

Tipo de artículo: Artículos originales
Temática: Ingeniería de software
Recibido: 30/11/2021 | Aceptado: 05/01/2022 | Publicado: 30/03/2022

Identificadores persistentes:
ARK: [ark:/42411/s8/a54](https://nbn-resolving.org/urn:nbn:org:ark:42411/s8/a54)
PURL: [42411/s8/a54](https://nbn-resolving.org/urn:nbn:org:ark:42411/s8/a54)

Design Thinking para resolver problemas con la selección de métricas relacionadas a la Calidad del software

Design Thinking to Solve Problems with the Selection of Metrics in Software Quality

Eduardo Armando Sutta Gonzales ^{1*}, Christian Larry Salazar Bobadilla ², Julio Cesar Aguilar Uñapilco ³, David Huittocollo Quispe ⁴

¹ Universidad Nacional de San Agustín. esuttag@unsa.edu.pe

² Universidad Nacional de San Agustín. csalazarb@unsa.edu.pe

³ Universidad Nacional de San Agustín. jaguilaru@unsa.edu.pe

⁴ Universidad Nacional de San Agustín. dhuitocollo@unsa.edu.pe

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

Resumen

En el presente artículo se va a utilizar la técnica Design Thinking para solucionar y evaluar el problema de la selección de métricas de calidad de un software, considerando algunas técnicas para la recopilación de las ideas que puedan dar soporte a la elaboración de una propuesta de solución al problema que en este caso es el aseguramiento de la calidad basadas en las métricas de evaluación definidas en los estándares actuales conocidos. Todo esto es posible gracias al apoyo de herramientas web que proporcionan plantillas para abarcar las fases del Design Thinking. También se hace el detalle de las fases empezando por la fase de Empatizar seguido de Definir, Idear, Prototipar, Testear y finalmente llegando a los resultados y las conclusiones encontradas luego de todo el proceso del Design Thinking. Se espera que los resultados ayuden a la construcción de software de calidad y sean utilizados en un proyecto real y en sus diferentes etapas haciendo que el software tenga éxito en el mercado.

Palabras clave: Design Thinking, Métricas, Calidad, Software, Herramientas, Estándares.

Abstract

In this article, the Design Thinking technique will be used to solve and evaluate the problem of the selection of software quality metrics, considering some techniques for the collection of ideas that can support the development of a solution purpose to the problem that in this case is the quality assurance based on the evaluation metrics defined in the current known standards. All this is possible thanks to the support of web tools that provide templates to cover the phases of Design Thinking. The details of the phases are also made starting with the Empathizing phase followed by Define, Devise, Prototype, Test and finally arriving at the results and conclusions found after the whole Design Thinking

process. The results are expected to help build quality software and be used in a real project and in its different stages making the software successful in the market.

Keywords: *Design Thinking, Metrics, Quality, Software, Tools, Standards.*

Introducción

Nuestra meta es buscar los diferentes conflictos sobre la calidad de software para lo cual se va a evaluar y dar solución a la generación de problemas cuando se elige las métricas de calidad de Software para ello se recopila información acerca de cómo se eligen las métricas de calidad para el desarrollo de software. El escoger una métrica de calidad resulta complicada debido al presupuesto reducido, la documentación puede extenderse demasiado y no se hace seguimiento debido del proceso de desarrollo del software. Es por ello que usar el Design Thinking nos permite identificar problemas que pueden ser resueltos por medio del uso de herramientas modernas, estándares alcanzables en la industria del desarrollo de software.

Materiales y métodos o Metodología computacional

El Design Thinking es una metodología que permite construir ideas en base a la función y a emociones desde la perspectiva del usuario final. Se generan soluciones que se basan en la idea de ponerse en el lugar del usuario final o sea para quien va dirigido el producto o servicio.

Se escogió esta metodología de trabajo porque permite ver las situaciones de diferentes partes del proceso de la calidad de software en donde participan diferentes equipos de trabajo cuyos integrantes desempeñan diferentes roles y así se pueden generar tantas ideas como sea posible.

Empatizar

Para la primera fase del Design Thinking se emplearon las siguientes técnicas: a) MoodBoard, b) Observación Encubierta, c) Que Como y Porque y d) Mapa de Actores. Estas técnicas permitieron identificar problemas desde la perspectiva de un encargado de la calidad en el desarrollo de software puesto que son ellos quienes seleccionan las métricas que mejor convengan al proyecto y que deben ser evaluadas para medir el aseguramiento de la calidad.

