

Simulando Proyectos de Desarrollo de Software Administrados con Scrum

Diego Alberto Godoy^a, Edgardo A. Belloni^b, Henry Kotynski^c, Héctor Dos Santos^d
Eduardo O.Sosa^e

Centro de Investigación en Tecnologías de la Información y Comunicaciones (C.I.T.I.C.)
Departamento de Ingeniería y Ciencias de la Producción-Universidad Gastón Dachary
Av. López y Planes 6519- Posadas, Misiones, Argentina. Teléfono: +54-376-4438677

diegodoy@citic.ugd.edu.ar, ebelloni@ugd.edu.ar, henrykotynski@citic.ugd.edu.ar,
hector2s@citic.ugd.edu.ar, eduardo.sosa@citic.edu.ar

Resumen

El presente trabajo presenta una línea de investigación que tiene por objetivo general la elaboración e implementación de un modelo que permita simular la metodología de gestión de proyectos *Scrum*. Concretamente, se presenta aquí un modelo para simular uno de los componentes esenciales de la gestión de un proyecto desarrollado con Scrum: el Subsistema de Control de Errores de Tareas correspondiente a la fase denominada *Development Game*, propuesta por la metodología referida. El modelo presentado se ha validado con datos de tres proyectos reales. Adicionalmente, se han diseñado y ejecutado una serie de experimentos sobre el modelo propuesto y se ha realizado un Análisis de Sensibilidad de sus variables más importantes. El modelo construido sirve como ayuda a administradores de proyectos considerados novatos en la aplicación de la metodología referida, permitiéndoles conocer de antemano las consecuencias de sus decisiones.

Palabras claves: Administración de Proyectos de Desarrollo de Software; Scrum; Dinámica de Sistemas.

Contexto

El trabajo presentado en este artículo tiene como contexto marco el proyecto de investigación denominado “Simulación como Herramienta para la Mejora de los Procesos de Software”, registrado actualmente en la Secretaría de Investigación y Desarrollo de la

Universidad Gastón Dachary (UGD)[†] y radicado en el Centro de Investigación en Tecnologías de la Información y Comunicaciones de dicha universidad.

Entre las líneas con mayores resultados dentro del proyecto referido, se encuentran las de: (i) “Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software que utilizan Programación Extrema”, y (ii) “Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software Bajo Scrum”. Este artículo se enfoca en la presentación de la última línea aludida.

Introducción

La complejidad de los sistemas de actuales, los cambios repentinos en el contexto de estos sistemas y las modificaciones en los requerimientos del cliente una vez que se ha comenzado el desarrollo de un proyecto de software, generan un ambiente donde la planificación, el desarrollo, la administración y el control del mismo resultan difíciles de estimar o evaluar.

Este tipo de escenarios requiere de metodologías de desarrollo de software que permitan generar resultados rápidamente. Entre las metodologías con este tipo de características se encuentra *Scrum*, la cual fue aplicada por primera vez por Ken Schwaber y Jeff Sutherland, quienes la documentaron en detalle en su obra “Agile Software Development with Scrum” [1]. Esta metodología centra su atención en las actividades de gerencia basándose principalmente en una

[†] Mediante Res. Rectoral UGD N° 04/I/12.-

planificación adaptativa y en el desarrollo incremental del software con entregas funcionales en breves períodos de tiempo.

Frente a escenarios y requerimientos cambiantes, contar con una herramienta que permita simular la gestión de proyectos de desarrollo de software con *Scrum*, representa una alternativa interesante para que los administradores puedan evaluar el impacto de sus decisiones sobre la gestión en el desarrollo del proyecto, sin influir o poner en riesgo el proyecto real y sus recursos.

Actualmente existen diferentes trabajos y herramientas que permiten simular la administración de proyectos de software, entre ellos se puede citar a los siguientes: “Dynamics of Agile Software Development”[2], “Modeling Agile Software Maintenance Process Using Analytica ITheory of Project Investment”[3], “Modelo Dinámico de Simulación de Proyectos de Software con XP” [4] y [5], “Agile Project Dynamics: A Strategic Project Management Approach to the Study of Large-Scale Software Development Using System Dynamics” [6].

Estas herramientas presentan modelos basados en diferentes metodologías ágiles, en los cuales son propuestos y evaluados diferentes escenarios, algunos de ellos genéricos. No obstante, en ninguno de tales trabajos se presenta un estudio que contemple las fases y variables particulares de la

metodología *Scrum*.

Objetivos Propuestos

Teniendo en cuenta lo planteado en la introducción en este trabajo se presentan los primeros avances en el diseño y la construcción de un Modelo de Simulación Dinámico basado el modelado sistémico que de soporte a la gestión de procesos de desarrollo de software que se basan en el uso de la metodología *Scrum*.

