



Mejores prácticas de calidad en el desarrollo de software integradas al conocimiento de la ingeniería

Best quality practices in software development integrated with engineering knowledge

Melhores práticas de qualidade em desenvolvimento de software integrado com conhecimento de engenharia

Germania del Roció Veloz-Remache ^I
g_veloz@epoch.edu.ec
<https://orcid.org/0000-0003-2865-8181>

Jorge Ariel Menéndez-Verdecia ^{II}
jorge.menendez@epoch.edu.ec
<https://orcid.org/0000-0001-5221-1326>

Linda Noralma Aguilar-Moncayo ^{III}
laguilar@epoch.edu.ec
<https://orcid.org/0000-0001-7644-1804>

Correspondencia: g_veloz@epoch.edu.ec

Ciencias Técnicas y Aplicadas
Artículo de investigación

***Recibido:** 15 de noviembre de 2020 ***Aceptado:** 21 de diciembre de 2020 * **Publicado:** 09 de enero de 2021

- I. Magister en Interconectividad de Redes, Master Universitario en Ingeniería de Software y Sistemas Informáticos, Ingeniero en Sistemas Informáticos Escuela Superior Politécnica de Chimborazo, Sede Morona Santiago, Macas, Ecuador.
- II. Diploma Superior en Formulación y Evaluación de Proyectos de Investigación, Magister en Gestión de Bases de Datos, Ingeniero Informático, Ingeniero Informático, Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.
- III. Magister en Gerencia de Redes y Telecomunicaciones, Ingeniera en Electrónica y Computación, Formación De Formadores, Escuela Superior Politécnica de Chimborazo, Sede Morona Santiago, Macas, Ecuador.

Resumen

La responsabilidad de la Academia en la formación de ingenieros de software debe orientarse a fortalecer sus habilidades y destrezas en el proceso de desarrollo de soluciones, tomando en cuenta a la "Calidad" como un factor importante para obtener productos que satisfagan las expectativas de los usuarios. La investigación presenta la aplicación de buenas prácticas de calidad en la construcción de proyectos que automatizan procesos académicos universitarios, analizando los errores más comunes que tienen los equipos de trabajo e identificando las mejoras en el proceso de desarrollo de software al aplicar mejores prácticas de calidad. Se aplicó a Scrum como mejor práctica de desarrollo, observación del manejo de las diferentes fases y revisiones frecuentes a la aplicación de sugerencias, notando que la habilidad evaluación constante de la calidad en los grupos de desarrollo maduró en cada fase.

Palabras-clave: Calidad de software; mejores prácticas; ingeniería de software.

Abstract

The responsibility of the Academy in the training of software engineers should be aimed at strengthening their skills and abilities in the process of developing solutions, taking into account "Quality" as an important factor to obtain products that meet the expectations of users. The research presents the application of good quality practices in the construction of projects that automate university academic processes, analyzing the most common mistakes that work teams have and identifying the improvements in the software development process by applying best quality practices. Scrum was applied as a best development practice, observation of the management of the different phases and frequent revisions to the application of suggestions, noting that the constant quality assessment skill improved by xxxx%.

Keywords: Software Quality; better practices; software Engineering.

Resumo

A responsabilidade da Academia na formação de engenheiros de software deve estar direcionada para o reforço das suas competências e competências no processo de desenvolvimento de soluções, tendo em consideração a "Qualidade" como um factor importante para obter produtos que satisfaçam as expectativas dos utilizadores. . A pesquisa apresenta a aplicação de boas práticas de

qualidade na construção de projetos que automatizam processos acadêmicos universitários, analisando os erros mais comuns das equipes de trabalho e identificando melhorias no processo de desenvolvimento de software por meio da aplicação de melhores práticas de qualidade. Scrum foi aplicado como melhor prática de desenvolvimento, observação da gestão das diferentes fases e revisões frequentes da aplicação de sugestões, observando que a capacidade de avaliar constantemente a qualidade nos grupos de desenvolvimento amadureceu em cada fase.

Palavras-chave: Qualidade de software; melhores práticas; Engenharia de software.