A. MoodBoard

Un MoodBoard permite plasmar las ideas de un problema o situación de manera gráfica y es puede ser utilizada como apoyo para distinguir los tópicos de los problemas de calidad en nuestro caso.



Figura 1. Mapa mental sobre Design Thinking.

B. Observación encubierta

Permite mezclarse con el entorno y captar y evaluar la información obtenida. En el entorno de desarrollo de software la elección de métricas muchas veces no se da por cuestiones de tiempo, o requerimientos inconclusos, en las empresas pequeñas es más difícil ver la rama de ingeniería de software porque no se cuenta con el

personal ni el presupuesto adecuado para asegurar la calidad, dando más prioridad al análisis y al desarrollo desordenado.

C. ¿Qué?, ¿Cómo?, ¿Por qué?

Tabla 1.¿ Resumen del Qué Cómo Por Qué?

¿Qué está Pasando?	¿Como está Pasando?	¿Por qué está pasando?
Las métricas no se gestionan correctamente.	El personal encargado no está usando formatos correctos para evaluar la calidad.	Por que no se entendió correctamente la métrica para elaborar formatos de evaluación de calidad.
No se usan herramientas modernas de gestión de métricas de calidad.	Se hace uso excesivo de documentación que se incrementa y no se maneja controla correctamente.	Las herramientas que existen para manejar documentación son caras.
Se eligen las mismas métricas de calidad para varios proyectos de desarrollo.	En los proyectos de desarrollo se utilizan métricas que se han usado en otros proyectos y se evalúan de la misma manera como lo hicieron anteriormente.	Las métricas de calidad se pueden desglosar en diferentes características que complican entender completamente la métrica de esta forma el personal no logra aplicar nuevas métricas por que creen que ya es suficiente con las que manejan.

D. Mapa de Actores

Listado de actores

Personas:

- Jefe de Proyecto.
- Programador.
- Diseñador.
- Usuario.
- Analista.
- Probador(tester).

Organizaciones:

- Empresa de software.
- Empresa distribuidora.

