

Metodologías Ágiles para el Desarrollo de Sistemas de Cómputo Distribuido

Oscar Martín Bianchi

mbianchi@ejercito.mil.arCIDESO¹, DIGID² - Ejército ArgentinoEST³, IESE⁴ - Ejército Argentino

Alejandro J. M. Repetto

arepetto@ejercito.mil.arCIDESO¹, DIGID² - Ejército ArgentinoEST³, IESE⁴ - Ejército Argentino

José Ignacio Ariznabarreta Fossati

ignacioariznabarreta@gmail.comEST³, IESE⁴ - Ejército Argentino**RESUMEN**

El Manifiesto Ágil, escrito en el 2001, establece las pautas para la utilización de las metodologías ágiles, las metodologías más populares y las lecciones aprendidas, para el desarrollo de software (1). Éstas surgen como alternativa a los procesos de desarrollo con alto costo en documentación y procesos excesivamente prolongados.

De la misma forma, las arquitecturas de computación distribuida y computación de grilla surgen como respuesta a la necesidad de aportar una solución y alternativa a los procesos con alta demanda de recursos.

En la Escuela Superior Técnica del Ejército, durante el año 2011, se llevó adelante el proyecto CoDiSe (Computación Distribuida para Seguridad Informática) (2), que sirvió como prototipo experimental para el proyecto CODEC (Computación Distribuida para Experimentación Científica) (3), el cual pretende dotar a los laboratorios de las distintas especialidades de la EST con un sistema capaz de procesar grandes cantidades de información o realizar cómputos a gran escala, sirviéndose de los infraestructura informática ociosa de la universidad.

El presente trabajo tiene por objetivo compartir las experiencias obtenidas en el uso de las metodologías ágiles en el ámbito académico, en el marco del desarrollo de un sistemas de computación distribuida como parte del dictado de la asignatura Lenguajes de Programación I.

Palabras Clave: *Computación Distribuida, Metodologías Ágiles, Computación en Grilla de Escritorio, Ingeniería del Software.*

CONTEXTO

Elegir una metodología de desarrollo de software apropiada es una preocupación siempre presente en la mayoría de los proyectos de IT, así como en los ámbitos académicos asociados a la misma actividad.

Las nuevas metodologías de desarrollo de software, a diferencia de las clásicas, son consideradas mucho más ágiles y fáciles de adaptar a los cambios. Éstas no hacen foco en el largo ciclo de desarrollo sino en iteraciones más cortas y ligeras, con el sello de una proximidad más acentuada a los *Stakeholders* (4).

Al respecto, las metodologías ágiles proveen la oportunidad de evaluar la dirección del proyecto a medida que se desarrolla el ciclo de desarrollo natural del sistema. Esto se alcanza a través del ritmo regular establecido de trabajo, conocido como iteraciones o *sprint*, al final de los cuales se puede presentar algún estado de terminación del producto (sistema en este caso).

Enfocándose en la repetición de ciclos de trabajo abreviados (o iteraciones) los equipos de trabajo tienen la posibilidad de corregir tempranamente desvíos al inicio de cada iteración, reduciendo de esta forma el impacto de cualquier error o falla cometido en las etapas previas (4) (5).

Asimismo, esta metodología de desarrollo en paralelo propuesta está basada en el

¹ CIDESO: Centro de Investigación y Desarrollo de Software

² DIGID: Dirección General de Investigación y Desarrollo

³ EST: Escuela Superior Técnica - Facultad de Ingeniería del Ejército Argentino

⁴ IESE: Instituto de Enseñanza Superior del Ejército - Universidad del Ejército Argentino

enfoque de inspeccionar y adaptar (*inspect-and-adapt*), solucionando o evitando problemas como la parálisis por análisis (*analysis paralysis*), la sobrecarga de información y el problema de la racionalidad limitada (*bounded rationality*), entre otros.

En ese aspecto, en los últimos años se ha observado un aumento exponencial en la práctica de metodologías ágiles de desarrollo de software. Esto se debe en gran parte al hecho que estas prácticas facilitan el trabajo en equipo, especialmente si los grupos están fragmentados o distribuidos, haciendo hincapié en el uso de herramientas de gestión como los *Sprint* y los *Scrum*.

Ambas herramientas ofrecen la posibilidad de simplificar la aproximación a desarrollos complejos, facilitando tanto la comprensión del problema, como la implementación y seguimiento de la solución. Estas cualidades hacen del *Scrum* una metodología de desarrollo viable para su utilización en el ámbito académico (6).

