



Innovación y Software

ISSN: 2708-0927

ISSN: 2708-0935

facin.innosoft@ulasalle.edu.pe

Universidad La Salle

Perú

Cahui Osis, Victor Fernando; Quispe Soto, Darwin; Condori Huarca, Andres; Chapi Suyo, Jesús
Casos de Estudio de Design Thinking en las etapas de Análisis y Diseño del Desarrollo de Software
Innovación y Software, vol. 3, núm. 1, 2022, Marzo-Agosto, pp. 17-29
Universidad La Salle
Perú

Disponible en: <https://www.redalyc.org/articulo.oa?id=673870840002>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Tipo de artículo: Artículos originales
Temática: Ingeniería de software
Recibido: 28/09/2021 | Aceptado: 25/10/2021 | Publicado: 30/03/2022

Identificadores persistentes:
ARK: <ark:/42411/s8/a50>
PURL: 42411/s8/a50

Casos de Estudio de Design Thinking en las etapas de Análisis y Diseño del Desarrollo de Software

Case Studies of Design Thinking in the Analysis and Design stages of Software Development

Victor Fernando Cahui Osis¹[\[0000-0003-1587-7545\]](#)*, Darwin Quispe Soto², Andres Condori Huarca³, Jesús Chapi Suyo⁴

¹ Universidad Nacional de San Agustín. Arequipa, Perú. vcahui@unsa.edu.pe

² Universidad Nacional de San Agustín. Arequipa, Perú. dquispes@unsa.edu.pe

³ Universidad Nacional de San Agustín. Arequipa, Perú. acondorihuar@unsa.edu.pe

⁴ Universidad Nacional de San Agustín. Arequipa, Perú. jchapis@unsa.edu.pe

* Autor para correspondencia: vcahui@unsa.edu.pe

Resumen

La aplicación de Design Thinking en las primeras etapas del ciclo de vida del desarrollo del software (análisis y diseño), permiten obtener requisitos de manera óptima y diseñar arquitecturas que soluciones del problema del cliente. Como objetivos de este trabajo, se presentan casos de estudio en donde se aplicó Design Thinking en las etapas de análisis y diseño. Como resultados se obtuvo que las ideas generadas utilizando Design Thinking ofrecieron soluciones novedosas, pero algunas de ellas fueron inciertas. Además, se detalla cómo se desarrollaron los casos y cuáles fueron los beneficios de utilizar Design Thinking. Como resultado final, la aplicación de Design Thinking proporciona un buen punto de partida para el desarrollo de un software.

Palabras clave: Análisis, Design Thinking, Diseño, Ingeniería, Software.

Abstract

The application of Design Thinking in the early stages of the software development life cycle (analysis and design), allows us to obtain requirements optimally and design architectures that solve the client's problem. As objectives of this work, case studies are presented where design was applied with the analysis and design stages in mind. As a result, it was obtained that the ideas generated using Design Thinking offered novel solutions, but some of them were uncertain. It also details how the cases were developed and the benefits of using Design Thinking were experienced. As a final result, the Design Thinking application provides a good starting point for software development.

Keywords Analysis, Design Thinking, Design, Engineering, Software.

Introducción

Existen una gran variedad de investigaciones que han sugerido que Design Thinking (DT) puede tener un impacto positivo en el desarrollo de software, ya que facilita una profunda comprensión de las necesidades del usuario y aumenta la colaboración del equipo.

Design thinking es un concepto que promete una mayor innovación a través de un enfoque de innovación más centrado en el usuario [12]. Para mejorar la comprensión y la solución de problemas, así como la atención hacia el diseño, los investigadores sugieren la integración de Design Thinking con el proceso de desarrollo de software [12]. Según [13], usar ambos enfoques juntos tiene un sentido particular porque Design Thinking proporciona una ruta hacia soluciones creativas y centradas en el ser humano, mientras que los métodos ágiles del desarrollo de software optimizan el proceso y permiten que el equipo se mueva rápidamente.