- Empresa usuario.
- Instituciones
- Gobierno

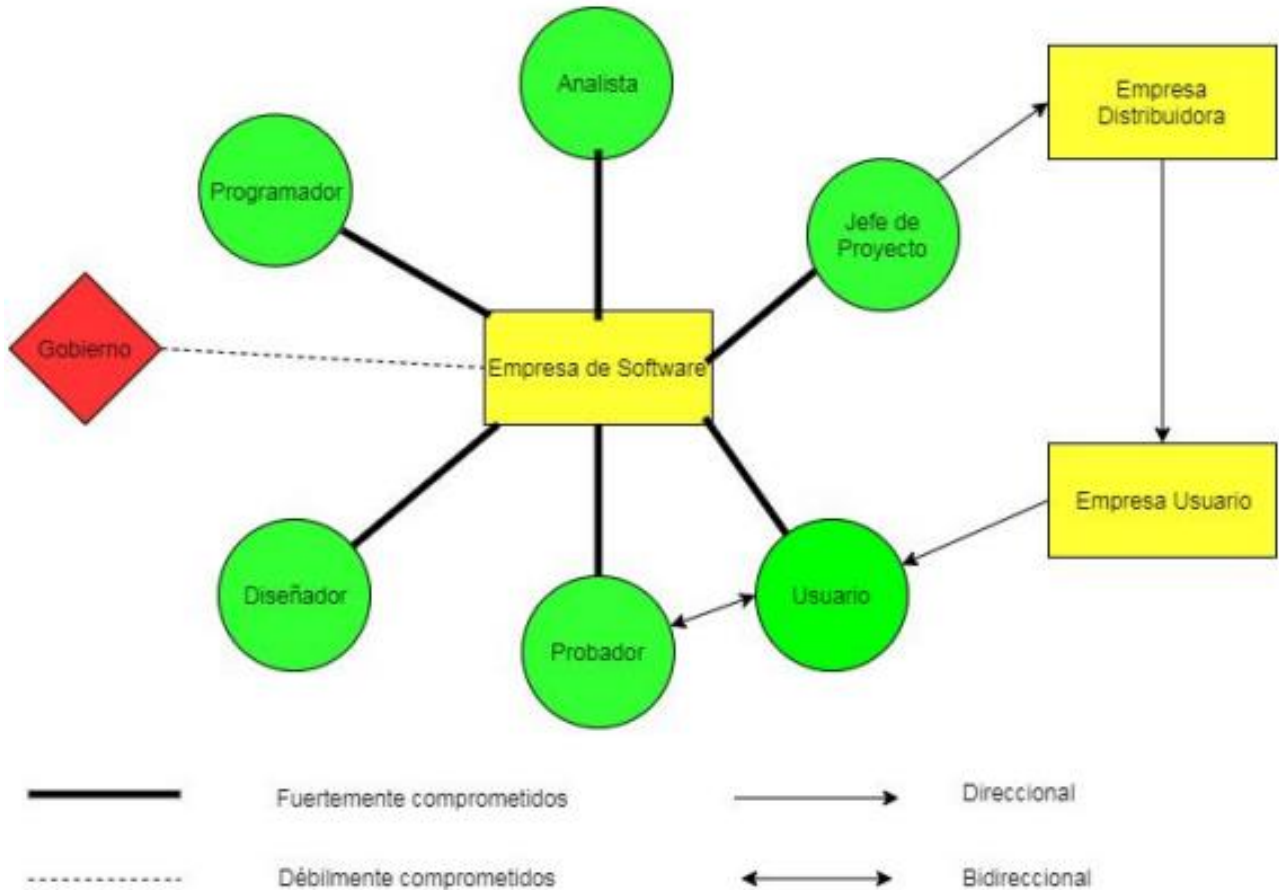


Figura 2. Mapa de actores

Definir

Para esta fase se usaron las siguientes técnicas: a) Un mapa mental, b) Point Views y c) Blue Print. El objetivo de esta fase es identificar las posibles mejoras que podemos proponer en la selección de métricas de la calidad partiendo de situaciones que no se desarrollan bien sin embargo estas deben ser alcanzables.

A. Mapa Mental

Realizar mapas mentales permite poder ordenar las ideas encontradas acerca del problema a resolver de la mejor manera entendible y poder interpretar conceptos.