En cuanto a la tecnología implementada, el uso de computación distribuida (CD) para solucionar problemas de rendimiento o disponibilidad de recursos (necesarios para grandes cómputos) es una tendencia en crecimiento. Se puede observar en áreas tan diversas como la experimentación científica (con proyectos como SETI^a y BOINC^b) y la seguridad cibernética (tanto en su uso como herramienta de testeo de nuevas tecnologías, como instrumento de *hacking*).

La incorporación de conceptos como “*Grid Desktop Computing*” (GDC) (7) y “*Cloud Computing*” a las tecnologías ya existentes de CD, la vuelven en una herramienta valiosa no sólo para el desarrollo de software de cómputo de alto rendimiento, sino también como herramienta pedagógica de enseñanza que aporta una variedad de conocimientos y conceptos amplia, fundamentales para la formación profesional de los estudiantes de las carreras de IT.

Teniendo presente esto, y como evolución de experiencias previas, en el año 2011 se

decide avanzar con un prototipo experimental de un sistema de CD, de la mano del proyecto CoDiSe (2), obteniendo en el 2012, como fruto de este trabajo el proyecto CODEC (3). Este proyecto tiene como propósito el desarrollo de un sistema de computación distribuida multipropósito construido sobre la base del sistema BOINC. El fin de CODEC es poner a disposición de la EST un sistema con gran capacidad de procesamiento basado en una arquitectura de Computación de Grilla de Escritorio (7), aprovechando a su vez la infraestructura informática de la universidad, bajo el paradigma Infraestructura como Servicio (IaaS – *Infrastructure as a Service*).

Esta variedad de conceptos y tecnologías hacen de la comprensión y ejecución de un proyecto como CODEC una tarea compleja para aquellos participantes del mismo. Por esto, y dadas las características del proyecto, específicamente su origen académico como trabajo práctico de la cátedra de Lenguajes de Programación I, el uso de metodologías ágiles surge como la alternativa natural.

INTRODUCCIÓN

Una de las características determinantes de CODEC es el hecho que cada uno de sus *sprint* es semestral (y no mensual, o bimestral como recomiendan los especialistas en el tema) coincidiendo con el ciclo académico de las materias asociadas. Por la misma razón, el proyecto posee en sí mismo una alta variabilidad en los equipos de trabajo (los que rotan anualmente), perfiles de los miembros, conocimientos, dedicación horaria e incluso área de experticia. Esto constituye un ambiente altamente hostil para el desarrollo de software, lo que llevo a la evaluación de diferentes aproximaciones y metodologías de desarrollo, surgiendo el enfoque ágil como la opción más próxima a las necesidades y naturaleza del proyecto

Entre las ventajas que nos aporta el enfoque ágil podemos mencionar la posibilidad de contar con versiones de alcance limitado en donde se podrá tener un producto tangible, el cual le será entregado en sus diferentes ver-

^a<http://setiathome.berkeley.edu/>

^b<http://boinc.berkeley.edu/>

siones en los *sprints* propuestos, así como una versión final en el tiempo acordado.

Con respecto a lo estrictamente técnico, la implementación y las pruebas se simplifican sensiblemente al subdividirse cada una de las tareas programadas en el sprint en hitos más atómicos.

Como antecedente, la metodología utilizada durante el año 2012 puede ser considerado un “híbrido” debido a que, si bien se utilizó la definición base de la metodología *scrum*, el marco en el que fue desarrollado el proyecto forzó a un *sprint* semestral, debido a que la realización del mismo se llevó a cabo en la cátedra “Lenguaje de Programación 1”, dictada semanalmente, en donde se realizaba el *scrum* semanal. Esto colaboró en la resolución de los problemas identificados a lo largo del proceso de desarrollo, ya que el hecho de contar con equipos reducidos y metas atómicas facilitaba la posibilidad de cumplirlas de forma autodidacta.

Por último, no podemos dejar de mencionar un aspecto fundamental: los recursos humanos. Sobre este particular, es importante señalar que esta metodología nos abstrae de la problemática de encontrar alumnos con diferentes capacidades, así como diferentes conocimientos (adquiridos de manera extra curricular o de forma autodidacta), además de la dedicación que se le brindaba a dicho proyecto, trayendo a consideración nuevamente, la situación de estudiantes de Ingeniería en Informática, así como de Ingenieros en Electrónica.

LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

La metodología utilizada en CODEC plantea una línea de investiga clara con respecto a la estrategia a utilizar, siendo esta una variación de la metodología ágil, adaptada a las necesidades del proyecto. Entre sus características más importantes podemos mencionar:

- *Scrum* reducido: en lugar de realizar *Scrum* diarios, se decidió llevar adelante una práctica de *Scrum* semanales. Esto se debió a las características de la asignatura, la cual tenía una carga horaria muy limitada, lo que

sumado a la disponibilidad de cada uno de los alumnos y docentes, dejaba un marco horario extremadamente estrecho. Por esta misma razón se decidió realizar los *scrum* una vez por semana durante el horario de dictado de clase.