El modelo permite, que los administradores de proyectos, puedan evaluar qué impacto tendrán sus decisiones de gestión sobre el mismo, a lo largo del desarrollo. De esta manera el modelo refleja el efecto del uso de prácticas de *Scrum* en proyectos de desarrollo de software y para ello, su estructura permite simular el desarrollo de un *Scrum* completo a la vez. Es decir, dado que *Scrum* presenta un estilo de desarrollo iterativo e incremental, dentro de un proyecto se negocian con el cliente varias entregas o versiones, por lo que el modelo, está destinado a simular un *Scrum* por vez.

Construcción del Modelo

En el desarrollo y construcción del modelo se utilizó la herramienta CASE VenSim PLE 5.4c (Versión Académica) [7] seguido las etapas de la Metodología de Dinámica de Sistemas [8], de esta forma, en la Fase de Conceptualización se ha construido el Diagrama Causal, incluyendo las variables que representan un proyecto

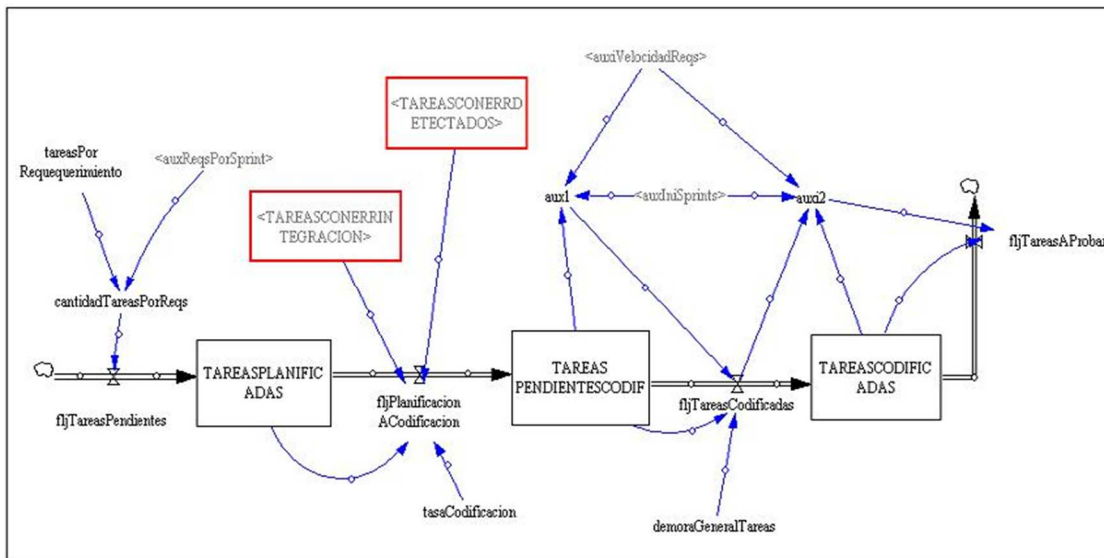


Diagrama1 - Sub-Sistema Desarrollo de Tareas

de software Scrum y sus interrelaciones. Luego, en la Fase de Formulación se ha traducido el Diagrama referido a un Diagrama de Forrester, el cual ha sido dividido en Subsistemas Conservativos de acuerdo con [9]. Finalmente, en la Fase de Evaluación se han realizado las corridas de validación y experimentales del modelo.

En el Diagrama 1, se presenta uno de los subsistemas más importantes del modelo: El Subsistema de Desarrollo de Tareas. En este Subsistema se muestra el proceso de planificación de las distintas tareas a codificar para satisfacer los distintos requerimientos del usuario seleccionados para cada *Sprint*. El subsistema se inicia a partir del número de requerimientos del usuario y una cantidad promedio de tareas necesarias para poder satisfacer a cada uno de estos. En la siguiente etapa, y en función del resultado de la primera etapa, a las tareas planificadas y pendientes de codificación se le suman aquellas tareas que tengan errores de Codificación y de Integración. En la última etapa se codifican o re-codifican las tareas provenientes de la etapa anterior. Las variables auxiliares Tasa de Codificación y Demora General de Tareas tienen respectivamente como función, seleccionar que tasa o porcentaje de las tareas planificadas se codificarán, y de agregar una demora temporal para el desarrollo de cada una de las tareas seleccionadas que finalmente se codifican y que luego serán probadas, generando de esta manera, Tareas con Errores de Integración o Con Errores de Programación. La variable auxiliar Velocidad de Requerimientos facilita la selección de la velocidad con la que se irán codificando las diferentes tareas planificadas, permitiendo esto, calcular la velocidad ideal a la cual podrá trabajar el Team a lo largo del sprint.