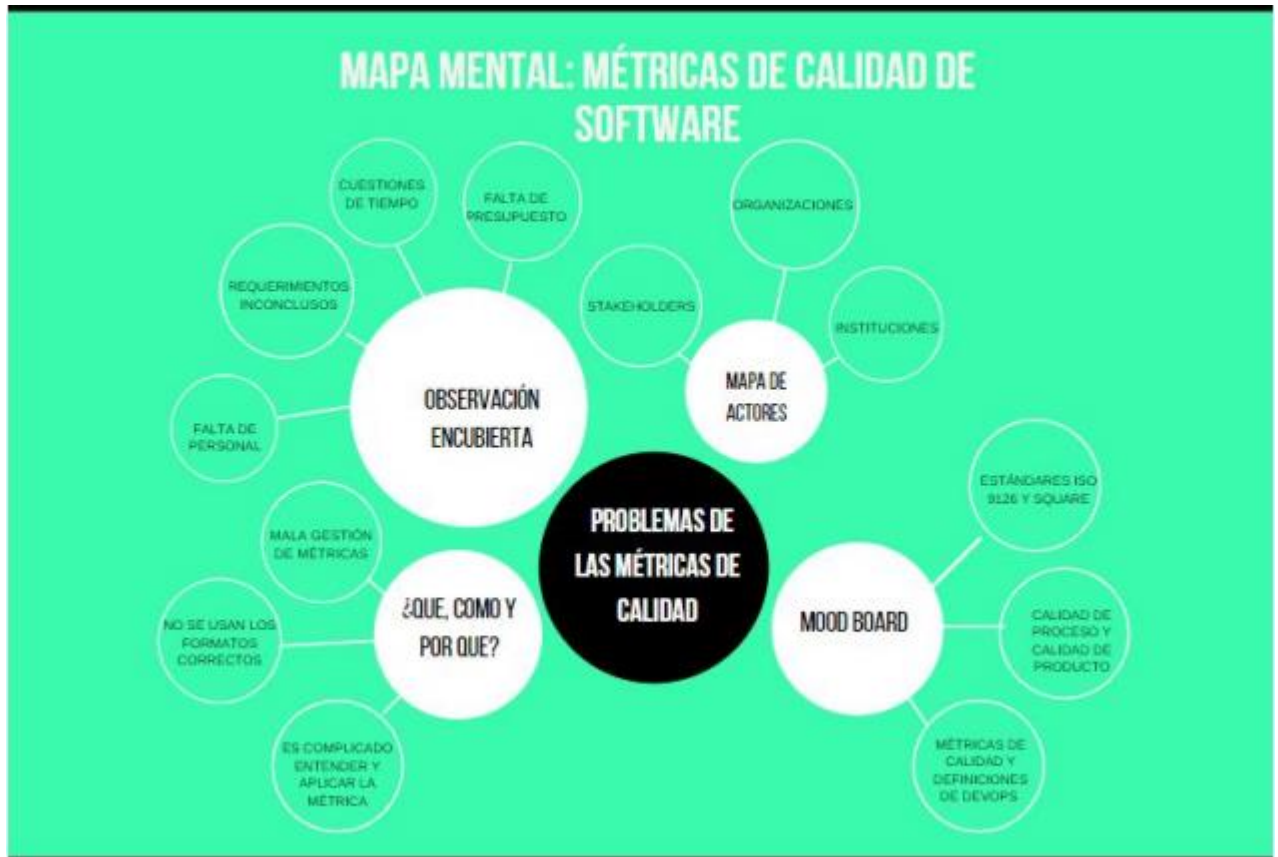


Figura 3. Mapa mental

B. Point Views

Mediante la técnica de Puntos de vista es uno de los pilares para construir ideas, para crear se usa la palabra necesita y por qué.

Se identificaron los siguientes points views.

- La empresa necesita asegurar la calidad de software porque le da más valor al producto.
- La empresa necesita usar métricas de software porque le da más valor al producto.

- La empresa necesita personal adecuado en calidad de software porque aumentará la calidad del software.
- Las empresas necesitan un mejor proceso de desarrollo porque eso aumentaría su competitividad.
- El analista necesita que se gestione correctamente las métricas de software porque tiene que elaborar mejores formatos para evaluar la calidad de software.
- El analista necesita mejores formatos de software porque la documentación creciente es difícil de manejar.
- El analista necesita mejores herramientas porque eso facilita su trabajo.
- La empresa necesita gestionar elegir las métricas correctas porque eso aumentará la calidad del proceso de software.
- Las empresas usuario necesita un correcto proceso de desarrollo del software porque necesita asegurar que le entrega un buen producto.

C. Blue Print

El blueprint que se presenta a continuación identifica procesos que se dan en el desarrollo de software que están directamente relacionados con las métricas de la calidad.

