarrollo aplicada estaba basada en el tradicional ciclo de cascada y era guiada por el desarrollo de diversos artefactos de documentación (requisitos, diseño y pruebas, entre otros). Los equipos de desarrollo tenían roles definidos y especializados (jefe de proyecto, analista, diseñador, etc.). Los proyectos a partir de entonces obtuvieron resultados disímiles, teniendo proyectos altamente exitosos con clientes muy satisfechos, pero también otros clientes altamente disconformes y alumnos frustrados.

Como una medida para mejorar los resultados de los proyectos, a partir del año 2002 se incorporaron al equipo docente ingenieros de software con varios años de experiencia asumiendo el rol de tutores para cada uno de los proyectos a realizar, logrando una mejora paulatina pero sostenida en el éxito de los proyectos como consecuencia de tener un mayor control sobre ellos y de la experiencia profesional aportada por los tutores. Sin embargo, la mejora más sustancial en los resultados se obtiene a partir del año 2005 cuando se reemplaza el enfoque metodológico por uno basado en gestión ágil extraída de metodologías tales como *Extreme Programming* o *Scrum*, modelo que se mantiene hasta la actualidad.

2.2. Bases conceptuales del el curso

Actualmente el curso utiliza un modelo ágil de gestión para sus proyectos. Las metodologías ágiles de desarrollo de software han tenido un impacto sustantivo en la industria mundial del software [4], y se basan en los “principios ágiles”[1], que presentamos en la Tabla 1.

Individuos e interacciones	por sobre	Procesos y herramientas.
Software funcional		Documentación exhaustiva
Colaboración con el cliente		Negociación de contratos
Responder al cambio		Seguir un plan

Tabla 1: Principios Ágiles

En CC61A se han implementado un sub-conjunto de prácticas de gestión basadas en metodologías ágiles, en particular *Extreme Programming*. Para la negociación de requerimientos se sigue el modelo de **contratos de alcance variable**, es decir, existe un protocolo de roles entre clientes y desarrolladores que otorga a ambas partes confianza en que el plan generado es factible, y que el producto a entregar será realmente valioso y de calidad. Esta relación se resume en las reglas descritas en la Tabla 2, inspiradas en la “*Declaración de deberes y derechos de desarrolladores y clientes*” [2].

	 Cliente	 Desarrollador
Desea maximizar	Valor recibido por cada semana de desarrollo	Calidad del trabajo realizado
Puede definir	Qué será implementado, y en qué prioridad, según las necesidades de su negocio	Cuánto se estima que demorará una tarea (idealmente)
Puede cambiar	Funcionalidades solicitadas por otras no implementadas de costo equivalente (canjear)	Sus estimaciones en base a nuevos descubrimientos

Tabla 2: Roles de clientes y desarrolladores para una relación ágil sinérgica

Nuestros alumnos trabajan con un **cliente in situ**, dado que el trabajo se realiza en las instalaciones de la organización cliente, con acceso a la contraparte, buscando constituir con ella o él un mismo equipo. La estructura de los equipos es horizontal, liderada por un coordinador, y donde cada integrante asume de manera integral tareas como análisis, diseño, implementación y control temprano de calidad. Las herramientas de gestión son bastante simples, basadas en planillas de cálculo que, con el mínimo de tiempo de capacitación y uso permitan obtener la mayor visibilidad del avance y riesgos del proyecto. La validación de los productos es realizada por el cliente a través de **pruebas de aceptación** definidas por él apenas la funcionalidad haya sido finalizada por los desarrolladores. Al final de cada ciclo se debe generar **un entregable pequeño** que entregue un incremento de software funcional y útil para el cliente. Finalmente, se promueve que nuestros alumnos adquieran una buena comprensión del problema de negocio del cliente, adoptando un **lenguaje común** con él, y pudiendo de esta manera definir soluciones más asertivas para su cliente.

2.2.1. El rol del ingeniero de software como vinculador entre el mundo real y la tecnología

Nuestros alumnos poseen un muy alto dominio en temas técnicos, lo que se explica tanto por los mecanismos de selección utilizados en su ingreso a la carrera como por el acento en destrezas matemático-técnicas de la malla curricular actual. Sin embargo en el mundo se está marcando una tendencia a requerir profesionales de la ingeniería de software integrales, hábiles para las relaciones interpersonales a la vez que buenos diseñadores e implementadores, y capaces de traducir múltiples dominios de negocio a soluciones tecnológicas efectivas [3][6].

