

07-002

UCAD Methodology: principles for assuring the development of digital solutions that are focused on the user

Erik Aranburu Zabalo; Ganix Lasa Erle; Daniel Reguera Bakhache

Escuela Politécnica Superior de Mondragon Unibertsitatea;

The so-called Industry 4.0 has caused the introduction of new digital technologies that increase and improve the production and potential of the new intelligent industry. For that, it's necessary a good communication between the machines and the users, so that they could exploit their abilities. Nevertheless, on the development of such new technologies the aim its usually the optimization of the system functionalities, setting aside the user on the development process and the improvement of aspects as the usability of the application. Therefore, there is lack of structured methods that have the users as the starting point.

With the objective of working on this lack, this communication presents the methodology User Centered Agile Design (UCAD), showing the 8 principles that compose it. These principles set out a working method for the creation of Human Machine Interaction (HMI) systems that are centred on the user. This way, it is expected to help designing new solutions centred on the user and creating interactions that fulfil their needs, getting a friendly and intuitive product.

Keywords: user centered design; interface; human machine interaction; methodology, industry 4.0

Metodología UCAD: Principios para garantizar el desarrollo de soluciones digitales que tienen en cuenta a los usuarios

La denominada industria 4.0 ha traído consigo la introducción de nuevas tecnologías digitales que aumentan y mejoran la producción y el potencial de la nueva industria inteligente. Para ello, es necesaria una buena comunicación entre las máquinas y los usuarios, para que éstos puedan explotar sus habilidades al máximo. Sin embargo, en el desarrollo de dichas nuevas tecnologías se tiende a optimizar las funcionalidades del sistema, dejando a un lado al usuario durante el proceso de desarrollo y la mejora de aspectos como la usabilidad de la aplicación. De modo que se observa una carencia de métodos estructurados que tengan como punto de partida a las personas.

Con el objetivo de trabajar sobre esta carencia, esta comunicación presenta la metodología User Centered Agile Design (UCAD), mostrando los 8 principios que la componen. Estos principios plantean un modo de trabajo de creación de sistemas de Interacción Hombre Máquina (IHM) que tiene como eje a las personas. De este modo, se pretende ayudar a diseñar nuevas soluciones centradas en el usuario y crear interacciones que cumplan con sus necesidades, consiguiendo así un producto amigable e intuitivo.

Palabras clave: *diseño centrado en el usuario; interfaz; interacción hombre-máquina; metodología; industria 4.0*

Correspondencia: Erik Aranburu Zabalo (e-mail: erik.aranburu@alumni.mondragon.edu)

Agradecimientos: Los autores agradecemos el apoyo recibido por el Diseinu Berrikuntza Zentroa (DBZ) de Mondragon Unibetsitatea y la Escuela Politécnica Superior de Mondragon Unibertsitatea.



Este obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-nd/4.0/>

1. Introducción

La llegada de la denominada Industria 4.0 ha supuesto un gran aumento de los entornos digitales que habilitan la innovación en los servicios y procesos productivos (Lee, 2014). Los objetos de esta nueva industria inteligente están conectados y mediante sistemas ciberfísicos se recogen datos, se almacenan y se estudian para ofrecer soluciones mejoradas que incrementan la productividad y explotan al máximo el potencial de las empresas. En dichos sistemas digitales, la participación de los usuarios es fundamental, por lo que la comunicación entre la máquina y el usuario es clave para a la hora de emplear sus habilidades y conocimientos. (Gorecky, 2014).

Ante esta situación la interacción humano máquina (IHM) se ha convertido en un tema de gran interés en el ámbito de la industria. Al igual que en otros sectores económicos el desarrollo de los sistemas software de dichas interacciones ha ido evolucionando a lo largo de los años. Tradicionalmente, el desarrollo de la comunicación se ha basado en crear softwares centrados en la optimización de la funcionalidad, dejando a un lado el usuario (Brhel, 2015). Sin embargo, en esta nueva era las personas seguirán siendo muy protagonistas en el uso de las máquinas, de tal modo que la tecnología llegue a estar fundamentada en que los usuarios puedan desarrollar y aprovechar al máximo sus habilidades. Por lo tanto, el desarrollo de software debe centrarse primero en las personas, conocer sus objetivos, conocimientos y necesidades para así crear un sistema que se adapta al usuario, en lugar de que los usuarios se adapten al sistema.