Teniendo en cuenta este punto de vista, los equipos de software podrían beneficiarse de la aplicación de Design Thinking, a su proceso de desarrollo en cada etapa. Sin embargo, todavía no está claro cómo Design Thinking puede ser incorporado al desarrollo de software en una organización grande y cómo se manifiesta durante el proceso de desarrollo de software, que es algo que se explicara con casos de estudio.

Metodología computacional

Design Thinking

Design Thinking (DT) es una manera de ofrecer una solución a un problema. Descomponemos un problema, lo dividimos en partes más pequeñas, las analizamos, pensamos mucho, sin límites, todo lo que podamos y todo lo que se nos ocurra, de manera empática y junto a otros miembros del equipo, entonces estaremos mucho más cerca de encontrar la solución que buscamos.[8]

Design Thinking no solo es aplicados en la gestión de proyectos, sino es multidisciplinario, para la realización de este artículo nos centraremos en el área de las etapas de desarrollo de Ingeniería de Software.

Design Thinking en Ingeniería de Software

En tiempos recientes, Design Thinking se ha utilizado para mejorar el proceso de desarrollo de software y otras áreas relacionadas a la informática, como por ejemplo: enseñar ingeniería de sistemas en puntos críticos para la seguridad, diseño de aplicaciones móviles, desarrollo de juegos, etc[5]. Sin embargo, según los informes, se ha sugerido que el Design Thinking podría usarse también para el aprendizaje basado en proyectos, la obtención de requisitos y la ciberseguridad. Además, el método ágil podría combinarse con el pensamiento de diseño para mejorar el desarrollo de software [5][1].

Según [3], los investigadores han estado discutiendo cómo facilitar la integración de Design Thinking al desarrollo de software con el fin de integrar conocimientos más diversos para abordar los desafíos apremiantes durante el proceso. Según [4], Design Thinking puede soportar el desarrollo de software de cuatro maneras.

Tabla 1: Posibles modelos de integración de Design Thinking en el proceso de desarrollo de tecnologías de información.

Nombre	Descripción
Modelo de proyecto dividido	Design thinking como un proyecto separado del desarrollo
Modelo de términos superpuestos	Design thinking como fase inicial del proceso con uno o más miembros del equipo de desarrollo que participan
Modelo de proyecto indefinido	Design thinking como fase inicial del proceso y una gran superposición del equipo Design thinking y el equipo de desarrollo
Moldeo toolbox	Los desarrolladores de métodos pueden usar para superar problemas que no pueden resolver con métodos de TI

Etapas de Design Thinking

Design Thinking se define normalmente desde una perspectiva abstracta de alto nivel, pero vista desde un punto práctico, existen diversas estrategias y formas para implementarlo "de forma práctica"[2]. Las siguientes son las etapas generalmente aceptadas como marco para Design Thinking [5], aunque los pasos y la nomenclatura específicos pueden cambiar según la fuente:

- Inspiración:
 - Empatizar: comprender lo que el usuario o el cliente necesita, en términos de lo que siente el cliente.
 - Definir: Establecer las necesidades del cliente en términos de un problema que se puede resolver.
- Ideación: La propuesta de una solución (producto, servicio, experiencia) que satisfaga las necesidades del cliente.
- Implementación:
 - Prototipo: Crear una solución que cubra una selección de las características requeridas por el usuario, para que el cliente final pueda experimentarlo.
 - Prueba: Utilización real de la solución, de manera que el cliente pueda proporcionar comentarios oportunos y el proceso pueda iterar.

Tabla 2: Etapas de Design Thinking [5].

Design Thinking
Empathize: Market research, interviews, user persona, user journey
Define: Point-of-view statement, Analysis/synthesis, problem statement
Ideate: Brainstorming, point-of-view analysis, <i>How might we templates</i>
Prototype: Storyboard, sketching, non-functional prototyping, functional prototyping
Testing: Minimum viable product rollout, user feedback grid

Design Thinking propone que a través de la empatía y definición, la experimentación, la creación de prototipos y la retroalimentación inmediata, los diseñadores son más propensos a tener éxito al resolver una necesidad, simplificando y humanizando la solución [6].

