

Modelo de Simulación Dinámico de Proyectos de Desarrollo de Software con Scrum

Diego Alberto Godoy¹, Cristian Henry Kotyński¹, Edgardo Aníbal Belloni¹, Héctor Dos Santos¹, Eduardo Omar Sosa

Centro de Investigación en Tecnologías de la Información y Comunicaciones
(C.I.T.I.C.)/Universidad Gastón Dachary (UGD)
Av. López y Planes 6519. Posadas –Teléfono: +54-376-4438677

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

Resumen

En este trabajo se sintetizan los avances de una línea de investigación, la cual tiene definido por objetivo esencial: la construcción y mejora continua de un modelo de simulación dinámico que de soporte a la metodología de gestión de proyectos desarrollados con *Scrum*.

Concretamente, se describen los subsistemas que componen actualmente el modelo propuesto y construido para simular los componentes esenciales de la gestión de un proyecto desarrollado con *Scrum*, así como también, se referencian los experimentos realizados con el modelo. El modelo construido resulta de utilidad para administradores de proyectos considerados novatos en la aplicación de la metodología referida, permitiéndoles conocer de antemano las consecuencias de sus decisiones.

Adicionalmente, se presentan nuevos objetivos y trabajos futuros para la línea.

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

Contexto

El presente trabajo tiene como marco el proyecto de investigación denominado “Simulación como Herramienta para la Mejora de los Procesos de Software”, registrado por la Secretaría de Investigación y Desarrollo de la Universidad Gastón Dachary (UGD) por

R.R. UGD N° 04/I/12, estando radicado en el Centro de Investigación en Tecnologías de la Información y Comunicaciones (CITIC) de dicha universidad por Disposición 10/CITIC/13, y ratificado en 2015 por Disposición 01/CITIC/15.

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”, (iii) Modelo de Simulación Dinámico de proyectos desarrollados con Metodologías Crystal” y (iv) “Dinámica de la Incorporación de prácticas de Test Driven Development a proyectos Ágiles”

Este artículo se enfoca en una síntesis del desarrollo bienal, y resultados, de la segunda línea aludida.

Introducción

La evolución continua del software actual, la necesidad de mantenimiento constante de los sistemas de información para satisfacer cambios dinámicos en los requerimientos, por parte del cliente durante el desarrollo de un proyecto de software, generan un contexto donde la planificación, la administración y el control del mismo, son difíciles de manejar.

Ciertamente, se ha promovido la aplicación de metodologías de gestión de proyectos de software ágiles, ante escenarios que responden al contexto que se acaba de describir; puesto que tales

metodologías permiten generar resultados rápidamente, promoviendo una evolución iterativa e incremental, y contando con un fuerte involucramiento de los usuarios. Entre estas metodologías se incluye *Scrum* [1].

La metodología *Scrum* pone énfasis en las actividades de gerenciamiento basándose principalmente en una planificación adaptativa y en el desarrollo incremental del software con entregas funcionales en breves periodos de tiempo. En tal sentido, ante requerimientos cambiantes y una planificación que deberá adaptarse dinámicamente respecto de tales cambios, resulta sumamente importante - para los *scrum masters* (principalmente), el *team* y el *product owner*- contar con una herramienta que permita simular la gestión de proyectos de desarrollo de software con *Scrum*, y poder evaluar así el impacto de las decisiones sobre la gestión en el desarrollo del proyecto, sin influir o poner en riesgo el proyecto real.

Se han propuesto diferentes modelos y herramientas que permiten simular la administración de proyectos de software – cotéjense, por ej. [2], [3], [4], [5], y [6]. Estas herramientas presentan modelos basados en diferentes metodologías ágiles, en los cuales son propuestos y evaluados diferentes escenarios, algunos de ellos genéricos. Sin embargo, estos trabajos están enfocados en metodologías clásicas en su gran mayoría, o en otras metodologías ágiles distintas a *Scrum*.

Objetivos

Uno de los objetivos de este trabajo es continuar con la evolución del diseño y desarrollo de un Modelo de Simulación Dinámico de Proyectos de Desarrollo de Software que utilizan Metodología *Scrum*.

El modelo propuesto y construido actualmente permite al *scrum master* y demás participantes de un proyecto, evaluar qué impacto tendrán sus decisiones de gestión sobre el mismo, a lo largo del desarrollo. Dicho modelo refleja el efecto del uso de prácticas de *Scrum* en proyectos de desarrollo de software. Es decir, dado que se promueve un enfoque de desarrollo iterativo e incremental, dentro de un

proyecto se negocian con el *product owner* varias entregas o versiones, por lo que el modelo, está destinado a simular un proyecto por vez.