Actualmente, varios autores mencionan la implementación de metodologías como el diseño centrado en el usuario (DCU) en el desarrollo de los sistemas software, pero carecen de procedimientos o definición de principios que sirvan para la implementación de la metodología en el ámbito industrial (Brhel, 2015). Con el objetivo de cubrir dicha carencia, se han definido los principios que formarán la metodología User Centered Agile Design (UCAD). Una metodología que propone un modo de trabajo para crear sistemas digitales que cumplen con las necesidades de los usuarios y los objetivos funcionales. Está basado en las bases de las metodologías Diseño Centrado en el Usuario (DCU) y el Desarrollo de Software Ágil (DSA).

En este artículo, se presentan los 8 principios que componen la metodología UCAD, con la intención de definir una guía básica para la creación de interfaces máquina-usuario intuitivos, amigables, usables y eficientes.

2. Antecedentes

En este apartado se muestra el desarrollo de los ocho principios de la metodología UCAD, el cual nace de la unión de dos metodologías, el Desarrollo de Software Ágil (DSA) y el Diseño Centrado en el Usuario (DCU). El primero de ellos, es un método centrado en el desarrollo de software, y actualmente es muy común en la ingeniería del software (Silva, 2012). Se basa en proveer un proceso dinámico caracterizado por ciclos iterativos y la participación puntual de agentes externos, de modo que, el Producto Mínimo Viable (PMV) se diseña lo antes posible y se va mejorando con las continuas evaluaciones (Highsmith, 2001). Partiendo de estas bases, en 2001 se creó el Agile Manifesto (Beck et al., 2001) y propone los siguientes 12 principios:

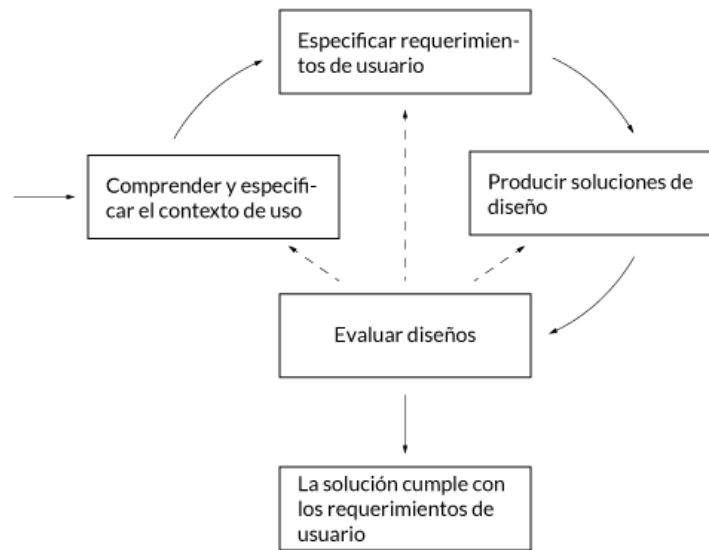
- Satisfacer al cliente mediante entregas tempranas y continuas de software de valor.
- Aceptar siempre los cambios de requerimientos, aunque se encuentre en la fase final de desarrollo.
- Realizar entregas de los avances en el software de forma frecuente y en plazos cortos.

- Trabajar conjuntamente los responsables del negocio y los desarrolladores.
- Construir los proyectos con personas motivadas, confiar en ellas y ofrecer un entorno adecuado para el trabajo.
- Realizar conversaciones cara a cara dentro del equipo para intercambiar opiniones e información.
- Emplear el avance del software como forma de medir el progreso.
- Promover desarrollos sostenibles y mantener siempre un ritmo de trabajo constante.
- Mantener una atención continua a la excelencia técnica y al buen diseño para mejorar en agilidad.
- Buscar la simplicidad, intentar minimizar las horas de trabajo.
- Organizar el equipo para construir las mejores arquitecturas, requerimientos y diseños.
- Reunirse regularmente para analizar cómo poder ser más efectivos y después adaptar el equipo a las mejoras propuestas.

Por lo tanto, como se ha podido observar en el Agile Manifesto, el DSA es una metodología centrada principalmente en las acciones relacionadas con la codificación y la gestión de proyecto. Sin embargo, no contempla los usuarios finales del producto desarrollado, sus necesidades, habilidades o motivaciones.

Actualmente, con el objetivo de aportar la perspectiva de los usuarios en los procesos de desarrollo de software, varios autores proponen la unión de la metodología DCU y el DSA (Blomkvist, 2005; Silva, 2012; Bertholdo, 2014; Brhel, 2015; Caballero, 2016). Los métodos ágiles buscan entregar al cliente el conjunto de funciones del software cuanto antes, mientras que el DCU propone dedicar más tiempo y esfuerzo en el análisis previo al desarrollo. El diseño centrado en las personas plantea la participación activa de los usuarios para comprender de mejor forma el usuario y sus necesidades y tareas, un diseño iterativo y de continuas evaluaciones y un equipo de trabajo multidisciplinar (Vredenburg, 2002). Por lo tanto, tal y como recoge la ISO 9241-210 (ISO, 2010), es un factor determinante para el desarrollo de sistemas interactivos (Fig. 1).

