



Pruebas de regresión de software: hallazgos iniciales en la industria y academia del Ecuador

Software regression testing: initial findings from the industry and academia in Ecuador

Teste de regressão de software: descobertas iniciais na indústria e na academia equatoriana

Raúl Hernán Rosero-Miranda^I

rrosoro@epoch.edu.ec

<https://orcid.org/0000-0002-2315-9773>

Omar Salvador Gómez-Gómez^{II}

ogomez@epoch.edu.ec

<https://orcid.org/0000-0002-2951-3833>

Correspondencia: rrosoro@epoch.edu.ec

Ciencias Técnicas y Aplicadas

Artículo de Investigación

***Recibido:** 30 de Septiembre de 2021 ***Aceptado:** 30 de Octubre de 2021 * **Publicado:** 23 de Noviembre de 2021

- I. Ingeniero en Informática, Doctor en Ingeniería (Ph.D.), Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador
- II. Ingeniero Sistemas y Computación, Doctor en Sistemas e Informática (Ph.D.), Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador

Resumen

Las pruebas de regresión de software son un tipo especial de prueba que se efectúa durante el desarrollo o mantenimiento de productos software. En este trabajo se presentan hallazgos iniciales sobre una encuesta nacional aplicada a actores de la academia (universidades) e industria (desarrolladores de software). En esta encuesta se examinan elementos como son: metodologías, técnicas, métricas, enfoques aplicados en estos dos sectores. Los resultados de la encuesta sugieren que en la academia y la industria ecuatoriana la metodología de desarrollo usada es la iterativo-incremental. La técnica más aplicada de pruebas de regresión es la priorización de casos de prueba; la métrica más utilizada es la eficiencia en sus tiempos de ejecución, entre otros hallazgos. Los resultados aquí reportados sirven como referente para considerar ajustes curriculares en carreras afines a la informática de las instituciones de educación superior del país, así como efectuar ajustes a los procesos de verificación en las empresas de desarrollo de software nacionales.

Palabras clave: Pruebas de regresión de software; pruebas de software; academia e industria ecuatoriana; ingeniería de software.

Abstract

Software regression testing is a special type of testing that is performed during the development or maintenance of software products. This paper presents initial findings on a national survey applied to actors from academia (universities) and industry (software developers). This survey examines elements such as: methodologies, techniques, metrics, and approaches applied in these two sectors. The results of the survey suggest that in Ecuadorian academia and industry, the development methodology used is iterative-incremental. The most widely applied regression testing technique is the prioritization of test cases; the most used metric is the efficiency measured as execution time, among other findings. The results reported here serve as a reference to consider curricular adjustments in computer science-related careers in the country's higher education institutions, as well as to adjust the verification processes in national software development companies.

Keywords: Software regression testing; software testing; Ecuadorian academia and industry; software engineering.

Resumo

O teste de regressão de software é um tipo especial de teste executado durante o desenvolvimento ou manutenção de produtos de software. Este artigo apresenta os resultados iniciais de uma pesquisa nacional aplicada a atores da academia (universidades) e da indústria (desenvolvedores de software). Esta pesquisa examina elementos como: metodologias, técnicas, métricas e abordagens aplicadas nesses dois setores. Os resultados da pesquisa sugerem que na academia e na indústria equatoriana a metodologia de desenvolvimento utilizada é iterativo-incremental. A técnica de teste de regressão mais amplamente aplicada é a priorização de casos de teste; a métrica mais utilizada é a eficiência em seus tempos de execução, entre outras constatações. Os resultados aqui relatados servem de referência para considerar ajustes curriculares nas carreiras relacionadas com a informática nas instituições de ensino superior do país, bem como para fazer ajustes nos processos de verificação nas empresas nacionais de desenvolvimento de software.

Palavras-chave: Teste de regressão de software; teste de software; indústria e acadêmica equatoriana; engenharia de software.

Introducción

Las pruebas de regresión de software permiten disminuir la posibilidad de inyectar defectos en versiones anteriores del producto software (ya sea en un enfoque de mantenimiento correctivo o adaptativo), o en etapas de desarrollo (construcción)(Rothermel & Harrold, 1993). Dado que las pruebas de regresión de software se ejecutan sobre versiones anteriores del producto, se dice que este enfoque favorece una mayor confiabilidad en el producto software producido o mantenido (Morales & Aguilar Vera, 2021).

