

Gestión Cuantitativa de Proyectos y Entrega Continúa en Entornos Ágiles

Dapozo, Gladys N.; Greiner, Cristina; Irrazabal, Emanuel; Medina, Yanina; Ferraro, María; Mascheroni, Agustín

Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura. Universidad Nacional del Nordeste
{gndapozo; cgreiner, eirrazabal, ferraro, yanina}@exa.unne.edu.ar

Resumen

Este proyecto de investigación aborda temas de gestión cuantitativa de proyectos y entrega continua en entornos ágiles buscando promover y/o generar métodos y herramientas que contribuyan a mejorar la calidad del proceso y del producto software. La automatización de los métodos de estimación permitirá a las empresas mejorar la precisión de la duración y costo de los proyectos de desarrollo bajo diferentes enfoques: tradicional y ágil, generando además información histórica que retroalimenta a los métodos de estimación. La gestión de incidentes, la sistematización de mecanismos de trazabilidad de los requerimientos y la incorporación de procedimientos para la entrega continua, contribuirá a mejorar la mantenibilidad del software, siendo este atributo uno de los más críticos en los ámbitos reales de producción de software. En todos los casos, se promueve el uso de técnicas de desarrollo ágil.

Palabras clave: Estimación software. Incidencias. Entrega Continua. Gestión de Requerimientos. Agilismo.

Contexto

Las diferentes líneas de I/D presentadas en este trabajo corresponden al proyecto PI-F10-2013 “Métodos y herramientas para la calidad del software”, acreditado por la Secretaría de Ciencia y Técnica de la Universidad Nacional del Nordeste (UNNE) para el periodo 2014-2017.

Introducción

Gestión cuantitativa

La gestión cuantitativa en el desarrollo de software proporciona una visión del grado de cumplimiento de metas, así como las causas que explican desviaciones significativas en procesos o productos [1]. La gestión de proyectos en base al conocimiento cuantitativo contribuye a la determinación de los aspectos de mayor relevancia, cuyo rendimiento afecta en forma significativa al logro de los objetivos del proyecto y la satisfacción de los clientes, obteniendo productos con un mayor nivel de calidad.

En este proyecto la gestión cuantitativa se enfoca en la estimación, la gestión de incidentes y en la trazabilidad de requerimientos.

Estimación

El desarrollo de software requiere de la estimación para controlar y administrar los recursos antes y durante el proyecto. Existen diferentes modelos de estimación que van desde las técnicas orientadas a los procesos hasta métodos paramétricos o algorítmicos, basados en datos históricos utilizados para definir y calibrar el modelo [2][3]. La técnica (o conjunto de técnicas) que se utiliza debe adaptarse a los datos disponibles y la naturaleza del problema en la estimación. En este sentido, es importante mantener diferentes técnicas de estimación de acuerdo con la metodología de desarrollo a utilizar.

La realidad de las empresas de la región NEA muestra que las áreas o empresas de software, en gran medida, desconocen los métodos paramétricos y

herramientas para la estimación de esfuerzo, duración y costo de los proyectos software [4], siendo lo usual la estimación por juicio de expertos.

En esta línea se estudian diversos métodos de estimación, principalmente aquellos orientados al desarrollo web. En particular, se consideró un método elaborado por el Centro de Estudios de Ingeniería de Software de la Universidad de la Frontera (CEIS-UFRO) que parte de una especificación de requerimientos basada en casos de uso y la productividad, llega a definir el costo de un producto de software transaccional web [5].

Gestión de incidentes

La medición de código fuente aporta a la gestión cuantitativa, por ejemplo, estableciendo relaciones en cuanto a la complejidad del software y la propensión a fallos, lo cual ofrece indicios sobre qué clases o porciones de código debe enfocarse la prueba de software.

Vinculado con el código fuente aparece el concepto de deuda técnica, la que se contrae cuando las actividades relacionadas con el desarrollo de software no se realizan con los niveles de calidad adecuados y por tanto puede disminuir la mantenibilidad del código u ocasionar mayores costos en el proyecto [6].