Fig. 1: ISO 9241-210 Diseño centrado en los usuarios



Con el objetivo de conocer cómo definen otros autores las bases teóricas que fundamentan la unión de las metodologías mencionadas, se ha realizado un estudio de la literatura actual. Se han analizado concretamente los principios definidos por cuatro autores (Blomkvist, 2005; Chamberlain, 2006; Silva, 2012; Brhel, 2015). En primer lugar, Blomkvist pone de manifiesto que un desarrollo de software ágil y centrado en el usuario debe cumplir con los siguientes principios: (i) la participación de los usuarios para conocer las necesidades de los usuarios, (ii) equipos multidisciplinares, (iii) representaciones simples en los diseños, (iv) desarrollo de sistema de forma evolutiva e iterativa y (v) el uso de los prototipos en la fase de desarrollo. En segundo lugar, Chamberlain define los siguientes 5 principios: (i) el usuario debe participar en el proceso de desarrollo, (ii) diseñadores y desarrolladores deben trabajar juntos y mantener una estrecha comunicación, (iii) los diseñadores deben transmitir sus ideas a los desarrolladores mediante prototipos y trasladar el feedback recibido por los usuarios, (iv) se debe ofrecer el tiempo necesario en la fase de análisis antes de la codificación para conocer las necesidades básicas de los usuarios, (v) la integración DSA/DCU debe realizarse con una gestión de proyecto cohesionado. En tercer lugar, tras realizar una revisión sistemática y con la intención de llevar a la práctica las propuestas teóricas de la unión de las metodologías, Silva propone las siguientes pautas: (i) trabajar el diseño antes de comenzar con el desarrollo, (ii) colaboración cercana entre diseñadores y desarrolladores, (iii) el uso de prototipos de baja fidelidad, (iv) continuas evaluaciones con los usuarios, (v) tener una visión holística del estado del proceso y (vi) el uso de herramientas de DCU como Personas, Historias de Usuario y Escenarios. Por último, Brhel tras su revisión de la literatura sobre diseño y desarrollo ágil de interfaces, plantea una serie de principios: (i) separar la búsqueda de producto y su desarrollo ofreciendo el tiempo necesario a cada uno, (ii) realizar un diseño y desarrollo iterativo e incremental, (iii) trabajar de forma paralela y entrelazada el diseño y desarrollo del producto, (iv) integrar en el proceso de forma activa a los diferentes stakeholders para que se identifiquen sus necesidades y recibir un feedback continuo y (v) emplear aproximaciones tangibles para comunicar las ideas y conceptos.

Tras el análisis del estado del arte actual sobre las metodologías DSA y DCU y su unión para el desarrollo de interfaces, se ha definido una serie de principios que a posteriori han sido el punto de partida para la creación de la metodología UCAD. En el siguiente apartado

se muestran dichos principios y se ha realizado una detallada explicación sobre cada una de ellas.

3. Resultado

La metodología UCAD se basa en los siguientes ocho principios:

1. Se diseña partiendo de las necesidades del usuario, sus motivaciones, aspiraciones y habilidades.
2. Integra al usuario durante todo el proceso, tanto para evaluar como para ser parte activa del diseño.
3. Equipo multidisciplinar. El equipo de trabajo está compuesto con personas de perfil de diseñador y programador. Donde trabajan conjuntamente desde principio a fin.
4. Combina el proceso analítico y sintético. Se intercalan continuamente fases de divergencia y convergencia, donde primero se crean diferentes vías y después se selecciona la más adecuada en base a los requisitos.
5. Trabaja con ciclos iterativos de diseñar, testear y evaluar. Tanto en las primeras fases con los bocetos a mano como con los diseños ya maquetados.
6. Tangibilizar las ideas cuanto antes, aunque sea con propuestas de baja fidelidad, para que se pueda testear y evaluar con los usuarios.
7. Toma en cuenta el contexto completo del sistema: las personas involucradas, el objetivo de la interacción, el tiempo y el lugar en el que ocurre la interacción, para ofrecer la información correcta, en el tiempo correcto, en el lugar correcto, en el modo correcto y a la persona correcta.
8. Busca que el producto sea comprensible, fácil de usar, amigable, claro, intuitivo, de fácil aprendizaje y que genere emociones positivas en el usuario.