Introducción

La calidad aplicada al desarrollo de software es una de las exigencias del mercado hoy en día, ya que no solo se requiere de aplicaciones que resuelvan una determinada problemática, sino que debe ofrecer un plus orientado a la satisfacción del cliente final. La calidad es un criterio subjetivo que requiere de la determinación de parámetros medibles para establecer el grado de calidad que tiene un proceso y/o producto, en cada una de sus fases, no solo se debe esperar a la evaluación del producto final.

En 2012, el Ministerio de Industrias y Productividad de Ecuador priorizó diez sectores industriales, uno de ellos el software ya que para el cumplimiento de la Agenda de Transformación Productiva se solicitó a la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (Senescyt) trazar el camino hacia la economía social del conocimiento y la innovación colaborativa, responsable y orientada a la satisfacción de las necesidades de la población (Ecuador & Secretaría Nacional de Planificación y Desarrollo, 2009). Por tanto, la formación universitaria es responsable de conseguir profesionales competentes y exitosos desde sus actividades académicas por ello, dentro del proceso de desarrollo de software se debe incorporar a la calidad en cada una de sus fases, ya que es indispensable saber qué se espera del producto final en cuanto a sus requerimientos funcionales y no funcionales, con lineamientos a la satisfacción del cliente, es así que la enseñanza de la “Calidad” a los futuros profesionales en área de Ingeniería de Software debe procurar motivar la creencia de su aplicación en cada uno de los pasos que conlleva a la solución de una problemática.

El presente trabajo recopila una serie de mejores prácticas aplicadas en el desarrollo de soluciones software realizadas en asignaturas de proyectos integradores de la carrera de Sistemas y Software,

permitiendo así determinar dentro de la investigación: errores concurrentes y posibles alternativas dentro de la aplicación de la Ingeniería de Software en soluciones reales en medio de parámetros de calidad fundamentales.

Metodología

“La calidad de un producto la determina el proceso usado para desarrollarlo”. (Humphrey,2011)

El estudio emplea a tres aplicaciones relacionadas a actividades académicas universitarias: Programas Analíticos (P1), Sílabo de Asignatura (P2) y Planificación Semestral de la asignatura (P3), donde sus equipos de desarrollo están conformados por dos o tres estudiantes de la carrera de Software cada uno.

Para la construcción de tres soluciones se empleó a Scrum como práctica ágil de desarrollo ya que las metodologías ágiles surgen para aplicarse principalmente en proyectos cuya naturaleza sea cambiante. (Víctor Hugo Mercado Ramos & Zapata Julian, 2015), donde la academia la emplea en la formación (Mina & Gallegos Doris, 2017). Así mismo se determina los estándares de calidad empleados en el mercado nacional como CMMI (CMMI Institute. 2017), ISO 25000 (Oktaba, H. s. f.), ISO

La definición de las métricas para evaluación de la calidad en el desarrollo se fundamenta en los componentes de Scrum como: Product Backlog, sprint backlog y burndown chart, la distribución de roles product owner, Scrum master y equipo de trabajo ejecutados por las unidades institucionales de la universidad, profesores tutores y estudiantes respectivamente (Yépez Juan David et al., 2015). También son utilizadas los diferentes tipos de sesiones establecidas en Scrum para la revisión de avances de los proyectos planteados con el objetivo de detectar errores, retroalimentar procesos y corregirlos oportunamente.

Se emplean herramientas como Planner de Office 365, correo electrónico, redes sociales como Facebook y WhatsApp con la finalidad de agilizar tareas de planificación y gestión de los proyectos, así como el planteamiento de métricas de calidad como estándar para todos los proyectos en construcción. BootStrap

A continuación, se muestra un resumen de errores frecuentes en los tres proyectos de desarrollo de software en estudio.

Errores detectados

El término error es definido como “una idea falsa o equivocada”, los productos software no pueden estar basados en ideas por lo cual se indicará que un error es un defecto de software que puede causar un fallo y el fallo es una salida no aceptable de una aplicación basada en los requerimientos funcionales. Por lo tanto, se describe los problemas detectados en función de los procesos realizados por los grupos de trabajo en cada una de sus fases.

