

De la Universidad a la industria del software y viceversa: Una experiencia centrada en la aplicación de nuevas tecnologías, adoptadas por la industria, a proyectos finales de asignaturas

Marcela Daniele, Marcelo Uva, Ariel Arsautte y Franco Brusatti

Departamento de Computación, FCEFQyN, Universidad Nacional de Río Cuarto,
Río Cuarto, Argentina. Email:
{marcela,uva,fbrusatti,aarsaute}@dc.exa.unrc.edu.ar

Resumen Durante la última década, la relación Universidad-Industria de desarrollo de software se ha vuelto cada vez más estrecha. Pequeñas, medianas y grandes empresas se acercan a menudo a las universidades en busca de analistas, programadores, ingenieros de software, etc. El crecimiento y expansión del mercado informático requiere recursos humanos calificados. Paralelamente, dicho crecimiento ha propiciado en la industria, la generación y adopción de nuevas metodologías de desarrollo, nuevas modalidades de trabajo por ejemplo desarrollos outsourcing. Junto con éstas, han surgido nuevas tecnologías brindando el soporte necesario para actividades de administración, gestión, planificación, implementación, diseño, testing y control en la producción del software, entre otras. En este trabajo se presenta una experiencia desarrollada durante los años 2014 y 2015 dentro de las asignaturas Análisis y Diseño de Sistemas e Ingeniería de Software. La propuesta consiste en el desarrollo de un proyecto-taller integrador que aplica un set de herramientas y tecnologías utilizadas en la industria del desarrollo de software actual, estableciendo un puente más directo entre formación académica y necesidad del mercado.

1. Introducción

Durante la última década, la relación Universidad-Industria de desarrollo de software se ha vuelto cada vez más estrecha. Pequeñas, medianas y grandes empresas se acercan a menudo a las universidades en busca de analistas, programadores, ingenieros de software, etc. El constante crecimiento y expansión del mercado informático, en todos los ámbitos, hace cada vez más visible la necesidad de contar con re-

cursos humanos calificados. Paralelamente, dicho crecimiento ha propiciado en la industria, la generación y adopción de nuevas metodologías de desarrollo y nuevas modalidades de trabajo por ej. desarrollos outsourcing. Junto con éstas, han surgido nuevas tecnologías brindando el soporte necesario para las actividades de administración, gestión, planificación, implementación, diseño, automatización del testing, seguimiento y control de las tareas que hacen a la producción del software, en-

tre otras. Muchas de éstas tecnologías poco a poco han ido incorporandose a las currículas de las carreras de informática. De esta manera se ha establecido una relación de necesidad entre Universidad e Industria de desarrollo de software en ambos sentidos.

En el marco de los proyectos PII-MEG CITA (Proyectos de Investigación e Innovación para el Mejoramiento de la Enseñanza de Grado) pertenecientes a la Universidad Nacional de Río Cuarto se han realizado una serie de propuestas en pos de detectar, analizar y ejecutar acciones concretas con el fin de realizar aportes para solucionar problemáticas observadas en las asignaturas de 3er. año de las carreras de Analista, Profesorado y Licenciatura en Ciencias de la Computación. Algunas de las acciones realizadas se resumen a continuación:

- Desarrollo e un Proyecto-taller de integración de todas las asignaturas de 3er. año (Análisis y Diseño de Sistemas, Base de Datos y Diseño de algoritmos. El problema observado en este caso radicaba en que los alumnos debían realizar un proyecto final para cada una de las asignaturas de 3er. año. En algunos casos, la carga horaria de la asignatura tenía contemplado esto y en otros no. Esto producía que alumnos quedaran libres en algunas materias por no poder cumplir con los requisitos de las mismas. O bien, ellos mismos optaran por hacer unas u otras, debido a la exigencia horaria requerida. Para evitar todo lo anterior, se diseñó un proyecto-taller integrador único para todas las asignaturas de 3er. año. Cada asignatura incluiría en éste los contenidos a evaluar que

requiriera. El proyecto-taller consistiría en el desarrollo de un sistema informático en donde la asignatura Base de Datos se encargaría de diseñar y generar la base de datos. La asignatura Diseño de Algoritmos debía establecer algún requerimiento que implicara la necesidad de la utilización de técnicas algorítmicas estudiadas durante la cursada y, finalmente Análisis y Diseño de Sistemas se encargaría del seguimiento y control del proceso de desarrollo. El dominio del proyecto sería definido por los cuerpos docentes de las tres asignaturas.CITA

- Definición de una plantilla para guiar la documentación en lo que respecta a proyectos desarrollados con metodologías ágiles.CITA
- Estudio y análisis de causas de retraso de la finalización de los Trabajos Finales CITA.
- Propuesta para realizar el Trabajo Final en los tiempos planificados CITA.