Design Thinking, por lo tanto, puede ser lo suficientemente flexible como para ser tomado como un marco general de desarrollo de soluciones que lo hace adecuado para ser utilizado en los procesos de desarrollo de software, que de hecho es un campo en el que se requiere un enfoque abierto, creativo y flexible para resolver problemas y encontrar oportunidades de mejora [1].

Design Thinking aplicado al Análisis de Requerimientos

Caso de Estudio [6]

Proyecto Capstone de la Universidad Tecnológica de Lappeenranta, aplicado en el curso Human-Centric SE. El objetivo principal de este artículo es permitir a los estudiantes explorar problemas y estimar posibles soluciones, los estudiantes eligieron un escenario donde su solución se desarrollaría, luego elaboraron un inicial plan de proyecto [6], en total son 6 proyectos a los que se evalúan obteniendo como resultado una mejora en las habilidades blandas, una mejor experiencia de trabajar en un proyecto.

Caso de Estudio [9]

Este artículo de investigación nos plantea la creciente necesidad de digitalizar todo para adaptarse a las tecnologías de tablets, smartphone, etc. El análisis de requerimientos siempre varía entorno a las necesidades de los usuarios, por lo es necesario poder extraer estos requerimientos de manera más dinámica. El análisis de requerimientos también requiere de procedimientos o metodologías que se adaptan a la ingeniería de software, para esta investigación se utiliza los 4 principios de Design Thinking [8], que aporta para mejorar los procesos creativos, la comprensión, el trabajo colaborativo en equipo y la retroalimentación en el proyecto.

Design Thinking aplicado al Diseño

Design Thinking es una herramienta poderosa que a través del diseño específico permite a las empresas hacer que los clientes hagan lo que quieran [2].

Caso de Estudio [7]

¿Por qué los profesionales de la industria de Tecnologías de la información se esfuerzan tanto para que cada detalle en el diseño del producto sea perfecto y piensen en los colores de los botones, etc.?

Para los programadores el color de un botón no es importante, ya que vista desde el punto de vista de desarrollo, es cambiar el valor a una variable, pero en la práctica, este es un factor importante, ya que en cierta medida determina si los clientes utilizarán su producto.

A veces, los usuarios no pueden encontrar instantáneamente un botón que están buscando y luego entran en juego dos escenarios:

1. Molestan, cierran la ventana de la aplicación.
2. Pasan bastante tiempo tratando de encontrar el botón que conduce a la funcionalidad que necesitan.

Ambos casos son malos para el dueño del sistema. Y ambos significan que el problema del usuario no se tuvo en cuenta en la etapa de diseño.

Design Thinking en esta etapa permite a las empresas evitar problemas como este.

- Paso 1 Empatizar: Nos ponemos en el caso de que tu eres la persona que entrara a esta aplicación o página web.
- Paso 2 Definir: Cuando se tiene un conocimiento profundo de cuál es el problema del usuario.
- Paso 3 Idear: Podemos sugerir algunas soluciones a este problema.
- Paso 4 Prototipar: Elegir y crear el mejor diseño posible para todos los elementos del producto.
- Paso 5 Probar: Mostramos a los usuarios el diseño de las interfaces. Como resultado, los usuarios finales no se confunden cuando utilizan su producto y eventualmente se convierten en sus clientes leales.

Los mismos colores de botón pueden alentar a los usuarios a tomar una acción que usted necesita para, por ejemplo, programar un seminario web o solicitar la versión de prueba de su producto.

O simplemente mire las notificaciones prominentes del navegador: cuando, por ejemplo, un sitio web de medios que está visitando le ofrece notificaciones instantáneas de noticias importantes.



Figura 1.

El primero es el botón Sí (permitir), al hacer clic en el que acepta recibir notificaciones push. Otro es el botón No, gracias (bloquear).

Estamos acostumbrados al hecho de que el color verde generalmente implica consentimiento, mientras que su color complementario, rojo, significa la disidencia. Como las señales de tráfico rojas dicen que algo está prohibido, un botón rojo dice que al hacer clic no estamos de acuerdo en recibir notificaciones automáticas del sitio web. El truco es que algunos medios ponen la etiqueta Sí en el botón rojo. También tenga en cuenta el orden: sí, generalmente va primero, pero no en este caso, haciendo un prototipo:



Figura 2.