Es por esto que al final de cada ciclo, los alumnos deben exponer frente a la comunidad de clientes, docentes y compañeros cuál es el problema real de negocio de su proyecto, y cómo su solución aporta valor a su cliente. Adicionalmente, se considera que en el cambiante ambiente tecnológico y de negocios del mundo es fundamental la adopción de un enfoque que potencie el aprendizaje acelerado y evaluación continua de los procesos y productos de desarrollo, que es una fortaleza de las metodologías ágiles.

2.3. Fases del curso

El curso posee una fase introductoria de tres semanas donde se introduce a los alumnos en el modelo del curso, sus bases metodológicas y reglas académicas. Es aquí que se conforman los equipos de trabajo y se realizan las primeras aproximaciones al cliente con el objeto de que el equipo pueda elaborar un plan inicial de trabajo y realizar una presentación donde explique cuál es el problema de negocio a resolver.

Posteriormente comienza la fase de proyectos, organizada en tres ciclos de cuatro semanas cada uno. Los alumnos deben trabajar 16 horas semanales en la organización cliente, 2 horas para gestión interna del equipo y 2 horas para las reuniones con su tutor, completando así una jornada semanal de 20 horas efectivas de dedicación al curso. Al final de cada ciclo cada equipo debe lograr un producto de software funcional que implemente características de real valor para el cliente. Este producto es demostrado al equipo docente, al resto del curso y a los clientes en una presentación formal. Una evaluación más detallada de cómo está implementado el software y la calidad de la documentación generada es realizada por cada tutor con su equipo en una reunión posterior. En los ciclos siguientes el equipo debe hacer crecer de manera incremental la base de código generada hacia el producto que finalmente es puesto en producción al final del proyecto. Podemos observar la estructura antes descrita en la Ilustración 1.

Fase \ Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Introducción al curso															
Ciclo 1															
Ciclo 2															
Ciclo 3															

Ilustración 1: Calendario del curso

2.4. Modelo evaluativo

El modelo evaluativo aplicado es bastante complejo, dado que tiene que armonizar las perspectivas de todos los involucrados en el proceso tales como son el equipo docente, los miembros del equipo de desarrollo y los clientes de los proyectos, tal como se puede apreciar en la Tabla 3. Al final de cada ciclo se recopilan las distintas evaluaciones y se pondera una nota final de ciclo generalmente grupal, y si lo amerita también individual. Las proporciones entre los ítems evaluados varían en cada ciclo, poniéndose el énfasis en el aspecto metodológico (proceso de desarrollo) en el ciclo 1 para luego en los ciclos siguientes enfatizar más el resultado obtenido (producto de software generado). Las presentaciones realizadas por los alumnos tienen una ponderación menor a los ítems anteriores pero fija a lo largo del curso, con lo que se quiere indicar que si bien no es el objetivo fundamental del curso el desarrollar

la capacidad de “vender” el valor generado en un proyecto, sí es un elemento relevante que no puede ser ignorado.

Evaluable \ Item	Producto	Proceso (Grupal e Individual)	Presentación
Equipo Docente	✓	✓	✓
Equipo (Coevaluación)		✓	
Cliente	✓	✓	

Tabla 3: Relación entre evaluadores e ítems evaluados en el curso

Adicionalmente a lo anterior se ha implementado una encuesta en la que al final de cada ciclo los alumnos evalúan de manera individual el apoyo recibido por el equipo docente y la relación que tienen con su cliente. De esta manera es posible mejorar la gestión pedagógica y que el equipo docente medie oportunamente en la relación con los clientes si la situación lo amerita. En general los alumnos tienen cierta dificultad para adaptarse a la forma de operar en el curso debido a la gran diferencia que tiene con todos los demás cursos de la carrera. Sin embargo, en términos generales, después de la primera iteración todos los equipos son capaces de alinearse de acuerdo con las reglas y obtener valiosos resultados.

2.5. Selección de organizaciones cliente

Las organizaciones clientes son seleccionadas a través de la convocatoria a un concurso semestral, en el cual deben cumplir con los siguientes requisitos:

- El problema de negocio debe estar suficientemente claro y definido para la empresa proponente de modo que permita una solución viable en el tiempo de desarrollo del curso. Sin embargo, debe existir un margen de indefinición que desafíe a los alumnos a dominar el problema de negocio y poder aportar a la solución.
- Poner a disposición de los alumnos la infraestructura de espacio y equipamiento adecuada para realizar su labor y comprometer horas suficientes de su propio personal para cumplir el rol de cliente de manera efectiva. Quien asume el rol del contraparte debe poseer un buen dominio del problema a resolver y las atribuciones organizacionales suficientes como para que sus decisiones sean respetadas.

