

Una estrategia centrada en el usuario y guiada por modelos para el desarrollo sistemático de módulos ERP

M.Á. Olivero¹, L. Morales-Trujillo¹, D. Lizcano², F.J. Domínguez-Mayo³

¹Grupo de Investigación IWT2, Universidad de Sevilla, España

²Escuela de Ciencias Técnicas e Ingeniería, Madrid Open University (MOU), España

³Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, España

{miguel.olivero, leticia.morales}@iwt2.org,
david.lizcano@udima.es, fjdominguez@us.es

Abstract. Los sistemas ERP están compuestos por diferentes módulos funcionales en el que cada uno aborda un área de negocio distinta. Los desarrollos que se llevan a cabo sobre estos módulos se gestionan de manera independiente por cada uno, lo que permite abordar de forma controlada un gran número de requisitos de información relacionados. En este contexto trabaja la startup G7Innovation para su producto iMEDEA. En este estudio hemos combinado dos metodologías, Design Sprint y NDT 4.0; consiguiendo sistematizar el desarrollo de módulos de sistemas ERP. Esta combinación permite usar Design Sprint para generar y validar el prototipado, y NDT 4.0 para hacer el estudio, análisis y diseño del software a desarrollar. Además, de acuerdo a las especificaciones definidas en NDT, se puede automatizar la generación de código del módulo ERP. Esta propuesta se ha validado en un caso de estudio en colaboración con la startup G7Innovation donde hemos aplicado ambas metodologías sobre Odoo, un sistema ERP de código abierto escrito en lenguaje Python. Gracias al uso de estas dos metodologías hemos producido un módulo afín a las necesidades de la clínica reduciendo costos, tiempos y fallos humanos.

Keywords: Design Sprint, ERP, NDT, Odoo.

1 Introducción

El proyecto iMEDEA es una solución desarrollada por G7Innovation [1]. Este proyecto está basado en el ERP Odoo [2] para facilitar la gestión de las unidades de reproducción humana. Al estar compuesta por diferentes módulos que pueden ser usados independientemente se ofrece mejor apoyo a los clientes del negocio, dando tiempos rápidos de respuesta a sus problemas y con un eficiente manejo de información [3]. Todas estas características permiten la toma oportuna de decisiones y disminución de los costos totales de las operaciones.

En este trabajo se presenta una propuesta metodológica que engloba dos metodologías, Design Sprint [4][5] y NDT [6] (Navigational Development Techniques);

metodologías ya existentes para el prototipado y la generación de código respectivamente con el fin de reducir la curva de aprendizaje necesaria para desarrollos simples.

En este trabajo nos centramos en los módulos de Odo, al ser el principal producto que desarrolla la startup G7Innovation, con quien validamos esta propuesta.

El resto de este artículo se organiza de la siguiente forma: se define el proyecto de I+D en la sección 2, las herramientas empleadas en la sección 3 y finaliza con las conclusiones en la sección 4.

2 Proyecto I+D

Esta guía está diferenciada en tres etapas: (1) *Descubrimiento del producto software*, donde aplicamos Design Sprint para conocer de primera mano las necesidades reales de los usuarios; (2) *Desarrollo*, donde se especifica técnicamente el proyecto, construimos y probamos el sistema de forma desatendida ad haciendo uso de NDT y transformaciones de modelos; y (3) *Operación*, para su puesta en marcha y su mantenimiento.

El *descubrimiento del producto* pretende conocer cuáles son las necesidades y motivaciones que llevan a buscar una solución. Para ello se plantean distintas propuestas con el objetivo de obtener diferentes enfoques que ayuden en la toma de decisiones, apoyándose la información que se ha obtenido de cada alternativa. Una vez se acuerda qué solución se desea alcanzar se boceta el prototipo. El prototipo debe ser lo suficientemente realista para poder finalmente hacer una validación de la propuesta de solución con el usuario. El uso de prototipos para validar el proyecto nos permite desechar las alternativas a bajo coste.

El *desarrollo* está dividido en tres hitos: *especificación técnica*, *construcción* y *verificación y validación*.

En la *especificación técnica* se definen la estructura y el comportamiento de nuestro sistema haciendo uso de modelos UML [7] y modelos IFML [8]. Conociendo el resultado que queremos obtener gracias a los prototipos y una vez definida la arquitectura técnica se inicia la etapa de construcción.

Durante la *construcción*, hacemos uso de operaciones M2T que nos permiten obtener código partiendo de los modelos de diseño. Con ello se genera código del modelo de datos de nuestro sistema, tomando el diagrama de clases del sistema de NDT como modelo fuente. De forma análoga se pueden obtener los códigos de las funciones gracias al modelo de requisitos funcionales y casos de uso que ofrece NDT. Al estar automatizada la generación de código se aumenta la productividad y se reduce la posibilidad de fallos humanos.

La *verificación y validación* de la calidad del proyecto es el último paso antes de poner el producto en producción. Durante esta etapa se deben llevar a cabo las pruebas funcionales y de aceptación, entre otras. Para alcanzar a un entorno estable de pruebas se hace uso de otro tipo transformaciones M2M que ofrece NDT. Esta metodología, a partir de su herramienta CASE, permite generar la documentación del plan de pruebas.

Concluye nuestra propuesta con la fase de *operaciones*. Esta fase es la responsable de que se garantice un buen funcionamiento del sistema y su infraestructura tecnológica en entornos de producción una vez que se despliega la aplicación [9].