Por lo tanto, nosotros, que no queremos empeñarnos en spam, intuitivamente hacemos clic en el botón rojo para rechazar la oferta de recibir notificaciones, pero, en cambio, comenzamos a recibirlas. En la parte de desarrollo de software de Diseño, el color no es el único parámetro que importa. El tamaño, la forma, la ubicación, el orden, las etiquetas de los botones y más son cosas a tener en cuenta, los botones son solo la punta del iceberg

Resultados y discusión

En [5] se tomó a un grupo de 17 estudiantes de tercer año provenientes de tres países diferentes, con especialización en Ciencias de la Computación. El grupo se organizó en cinco equipos de tres miembros y dos equipos formados por cuatro estudiantes. Luego del desarrollo de las actividades correspondientes al Design thinking se obtuvo lo siguiente:

- Empatizar: Los equipos encontraron una necesidad que pueda satisfacerse mediante un sistema de software. Pero con la condición que el equipo tiene que consultar a sus amigos y descubrir cuáles son sus necesidades. Luego de la identificación de la necesidad, se le pidió que identifiquen quién es el usuario que padece esa necesidad.
- Definir: una vez que se identifica la necesidad y el grupo objetivo, los estudiantes exploraron más a través de entrevistas para delinejar una mejor definición de la situación que quieren resolver. Las metas y los objetivos generalmente se escriben en forma de oraciones inspiradas en historias de usuarios. Una vez concluida esta etapa los estudiantes ya tenían claro cuál es la solución.

- Idear: Los equipos proponen soluciones, verifican la viabilidad, comparan tecnologías y presentan una idea para desarrollar. En esta fase, se observó que después de ejercitarse la definición del problema y las soluciones de lluvia de ideas, un equipo podría transformar la idea original en otra diferente.
- Prototipo: Para la elaboración real de una solución de trabajo, los equipos utilizaron la tecnología de su elección, lo que conduce a una variedad de lenguajes de programación y una pila de tecnología muy diversa.
- Prueba: Etapa en la cual los estudiantes mostraron todos sus productos.

Como resultado se obtuvo lo siguiente:

- El equipo que utilizó Design Thinking utilizó una metodología de creatividad e innovación para proporcionar una solución de software. Además, los proyectos resultantes del equipo de Design Thinking resultaron ser más innovadores, pero más inciertos.

En [6] nos demuestra cuáles fueron los resultados de aplicar Design thinking en los cursos Capstone, desde la perspectiva de los alumnos se describe lo siguiente:

- Personas: descubrieron que la creación de clientes en las etapas de análisis y diseño inspira a los estudiantes de ingeniería a ver los problemas desde la perspectiva de un grupo de usuarios, en lugar de desde una perspectiva tecnológica y propia.
- Storyboard: descubrieron que expresar ideas con comportamientos y acciones reales de los usuarios a través de storyboards, ayuda a los estudiantes a comprender mejor quién, qué, cuándo, dónde y cómo se usa la aplicación e identificar las implicaciones sociales y éticas.
- Mapa de usuario: descubrieron que el mapa del usuario ayudó a los estudiantes a mejorar y mejorar su pensamiento de diseño al ayudarlos a comprender las emociones de los usuarios de la aplicación en las primeras etapas. En la sesión de tutoría, los estudiantes señalaron que a través del mapa de viaje del usuario lograron comprender los sentimientos más profundos de los usuarios de la aplicación, que de lo contrario podrían haber sido difíciles.
- Creación de prototipos: descubrieron que los estudiantes encontraron que el diseño del prototipo era muy útil para mejorar su proceso de diseño. Por ejemplo, uno de los equipos señaló durante la sesión de tutoría que recibir comentarios de los usuarios durante el proceso del prototipo fue útil para delinear la interfaz de usuario.