Construcción del Simulador

Para la construcción del simulador 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 de software *Scrum* y sus interrelaciones. Luego, en la Fase de Formulación se ha traducido el Diagrama referido a un Diagrama de Forrester. Finalmente, en la Fase de Evaluación se han realizado las corridas de validación y experimentales del modelo.

Dada la complejidad del sistema y teniendo como base los trabajos de un Modelo Dinámico de Simulación de Proyectos de Software con XP [5] y un Modelo Dinámico Reducido [9] el sistema modelado se dividió en diferentes Subsistemas, de acuerdo con [10]. Se diseñó un Modelo de simulación que comprendiera la mayor cantidad de aspectos posibles.

Cada subsistema tiene preponderancia de influencia en cada uno de las fases de *Scrum*. A continuación, se presenta una breve descripción de cada Subsistema (Figura 1).

1. **Planificación (*Pregame Phase*):** Este subsistema es representa la base para poder iniciar el proceso de simulación. Dentro de los parámetros que especifica el *Scrum Master* están: el inicio, la duración, la cantidad de puntos de historia a realizar y la velocidad ideal estimada de trabajo diario para cada *Sprint*.
2. **Producción (*Development Phase*):** Este subsistema recoge la dinámica del trabajo realizado al momento de definir la cantidad de Puntos de Historia a incluir en el *Sprint* y de cómo estos se completarán bajo condiciones ideales.

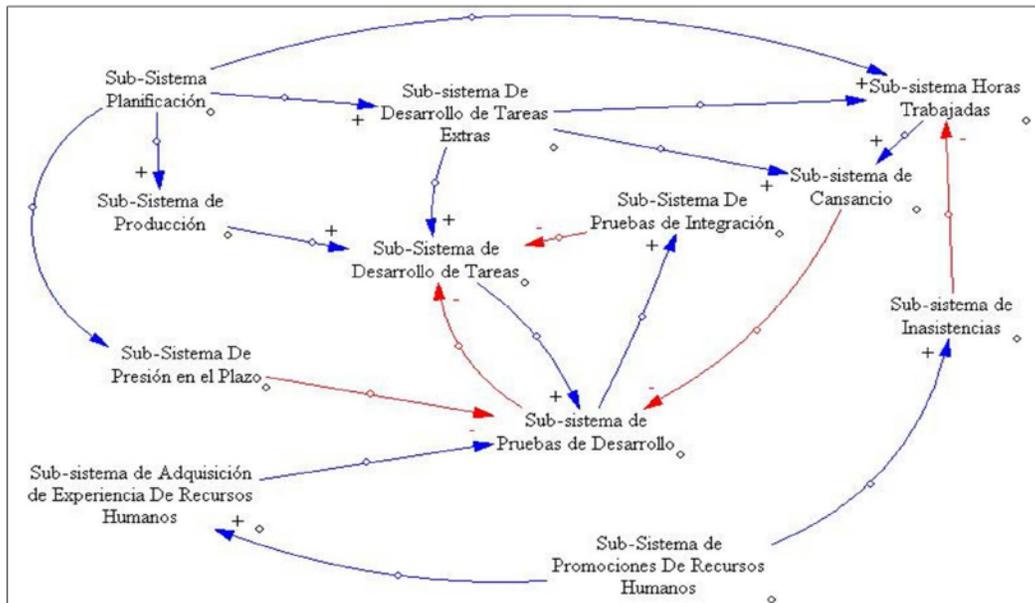


Figura 1 - Diagrama de Influencias entre Subsistemas

3. **Desarrollo de Tareas (*Development Phase*):** Representa la dinámica del trabajo realizada al inicio de cada Sprint en lo relacionado al Diseño y Codificación de tareas.
4. **Pruebas de Desarrollo (*Development Phase, PostGame Phase*):** Este subsistema permite modelar el comportamiento donde se llevan a cabo diversas pruebas que para detectar el número de tareas codificadas que presenten errores de lógica o errores de codificación. El Subsistema comienza a partir de un conjunto de pruebas a realizar que resulta del producto entre un número de pruebas por tarea y la cantidad de tareas codificadas que se generaron en el subsistema explicado en la sección subsistema de desarrollo de Tareas. La cantidad de pruebas a realizar y que tendrá efecto directo sobre el comportamiento del Subsistema las determina el *Scrum Master*.
5. **Pruebas de Integración (*PostGame Phase*):** Expresa la lógica previa a la integración final de las diferentes tareas codificadas. El Subsistema tiene su origen en el Subsistema de Control de Errores de Desarrollo más precisamente en aquellas tareas que no han tenido errores lógicos y ni de codificación, así, estas tareas programadas son sometidas a ciertas pruebas de integración.
6. **Presión en el Plazo (Todas las fases):** Este subsistema modela la presión que sufre el *Team* en la medida que transcurre el tiempo durante el desarrollo del proyecto y en la medida que la finalización del mismo se acerca. En la medida que el tiempo transcurre el plazo acordado para finalizar es cada vez menor y esto hace que aumente la presión.
7. **Desarrollo de Tareas Extras (*Development Phase*):** modela las tareas extras que pudieran surgir por errores o fallas en el análisis de requerimientos realizados al inicio del Sprint y que son detectadas una vez iniciado el Sprint, más aquellas tareas extras planificadas por el *Scrum Master* y el *Team*.
8. **Promociones de RRHH (Todas las fases):** Representa la Promoción de los integrantes del *Team* a lo largo del proyecto. Se consideran aquí los niveles de experiencia: Juniors, Juniors Promocionales y Seniors a partir del trabajo presentado en [11]. Este Subsistema está vinculado directamente con el de Búsqueda y Reclutamiento, ya que este último provee de las entradas necesarias a este