Validación

En la validación se utilizaron datos reales de tres Proyectos realizados con *Scrum*. El primero “Automatización de sistemas de desarrollo ágil Scrum: Team& Role”, cuyos detalles se pueden ver en [10]. El segundo “Aplicação Do Processo Ágil De Gerenciamento Scrum No

Desenvolvimento De Um Jogo Digital – Estudo De Caso Em Empresa De Software” [11]. El tercero de los casos utilizados Método Ágil Scrum Aplicado Al Desarrollo De Un Software De Trazabilidad” [12].

Sprint	Requerimientos por Sprint	Tareas Promedio por Requerimiento
Plannig Meeting	1	1
1	8	8
2	8	8
3	12	12
4	9	9
5	9	9
6	7	7
7	10	9

Tabla 1 – Requerimientos Por Sprint del proyecto presentado en [10]

En el Gráfico1 se muestra el número de requerimientos de cada *Sprint* de acuerdo a los datos exhibidos en la segunda columna de la Tabla 1. Los valores de este gráfico sirven de inicio al subsistema del Diagrama 1 y son el resultado del número medio de tareas multiplicado por el número de requerimientos del usuario.

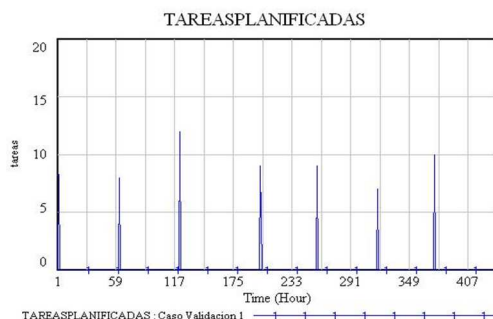


Gráfico1 - Tareas Planificadas por Sprint

En el Gráfico2 se puede observar el Burdown-Chart para todos los Sprints que componen el proyecto. El gráfico muestra como el número de tareas pendientes de codificación desciende desde los valores iniciales (Gráfico1), hasta llegar a cero, alcanzándose el nivel de Codificación planificado para cada Sprint (Gráfico2). Se aprecia también, como el nivel de codificación de Tareas se desarrolla de manera similar al presentado en el caso de validación [10].

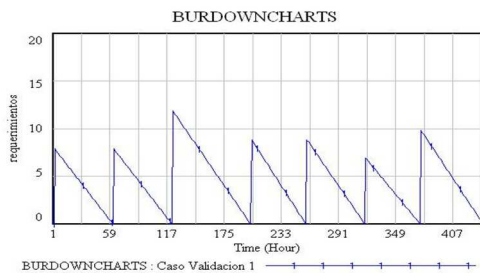


Gráfico2-Burdown-Chart Por Sprint

Experimentos realizados

En un experimento realizado basado en el primer caso de validación [10] se consideraron 63 requerimientos, distribuidos en 7 Sprints, resultando 9 requerimientos por Sprint. El número de integrantes no se modificó permaneciendo en 5 compuesto de la siguiente manera: 4 integrantes en el Team y 1 Stakeholder. La unidad de avance de tiempo elegida para la corrida fue de 1 hora. Considerando el número de requerimientos calculados para cada Sprint y asumiendo una tarea por cada requerimiento, se puede observar en el Gráfico4 los valores de tareas a planificar en cada sprint.

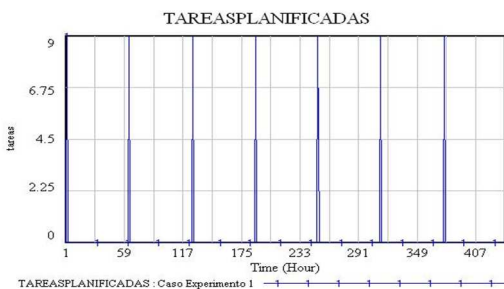


Gráfico3 - Tareas Planificadas Por Sprint

Por otra parte, en el Gráfico4 se puede observar el Burdown-Chart y como la cantidad de tareas pendientes descenden a medida que avanza el tiempo, y tomando el valor del número planificado (Gráfico3) de tareas para cada Sprints que componen el caso simulado en el experimento.

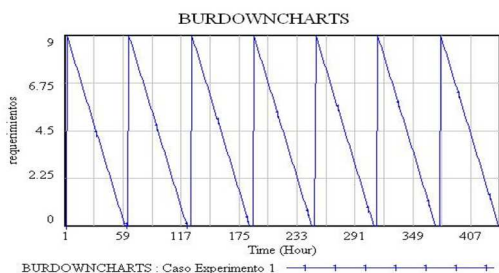


Gráfico4- Burdown-Chart Por Sprint

Resultados

En este trabajo se presentan avances en la construcción de un “Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software administrados con Scrum”. Al enfocarse en las particularidades de Scrum, este trabajo se diferencia de otros en los que se modelan proyectos con Metodologías Tradicionales y en los que se modelan las generalidades de las Metodologías Ágiles[2][3], o bien se modela la especificidad de una de éstas últimas como en[4]y [5].