En [6] desde la perspectiva de los profesores resaltaron las siguientes lecciones:

- El diseño del curso fue muy efectivo para motivar a los estudiantes a participar, los propios estudiantes eran los encargados de sus proyectos. Al utilizar Design Thinking en las etapas de diseño aumentó la motivación de los estudiantes al percibir que lo que diseñan es lo que realmente quiere el usuario.

En [3] los autores aplicaron una serie de preguntas a una empresa de desarrollo de software dividida en tres secciones. La primera sección tenía como objetivo recopilar información demográfica sobre los participantes.

La segunda sección cubrió la experiencia general del entrevistado con DT. La tercera sección tenía como objetivo recopilar datos sobre el valor de DT para el proceso de desarrollo de software y su implementación.

Tabla 3: Estructura de preguntas realizadas a los trabajadores de una empresa de software sobre DT [3]

Part I – Demographics
Age, name
Purpose and task of the team
Role and background of the interviewee
Part II – Experience with DT
What's your experience with Design Thinking?
What is your view point on Design Thinking?
Part III – DT in software development projects
How does Design Thinking support you and your team?
How is Design Thinking implemented?
What activities are you involved in?
How is your involvement with the user?

En [3] los autores aplicaron el cuestionario mencionado anteriormente y como resultado del análisis de las respuestas obtenidas identificaron tres temas principales y ocho categorías que describen las manifestaciones del Design Thinking dentro de la empresa de software. La Tabla de continuación muestra los temas y categorías relacionadas.

Tabla 4: Temas y categorías de la manifestación del Design Thinking derivado del análisis temático en [3]

Understanding	Implementation	Connection to Agile
Interpretation	DT enablers	Agile activities
Experience	DT barries	
Perceived values and benefits	Goals	Agile values and principles
	formats	

La comprensión del tema del Design Thinking describe cómo el Design Thinking se manifiesta dentro de las mentes de los empleados. Este tema está cubierto por las categorías interpretación, experiencia, valores y beneficios percibidos. La interpretación de la categoría explica cómo los empleados describen y ven el Design Thinking. La categoría de experiencia cubre lo que los empleados saben sobre el proceso y las herramientas y cuánto han usado y facilitado el pensamiento de diseño. Finalmente, la categoría de valores y beneficios percibidos describe por qué los empleados usarían el Design Thinking.

El tema de conexión a ágil describe cómo el Design Thinking apoya el desarrollo ágil de software. Este tema está cubierto por las categorías de actividades ágiles y valores y principios ágiles. La categoría de actividades ágiles describe qué actividades concretas en el ciclo de desarrollo ágil pueden ser apoyadas por Design Thinking. La categoría valores y principios ágiles describe qué valores y principios ágiles son compatibles con el uso del pensamiento de diseño.

Este estudio tuvo como objetivo comprender cómo DT puede apoyar el desarrollo de software en una empresa de TI y cómo se manifiesta durante el proceso de desarrollo. El primer hallazgo que surgió fue la correlación entre la comprensión del DT y los roles profesionales. Por ejemplo, la interpretación de DT como un proceso o caja de herramientas fue dominante entre los desarrolladores. Ejecutar fases DT más largas no tenía sentido para ellos ni contaban con el conocimiento necesario para hacerlo. Incluso mencionaron la situación en la que DT y su mentalidad no serían de ninguna ayuda, por ejemplo, al solucionar un bug.

La interpretación de DT como una mentalidad fue dominante entre los diseñadores y gerentes. Curiosamente, varios entrevistados se mostraron reacios a la idea de considerar DT solo como un proceso, en lugar de una mentalidad. Sin embargo, no se encontró ningún estudio que afecte negativamente al uso de DT solo como herramienta o proceso. Por

el contrario, las investigaciones existentes junto con los hallazgos de este trabajo sugieren que el nivel de aplicación de DT depende del objetivo del proyecto y la experiencia del equipo [10], [11].