Requerimentación

Un error frecuente de los equipos de desarrollo es pensar que el usuario solo suministra los requerimientos en las primeras fases, en la tabla 1, se describe los problemas que se presentan en la fase de requerimentación, donde se puede notar que emplean procesos informales de adquisición de requerimientos a través de entrevistas personales, las mismas que no se encuentran claramente diseñadas y no formalizan lo adquirido en los diferentes encuentros con el cliente que son mínimos

Tabla 1: Errores frecuentes en la fase de requerimentación

Problema Detectado	P1	P2	P3
<i>Proceso definido para automatizar</i>	Parcialmente definido	Parcialmente definido	Parcialmente definido
<i>Forma de adquisición del requerimiento</i>	Informal Poca planificación de la entrevista	Informal Media planificación de la entrevista	Informal Poca planificación de la entrevista
<i>Interpretación del requerimiento</i>	A conveniencia del desarrollador	A conveniencia del desarrollador	A conveniencia del desarrollador
<i>Establecimiento de prioridades</i>	Basado en la experiencia del desarrollador, no en la necesidad del cliente	Basado en la experiencia del desarrollador, no en la necesidad del cliente	Basado en la experiencia del desarrollador, no en la necesidad del cliente
<i>Visitas al cliente</i>	Mínima	Regular	Mínima

Al no haber documentado, se supone la resolución del problema de manera superficial, debiendo nuevamente planificar otras reuniones, lo cual genera planificaciones no reales y pérdida de recursos.

Diseño de Base de Datos

Una de las formas para determinar los errores cometidos en el diseño de las bases de datos es la cantidad de rediseños que tiene la misma. Al momento en que se realizan las reuniones de avance de los proyectos para revisión del diseño de la base de datos se encontraron las observaciones identificadas en la tabla 2, donde el error más común que poseen los equipos de trabajo es la identificación de relaciones recursivas y el manejo de identificadores, lo cual se genera en un problema para la escalabilidad de la base de datos a futuro, implicando un rediseño de impacto en un posible mantenimiento o actualización.

Tabla 2: Errores frecuentes en el diseño de la base de datos

Problema Detectado	P1	P2	P3
<i>Relaciones recursivas</i>	2 de tres identificadas	Una de dos identificadas	dos de dos identificadas
<i>Nivel de empleo de identificadores</i>	Medio	Medio	Medio
<i>Identificación de cardinalidad de las relaciones entre las entidades.</i>	Relaciones de 1-n	Relaciones de 1-n	Relaciones de 1-n
<i>Confusión entre Entidades y Relaciones</i>	2 errores encontrados	1 error encontrado	3 errores encontrados

Interfaz

Tabla 3: Errores frecuentes en la codificación

Problema Detectado	P1	P2	P3
<i>No respetar interfaz definida</i>	4	1	5

Pruebas

Los integrantes del equipo de desarrollo eran quienes ejecutaban las pruebas de las diferentes funcionalidades con datos ya conocidos, no con datos nuevos que permitan la identificación de fallas funcionales.

Documentación

Generar la documentación es uno de los criterios que los equipos de desarrollo manifiestan un grado alto de resistencia, y se identifica claramente este fenómeno por la falta de actualización de los manuales técnico y de usuario. La documentación en las prácticas ágiles puede verse afectada, por el objetivo de aumentar la productividad en el proceso de desarrollo de software, donde la documentación puede ocasionar demoras en las entregas de funcionalidades.

Al realizar las reuniones de avance de los proyectos P1,P2,P3, se determina que el 95% del tiempo consumido en el desarrollo del software se lo orienta a la programación, el 4% a la gestión y un 1% a la documentación, implicando que se deba revisar código fuente, diseños y otros recursos para que la documentación sea realizada. Este problema conlleva a no poder entregar funcionalidades con su documentación en el tiempo planificado o afectando al mantenimiento o conocimiento de lo realizado en caso de cambio o integración de un nuevo miembro al equipo de desarrollo.