3.1. Diseñar partiendo de los usuarios

Conocer el usuario siempre debe ser el punto de partida. Por un lado, es importante comprender cuál es el perfil del usuario, sus conocimientos, habilidades, necesidades y motivaciones en el uso del producto. Por otro lado, conocer las tareas que realiza y los hábitos ayuda a obtener una información clave para comprender cómo debe actuar el sistema. Gracias a este análisis se definen una serie de características que ayudan a crear un sistema adaptado al usuario, de manera que pueda emplear sus conocimientos y potenciales al máximo. En esta fase se aconseja utilizar herramientas como Personas o Escenarios.

3.2. Integrar al usuario durante todo el proceso

Tal y como se ha mencionado en el principio anterior, el usuario debe ser el punto de partida del proceso. No obstante, la participación del usuario no debe finalizar en esa fase, sino que debe continuar a lo largo de todo el proyecto. El diseño y desarrollo de las interfaces debe ser un proceso iterativo de continuas evaluaciones, de modo que la participación del usuario es muy determinante. Permite recibir el feedback de cada avance en el diseño y ayuda a mantener la perspectiva del usuario siempre en mente, así aumenta la probabilidad de realizar un diseño adaptado a las personas.

3.3. Equipo multidisciplinar

El equipo de trabajo debe estar formado por personas de perfiles tanto de diseñador como de desarrollador, para que participen conjuntamente en todas las fases del proceso, y así

permitir que las aptitudes de cada perfil se unan para crear un equipo mucho más completo, con notables beneficios:

- Tener opiniones de expertos de diferentes campos de conocimiento y trabajando conjuntamente ayuda a evitar pérdidas de tiempo en rediseños y así optimizar la duración del proyecto.
- Conseguir un diseño más creativo, gracias al aporte de diferentes puntos de vista que puedan ofrecer los diferentes perfiles.
- Poder ofrecer una experiencia más completa, porque el diseñador entiende lo que el desarrollador es capaz de crear.
- Aprender sobre conocimientos sobre diseño/desarrollo y viceversa.

3.4. Fases de divergencia y convergencia

Un proyecto basado en la metodología UCAD debe combinar procesos analíticos y sintéticos en todas sus etapas. Primero, en la fase de análisis, se intentaría reunir la mayor cantidad de información posible, para después filtrarlo y dirigirlo para crear una vía de trabajo. Después, en la fase de diseño, en una fase divergente se plantearían diferentes tipos de ideas, intentando crear la mayor cantidad posible y después converger en ideas concretas partiendo del feedback recibido por los usuarios. Por último, en la fase de desarrollo, plantearían diferentes modos de llevar a cabo lo planteado en pasos anteriores, y se sintetizaría en base a los requisitos establecidos.

3.5. Ciclos iterativos de diseñar, testear y evaluar

La metodología plantea que los avances realizados durante las fases sean contrastados con los usuarios, de modo que en lugar de crear soluciones definitivas y después testearlas, se aconseja que tras obtener un resultado o avance mínimo se contraste con los usuarios y seguir diseñando partiendo de esas evaluaciones. Esta filosofía de trabajo debe establecer tanto en fases iniciales de diseño como en la fase de desarrollo. Los testeos comienzan desde las propuestas de la nueva arquitectura de navegación, después con los diseños de las propuestas visuales y por último con los desarrollos codificados. Gracias a estos ciclos iterativos, se consigue evitar grandes rediseños, optimizar la duración del proyecto y conseguir un resultado adaptado a las necesidades de los usuarios.

3.6. Tangibilizar las ideas

En relación con el principio anterior, se recomienda tangibilizar las diferentes propuestas que vayan surgiendo, aunque sean de baja fidelidad. Este modo de trabajo proporciona una serie de ventajas que al final del proceso afectan notablemente en el resultado. Por un lado, el hecho de hacer tangibles los diferentes conceptos que vayan surgiendo asegura que esas ideas no queden en el aire o se vayan olvidando a medida que avanza el proyecto. Por otro lado, ayuda a comunicarse entre los compañeros de equipo, favorece la comprensión de los conceptos por personas de cualquier perfil, lo cual siendo un equipo multidisciplinar resulta primordial. Por último, las propuestas de baja fidelidad a menudo fomentan la participación de otras personas en la valoración de las mismas, puesto que siempre incita más proponer mejoras a versiones iniciales que a los finales.