Conclusiones

- Los métodos de Design Thinking son métodos prácticos para involucrar al equipo de desarrollo de software y mejorar su experiencia y habilidades blandas (por ejemplo, el trabajo en equipo) importantes en la industria.
- Cada vez que esté planeando un nuevo producto o un nuevo diseño, aparte de Design Thinking, podemos usar las pruebas A / B, que también se llaman pruebas divididas. Lo mismo se aplica al desarrollo de nuevas características.
- Design thinking es un enfoque para crear productos de software personalizados. Reúne los objetivos comerciales de los clientes y las necesidades del usuario final y los integra en una solución perfectamente adaptada que aborda el problema de una manera efectiva.
- Design thinking mejora el proceso de entendimiento por parte de los desarrolladores hacia los usuarios, esto es porque se busca empatizar con el usuario final.
- En la Universidad Tecnológica de Lappeenranta se aplica Design Thinking a seis proyectos distintos, y se pudo obtener resultados positivos, los que más resaltan es la mejor productividad y la disminución de abandonos por parte de los estudiantes.
- Los proyectos realizados utilizando Design Thinking pueden resultar muy innovadores pero pueden ser inciertos.
- El uso de Storyboard y mapas de usuario ayudan considerablemente a entender que es lo que necesita el usuario final.
- La aplicación Design Thinking en una empresa de TI da a conocer la correlación entre la comprensión del DT y los roles profesionales. Es decir, no es lo mismo Design Thinking para un gerente que para un desarrollador, cada uno lo ve y lo aplica desde su perspectiva.
- No se puede generalizar o estandarizar Design Thinking, ya que este es adaptable al entorno en donde se vaya a aplicar.

Referencias

- [1] K. Thoring, M. Mülle, Understanding Design Thinking: a Process Model Based on Method Engineering. International Conference On Engineering And Product Design Education 8 & 9 September 2011, City University, London, UK
- [2] J. Denning, The Profession of IT Design Thinking, Communications of the ACM, December 2013
- [3] F. Dobrigkeit, D. de Paula Design, Thinking in Practice: Understanding Manifestations of Design Thinking in Software Engineering, Hasso Plattner Institute Potsdam, Germany. ESEC/FSE '19, Tallinn, Estonia, August 2019
- [4] T. Lindberg, E. Köppen, I. Rauth, C. Meinel, On the perception, adoption and implementation of Design Thinking in the IT industry.
- [5] L. Corral, I. Fronza, Design Thinking and Agile Practices for Software Engineering: An Opportunity for Innovation.
- [6] M. Palacin-Silva, J. Khakurel, Infusing Design Thinking into a Software Engineering Capstone Course. 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)
- [7] J Sumato, What Design Thinking Is and How It Is Used in Software Development. Recuperado 23 octubre, 2019, de <https://medium.com/@sumatosoft/what-design-thinking-is-and-how-it-is-used-in-software-development-37cf8c581675>
- [8] T. Brown, Design thinking, Harvard business review, 2008, vol. 86, no 6, pp. 84.
- [9] A. Carrel, A. Lauenroth, Using Design Thinking for requirements engineering in the context of digitalization and digital transformation: a motivation and an experience report. The Essence of Software Engineering. Springer, Cham, 2018. pp. 107-120.
- [10] Franziska Dobrigkeit, Danielly de Paula, and Matthias Uflacker. 2018. InnoDev A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup. In Design Thinking - Research Looking Further:

Design Thinking Beyond Solution-Fixation, Hasso Plattner, Christoph Meinel, and Larry Leifer (Eds.). Springer, 199–228.

[11] H. Ximenes, I. Alves, and C. Araújo. 2015. Software Project Management Combining Agile, Lean Startup and Design Thinking. In Design, User Experience, and Usability: Design Discourse, Aaron Marcus (Ed.). Vol. 9186. Springer International Publishing, Cham, 356–367. http://link.springer.com/10.1007/978-3-319-20886-2_34

[12] T. Brown. 2008. Design thinking. Harvard Business Review 86, 6 (2008), 84. <http://www.unusalleading.com/wp-content/uploads/2009/12/HBR-on- Design-Thinking>

[13] W. Brenner, F. Uebenickel, T. Abrell.. Design Thinking as Mindset, Process, and Toolbox. In Design Thinking for Innovation, Walter Brenner and Falk Uebenickel (Eds.). Springer International Publishing, Cham, 3–21, 2016