- *Sprint* semestral: cada una de las iteraciones se basa en una fase del proyecto, cada una de las cuales tiene la duración de un ciclo lectivo.

- Generación de una métrica que permita evaluar los progresos del proyecto y la efectividad del trabajo realizado durante cada iteración con respecto a la anterior. Sobre este último punto, la métrica propuesta evaluara una serie de cuestiones entre las que cabe mencionar:

- Formación de equipos: los equipos han de ser formados de teniendo en cuenta las habilidades de cada miembro, así como los conocimientos transversales que deben poseer.

- Trabajo interdisciplinario: dándose la particularidad que alumnos de dos carreras diferentes forman parte de CODEC, esta oportunidad es aprovechada para evaluar el trabajo en equipo entre futuros profesionales de distintas especialidades, tomando como parámetro el rendimiento de cada grupo individualmente.

- Evaluación de conocimientos: siendo de interés tanto para la cátedra como para la institución, la evaluación de conocimientos de cada participante del proyecto, previa y posterior al *sprint*, ofrece un claro parámetro del grado de asimilación de los conocimientos y aptitudes que pretende ofrecer el proyecto.

- Éxito individual de los grupos: cada *sprint* planteara una serie de objetivos a cumplir (generalmente asociados a pequeños ciclos de desarrollo o componentes de alcance acotado), considerando el éxito de cada grupo en el trabajo en el grado de cumplimiento de cada tarea asignada.

- Integración del trabajo grupal: la tarea de integración del trabajo de cada grupo (su dificultad, nivel de éxito, resultados en general, etc.) ofrece una clara perspectiva para evaluar el apego a los estándares de integración fijados así como la facilidad de

interpretación del trabajo de terceros entre los distintos grupos.

RESULTADOS Y OBJETIVOS

Los resultados obtenidos durante el 2012 en lo que hace al rendimiento del curso revelaron que el uso de las metodologías ágiles desarrollo lograron:

- Facilitar la comprensión de la temática, y el trabajo a realizar.
- Mejor aprovechamiento del tiempo dedicado al proyecto y flexibilidad a los cambios.
- Sencillez en el seguimiento del progreso, la supervisión y el soporte a los alumnos/desarrolladores.

En lo que hace a las mejoras que esperamos obtener respecto al rendimiento del curso, haremos referencia al incremento en la calidad del software, lo cual es consecuencia de la implementación de tareas, lo que trae una reducción en cuanto a la complejidad y, por ende, soluciones más rápidas ante cualquier eventualidad (como la metodología de trabajo ágil propone).

En la siguiente iteración se pretende cumplir los siguientes objetivos:

- Reducción de riesgos, al poder atomizarse el trabajo y la posibilidad de elegir comenzar por las tareas más críticas, se reducen los riesgos fracaso del proyecto.
- Disminución del tiempo de desarrollo, se pueden ir realizando entregas medida que vayan estando disponibles las diferentes partes/objetivos/sprint del proyecto.
- Análisis previo para identificar, de forma prematura, las tareas críticas en cuanto a la realización, para que las mismas sean abordadas desde un comienzo, reduciendo los riesgos de fracaso del proyecto.
- La liberación de productos finalizados (*releases*) por ciclo. Es por ello que al realizar diferentes betas/*releases* a medida que se van cumpliendo a los diferentes *sprints* del proyecto, esperamos lograr una disminución del tiempo de desarrollo del sistema.

FORMACIÓN DE RECURSOS HUMANOS

Una marca distintiva de CODEC es el hecho de haber nacido de la colaboración de dos centros de investigación dentro del Ejército Argentino (3). Y la particularidad que uno de dichos centros es parte funcional de la EST. Esto pone al proyecto en un ámbito privilegiado para la formación de recursos humanos.

Por un lado, el CIDESO tiene amplia experiencia en la formación de recursos humanos en el terreno de la investigación aplicada en sistemas de información de diversa índole, incluyendo sistemas de simulación para el adiestramiento, sistemas de información geográfica, sistemas de visualización, sistemas inteligentes, sistemas móviles, sistemas de comunicación de alta complejidad y sistemas de cómputo de alto rendimiento. Por el otro, la EST tiene experiencia en investigación básica en el terreno de la ciencia de la informática, asociado estrechamente al hecho de poseer la carrera de Ingeniería Informática como parte de su oferta académica.