3.7. Contexto completo del sistema

UCAD propone diseñar tomando en cuenta todo el contexto que rodea el sistema, a diferencia de los métodos tradicionales de diseño y desarrollo de software donde solo se centran en la propia interfaz. El contexto del sistema considera todos los elementos que toman parte en la interacción: el usuario, el tiempo, el lugar, las acciones a llevar a cabo y el

modo para realizarlo, es decir, la propia interfaz. Este modo de trabajo, aporta una visión holística que ayuda a comprender de forma integral los diferentes escenarios en los que se interactuará con el sistema, de forma que el nuevo producto sea capaz de adaptarse a todas ellas.

3.8. Usabilidad y experiencia de usuario

El objetivo principal de la metodología UCAD es conseguir un producto usable y que genere una experiencia de uso positiva. La experiencia de usuario engloba a todo los elementos que afectan en la interacción y se centra en las emociones y motivaciones de los usuarios, para conseguir por ejemplo, productos que generan emociones de tranquilidad y seguridad. Sin embargo, para conseguir cumplir con una experiencia positiva es necesario crear una interfaz usable. El término de la usabilidad está más enfocado a la funcionalidad de la aplicación, y se determina con criterios más cuantificables, como el tiempo o el número de acciones realizadas con éxito. Por lo tanto, mediante esta visión de trabajo se consigue un diseño útil, usable, intuitivo, comprensible, amigable y que habilita una experiencia de uso que genera emociones positivas y cumple con las motivaciones de los usuarios.

4. Conclusiones y líneas futuras

Mediante la definición de los principios de la metodología UCAD se define una base teórica para su posterior implementación en un proceso de diseño y desarrollo de software. Los principios definidos dictan unas pautas que concretan un modo de trabajo dirigido a los sistemas ciber-físicos de la industria 4.0, por lo que sirven de guía para proyectos reales de dicho contexto. De este modo, se cubre la carencia percibida en cuanto a definición de principios para el desarrollo de sistemas en la industria.

Gracias a los principios definidos y el método con las fases del proceso expuestos en la metodología UCAD, se pretende transferir el conocimiento a los proyectos de desarrollo IHM para el ámbito de la industria. Por lo tanto, se empleará el modo de trabajo planteado en casos reales, consiguiendo así IHMs que se adaptan a las necesidades de los usuarios, y no usuarios que se adaptan a los IHMs. De este modo se habilitarán sistemas ciber-físicos con una funcionalidad óptima pero adaptados al usuario, lo cual les permita aplicar sus conocimientos y explotar todo el potencial de esta nueva industria inteligente.

Los proyectos realizados servirán como casos de estudio para la posterior evolución de los principios y la metodología, los cuales irán transformando en base a los resultados obtenidos y las necesidades identificadas. Asimismo, UCAD deberá evolucionar junto con las nuevas tecnologías que surgirán en los próximos años, ya que, el desarrollo software debe estar predispuesto y preparado para la transformación.

5. Bibliografía

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Greening, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). The agile manifesto. Obtenido de www.agileAlliance.org
- Bertholdo, A. P. O., da Silva, T. S., Melo, C. D. O., Kon, F., & Silveira, M. S. (2014, June). Agile usability patterns for UCD early stages. In *International Conference of Design, User Experience, and Usability* (pp. 33-44). Springer International Publishing. doi: 10.1007/978-3-319-07668-3_4
- Blomkvist, S. (2005). Towards a model for bridging agile development and user-centered design. *Human-centered software engineering—integrating usability in the software development lifecycle*, 219-244. doi: 10.1007/1-4020-4113-6_12

- Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, 61, 163-181. doi: 10.1016/j.infsof.2015.01.004
- Caballero, L., Moreno, A. M., & Seffah, A. (2016). How Agile Developers Integrate User-Centered Design Into Their Processes: A Literature Review. *International Journal of Software Engineering and Knowledge Engineering*, 26(08), 1175-1201. doi: 10.1142/S0218194016500418
- Da Silva, T. S., Martin, A., Maurer, F., & Silveira, M. (2011, August). User-centered design and agile methods: a systematic review. In *Agile Conference (AGILE), 2011* (pp. 77-86). IEEE. doi: 10.1109/AGILE.2011.24
- Gorecky, D., Schmitt, M., Loskyll, M., & Zühlke, D. (2014, July). Human-machine-interaction in the industry 4.0 era. In *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on* (pp. 289-294). IEEE. doi: 10.1109/INDIN.2014.6945523
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127. doi: 10.1109/2.947100
- Lee, J., Kao, H. A., & Yang, S. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp*, 16, 3-8. doi: 10.1016/j.procir.2014.02.001
- Vredenburg, K., Mao, J. Y., Smith, P. W., & Carey, T. (2002, April). A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 471-478). ACM. doi: 10.1145/503376.503460