A medida que el proyecto crece en cantidad de requerimientos, tamaño del código o personas involucradas, el éxito resulta cada vez más difícil de alcanzar. Es importante que los equipos se mantengan dentro de presupuestos y tiempos planificados. Una forma efectiva para alcanzar las metas es que los equipos cuenten con herramientas que les permitan sistematizar aspectos tales como peticiones de cambios, incidencias y bugs. SWEBOK define como petición de cambio (Change Request, CR) a “una solicitud para ampliar o reducir el alcance del proyecto; modificar las políticas, procesos, planes o procedimientos; modificar los costos o presupuestos; o revisar los horarios. Una fuente de CR es

el inicio de medidas correctivas en respuesta a informes de problemas [7].

La International Software Testing SWEBOK define un bug como “un defecto en el código fuente. Un paso, proceso o definición de datos incorrectos en el programa. La codificación de un error humano en código fuente” [7].

En la literatura especializada surge que la asignación de incidentes es un aspecto crítico. En [8] se describe cómo la tarea de asignar un CR puede ser costosa en tiempo, y la reasignación doblemente costosa, señalando la falta de uso de una herramienta automatizada que asista en esta tarea, y la necesidad de considerar la información histórica para tomar decisiones efectivas, tales como la carga de trabajo actual y conocimiento de habilidades de sus desarrolladores. En [9] se describe la asignación de bug como un proceso social debido a la complejidad que implica determinar la persona más adecuada para resolver el problema, señalando la importancia de tener una base de conocimiento sobre la experticia de los desarrolladores y herramientas que abarquen información socio-técnica de la organización. En [10] se menciona el conocimiento de los desarrolladores y la repercusión en la carga de trabajo del equipo como desafíos claves en la asignación de un bug o incidentes.

Trazabilidad

Los procesos de evolución del software (adaptación a nuevos requerimientos), varían considerablemente dependiendo del tipo de software a mantener, los procesos de desarrollo utilizados en una organización y el personal implicado en el proceso. Por tanto, es imprescindible contar con una herramienta de gestión de requerimientos que acompañe este proceso, para minimizar el impacto de los cambios en el sistema y lograr que los mismos se realicen de manera planificada y controlada. Se reconoce que realizar el seguimiento a los requerimientos a lo largo del proceso de desarrollo de software no es tarea fácil. Todo artefacto

de software cambia en el tiempo por la evolución en las necesidades de los usuarios. Para minimizar el impacto causado por dicha evolución, la práctica de la trazabilidad ha sido estudiada e implementada con diferentes modelos y técnicas que permiten lograr mayor calidad en los productos de software [11]. El desafío es sin duda, acompañar la gestión de cambios con el suficiente nivel de trazabilidad que logre el equilibrio entre calidad de software y tiempo dedicado a la gestión.

Entrega continua

La Entrega Continua de Software, en inglés Continuous Delivery (CD), se define como un enfoque en el cual los equipos mantienen la producción de software en ciclos cortos de tiempo, asegurando que el producto pueda ser lanzado de manera fiable en cualquier momento [12]. La idea es poder lanzar a producción un producto software libre de defectos “*con solo apretar un botón*” [13].

Existe un concepto similar al CD, que es el de Despliegue Continuo, en inglés Continuous Deployment (DC). El DC es una actividad que consiste en lanzar cambios continuamente al ambiente de producción [14]. La principal diferencia se encuentra en la fiabilidad a la hora de lanzar una nueva versión del producto: el DC busca integrar código a producción una, dos, y muchas más veces en el mismo día, en cambio, el CD se centra en hacerlo con la certeza de que el producto que se está lanzando a producción tiene un alto grado de calidad y se encuentre libre de defectos.

En el intento de implementar el CD, muchas organizaciones terminaron alcanzando solamente el DC. Por ejemplo, algunas empresas como Facebook, Atlassian, IBM, Adobe, Microsoft y Flickr, han tenido éxito en la implementación de diferentes enfoques para realizar entregas en periodos cortos de tiempo. Sin embargo, diferentes autores señalan que el proceso de lanzar

rápidamente productos software a producción, aún presentan inconvenientes por resolver [15].

Uno de los principales problemas estaría en la calidad del producto software. Ésta puede disminuir, dado que, al realizarse los despliegues del sistema con mayor frecuencia, aparecen más defectos en el producto [13]. Por tanto, es esencial desarrollar un enfoque de priorización de los diferentes aspectos en la calidad del producto software, teniendo en cuenta la forma de trabajo actual de las empresas de desarrollo software.