En los laboratorios de la institución se dispone del material y los recursos humanos específicos para la investigación en tanto en el área de la seguridad, comunicaciones así como en las asociadas a las tecnologías de CD. También cabe mencionar que cuenta con expertos informáticos, matemáticos y criptógrafos que dan cuerpo muchas veces a las investigaciones de los trabajos finales de carrera que se realizan en los posgrados que brinda la universidad.

Tanto el CIDESO como los laboratorios de la EST, a través del dictado de materias de grado en Ingeniería Informática, aportan recursos humanos a la misma universidad. Es así que investigadores de los laboratorios dan cátedras en la EST y, de manera análoga, alumnos de la escuela aportan sus análisis a los laboratorios a través de trabajos prácticos de laboratorio, prácticas profesionales supervisadas o tesis y tesinas de grado y posgrado.

En particular, el prototipo experimental del cual nace la idea de CODEC fue implementado por los alumnos de tercer año de Ingeniería Informática y Electrónica como

trabajo práctico de laboratorio de la materia Lenguajes de Programación I, y tutorado por investigadores del CIDESO y el CriptoLab, que a la vez son docentes de dicha cátedra.

Para el próximo paso, se pretende continuar con esta interacción fluida entre los centros de investigación y el alumnado, formado profesionales con conocimientos de campo en el terreno de la computación de alto rendimiento y un conocimiento acabado sobre temas criptográficos.

Además, al expandirse el sistema para ser aplicado en cualquier problema que requiera altos niveles de cómputo, se pretende incorporar alumnos y docentes de otras cátedras, de cualquiera de las ingenierías que se dictan en la Facultad.

Por otro lado, se continuará con la formación de profesionales en investigación utilizando estas nuevas tecnologías en los dos laboratorios participantes a través de la utilización de la GDC.

Así, pues, se formarán recursos humanos de todos los niveles, grado, posgrado o investigadores activos, incorporando más alumnos a los laboratorios y, potencialmente, becarios que se dediquen de modo formal (no sólo académico) a la profundización de los modelos propuestos.

5- BIBLIOGRAFÍA

1. **Cockbun, A.** *Agile Software Development*. s.l. : Addison-Wesley, 2001.

2. **Bianchi, Oscar Martín y Repetto, Alejandro.** *Sistema de Computacion Distribuida para Seguridad Informatica*. Posadas, Misiones : Workshop de Investigadores en Ciencias de la Computación 2012, 2012.

3. **Bianchi, Oscar Martín y Bönke, Matías Miguel.** *Computación Distribuida para Experimentación Científica*. Mar del Plata, Buenos Aires, Argentina : RIIPRO Joven - 3er CIIP, 2012.

4. **Abrahamsson, P., y otros.** *Agile software development methods Review and analysis*. s.l. : VTT Publications, 2002.

5. **Canós, José H., Letelier, Patricio y Penadés, M. Carmen.** *Métodologías Ágiles*

en el Desarrollo de Software. s.l. : Universidad Politécnica de Valencia, 2010.

6. **K., Schwaber, M., Beedle y R.C, Martin.** *Agile Software Development with SCRUM*. s.l. : Prentice Hall, 2001.

7. **Ahmar, Abbas.** *Grid Computing Technology: An Overview*. s.l. : Abbas, 2004.

8. *Grid Desktop Computing for Constructive Battlefield Simulation*. **Repetto, Alejandro J. M.** San Salvador de Jujuy : Red UNSI, 2009. XV Congreso Argentino de Ciencias de la Computación (CACIC 2009). 978-897-24068-4-1.

9. *Hybrid Architecture for Constructive Interactive Simulation: Evaluation and Outcomes*. **Repetto, Alejandro J. M.** Orlando, FL : s.n., 2010. /ITSEC'10, Interservice/Industry Training, Simulation and Education Conference.

10. **Berstis, Viktors.** *Fundamentals of Grid Computing*. s.l. : Redbooks Paper, IBM Corp, 2002.

11. **Constantinescu-Fulop, Nicolae-Zoran.** *A Desktop Grid Computing Approach for Scientific Computing*. s.l. : Department of Computer and Information Science Faculty of Information Technology, Norwegian University of Science and Technology, 2008.

12. **Computerized Bussiness Solutions.** *Centralized vs Distributed Computing: White Paper*. 2007.

13. **Opiyo, Elisha T. O., y otros.** *Computing Research Challenges and Opportunities with Grid Computing*.

14. **Neves, Rodrigo, y otros.** *Parallel and Distributed Computing BOINC Grid Implementation*.