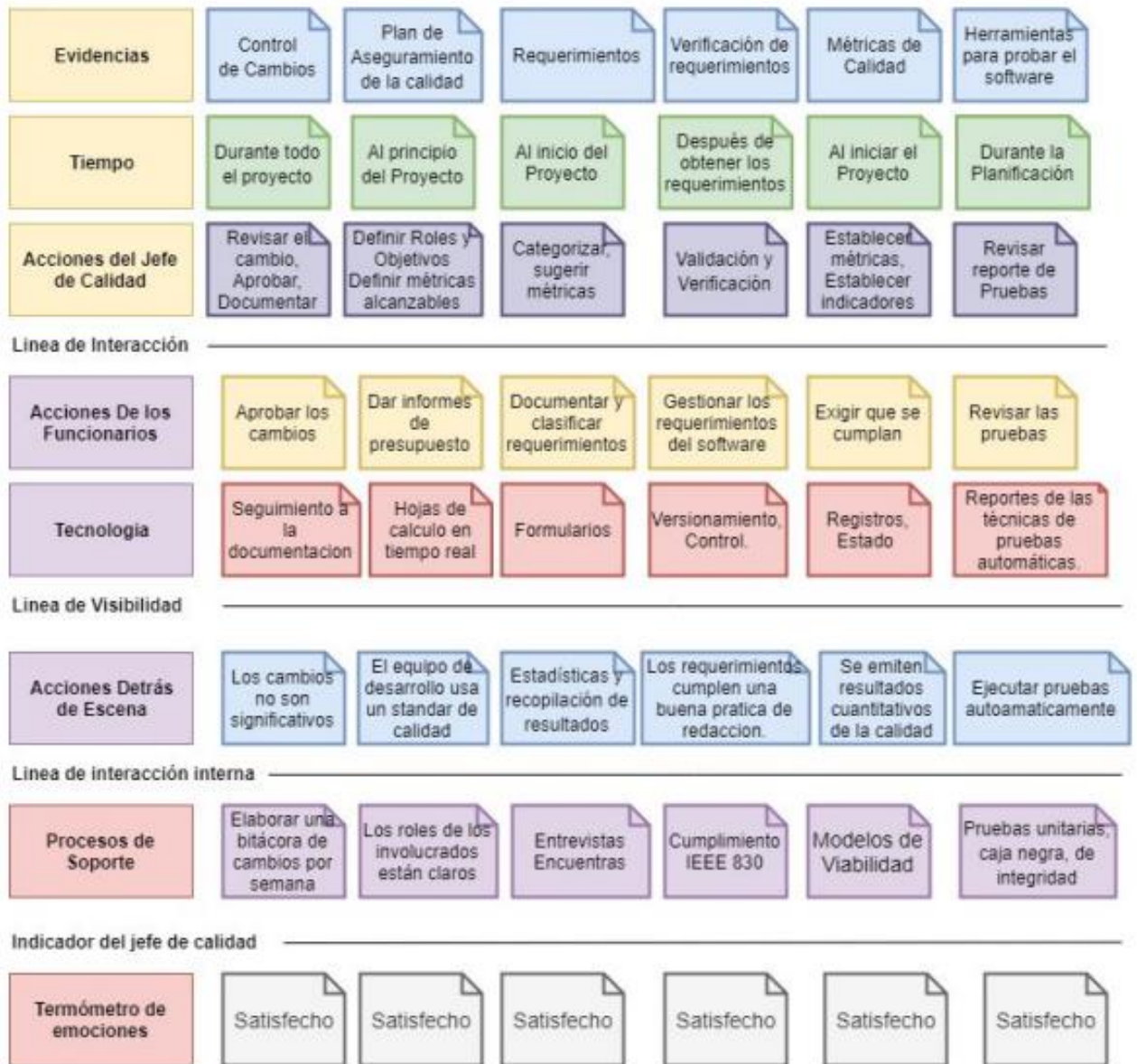


Figura 4. Diseño del Blue Print.

El blue print determina que la documentación de un proyecto se extiende en el tiempo y hace falta encontrar alguna herramienta o definir procesos de revisión y seguimiento.

También que se requiere utilizar un estándar para la mejorar la calidad de los requerimientos puesto que deben de ser entendidos por todos los involucrados y en esa situación es necesario mantener buenas prácticas de documentación de requerimientos para que se mantenga la calidad en el proceso. También se ve la necesidad de manejar un control sobre el resultado de las pruebas además sería de gran utilidad tener automatizada la realización de las pruebas para evitar una sobrecarga de trabajo. Esto tiene incidencia en la calidad ya que los resultados de las pruebas se pueden utilizar como entradas para las métricas de la calidad del software.

Idear

Permite definir los conceptos esenciales a tratar ideas simples de entender y que son los elementos conceptuales que surgen del grupo encargado de resolver el problema en base a lo aprendido o saberes previos.

A. Lluvia de Ideas



Figura 5. BrainStorm.

Con esta técnica de Se puede desprender que se deberían tomar las siguientes acciones para resolver el problema de las métricas de calidad:

1. Manejar un estándar para recoger requerimientos.
2. Revisar que los requerimientos estén completos.
3. Capacitaciones a los involucrados en la calidad.
4. Manejar formatos definidos por el usuario al inicio del proyecto.
5. Incluir que se busca conseguir usando la métrica en el proyecto.
6. Construir una herramienta para gestión de métricas.
7. Usar un estándar de calidad alcanzable.
8. Manejar un seguimiento a la documentación de forma automática.
9. Definir los roles al inicio y de forma clara.
10. Definir qué pruebas se van a realizar.
11. Realizar reportes de pruebas.
12. Gestionar un buen grupo personal para afrontar los proyectos futuros.
13. Hacer un estudio para ver la profundidad del problema y envase a eso tomar las medidas preventivas.
14. Crear reglas y política de comunicación realistas.
15. Apoyo en las tecnologías para para automatizar y disminuir el tiempo de estos procesos.
16. Apoyo en las tecnologías para buscar métricas realistas.
17. Buscar normas que se ajusten al problema.
18. Planear un buen recurso del tiempo y gestión de este.
19. Monitorear constantemente si las métricas son las correctas.
20. Tener una bitácora de los errores para posteriormente corregirlos.
21. Fundamentar las métricas usadas en el software.
22. Informar constantemente a los miembros del equipo sobre el funcionamiento correcto de las métricas.