Para adoptar este enfoque, en la literatura se utiliza el concepto de Tubería de Despliegue (DP - Deployment Pipeline). El DP es un estándar para automatizar el proceso de CD [16]. A pesar de que cada organización puede variar en la implementación de este estándar, el mismo se conforma de las siguientes actividades: Instalación, Compilación, Pruebas de Aceptación, Pruebas de Capacidad, Pruebas Manuales, Liberación a Producción.

Líneas de investigación y desarrollo

En la línea de Gestión Cuantitativa se propone:

Estimación

- Siguiendo con la línea de resultados obtenidos en [17] se trabajará en el desarrollo de una herramienta para automatizar la estimación de costos en proyectos web aplicando métodos de estimación ágiles. Con esta herramienta y la elaborada anteriormente, se podrán realizar estimaciones con diferentes técnicas orientadas a distintas metodologías de desarrollo, además de generar información histórica para mejorar la precisión.

Trazabilidad

Desarrollar una aplicación de gestión de proyectos en entornos de la Administración Pública para asegurar la

trazabilidad de los requerimientos a las actividades, tareas y el código fuente.

- *Gestión de incidentes*

Diseñar y construir una herramienta que permita realizar el seguimiento de las incidencias de los proyectos de software, aportando información de las características socio técnicas de los equipos involucrados. La herramienta permitirá gestionar un histórico de las incidencias, desde su reporte inicial hasta su corrección, siendo configurable el ciclo de vida de las mismas para que se adapte a las necesidades de la organización. A su vez, tendrá en cuenta la experiencia, carga de trabajo actual e involucramiento de los desarrolladores.

En la línea de *Entrega Continua en entornos ágiles*:

- Mejorar el algoritmo de pruebas para el desarrollo de compatibilidades web implementado en [18] y evitar escenarios de falsos positivos.
- Construir una técnica de Testing Continuo en base al framework para el desarrollo de pruebas REST [19].
- Desarrollar nuevos Estudios de Caso al implantar técnicas ágiles en entornos de la Administración Pública [20][21].

Resultados obtenidos

Los principales resultados de las actividades desarrolladas en estas líneas son:

- Se evaluaron cuatro metodologías específicas para la estimación de proyectos web y se realizó una aplicación que automatiza, simplifica y guía la aplicación de las mismas, que fue validada con datos experimentales. Los resultados indican que, a mayor especificidad en cuanto al tipo de proyecto y la adecuada configuración de parámetros que identifican el entorno de la aplicación, el ajuste de la estimación es mayor. Se espera que esta herramienta contribuya con la calidad del software permitiendo mejorar la

estimación de proyectos web, facilitando los cálculos, realizando un seguimiento de las estimaciones y aportando información histórica como feedback para estimaciones posteriores [22].

- Basado en los resultados obtenidos en [5], se realizó una aplicación web (CostEs) para estimar costos de desarrollo que implementa la técnica de estimación temprana y proporciona valores de tamaño, esfuerzo y costo [23][24]. La herramienta fue validada con los datos de un proyecto web finalizado, el portal del Gobierno de Corrientes, obteniendo resultados satisfactorios. Como punto fuerte, CostEs almacena toda la información referida al proyecto y al proceso de estimación, y de esta forma permite ir mejorando las estimaciones futuras.

En la línea de *Entrega Continua en entornos ágiles*:

- En [18] se describe el desarrollo de una herramienta para pruebas de compatibilidad entre navegadores web. La técnica propuesta es una iniciativa para automatizar el proceso de pruebas de compatibilidad mediante un algoritmo de comparación de imágenes. Para validarla, ha sido implementada en una empresa de desarrollo software de escala nacional que trabaja con desarrollo continuo de software. Los resultados demostraron que la herramienta permite acelerar el proceso de pruebas, mediante la automatización de pruebas de compatibilidad web. Por un lado, los tiempos de ejecución de pruebas disminuyeron un 82%, y esto también redujo el tiempo total del proceso de liberación de las versiones del sitio.
- En [19] se presentó un framework que permite el desarrollo de pruebas automatizadas para servicios REST. El tiempo de ejecución de un lote de pruebas que llevaba 3 horas, realizado manualmente con el equipo de 15 personas, pasó a ser de 15 minutos en

un servidor de integración continua 100% automatizado.