El primer punto es la causa fundamental para el rechazo de la mayor cantidad de proyectos; es poco lo que nosotros podemos hacer para que un potencial cliente clarifique su idea acerca de sus propias necesidades. En cuanto al segundo punto, sin embargo, hemos construido un documento de llamado a concurso de proyecto donde se establece muy claramente los compromisos que deben asumir los clientes, y la postulación es por estricta invitación. De esta forma hemos logrado reglas claras y un compromiso personal de los clientes.

Las empresas deben cancelar un monto por su participación como cliente, como una manera de asegurar un compromiso efectivo con el proyecto y poder financiar en parte los costos adicionales de traslado que realizan los alumnos y el pago a los tutores.

Por último se evalúa la disposición de los clientes a cumplir las reglas establecidas para el proyecto, buscándose así un ambiente propicio para el aprendizaje de los alumnos y la obtención de un buen producto para los clientes.

3. CONCLUSIONES

Desde la implementación del modelo ágil de gestión en el año 2005, CC61A ha atendido a 12 organizaciones, desde grandes empresas de alta tecnología hasta proyectos académicos de la universidad, completando 19 proyectos en áreas tan diversas como Sistemas de Información Administrativos, aplicaciones móviles, o *Bussiness Intelligence*. Nuestro porcentaje de éxito ha sido de un 94,7%, con sólo un proyecto que no logró una solución para su cliente. Los buenos resultados han permitido ir generando una positiva red de contactos entre el DCC y la industria, y por ende una vitrina para nuestros alumnos quienes en muchos casos han sido contratados a partir de esta experiencia. Destaca además de este curso que mientras la tendencia en muchas de las más destacadas experiencias académicas internacionales para enseñanza de los métodos ágiles se suelen utilizar problemas ficticios (ver por ejemplo [5]), aquí se abordan de manera efectiva problemas del mundo real, lo que da mayor profundidad al aprendizaje obtenido.

Por otra parte, este curso exige a los alumnos una dedicación y profesionalismo que en general antes no habían experimentado durante su vida académica. Esto se suma a que varios de nuestros alumnos a esta altura de su carrera ya están trabajando, por lo cual existe una disposición inicial no muy positiva a dedicar 20 horas efectivas a un curso. En este sentido hemos logrado sin embargo que en la mayor parte de los casos, al finalizar el curso se logre una valoración sobre lo aprendido y la experiencia vivida.

Entre los desafíos que podemos identificar se encuentra la necesidad de darle una mejor infraestructura organizacional al curso que permita una mejor difusión interna y externa de la calidad de la experiencia. Por ejemplo, para el DCC es importante abrirse a más cantidad y más variadas organizaciones clientes cada semestre, pudiendo así enriquecer el espectro de experiencias para nuestros alumnos y estar a la punta en los problemas tecnológicos y de negocio que surgen cada año. Sin embargo existe una natural reticencia de quienes no conocen el curso a realizar todas las inversiones que se requieren (principalmente tiempo y recursos), por lo cual el grupo de postulantes de buena calidad para cada semestre si bien en general es suficiente, no es todo lo numeroso que podríamos desear para poder seleccionar los mejores prospectos de proyectos para nuestros alumnos.

4. REFERENCIAS.

- [1]. Beck et al. *Manifesto for Agile Software Development* <http://agilemanifesto.org/>
- [2]. Beck, K., Fowler, M, *Planning Extreme Programming*, Addison-Wesley Professional, 1st edition, 2000.
- [3]. Collet, S. *Hot Skills, Cold Skills The IT worker of 2010 won't be a technology guru but rather a 'versatilist.'* <http://www.computerworld.com/action/article.do?command=viewArticleTOC&specialReportId=9000100&articleId=112360>
- [4]. Forrester Research, Inc. *Agile Processes For The Enterprise*,. June 9, 2005 Teleconference
- [5]. Hedin, G., Bendix, L., Magnusson, B., "Introducing Software Engineering by means of Extreme Programming" Proceedings of the 25th International Conference on Software Engineering, pp. 586- 593, 2003
- [6]. Marasco, J. *Software development productivity and project success rates: Are we attacking the right problem?* <http://www.ibm.com/developerworks/rational/library/feb06/marasco/index.html>
- [7]. Standish Group International. *The Chaos Report*; www.standishgroup.com