Prototipar

Nosotros proponemos que se use una herramienta que haga seguimiento al cumplimiento de las métricas de calidad permitiendo al encargado de la calidad tener un control de toda la documentación necesaria para evitar tener que realizar búsquedas y correcciones que no se necesita hacer.

Asimismo, se podrán gestionar diferentes métricas de calidad de estándares de calidad que sean alcanzables en el tiempo así las métricas pueden ser comprensibles por todo el equipo de trabajo encargado de la calidad del software sin que en los proyectos se usen los mismos estándares de calidad. IEEE 830 , ISO 9001, ISO 9126.

Definir roles y responsabilidades desde el inicio del proyecto en cuanto a la calidad por lo que se deberían manejar perfiles de trabajo para el aseguramiento de la calidad del producto donde se describa sus funciones y las comunicaciones que debe manejar para cumplir con sus labores. Por Ejemplo: Perfil del responsable de la calidad:

- Realiza tareas de trabajo con procedimientos e instrucciones de trabajo relativas a los requisitos del estándar seleccionado para el proyecto y de seguridad.
- Actúa como enlace entre los analistas y programadores para asegurar la calidad del producto.
- Selecciona las métricas de calidad para un proyecto, hace seguimiento a la documentación elabora los formatos de trabajo y entrega de resultados de calidad del software.
- Gestiona todos los procesos de calidad designa responsabilidades y revisa el cumplimiento de las métricas de calidad.
- Selecciona herramientas de seguimiento de documentación y la emisión de los informes de estado correspondientes.
- Realiza seguimiento a la ejecución de las pruebas y es encargado de aprobar los reportes correspondientes.

Definir medios de comunicación en la nube para el desarrollo del proyecto de forma que se puedan reducir costos así mismo se podrían establecer formatos que puedan ser usados durante todas las fases del desarrollo y que puedan ser reusadas y modificadas en otros proyectos evitando así hacer uso excesivo de papel así mismo automatizar el proceso de realización de pruebas y la emisión de los reportes correspondientes.

Usar Herramientas modernas de seguimiento de documentación (Gratis o Pago) que se encuentran en la nube. Realizar las pruebas durante todo el ciclo de desarrollo con la inspección del encargado de calidad.

Testear

Se observó que la herramienta funciona correctamente pero el escoger la herramienta genera dificultad, la instalación y la integración al proyecto en cuestión demora tiempo y al optar por herramientas de pago estas cuestan y en algunos

casos piden mensualidad. Se observó que los pasos propuestos funcionan como guía para la realización de un buen aseguramiento de calidad.

Resultados y discusión

El prototipo creado servirá para la mejora de la calidad en empresa, o grupos de desarrollo de software. Las pautas anteriores a vista de un desarrollador son las mejores elecciones para mejorar la calidad de software. viendo que soluciona parcialmente la déficit de calidad y elección de métricas del software.

También en el prototipo de tendría dificultad a la hora de encontrar una herramienta o grupo de herramientas para gestionar la calidad al existir muchas en el mercado y alguna gratuidad y de pago. siendo en algunos casos complicado la instalación y en otros el coste.

El uso de la técnica Design Thinking te da normas más claras a la hora de generar un buen protocolo de aseguramiento a la hora de elegir métricas de calidad. Al ser basado en prototipos se minimiza el costo y riesgo del fracaso del prototipo.

La herramienta llamada mapa de Actores de la fase de empatizar nos da una visión general de cómo es que actúan y se relacionan los actores en un proceso normal de aplicación de calidad de software en una empresa de desarrollo, nos ayuda mucho para detectar problemas que pueden existir dentro del equipo de desarrollo, posibles cuellos de botella en el flujo de información; Posteriormente a la realización de esta técnica se definen estrategias con las redes sociales que existen y se fortalecen las relaciones que se requieran.

Conclusiones

- Usar la técnica de Design Thinking facilitó el encontrar dificultades en el manejo de la documentación del proyecto durante las fases de empatizar y definir por que la documentación se da durante todo el desarrollo y le es de utilidad a todos los involucrados ya que se pueden dar cambios que no se pueden documentar de la forma correcta sin tener procedimientos claros definidos y los encargados de esta actividad ven barreras que no pueden superar en el proyecto.