Diversos enfoques de pruebas de regresión han sido considerados en la literatura. Por ejemplo, Rothermel et al., (Rothermel & Harrold, 1994) y (Engström et al., 2010) presentan propuestas de evaluación de las técnicas clásicas. Una caracterización de enfoques clásicos de pruebas de regresión de software se discute en (Yoo & Harman, 2010) y (Khatibsyarbini et al., 2018), donde los autores realizan una revisión de la literatura.

Considerando aspectos de desarrollo y mantenimiento del producto software, como por ejemplo el manifiesto ágil(*Manifesto for Agile Software Development*, s. f.), algunos enfoques de

pruebas de regresión son considerados en el contexto de la integración continua de software o el desarrollo guiado por pruebas TDD (Ebert & Paasivaara, 2017) (Rosero, Gómez, et al., 2017).

En los años recientes, aportes en el ámbito de la inteligencia artificial han sido considerados, entre ellos, algoritmos de aprendizaje máquina, redes neuronales, o algoritmos de heurísticas de búsqueda han sido considerados también como parte de la evolución tecnológica aplicada a las técnicas de pruebas de regresión de software(Kumar et al., 2014),(Parsa & Khalilian, 2009).

En el contexto internacional, se observa que la aparición de nuevos enfoques de pruebas de regresión de software en enmarcan principalmente en el contexto académico (Rosero et al., 2016)(Gómez et al., 2016). Con respecto al contexto del Ecuador, los autores en (Mina & Barzola, 2017) indican que la academia debe actualizar los contenidos curriculares en carreras afines a la informática para que sus egresados ya en el ámbito de la industria, dominen metodologías tales como: SCRUM(Khalid et al., 2020), XP(Taibi et al., 2017), KAMBAN (Ikonen et al., 2011), así como también, puedan realizar experimentos evaluando la calidad y productividad de los productos de software utilizando métodos flexibles. En cuanto al ámbito de la academia, los autores en (Arcos et al., 2018) observan que tradicionalmente el enfoque de programación utilizado es el paradigma orientado a objetos. Se observa el uso de estrategias colaborativas como la programación en pareja. Si bien existen algunos trabajos en el ámbito de la ingeniería de software en el contexto industrial y académico, no hemos encontrado trabajos que reporten el uso de las pruebas de regresión en la industria y la academia del Ecuador.

En el presente trabajo se reportan hallazgos preliminares acerca de la primera encuesta a nivel nacional sobre la enseñanza y la aplicación en la industria de las pruebas de regresión de software. Para ello se ha empleado una muestra de 10 instituciones de educación superior (IES) registradas en la Secretaría Nacional de Educación Superior, Ciencia tecnología e innovación del Ecuador (SENESCYT)(*SENESCYT - Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación, s. f.*), así como, una muestra de 20 empresas desarrolladoras de software (EDS) registradas en la Cámara de Innovación y Tecnología Ecuatoriana (CITEC) (*CITEC – Cámara de Innovación y Tecnología Ecuatoriana, s. f.*). El objetivo de esta encuesta es conocer bajo dos perspectivas (academia e industria), las técnicas que se enseñan y se aplican, las metodologías, las métricas, así como los niveles en el proceso de desarrollo software en los cuales las pruebas de regresión de software son enseñadas y aplicadas.

El resto del documento se encuentra organizado de la siguiente manera en la sección 2 presentamos un panorama genérico acerca de las pruebas de regresión de software. En la sección 3 se describe el método seguido para el desarrollo y ejecución de la encuesta. En la sesión 4 se presentan los resultados obtenidos. Posteriormente, en la sección 5 se discute los resultados encontrados para finalmente, en la sección 6 presentar las conclusiones.

Vista panorámica de las pruebas de regresión de software

De manera general las pruebas de regresión consisten en re ejecutar todos los casos de prueba disponibles, ya sea en etapas de desarrollo o en mantenimiento. Proceso que es costoso en recursos y tiempo, afectado por los diferentes enfoques que se adopten, entre éstos se consideran: la minimización, la selección, la priorización y la optimización de casos de prueba. A continuación, se describe cada uno de estos enfoques.

Pruebas de regresión por minimización

De acuerdo con la técnica de minimización (Wong et al., 1995) ésta consiste básicamente en la eliminación de casos de prueba redundantes, repetidos u obsoletos, formalmente, Sea P un producto de software y T el conjunto de casos de prueba asociados con P . En la minimización, el problema consiste en obtener un subconjunto de casos de prueba T' que no incluya casos de prueba redundantes u obsoletos para verificar P .