En este trabajo se presenta una experiencia desarrollada durante los años 2014 y 2015 dentro de las asignaturas Análisis y Diseño de Sistemas e Ingeniería de Software. La propuesta consiste en el desarrollo de un proyecto-taller en donde, además de integrar los contenidos teórico-prácticos abordados en las asignaturas de tercer año, se incorporen un conjunto de tecnologías utilizadas en la industria del desarrollo de software actual.

El resto del trabajo está organizado de la siguiente manera: en la sección 2 se presentan los fundamentos de la propuesta y se plantean los objetivos. La sección 3 presenta la metodo-

logía de desarrollo aplicada. En la sección 4 se presenta la propuesta, planificación, tecnologías y herramientas utilizadas. En la sección 5 se realiza una evaluación de la propuesta, y finalmente, las conclusiones.

objetivos. La sección 3 presenta la metodología de desarrollo aplicada. En la sección 4 se presenta la propuesta, su planificación y las tecnologías y herramientas utilizadas. En la sección 5 se realiza una evaluación de la propuesta, y finalmente, las conclusiones.

2. Fundamentos

3. Metodología de desarrollo aplicada a la propuesta

En el marco de las asignaturas Análisis y Diseño de Sistemas e Ingeniería de Software, ambas del tercer año de las carreras Analista, Profesorado y Licenciatura en Ciencias de la Computación de la U.N.R.C, se presentan diferentes metodologías para el desarrollo de sistemas informáticos. Algunos de los ciclos de vida del software estudiados son: el ciclo de vida lineal o secuencial, el método del análisis estructurado de Jourdon [1], desarrollo basado en componentes, diseño por contratos [2] de Bertrand Meyer, el Proceso Unificado [3] y metodologías ágiles.

La asignatura Análisis y Diseño de Sistemas cuenta con una planificación de 180 horas. Como se explicó en la sección FUNDAMENTOS, en trabajos anteriores se propuso integrar este proyecto a los proyectos de fin de materia de todas las asignaturas de 3er. año. En esta oportunidad lo que se presenta es una modificación a este proyecto-taller

meditante la incorporación de un set de herramientas que la industria de desarrollo de software utiliza actualmente. Dichas tecnologías están intimamente ligadas a la aplicación de metodologías ágiles. Es por ello que el cuerpo docente de la asignatura ha seleccionado la metodología ágil SCRUM CITA la cual se explica brevemente en el siguiente apartado.

3.1. Scrum

Scrum es una metodología ágil que permite trabajar en ambientes muy cambiantes, permitiendo permanentes replanteamientos. Por otro lado permite reducir el tiempo de producción y de comercialización del producto obtenido, aporta un gran beneficio o valor agregado al cliente, minimiza los riesgos de desperdiciar esfuerzo/tiempo en la construcción de artefactos que no serán usados o que no son fundamentales para el cliente. Facilita también la comunicación entre todos los integrantes del proyecto. La documentación producida dentro de un proyecto SCRUM es relativa al usuario, dueño, producto y equipo. Scrum es más bien un marco de trabajo que reposa sobre la premisa de que el equipo de desarrollo conocerá la mejor manera de resolver el problema que se le presenta. La reunión de planificación de cada grupo de requerimientos a producir se describe en términos del resultado deseado, en lugar de un conjunto de criterios de ingreso, definiciones y tareas. Scrum se basa en una auto-organización, con un equipo multifuncional y sin líder (dentro del equipo). El equipo es apoyado por dos individuos quienes ocupan los roles de Scrum Master y Product Owner. El Scrum Master es una especie de entrenador para el equipo, su

función es ayudar a los miembros del mismo a utilizar el marco que ofrece la metodología para conseguir un nivel alto de productividad. Mientras que el Product Owner representa los negocios, clientes o usuarios. Éste guía al equipo hacia la construcción del producto adecuado. Los proyectos Scrum avanzan en orden a la definición de los sprints, que son las iteraciones que poseen una duración de entre dos y cuatro semanas. En el inicio de cada sprint, los miembros del equipo se comprometen a producir un cierto número de características que enumeran en el artefacto conocido como el Product Backlog del proyecto. Al final de cada sprint, cada funcionalidad (conocida como *historia*) debe estar codificada, probada e integrada a una versión demo del sprint anterior.

Luego se realiza una revisión, y por último se demuestra la nueva funcionalidad frente al Product Owner y las otras partes interesadas que proporcionen información requerida para el siguiente sprint. Las iteraciones han de continuar hasta obtener el producto deseado. Como se puede observar de lo expuesto, SCRUM establece una forma de trabajo en donde todo el equipo se auto-regula y en donde no hay de antemano una documentación establecida a priori. Cada equipo utilizará los elementos que le sean necesarios para poder llevar adelante el grupo de *historias* con el cual se ha comprometido. Algunos podrán utilizar diagramas UML[] otros utilizarán diagramas de flujos de datos, etc. En la figura 1 se esquematiza todo el proceso.