subsistema. Cuando se produce la contratación de un nuevo integrante y este no posee la experiencia suficiente en la metodología se lo considera Junior Contratado, y en la medida que se considere oportuno es promovido a los siguiente niveles de experiencia.

9. **Adquisición de Experiencia de R.H. (Todas las fases):** El Subsistemas presentado en este apartado modela la adquisición de experiencia del *Team* a lo largo del proyecto. La acumulación total de la experiencia es la resultante de ponderar la experiencia de los integrantes y de una tasa de aprendizaje que es establecida por el *Scrum Master*. En la medida que las tasas de aprendizaje sean más altas, más rápido el equipo adquirirá experiencia. Este Subsistema está vinculado al Subsistema de Presión en el Plazo y al de Recursos Humanos, ejerciendo influencia sobre el primero y siendo afectado por la cantidad de integrantes que se modele en el segundo.
10. **Cansancio de RRHH (Todas las fases):** Este apartado modela el cansancio que sufre el *Team* en la medida que avanza el tiempo de desarrollo del proyecto. El *Scrum Master* determina la tasa de cansancio que estima tendrán los integrantes del *Team* en función de la cantidad de horas diarias trabajadas. El valor de la misma surge de la suma de horas normales trabajadas más las horas extras.
11. **Horas Trabajadas de R.H. (Todas las fases):** Modela y representa el total de horas trabajadas por el *Team* a lo largo del proyecto. Las horas totales trabajadas surgen de la suma de horas normales y de horas extras diarias trabajadas por el *Team* durante el proyecto.
12. **Inasistencias de R.H. (Todas las fases):** Modela la generación de Inasistencias no acordadas por parte de los integrantes del *Team* con el *Scrum Master*. El comportamiento del

Subsistema se inicia a partir de la probabilidad de que una inasistencia ocurra. Dicha probabilidad debe ser establecida por el *Scrum Master*.

13. **Variables auxiliares y Constantes:** En este subsistema se encuentran el resto de las variables auxiliares del sistema.

Validación y Experimentos

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 [12]. El segundo “Aplicação Do Processo Ágil De Gerenciamento Scrum No Desenvolvimento De Um Jogo Digital – Estudo De Caso Em Empresa De Software” [13]. El tercero de los casos utilizados Método Ágil Scrum Aplicado Al Desarrollo De Un Software De Trazabilidad” [14].

La validación se ha realizado siguiendo una combinación de los enfoques presentados en [15]. Como criterio objetivo de validación, se ha utilizado la correspondencia entre estructura y comportamiento, en este caso las situaciones reales que ocurren en los proyectos con los bucles de retroalimentación y los arquetipos sistémicos. Además de evidencia empírica de varios proyectos. Por otro lado, también en [15] se hace referencia a la validación por criterio de utilidad, en donde se especifica que si el modelo fue desarrollado para crear escenarios alternativos y tomar decisiones, entonces la utilidad depende de si el modelo logra construir el escenario adecuado, si puede establecerse que el escenario es realista y si permite tomar decisiones informadas. Por esta razón, se tomaron los criterios mencionados para la validación.

Si bien los proyectos de validación no informan datos para todas las variables que contempla el simulador, sirvieron para validar el comportamiento de las variables esenciales.

En cuanto a los experimentos, se han realizado cuatro, los cuales van de menor a mayor complejidad en cuanto a la cantidad de variables que se contemplan. Muestras

de los casos de validación y experimentación realizados se pueden revisar en [16], [17] y [18].