Pruebas de regresión por selección

En esta técnica se centra en la detección de las partes modificadas de un programa de software y los casos de prueba asociados a esa sección de código, que normalmente funciona con base en el análisis estático de caja blanca. Según Kumar et al., (2014), dado un programa P , la versión modificada de P , P' y un conjunto de pruebas T , el problema consiste en determinar un subconjunto de T , T' , con el que se pruebe P' (Rothermel & Harrold, 1996).

Pruebas de regresión por priorización

En la priorización (Panigrahi & Mall, 2010), el objetivo es determinar una permutación ideal de secuencias de casos de prueba para mejorar el rendimiento de la técnica o enfoque de regresión,

basado en aspectos como por ejemplo, su capacidad de detección de fallos. Formalmente, dado un conjunto de pruebas T , el conjunto de permutaciones de T , PT y una función de PT a números reales $f: PT \rightarrow R$, luego encontrar T' perteneciente a PT donde $(\forall T'')(T'' \in PT) (T'' = T') [f(T') \geq f(T'')]$.

Pruebas de regresión por optimización

Esta estrategia utiliza combinaciones de los enfoques antes mencionados. También se utilizan técnicas de optimización multiobjetivo e inteligencia artificial como aprendizaje máquina (Mohanty et al., 2010), algoritmos de agrupamiento supervisado y no supervisado basado en patrones como por ejemplo, de similitud de casos de prueba (Rosero, Gómez, et al., 2017).

Método

En esta sección presentamos las preguntas de investigación, así como las características del cuestionario aplicado. El objetivo principal de este trabajo se centra en determinar los aspectos relevantes sobre las metodologías, técnicas y métricas tomando como referente el marco de características de las pruebas de regresión consideradas en (Rosero et al., 2016)., y considerando como muestra un conjunto de Instituciones de Educación Superior (IES) así como Empresas de Desarrollo de Software (EDS) del Ecuador. Con respecto a la academia, se plantean las siguientes preguntas de investigación:

Preguntas de investigación planteadas en el contexto académico

Con respecto al contexto académico, se definieron las siguientes cinco preguntas:

- **RQA₁**. ¿Cuál es la denominación de la carrera?
- **RQA₂** ¿La carrera considera asignaturas relacionadas con temas de pruebas de software en su malla curricular?
- **RQA₃** En el programa analítico de las asignaturas relacionadas con temas de pruebas de software, ¿Se considera aspectos relacionados con las pruebas de regresión de software (modelos, marcos de trabajo, enfoques, técnicas, indicadores, métricas)?
- **RQA₄** ¿Qué recursos educacionales se utilizan para la enseñanza de las pruebas de regresión de software?

- **RQ_A5** ¿Se considera algún modelo para la enseñanza de pruebas de regresión de software?

Preguntas de investigación definidas para el ámbito de la industria

En cuanto al ámbito de la industria, se definieron las siguientes diez preguntas:

- **RQ_{I1}** ¿Qué tipo de producto software desarrolla la empresa?
- **RQ_{I2}** ¿Qué metodología de desarrollo software se aplica en su empresa?
- **RQ_{I3}** Dentro del modelo conocido como V, ¿Qué niveles de pruebas aplica en su metodología de desarrollo?
- **RQ_{I4}** ¿Qué planes de prueba ejecuta de acuerdo con el modelo V?
- **RQ_{I5}** ¿Qué técnica aplica para el proceso de pruebas en el desarrollo del producto?
- **RQ_{I6}** ¿Qué enfoque aplica para la aplicación de las pruebas de regresión?
- **RQ_{I7}** ¿Qué técnica de prueba de regresión aplica en el proceso de construcción de software?
- **RQ_{I8}** ¿Qué métricas utiliza cuando ejecuta las pruebas de regresión?
- **RQ_{I9}** ¿Tiene automatizado el proceso de pruebas de regresión?
- **RQ_{I10}** ¿Qué herramientas utiliza para automatizar el proceso de las pruebas de regresión?

Las preguntas antes descritas se hicieron disponibles a través de dos cuestionarios en línea diseñados para recabar la información correspondiente al contexto académico e industrial. En cuanto al contexto académico, la unidad de análisis corresponde a directores, coordinadores y profesores pertenecientes a IES ecuatorianas quienes imparten asignaturas relacionadas con tópicos de pruebas de software, en particular las pruebas de regresión de software. Con respecto a la unidad de análisis usada para el contexto de la industria, ésta se conforma por una muestra de desarrolladores, así como gerentes de productos software.