- El paso a seguir generado por Design Thinking se considera buenas prácticas de calidad a la hora de desarrollar software.
- Design Thinking ayuda a explorar más ideas y más rápido de lo que podrías utilizando otro concepto, adopta un enfoque centrado en el ser humano con consideraciones comerciales y tecnológicas, la innovación debe tener en cuenta el comportamiento humano, las necesidades y preferencias.
- Los resultados de la aplicación de Design Thinking son mejores si se usan templates o plantillas de trabajo que se encuentran en la web y realizar las fases con métodos tradicionales es un poco tedioso, así que es indispensable valerse de alguna herramienta web para realizar todos los diseños que son requeridos.

Referencias

- [1] Brown, Tim. "Design thinking." Harvard business review 86.6 (2008): 84. Available: <https://fusesocial.ca/wp-content/uploads/sites/2/2018/06/Design-Thinking.pdf> [Accessed November 01, 2019].
- [2] Herbsleb, James, et al. "Software quality and the capability maturity model." Communications of the ACM 40.6 (1997): 30-40. Available: <https://pdfs.semanticscholar.org/90cd/732b24707975d7adde20a58c302f23e3dc14.pdf> [Accessed November 01, 2019].
- [3] Slaughter, Sandra A., Donald E. Harter, and Mayuram S. Krishnan. "Evaluating the cost of software quality." Communications of the ACM 41.8 (1998): 67-73. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.450.3664&rep=rep1&type=pdf> [Accessed November 01, 2019].
- [4] Haag, Stephen, MmK Raja, and Lawrence L. Schkade. "Quality function deployment usage in software development." Communications of the ACM 39.1 (1996): 41-49. Available: https://www.researchgate.net/profile/Stephen_Haag/publication/220420894_Quality_Function_Deployment_Usage_in_Software_Development/links/53f4b5de0cf2888a749114b7/Quality-Function-Deployment-Usage-in-Software-Development.pdf [Accessed November 01, 2019].

[5] Sánchez, Ana María García, and Dtor Manuel García Clavel. "Evaluación de métricas de calidad del software sobre un programa Java." Universidad Complutense de Madrid, Madrid (2010). Available: https://eprints.ucm.es/11487/1/Proyecto_Fin_de_M%C3%A1ster.pdf [Accessed November 01, 2019].

[6] Müller, Roland M., and Katja Thoring. "Design thinking vs. lean startup: A comparison of two user-driven innovation strategies." *Leading through design* 151 (2012): 91-106. Available: https://s3.amazonaws.com/academia.edu.documents/40692530/Leading_Innovation_through_Design_Procee20151208-21966-ccwlds.pdf?response-content-disposition=inline%3B%20filename%3DLeading_Innovation_through_Design_Procee.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20191101%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20191101T234613Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=e472010c8014e117a49f4a85a6f0363cd5436b9f4a05ddfb08adbc0ef4b3c868#page=181 [Accessed November 01, 2019].

[7] Gaffney Jr, John E. "Metrics in software quality assurance." *Proceedings of the ACM'81 conference*. ACM, 1981. Available: https://people.dsv.su.se/~joco2917/ft_gateway.102cfm.pdf [Accessed November 01, 2019].

[8] Caldiera, Victor R. Basili, Gianluigi, and H. Dieter Rombach. "The goal question metric approach." *Encyclopedia of software engineering* (1994): 528-532. Available: https://s3.amazonaws.com/academia.edu.documents/40605563/gqm.pdf?response-content-disposition=inline%3B%20filename%3DThe_Goal_Question_Metric_Approach.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20191101%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20191101T234948Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=1de103e59d63c998e9d0ecd4f369abbc63aa69cfb9be9325e38d90a051474b3a [Accessed November 01, 2019].

[9] Kolko, Jon. "Design thinking comes of age." (2015): 66-71. Available: https://enterpriseproject.com/sites/default/files/design_thinking_comes_of_age.pdf [Accessed November 01, 2019].

[10] Steinbeck, Reinhold. "El «Design Thinking» como estrategia de creatividad en la distancia." *Comunicar* 19.37 (2011): 27-35. Available: <https://www.redalyc.org/pdf/158/15820024004.pdf> [Accessed November 01, 2019].