Tabla 4: Errores frecuentes en la documentación

Problema Detectado	P1	P2	P3
<i>Actualización de manuales</i>	Mínima	Mínima	Mínima
<i>Detalle de componentes (parámetros, servicios web empleados, librerías o paquetes empleados)</i>	Bajo	Bajo	Bajo
<i>Claridad y Síntesis</i>	Mínima	Mínima	Mínima
<i>Eficiente empleo de Lenguaje técnico</i>	Medio	Medio	Medio

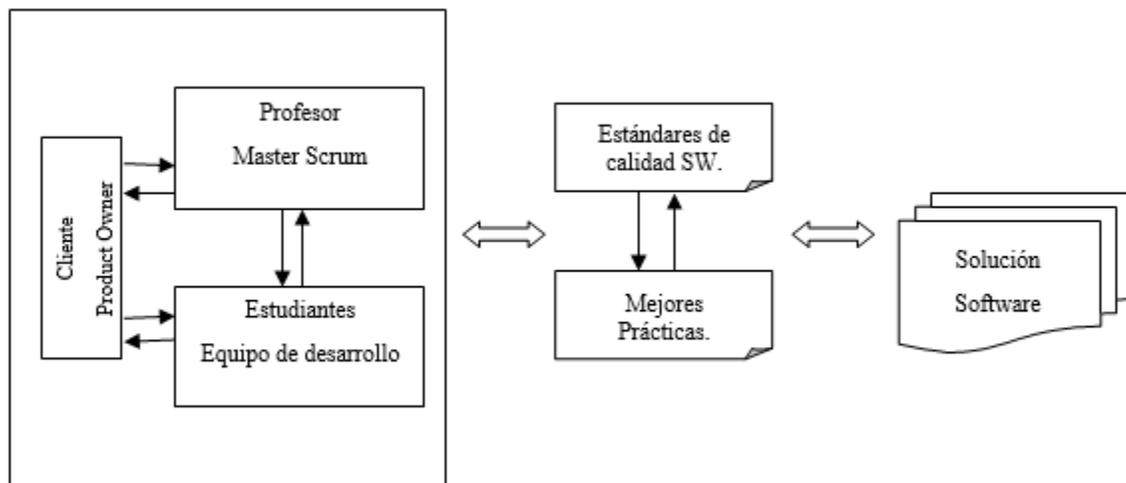
En la tabla 4, se presenta los problemas más comunes en la documentación de los tres proyectos en estudio, determinando que se requiere describir componentes esenciales en las historias de usuario como librerías, clases, bases de datos, servicios web entre otros. El lenguaje empleado requiere ser claro y concreto en base a los conocimientos técnicos incorporados.

Mejores prácticas aplicadas

Uno de los papeles que posee el profesor de las asignaturas relacionadas a la Ingeniería de Software es permitir que los estudiantes experimenten diferentes escenarios y distingan los beneficios de aplicar buenas prácticas convirtiéndose estas, en una herramienta que contribuyen a mejorar sus habilidades en la construcción de un producto software, así como también a que las fases de desarrollo se las ejecute orientadas a la obtención de productos de calidad.

En la figura 1 se plantea un modelo de enseñanza de la calidad en el proceso de desarrollo de software.

Figura 1: Modelo de aplicación de mejores prácticas de calidad en el desarrollo de software



En la Figura 1, se determina la importancia de que el cliente sea parte del proceso de desarrollo y su participación sea activa, donde el Scrum master (profesor), será quien monitorea y evalúa a través de las métricas de calidad planteadas al inicio de cada proyecto, al igual que el equipo de desarrollo (estudiantes) aportan en un nivel operativo inferior al cumplimiento de la calidad en cada fase.

Requerimentación

- Análisis previo de los objetivos del proyecto. Las actividades deben estar basadas en un análisis previo del problema a resolver, tomando en cuenta las prioridades del cliente en la entrevista inicial. Esta entrevista debe estar en co-participación del Scrum máster (profesor) quien será mediador entre el cliente y el equipo de desarrollo (estudiantes), permitiendo que los estudiantes actúen libremente, pero seguros de una participación oportuna por parte del profesor, adquiriendo habilidades de planificación y gestión al mismo tiempo.
- Planificación real basada en los objetivos del proyecto. Se definirá una lista de requerimientos y prioridades, las mismas que se trasladen a una planificación realizada por el equipo de desarrollo en base a los horarios disponibles para dicho proceso. El Scrum master (profesor), deberá verificar si la apreciación de la planificación en cuanto a utilización de recursos es factible o no, sugiriendo ajustes de la planificación de ser el caso.