3 Tecnologías Usadas

En la fase de descubrimiento del producto se ha hecho uso de la herramienta “Power-Point” para el desarrollo y validación de los prototipos. Aunque existen numerosas alternativas, con las que crear prototipos rápidamente, hemos elegido esta por su gran flexibilidad. Usando el modo patrón de esta herramienta se pueden crear plantillas de las pantallas, que posteriormente se pueden completar añadiendo el contenido que queramos validar. Además, esta herramienta permite añadir vínculos con los objetos de modo que se puede simular la navegación entre las pantallas usando “botones”. Esto nos ha permitido que los prototipos diseñados sean bastante realistas para validar la propuesta de solución con el usuario. Gracias a que conocemos la interfaz de Odo, y las limitaciones tecnológicas, hemos generado prototipos realistas a bajo coste. Esto nos permitía fácilmente desechar las alternativas que no se ajustaban a las expectativas del usuario o del cliente. En este punto se podría haber hecho uso de Odo Studio. Una herramienta que facilita la generación de módulos con una interfaz gráfica tipo Drag & Drop. A pesar de su ventaja, no la contemplamos por la dificultad encontrada al querer reutilizar el código generado o adaptarlo para integrarlo con API privadas.

La especificación técnica, dentro de la fase de desarrollo se ha llevado a cabo haciendo uso de NDT-Suite [10]. NDT-Suite es un conjunto de herramientas que facilita el uso de la metodología NDT, que está integrada sobre la herramienta Enterprise Architect [11]. Con esta herramienta y partiendo del prototipo acordado, hemos podido definir los diagramas de clase del sistema, actores, casos de uso con sus requisitos de interacción y algunos requisitos no funcionales.

Gracias a la posibilidad de realizar transformaciones M2M [12] y M2T [13] que ofrece NDT-Suite obtuvimos de forma automática algunos de estos requisitos, código Python que usamos como base para la construcción y la documentación del plan de pruebas para la verificación y validación.

Para la construcción se ha utilizado el código autogenerado por las transformaciones M2T como base. No obstante, sigue siendo necesario adaptar algunos parámetros de configuración del código en Odo para su integración con el sistema. Al estar automatizada la generación de gran parte de este código conseguimos aumentar la productividad y reducir la posibilidad de fallos humanos.

La verificación y validación del sistema que se genera se lleva a cabo usando la herramienta Katalon, ejecutando el plan de pruebas de sistema que genera NDT a partir de los definidos. Es relevante resaltar que las pruebas de integración de este módulo con los ya existentes no ha sido posible autogenerarlas con el uso de estas metodologías, a pesar de la importancia que tiene en este tipo de desarrollos modulares.

Una vez G7Innovation entregó el sistema validado y verificado, el equipo informático de la clínica se responsabilizó de llevar a cabo la fase de operaciones.

4 Conclusiones

La propuesta de esta metodología surge como una necesidad para facilitar la comunicación con los clientes, y permitir un desarrollo que sea amigable con el usuario sin

que ello suponga un incremento en el coste. Tras aplicar estas metodologías hemos podido generar un producto más cercano al usuario y automatizando en gran medida su construcción. Los prototipos pudieron ser muy realistas ya que conocíamos la tecnología sobre la que desarrollar, y sus limitaciones. El uso de estas metodologías permite reducir necesidad de cambios en el código en estados avanzados del proyecto ya que se validan las necesidades del cliente con el prototipo. Y los errores humanos gracias a la generación automática del código. No obstante, obliga a adaptar el código y ejecutar pruebas de integración mucho más exhaustivas para verificar la compatibilidad del código generado.

Agradecimientos

Este trabajo está parcialmente financiado por el proyecto POLOLAS (TIN2016-76956-C32-R) y bajo el programa de estancias pre/postdoctorales del Plan Nacional I+D+i en UDIMA.

Referencias Bibliográficas

1. G7Innovation Accesible in: <http://www.g7innovation.com/> (last access: April 2019)
2. Odo. Accesible in: https://www.odoo.com/es_ES/ (last access: April 2019)
3. Delgado, J., & Marín, F. (2000). Evolución en los sistemas de gestión empresarial. Del MRP al ERP. *Economía industrial*, 331(1), 51-58.
4. Banfield, R., Lombardo, C. T., & Wax, T. (2015). Design sprint: A practical guidebook for building great digital products. " *O'Reilly Media, Inc.*".
5. Knapp J., Zeratsky J., Kowitz B. (2018). Sprint: El método para resolver problemas y testar nuevas ideas en solo 5 días. *España: Conecta*.
6. Escalona, M. J., Mejías, M., & Torres, J. (2004, September). Developing systems with NDT & NDT-Tool. In *13th ISD, Vilna, Lithuania* (pp. 149-59).
7. OMG (2015, March). OMG Unified Modeling Language, v2.5. Accesible in: <http://www.omg.org/spec/UML/> (last access: April 2019).
8. Brambilla, M., Fraternali, P., et al. (2014). The interaction flow modeling language (ifml), version 1.0. Technical report, *Object Management Group (OMG)*
9. Hertvik, J. (2014). *What Does IT Operations Management Do? (ITOps)*. Joe Hertvik: Tech Machinist. Accesible in: <http://joehertvik.com/> (last access: April 2019)
10. García J., Escalona M.J., Domínguez-Mayo F.J., Salido A. (2014). NDT-Suite: A Methodological Tool Solution in the Model Driven Engineering Paradigm. *Journal of Software Engineering and Applications*, 7 (1), 206-217
11. Enterprise Architect. Accesible in: <https://sparxsystems.com/products/ea/> (last access: April 2019)
12. OMG. (2008), "Documents Associated with Meta Object Facility (MOF) 2.0 Query/View/Transformation", *Object Management Group*.
13. OMG. (2008), "MOF Model to Text Transformation Language (MOFM2T), 1.0", *Object Management Group*.