El modelo presentado puede ser utilizado como herramienta para analizar el efecto que tiene el uso de prácticas de Scrum en proyectos de software. Para ofrecer flexibilidad, el modelo permite la alteración de valores durante su ejecución. Entre ellos los de: la cantidad de requerimientos del cliente; las fechas de comienzo y entrega de cada iteración; la cantidad de requerimientos a ser desarrollados en cada iteración; la cantidad de programadores; las horas extras agregadas por día, y el porcentaje de tiempo destinado a cada Sprint.

La validación realizada -con datos tomados de los proyectos [10], [11] y [12]como casos de “entrenamiento” se evalúa como positiva ya que el modelo construido se comportó de acuerdo con los datos de proyectos reales. Luego de validar el modelo se ha llegado a la conclusión de que el mismo cumple con sus objetivos y puede ser utilizado como herramienta para evaluar diferentes decisiones de gestión sobre proyectos de software desarrollados con Scrum.

Como trabajo futuro se espera avanzar, con la construcción de los demás subsistemas restantes, como por ejemplo: el de Desarrollo de Tareas; el de Recursos humanos y de Presión en el plazo, entre otros. Otra actividad prevista es la de compilar datos de proyectos de desarrollo de software en donde se utilizó Scrum, con el fin abarcar mayor cantidad proyectos. Adicionalmente, se prevé comparar la utilización de los simuladores de proyectos en Scrum y en XP[4] y [5], desarrollados anteriormente.

Formación de Recursos Humanos

El equipo de trabajo se encuentra formado por tres investigadores con distintos niveles de posgrado, un investigador con nivel de grado, y cuatro estudiantes en período de realización de trabajos finales de grado en el contexto de las carreras de Lic. en Sistemas de Información y de Ingeniería en Informática de la UGD. Actualmente, el número de tesinas de grado aprobadas en el contexto de este proyecto, es de uno, y otras dos en proceso de desarrollo.

Bibliografía

- [1] Ken y Sutherland, Jeff. Schwaber, *Agile Software Development with Scrum*, Primera ed.: Prentice Hall, 2001.
- [2] Kim E. Van, Kishore Sengupta, and Luk N. Van., "Dynamics of Agile Software Development," *International Conference of the System Dynamics Society*, 2009.
- [3] Konga, Li, Liu y Chen, Jing Xiaoying, "Modeling Agile Software Maintenance Process Using Analytical Theory of Project Investment," *International Conference on Advances in Engineering 2011*, 2011.
- [4] Tamara Kasiak y Godoy Diego Alberto, "Simulación de Proyectos de Software desarrollados con XP," *XIV Workshop de Investigadores en Ciencias de la Computación.*, 2012.
- [5] Godoy Diego Alberto y Kasiak Tamara., "Modelo dinámico de simulación para la gestión de proyectos de software desarrollados con XP," in *Actas XVIII Congreso Argentino de Ciencias de la Computación*, 2012, p. 10.
- [6] Firas Glaiel, "Agile Project Dynamics: A Strategic Project Management Approach to the Study of Large-Scale Software Development Using System Dynamics," Massachusetts Institute of Technology , Tesis de Máster 2012.
- [7] Ventana System Inc. (2013) Vensim. [Online]. <http://www.vensim.com>
- [8] J Aracil, *Dinámica de Sistemas*. Madrid, España: Alianza Editorial, 1997.
- [9] Torrealdea J, *Dinámica de Sistemas. Elementos y Estructuras de un Modelo. Construyendo modelos.:* Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco, 2003.
- [10] Héctor MudarraTeruel, "Automatización de sistemas de desarrollo ágil Scrum: Team & Role - ," sitio Web temoa : Portal de Recursos Educativos Abiertos (REA), Memoria del Proyecto de Fin de Carrera de Ingeniería Informática Bellaterra, Junio de 2010 2010.
- [11] Tiago Keller Ferreira, "Aplicação Do Processo Ágil De Gerenciamento Scrum No Desenvolvimento De Um Jogo Digital.," UFSM, Informática/UFSM - Biblioteca Digital de Trabalhos de Graduação. , Estudo De Caso Em Empresa 2008.
- [12] MaríaLaura. Citón, "Método Ágil IScrum Aplicado Al Desarrollo De Un Software De Trazabilidad," Universidad de Mendoza - Facultad de Ingeniería, Mendoza, Trabajo Final de Carrera Ingeniería en Informática 2006.