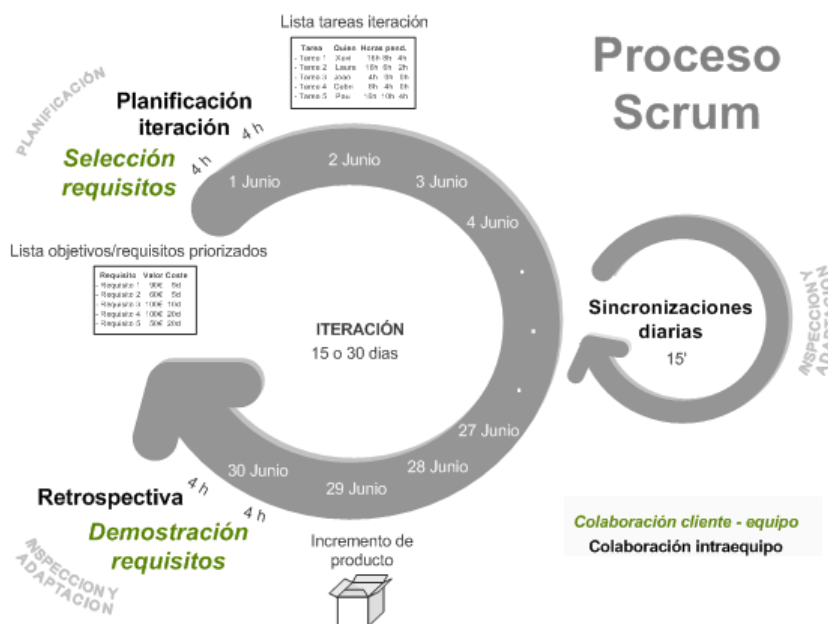


Figura 1. Proceso Scrum

4. Descripción de la propuesta

En este trabajo se presenta una experiencia desarrollada durante los años 2014 y 2015 dentro de las asignaturas Análisis y Diseño de Sistemas e Ingeniería de Software perteneciente a las carreras de computación dictadas dentro de la Universidad Nacional de Río Cuarto. La propuesta consiste en el desarrollo de un proyecto-taller en donde, además de integrar los contenidos teórico-prácticos estudiados, se incorporen un conjunto de tecnologías y herramientas utilizadas en la industria del desarrollo de software actual, permitiendo establecer un mejor puente entre el futuro egresado y las empresas de desarrollo de software.

4.1. Objetivos de la propuesta

- Integrar en un proyecto de desarrollo de software los contenidos estudiados en las asignaturas pertenecientes al tercer año de las carreras de Analista, Profesorado y Licenciatura en Ciencias de la Computación. La planificación CITA del proyecto esta fuertemente articulada con la planificación de la asignatura.
- Aplicar una metodología de desarrollo ágil como Scrum.
- Simular un ambiente de trabajo con características similares a las de un ambiente de trabajo real. Para ello se asignan responsabilidades específicas a cada uno de los integrantes de cada grupo. Si bien los objetivos particulares para cada grupo deben realizarse en equipo, se realiza un seguimiento particular de cada uno de los alumnos.

Su integración con el resto, su compromiso y forma de trabajo.

- Desarrollar en los alumnos habilidades para poder adoptar y sacar provecho de un set de herramientas utilizadas en la industria del software.

4.2. Organización y planificación

El taller se organiza para ser llevado adelante durante el primer cuatrimestre del año. Cabe señalar que dentro del grupo de docente de la asignatura, se ha seleccionado a un docente con dedicación part-time para la coordinación general del taller. La selección de este docente fue motivada por la vinculación que posee el mismo con la industria de desarrollo. Este docente es fundador de una empresa de desarrollo de software local y está habituado a formar parte de grupos de trabajo con desarrolladores a nivel internacional. // Uno de los objetivos del taller es la aplicación de una metodología ágil. Es por ello que el proyecto estará guiado por Scrum. En cada una de las fases del ciclo de vida, se aplicará una herramienta específica detallada en la sección CITA. No es objetivo de este proyecto el que los alumnos realicen un trabajo que se exceda en cuanto a complejidad y tiempo que les pudiese insumir. El mismo se focaliza en que cada grupo pueda completar el trabajo, usando las tecnologías propuestas. A continuación se presenta la planificación para el proyecto-integrador durante el año 2015.