Como estrategia de validación del cuestionario, utilizamos la revisión de expertos siguiendo un enfoque de grupo focal (Molléri et al., 2016). En este enfoque, durante cuatro sesiones de trabajo los autores del presente trabajo confeccionaron las preguntas del cuestionario. En cada contexto se realizó una prueba piloto aplicada a los mismos autores con el fin de medir los

tiempos en completar los ítems de dicho cuestionario y realizar un último refinamiento sobre las preguntas.

En cuanto al método de muestreo, se utilizó el muestreo accidental no probabilístico por conveniencia. En este tipo de muestreo los participantes son seleccionados con respecto a una serie de contactos previamente conocidos, puntualmente se enviaron invitaciones para completar el cuestionario a diversos docentes y coordinadores de carreras afines al desarrollo de software (academia), así como a gerentes de productos y profesionales del desarrollo de software (industria). Este tipo de muestreo es comúnmente utilizado en encuestas realizadas en el ámbito de la ingeniería de software (Molléri et al., 2016), (Arcos et al., 2018).

Ejecución

Una vez diseñado los cuestionarios, estos fueron puestos en línea a través de la plataforma de Google (Martin et al., 2017). Ambos formularios se mantuvieron en línea durante los meses de julio y agosto del año 2021. En total se completaron de manera satisfactoria 10 formularios en el área de las instituciones de educación superior y 20 en el área de las empresas desarrolladoras de software.

Resultados

Posterior al llenado de los formularios, se procedió al preprocesamiento de la información, así como al análisis de ésta. Las respuestas a los cuestionarios se pasaron a una hoja de cálculo donde se efectuaron una serie de análisis, mismos que permiten responder a las preguntas de investigación antes planteadas. A continuación, se presentan los resultados organizados según cada contexto (academia e industria).

Academia

En el ámbito académico, se observa que quienes respondieron pertenecen a tres titulaciones registradas en la SENESCYT (Secretaría de Educación Superior, Ciencia, Tecnología e Innovación), que son: ingeniería en software con 60%, ciencias de la computación e ingeniería en computación, ambas con porcentajes del 20% (pregunta **RQA₁**).

Con respecto a la segunda pregunta **RQA₂**, se consideró los posibles nombres de aquellas asignaturas que conformen parte de la oferta curricular de la carrera, por ejemplo: calidad del

software, pruebas de software entre otros cursos. De las respuestas observadas, se desprende que mayoritariamente las carreras afines cuentan con asignaturas como pruebas de software y calidad de software con el 80%, El 40% con asignaturas con nombres de verificación y validación de software, y un 20% con asignaturas como ingeniería de software en la que se tratan temas de manera más general, como se visualiza en la Tabla 1.

Tabla 1. Asignaturas que se imparten afines a las pruebas de software

Asignatura que se imparten en las carreras EIS	
Asignatura	Respuestas
Pruebas de software	80%
calidad de software	80%
Ingeniería de Software	40%
Análisis y pruebas de software	40%
verificación y validación de software	40%
pruebas de software base	40%
análisis y diseño de software	20%

Fuente: Autores

Con respecto a la pregunta de investigación **RQA₃**, en la que se consultó sobre las métricas que se enseñan dentro de temáticas relacionadas a las pruebas de regresión, y si éstas se aplican en etapas de mantenimiento o durante el proceso de construcción. Al respecto la Tabla 2 indica que se enseñan las técnicas de optimización y priorización como las más relevantes, sin embargo, las métricas menos usadas que se observan son las relacionadas con la eficiencia, efectividad y precisión, indistintamente de la etapa (desarrollo o mantenimiento).

Tabla 2. Temas relacionados con pruebas de regresión de software

Temáticas impartidas	
Tema	Respuestas
Métricas (cobertura de código, capacidad de detección de errores, costos, similitudes)	100%
Enfoque de la etapa de aplicación de las pruebas de regresión de software (progresivo, correctivo)	80%

Técnicas de pruebas de regresión de software (minimización, priorización, selección y optimización)	80%
Indicadores (eficiencia, efectividad, precisión)	20%

Fuente: Autores

Con respecto a la pregunta **RQA4**, la información recabada indica que mayoritariamente los recursos educacionales que usa la academia son los libros, secundado por los artículos científicos con. Se observa también el uso de códigos de productos software existentes, como la muestra en la Tabla 3.