- En [20] y [21] se describe la implementación de la metodología de gestión de proyectos SCRUM en dos oficinas de sistema de administraciones públicas provinciales en Corrientes y Misiones. Como primer paso, se indicaron las características institucionales con el fin de establecer un diagnóstico del equipo de trabajo actual y el plan de implantación de SCRUM, así como la enumeración de las lecciones aprendidas al construir el plan. Como resultado de ello, la implementación tuvo una acogida general positiva por parte de todo el equipo de desarrollo y sus directivos, demostrando interés y predisposición para experimentar una nueva forma de trabajo en desarrollo ágil de software.

Finalmente, en el marco del proyecto como una línea de trabajo emergente, se trabajó durante el año 2016 en un proceso de explotación de información adaptado del modelo propuesto en [25], el cual combinó las etapas de la metodología CRISP-DM con actividades de la metodología COMPETISOFT. En [26], [27] y [28] se publicaron los resultados; abordando desde el punto de vista de la ingeniería de procesos una propuesta tecnológica para la Analítica Académica en la Facultad de Ciencias Exactas y Naturales y Agrimensura de la Universidad Nacional del Nordeste. Para ello se explotó la información de los alumnos de la carrera Licenciatura en Sistemas de Información disponibles en el sistema SIU guaraní, de amplio alcance nacional.

Formación de recursos humanos

En el Grupo de Investigación sobre Calidad de Software (GICS) están involucrados 5 docentes investigadores, 1 becario de investigación de pregrado y 1 tesista de doctorado. Tres alumnos de la carrera finalizaron la misma con proyectos vinculado a estos temas.

Referencias

- [1] Gou, L.; Wang, Q.; Yuan, J.; Yang, Y.; Li, M.; Jiang, N. "Quantitative defects management in iterative development with BiDefect". *Software Process Improvement and Practice*, 14(4), 227-241. 2009.
- [2] Andrés, J.; Fernandez-Lanvin, D.; Lorca, P. "Cost estimation in software engineering projects with web components development". 2015, vol.82, n.192
- [3] Mendes, E. "Using knowledge elicitation to improve Web effort estimation: Lessons from six industrial case studies," *Software Engineering (ICSE)*, 2012.
- [4] Dapozo, G. N.; Greiner, C. L.; Irrazábal, E.; Medina, Y; Ferraro, M. A.; Lencina, A. B. "Características del desarrollo de software en la ciudad de Corrientes". *CACIC* 2015.
- [5] Díaz Villegas, J. E.; Robiolo, G. "Método de estimación de costos de un producto de software Web". 43 JAIIO - ASSE 2014 - ISSN: 1850-2792
- [6] Cunningham, W. "The wycash portfolio management system," in *OOPSLA '92: Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum)*, 1992.
- [7] Bourque, P.; Fairley, R. E. *SWEBOK*, Tercera ed., IEEE, 2014, pp. 82-87.
- [8] P. Anselmo da Mota y Y. Cerqueira. "Towards Understanding Software Change Request". *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, 2013.
- [9] P. Guo y T. Zimmermann. "Not my bug! and other reasons for software bug report reassignments". *CSCW '11 Proceedings of the ACM 2011 Conference on*, 2011.
- [10] G. Bortis y A. van der Hoek. Teambugs: a collaborative bug. *CHASE '11 Proceedings of the 4th International*, pp. 69-71, 2011.
- [11] M. Tabares, f. Arango, R. Anaya. Una Revisión de Modelos y Semánticas Para la Trazabilidad de Requisitos. *Revista EIA*, ISSN 1794-1237 Número 6, p. 33-42. Diciembre 2006. Escuela de Ingeniería de Antioquia, Medellín. Colombia.
- [12] L. Chen, "Continuous Delivery: Huge Benefits, but Challenges Too" in *IEEE Software* 03/2015. V. 32(2).