Diseño de Bases de Datos

- Rediseño de la base de datos. Desde el diseño inicial de la base de datos el equipo de trabajo en conjunto con el Scrum master (profesor), deberán esquematizar la información manejada a través de la definición del proceso a automatizar. Se debe verificar entidades, relaciones, reglas de negocio y roles permitiendo un diseño escalable y mantenible.

Pruebas

El número sprint realizados, así como las historias de usuario con aceptación del cliente deben ser analizadas con la finalidad de crear una lista de funcionalidades ya cumplidas.

Se realizan pruebas entre equipos de desarrollo de los proyectos P1,P2, y P3; facilitando así la determinación de posibles errores, informando los mismos en las reuniones de avance al Scrum Master, quien designará responsabilidades y planes de acción en conjunto con el equipo de desarrollo.

Documentación

- Establecimiento de estándares de documentación. Se plantea al inicio del cada proyecto un estándar de documentación orientada a la elaboración de informes, codificación, manejo de bases de datos y comunicación interna o externa a los equipos de desarrollo.
- Actualización de Manuales. Cada reunión de avance requería presentar la nueva versión de los manuales técnico y de usuario, con las actividades ejecutadas y evaluadas ya terminadas. El Scrum master realiza esta verificación, hasta que el equipo de desarrollo consiga documentar al mismo ritmo que realiza la programación de los sprint, así como también las diferentes pruebas de usuario con su estado y sugerencias.

Gestión

Se dice que el 80% de re-trabajo se debe al 20% de los defectos de un producto software, así como también el uso de prácticas para la mejora del equipo y los profesionales puede reducir los defectos en un 75% o más. Si se encuentra un problema después de haber entregado el producto es 100 veces más costoso que detectar y corregirlo en una etapa de diseño o desarrollo (Garzás, 2011).

- Gestión frecuente de las expectativas del cliente. La relación estrecha entre el cliente y los grupos de trabajo, facilitaba conocer sus principales necesidades y proyectarlas en el desarrollo integrando diseño escalar y adaptable a los cambios.
- Productividad y Calidad. Los equipos de desarrollo al inicio presentan un grado de incertidumbre debido a la cantidad de requerimentación presente, así como la probabilidad de cambios en los mismo, por la falta de definición de los procesos relacionados a sus aplicaciones, pero poco a poco van ganando la destreza de liderazgo, integrándose entre ellos; lo que conduce al trabajo colaborativo creando código que puede ser adaptado de manera eficiente entre los grupos. Así también se planifica la calidad, se la ejecuta en cada uno de los pasos como: diseño de bases de datos, diseño de interfaz, reutilización de clases y código, se la evalúa por roles desde el equipo de trabajo, scrum master y scrum owner, logrando el filtro y corrección correspondiente oportuna.
- Equipo Motivado y Seguro. Cada grupo de desarrollo debe realizar tareas de investigación y aplicación de la misma en las diferentes fases de desarrollo de los proyectos en estudio,

aspecto que permitió conseguir estudiantes seguros en la aplicación de conocimientos crear un nuevo nivel de estudiantes que pueden ser soporte del Scrum master en la formación de nuevos integrantes a los grupos de desarrollo.

Discusión

La Revista chilena de ingeniería, en su artículo “Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico”, menciona la importancia del aseguramiento de la calidad en el desarrollo de software utilizando métodos que permitan recabar, evaluar y divulgar datos sobre el proceso de ingeniería de software ; teniendo similitud en el presente trabajo en la importancia y el uso de una metodología adecuada que permita mejorar la calidad del producto final y difiere en que se utiliza diferentes métodos mientras que en el presente artículo se aplicó la metodología SCRUM, luego haber sido elegida luego de una revisión bibliográfica minuciosa.

El artículo titulado “Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles”, realizado por Víctor Hugo Mercado-Ramos¹Julián Zapata²Yony Fernando Ceballos se relacionan en el estudio de diferentes metodologías que aseguren la calidad del software, mencionan aspectos como casos pruebas, documentación (no excesiva) son determinantes para entregar un producto con calidad.