Resultados y Trabajos Futuros

En este trabajo se presentaron los subsistemas de un modelo propuesto para la Simulación Dinámica de Proyectos de Desarrollo de Software que utilizan Metodología *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 –cotéjense, por ej.: [10] y [3], o bien, se modela la especificidad de una de éstas últimas, como por ej. en [4] y [5]. Este modelo es de utilidad para el líder del proyecto y su equipo a la hora de analizar el efecto del uso de la metodología *Scrum* en su proyecto de desarrollo de software.

La validación realizada de diversos proyectos [12], [13] y [14] como casos de “entrenamiento” se evalúa como positiva ya que el modelo construido se comportó de acuerdo con los datos de proyectos reales. El mismo cumple con sus objetivos y puede ser utilizado como herramienta para evaluar diferentes decisiones de gestión.

En relación a nuevos objetivos y trabajos futuros, se espera en -primer término- avanzar con la construcción de otros subsistemas, como por ej. el de Cálculo de Costos y el de Comunicación en el *Team*.

Por otra parte, se planea experimentar con el simulador en cátedras que –como parte de su intensidad de formación práctica– promuevan actividades de proyecto y desarrollo de software utilizando metodologías ágiles. Resulta también necesario construir bases de datos de proyectos de software reales desarrollados con *Scrum*, ya que actualmente resulta difícil contar con datos *post mortem* de proyectos gestionados con métodos ágiles, en general, y *Scrum* en particular. Además, se prevé comparar la utilización de los simuladores de proyectos en *Scrum* y en XP [4] y [5], con el fin de verificar la existencia de aspectos estructurales y de comportamiento

comunes. En este sentido, se persigue un enfoque button up en contraposición al top down planteado en [6].

Recursos Humanos

El equipo de trabajo se encuentra conformado por un Doctor en Ciencias Informáticas, un Maestreado en Ingeniería de Software (con tesis en evaluación final), un Maestreado en Ingeniería Web, un Ingeniero en Informática, y cuatro estudiantes en período de realización de trabajos finales de grado en el contexto de la carrera de Ingeniería en Informática de la UGD.

Actualmente, el número de tesinas de grado aprobadas en el contexto del proyecto marco es de tres –dos en la línea descripta en este trabajo–, 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] Diego Alberto Godoy and Tamara Kasiak, Modelo dinámico de simulación para la gestión de proyectos de software desarrollados con XP, Octubre 2012, XVIII Congreso Argentino de Ciencias de la Computación. Extensión: 10 p.
- [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] Miguel Toro, Mercedes Ruiz Isabel Ramos, "Modelo Dinámico Reducido," Universidad de Sevilla, Técnico LSI-2001-01, 2001.
- [10] J Torrealdea, *Dinámica de Sistemas. Elementos y Estructuras de un Modelo. Construyendo modelos.:* Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco, 2003.
- [11] Nuria Calvo Babío, "Análisis de la simulación dinámica como herramienta de apoyo directivo a la gestión del talento en organizaciones de consultoría," *Revista Europea de Dirección y Economía de la Empresa*, vol. 18, no. 1, pp. 135-154, 2009.
- [12] Héctor Mudarra Teruel, "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.
- [13] 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.
- [14] María Laura Citón, "Método Ágil I Scrum Aplicado Al Desarrollo De Un Software De Trazabilidad," Universidad de Mendoza - Facultad de Ingeniería, Mendoza, Trabajo Final de Carrera Ingeniería en Informática 2006.
- [15] L., & Bartó, Godoy, "Validación y valoración de modelos en la Dinámica de Sistemas.," *Revista Argentina de Enseñanza de la Ingeniería*, 2002.
- [16] Diego Alberto Godoy, Edgardo A. Belloni, Henry Kotynski, Hector H Dos Santos, and Eduardo Omar Sosa, "Simulando Proyectos de Desarrollo de Software Administrados con Scrum," in *XVI Workshop de Investigadores en Ciencias de la Computación RedUNCI*, Ushuaia, 2014.
- [17] Diego Alberto Godoy, Edgardo A. Belloni, Eduardo Omar Sosa, Henry Kotynski, and Juan de Dios Benítez, "Evaluación de alternativas de gestión en proyectos de software desarrollados con scrum utilizando dinámica de sistemas," in *XX Congreso Argentino de Ciencias de la Computación*, 2014, p. 10.
- [18] Diego Alberto Godoy, Eduardo Omar Sosa, Edgardo Belloni, and Henry Kotynski, "Simulación Dinámica de Gestión de Tareas en Proyectos Desarrollados Con Scrum," in *II Congreso Nacional de ingeniería informática/ingeniería de sistemas (CoNaIISI)*, Universidad Nacional de San Luis, San Luis, 2014, 2014.