- [13] J. Humble and D. Farley. "Continuous delivery: reliable software releases through build, test, and deployment automation", 1st ed. Boston, US: Pearson Education, 2010.
- [14] H. H. Olsson, H. Alahyari, and J. Bosch. "Climbing the 'Stairway to Heaven'-A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software" in 38th EUROMICRO Conference, 2012, pp. 392-399.
- [15] B. Fitzgerald and K. Stol, "Continuous Software Engineering and Beyond: Trends and Challenges," in 1st International Workshop on Rapid Continuous Software Engineering, 2014.
- [16] O. Prusak. "Continuous Testing: The Missing Link in the Continuous Delivery Process". Blaze Meter. 2015.
<https://blazemeter.com/blog/continuous-testing-missing-link-continuous-delivery-process>
- [17] G.N. Dapozo, Y. Medina, A. Lencina. "La práctica de la estimación en empresas y áreas de Sistemas". Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação. v.1, n. 4. 2015.
- [18] M. A. Mascheroni, M. K. Cogliolo, E. Irrazabal. "Automatización de pruebas de compatibilidad web en un entorno de desarrollo continuo de software". 45 JAIIO - ASSE 2016 - ISSN: 2451-7593 – Pág. 51-63.
- [19] M. A. Mascheroni, E. Irrazabal. Framework para la creación y ejecución de pruebas automatizadas sobre servicios REST. XXII Congreso Argentino de Ciencias de la Computación. CACIC 2016.
- [20] R. López Lovera, R. Morante, E. Irrazabal. Plan de implementación de la metodología SCRUM y primeros resultados en la Dirección de Desarrollo de Sistemas de la Municipalidad de Posadas. 45 JAIIO - SIE 2016 - ISSN: 2451-7534 Pág. 30-39.
<http://45jaiio.sadio.org.ar/sites/default/files/SIE-05.PDF>
- [21] C. Pinto Luft, E. Irrazabal. Hacia la implantación de metodología SCRUM en la Inspección General de Personas Jurídicas de Corrientes: primeras lecciones aprendidas. Anales del IV Seminario Argentina-Brasil de Tecnologías de la Información y la Comunicación. Universidad Nacional del Nordeste. ISBN 978-987-3619-15-1. 2016.
- [22] O. G. Pedrozo Petrazzini, C. Greiner. Herramienta para apoyar la estimación en el desarrollo de aplicaciones web. 45 JAIIO - EST 2016 - ISSN: 2451-7615 – Pág. 116-1130
<http://45jaiio.sadio.org.ar/sites/default/files/EST-1873.pdf>
- [23] Lencina, A.; Dapozo, G. N.; Medina, Y. Estimación de costo en el desarrollo de software. XXII Reunión De Comunicaciones Científicas y Tecnológicas 2016. Campus Corrientes. 4 y 15 de junio de 2016.
- [24] Lencina, A; Medina, Y.; Dapozo, G. N. Aplicación para estimar costos en proyectos de software. 45 JAIIO - ASSE 2016 - ISSN: 2451-7593 – Pág. 181-192.
<http://45jaiio.sadio.org.ar/sites/default/files/asse-12.pdf>
- [25] J. Vanrell. Un Modelo de Procesos para Proyectos de Explotación de Información. Tesis de Maestría en Ingeniería en Sistemas de Información. Buenos Aires. 2012.
- [26] Lopez, M.E.A., Dapozo, G.N., Greiner, C.L. Modelo de proceso para proyectos de explotación de información en el ámbito académico. XXII Reunión De Comunicaciones Científicas Y Tecnológicas 2016. Campus Corrientes. 2016.
- [27] Lopez, M.E.A., Dapozo, G.N., Greiner, C.L. Propuesta tecnológica como apoyo a procesos de Analítica Académica en FaCENA-UNNE. Anales del IV Seminario Argentina-Brasil de Tecnologías de la Información y la Comunicación. Universidad Nacional del Nordeste. ISBN 978-987-3619-15-1. 2016.
- [28] Lopez, M.E.A., Dapozo, G.N., Irrazabal, E.A., Greiner, C.L. Proceso de Explotación de Información para Analítica Académica en FaCENA-UNNE. Revista Latinoamericana de Ingeniería de Software. ISSN 2314-2642. Vol 4, No 6 (2016).