Los autores Alex Espejo Chavarría, Sussy Bayona Oré y Carlos Pastor presentan información muy interesante en su trabajo titulado: “Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP”, en el que presentan un caso de estudio de la aplicación de un modelo para mejorar la calidad del proceso de desarrollo software en una empresa desarrolladora de software que ha sido certificada en CMMI utilizando las metodologías PSP y TSP y el modelo CMMI con lo que se logra reducir el coste de calidad, reducir los defectos, la desviación de coste y se incrementa la satisfacción del usuario.

Conclusiones

La formación de los profesionales en el área de la Ingeniería de Software requiere la realización de proyectos académicos de media complejidad, basados en ámbitos reales de desarrollo, donde

puedan adquirir y mejorar las habilidades y destrezas adquiridas en cada nivel de su formación universitaria con la experiencia que el mercado laboral necesita.

Al realizar la aplicación de mejores prácticas de calidad dentro del proceso de desarrollo y conservar los equipos de trabajo madurando cada proyecto en estudio, permitió en los grupos de desarrollo, el compromiso consciente que tiene con la realidad nacional, siendo ellos, los que pueden ser soporte en la formación de nuevos integrantes hasta llegar a la obtención de un producto software de calidad, que satisfaga las necesidades de los clientes, pero que además a ello, los alimente en cuanto a experiencia profesional..

Referencias

1. Carrizo, D & Alfaro,A (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. Recuperado de https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-33052018000100114
2. CMMI Institute. (2017). CMMI Maturity Profile Report. Recuperado a partir de <http://partners.cmmiinstitute.com/wp-content/uploads/2017/04/Maturity-Profile-Ending-in-Dec-31-2016.pdf>
3. Díaz, J. R. (2009). Las metodologías ágiles como garantía de calidad del software. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 5(3), 40-43.
4. Ecuador, & Secretaría Nacional de Planificación y Desarrollo. (2009). Plan nacional para el buen vivir, 2009-2013: Construyendo un estado plurinacional e intercultural. Secretaría Nacional de Planificación y Desarrollo, SENPLADES.
5. Espejo, A; Bayona, S; & Pastor (2016). Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP . Recuperado de http://www.scielo.mec.pt/scielo.php?pid=S1646-98952016000400006&script=sci_arttext&tlng=es
6. Garzás, J. (2011). Algunos datos estadísticos de la calidad software - Javier Garzás. Recuperado 1 de julio de 2017, a partir de <http://www.javiergarzas.com/2011/03/datos-estadsticos-calidad-software.html>
7. Guaycha, W. (2016). Estudio de Calidad para el proceso de desarrollo de software de empresas de Guayaquil. Universidad de Guayaquil, Guayaquil. Recuperado a partir de <http://repositorio.ug.edu.ec/bitstream/redug/16189/1/TESIS%20Revision%20II.pdf>

8. Mercado-Ramos, V.H., Zapata, J., & Ceballos, Y.F. (2015). Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles. *Rev.investig.desarro.innov*, 6(1), 73-83. Recuperado de : <https://dialnet.unirioja.es/servlet/articulo?codigo=6763102>
9. Mina, E. M., & Gallegos Doris. (2017). La industria del software en Ecuador: Evolución y situación actual. *Espacios*, 38(57), 25.
10. Ministerio de Industrias y Productividad del Ecuador. (2013). El software una industria priorizada en Ecuador. *País Productivo*, 7. Recuperado a partir de <http://www.industrias.gob.ec/wp-content/uploads/downloads/2013/08/REVISTA-7.pdf>
11. Oktaba, H. (s. f.). SQUARE: Modelo actualizado de las características de calidad. UNAM. Recuperado a partir de <https://sg.com.mx/content/view/990>
12. Víctor Hugo Mercado Ramos, & Zapata Julian. (2015). Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles—Dialnet. *Revista de Investigación Desarrollo e Innovación: RIDI*, 6(1), 73-83.
13. Ubeda, R. (2009). Métodos ágiles para el desarrollo de software. Universidad Politécnica de Catalunya, España. Recuperado a partir de <https://core.ac.uk/download/pdf/41798414.pdf>
14. Yépez Juan David, Pardo César, & Gómez Omar. (2015). Revisión sistemática acerca de la implementación de metodologías ágiles y otros modelos en micro, pequeñas y medianas empresas de software | Yepes González | *Revista Tecnológica—ESPOL. RTE Revista Tecnológica ESPOL*, 28(5), 464-479.