4.3. Tecnologías y herramientas utilizadas

La industria del software se encuentra en continua expansión. Esto requiere

re la incorporación de nuevas tecnologías para generar nuevas tecnologías. Sólo incorporando nuevas herramientas de soporte para estos procesos es que se logra producir sistemas para millones de usuarios como por ejemplo las redes sociales Facebook o Twitter. Las herramientas utilizadas en este proyecto fueron las siguientes:

- Github:CITA Es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git [14]. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.
- Git:CITA Es un sistema de control de versiones distribuido, libre y gratuito.
- Java 8: CITA Es un lenguaje de programación informática de propósito general que es concurrente, basado en clases y orientado a objetos.
- Postgresql: [17] Es un sistema de base de datos objeto-relacional que tiene las características de los sistemas de base de datos propietarios tradicionales con mejoras de los sistemas de base de datos de la nueva generación.
- ActiveJDBC: CITA rápido ORM (Object Relational Mapping) para desarrollo ágil. Es una implementación en el lenguaje Java del patrón arquitectural Active Record.
- JUnit: CITA Es un framework para escribir tests (pruebas de software). Es una implementación de la arquitectura xUnix para unit testing.
- Spark:CITA Es un framework inspirado en Sinatra para crear aplicaciones web con Java 8.
- Sinatra:CITA es un framework para aplicaciones web de software libre y código abierto con una DSL (Domain Specific Language) escrito en Ruby on Rail.
- PivotalTracker: CITA Es una interesante herramienta para la gestión de proyectos ágiles.
- Maven: herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant (y en menor medida a PEAR de PHP y CPAN de Perl), pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.
- GanttProject: CITA herramienta gratuita para crear una completa planificación de un proyecto de forma visual. Todo queda bajo control en GanttProject, desde los recursos necesarios en forma de personal, los días festivos, hasta dividir el proyecto en un árbol de tareas y asignar a cada uno los recursos oportunos.

5. Evaluación de la propuesta y resultados

6. Conclusiones

Referencias

1. Tom DeMarco. Libro: Structured Analysis and System Specification, 1979.
2. Pressman, Roger. Software Engineering: A Practitioner's Approach. 7ma edition. McGraw Hill. 2006.
3. An Integrated Approach to Software Engineering. Pankaj Jalote. Springer 2006.
4. Kent Beck, Mike Beedle y otros. "Agile Manifesto and agile principles". www.agilemanifesto.org. Febrero 2001.

5. Scrum in Action: Agile Software Project Management and Development. Andrew Pham, Phuong Van Pham. Course Technology Ptr. 2011.
6. Marcela Daniele. Nicolás Florio. Daniel Romero. Definición y uso de Plantillas Genéricas para la descripción de Casos de Uso. Proyecto de Innovación e Investigación para el Mejoramiento de la Enseñanza (PIIMEG 2004), Secretaría de Ciencia y Técnica y Secretaría Académica, Universidad Nacional de Río Cuarto. RR N° 302/2004.
7. Marcela Daniele. Daniel Romero. Evolución de Plantillas Genéricas para la descripción de Casos de Uso a Plantillas Genéricas para Análisis y Diseño. Proyecto de Innovación e Investigación para el Mejoramiento de la Enseñanza (PIIMEG 2006), Secretaría de Ciencia y Técnica y Secretaría Académica, Universidad Nacional de Río Cuarto. RR N° 109/2005.
8. Marcela Daniele. Daniel Romero. "La enseñanza de gestión de proyectos de software y la aplicación de herramientas que favorezcan su automatización". Secretaría Académica y de Ciencia y Técnica, Universidad Nacional de Río Cuarto. RR N° 499/06. (01/08/2006, 31/07/2008).
9. Marcela Daniele, Fabio Zorzan, Paola Martellotto, Mariana Frutos, Marcelo Uva, Ariel Arsaute, F. Brusatti, J. Guazzone, S. Angeli. "Estimación y Planificación de Proyectos de Software versus duración de proyectos finales en la carrera Analista en Computación". Secretarías Académica y de Ciencia y Técnica, Universidad Nacional de Río Cuarto. RR N° /11. (01/02/2011, 31/12/2012).
10. Marcela Daniele, Fabio Zorzan, Paola Martellotto, Marcelo Uva, Ariel Arsaute, Mariana Frutos. "Causas que producen que los estudiantes de Computación retrasen la culminación de su Trabajo Final. Secretarías Académica y de Ciencia y Técnica, Universidad Nacional de Río Cuarto. Presentado en abril de 2013. En evaluación.
11. Fabio Zorzan, Mariana Frutos, Ariel Arsaute, Marcela Daniele, Paola Martellotto, Marcelo Uva, Carlos Luna "Delayed Completion of Final Project of the Career Computer Analyst: Seeking its Causes". XX Congreso Iberoamericano de Educación Superior (CIESC 2012), en el Marco de la XXXVIII Conferencia Latinoamericana en Informática – CLEI 2012 - Octubre 1 al 5 de 2012 - Medellín, Colombia. ISBN 978-1-4673-0792-5.