Tabla 3. Recursos educacionales en la enseñanza de las pruebas de regresión de software

Recursos educacionales utilizados en la enseñanza de las pruebas de regresión de software	
Recurso educacional	Respuestas
Presentaciones	100%
Capítulos de libros	100%
Artículos científicos	80%
Tutoriales	60%
Código fuente	60%
Videos	40%
Sistemas de gestión de aprendizajes	20%
MOOCs	20%
Rede sociales	20%
Talleres	20%

Fuente: Autores

Las respuestas a la pregunta **RQA5**, indica que los modelos y métodos de enseñanza más representativos por la academia están en el de aprendizaje basado en proyectos, y el modelo de la clase invertida con un 80% de aplicabilidad, secundados por los talleres y aprendizaje basado en problemas con un 60%, entre los modelos más representativos, como se observa en la Tabla 4.

Tabla 4. Modelos de enseñanza de las pruebas de regresión de software

Modelo de enseñanza de las pruebas de regresión de software	
Modelo	Respuestas
Aprendizaje basado en proyectos	80%
Clase invertida	80%
Aprendizaje basado en problemas	60%
Talleres	60%
Retos	20%
Controles de lectura	20%

Fuente: Autores

Industria

En esta sección se presentan la recopilación de resultados del cuestionario aplicado a este sector.

La información de respuesta a la pregunta RQI₁, muestra que mayoritariamente las aplicaciones de corte transaccional son las más desarrolladas en la industria ecuatoriana. Secundado por aplicaciones de inteligencia de negocios y finalmente aquellas más comunes como carritos de compra a través de la web, ver Tabla 5.

Tabla 5. Dominios de las aplicaciones en la industria

Dominio de aplicación de los productos software de la industria	
Dominio	Respuestas
Aplicaciones de procesamiento transaccional (inventarios, sistemas de recursos humanos)	44%
Aplicaciones web (carritos de compras)	38%
Aplicaciones en tiempo real (sistemas de producción)	31%
Aplicaciones de gestión de la información (inteligencia de negocios)	31%
Aplicaciones sobre dispositivos	19%
Sistemas multimedia	13%

Fuente: Autores

Con respecto al modelo de desarrollo más popular en la industria de desarrollo de software, RQI₂, se observa es el iterativo incremental con un 56%, seguido del modelo clásico con un 38%. Como se muestra en la Tabla 6.

Tabla 6. Metodologías de desarrollo que aplica la industria

Metodología de desarrollo de software	
Metodología	Respuestas
Iterativo incremental	56%
Tradicional	38%
Scrum	6%

Fuente: Autores

En referencia a la pregunta RQI₃, se observa que las pruebas de aceptación, unitarias y de integración (con 81%, 69% 56%, respectivamente) son las más utilizadas por el sector industrial, dejando en último lugar a las pruebas de regresión con un 25%, ver Tabla 7.

Tabla 7. Niveles de pruebas según metodología

Niveles de pruebas aplicados	
Nivel	Respuestas
1. Pruebas de aceptación	81%
2. Pruebas del sistema	44%
3. Pruebas de integración	56%
4. Pruebas de pruebas unitarias	69%
5. Pruebas de pruebas de regresión	25%

Fuente: Autores

Con respecto a la respuesta de la pregunta RQI₄, los planes que se llevan a cabo en el proceso de desarrollo software son: el plan de prueba de aceptación, seguido por los planes de prueba unitarios y de integración, por último, se encuentran los planes de prueba de regresión, según se observa en la Tabla 8.

Tabla 8. Plan de pruebas vs Nivel del modelo V

Planes de prueba de acuerdo con el nivel del modelo V	
Plan de prueba	Respuestas
1. Aceptación	63%
2. Sistemas	44%
3. Integración	50%
4. Unitarias	50%
5. Regresión	19%

Fuente: Autores

Las empresas de desarrollo de software del Ecuador consideran que la técnica de caja blanca es las más apropiada para su proceso de pruebas, que es la respuesta a la pregunta RQI₅, cuyo resultado se muestra en la Tabla 9.

Tabla 9. Técnicas aplicas en el proceso de pruebas

Técnicas de pruebas aplicadas	
Técnica	Respuestas
Caja Blanca	63%
Caja Negra	56%

Fuente: Autores

Las respuestas recogidas a través del cuestionario con respecto a la siguiente pregunta (RQI₆), muestran que el 63% aplica un enfoque progresivo, mientras que el 75% utilizan un enfoque correctivo, en etapa de mantenimiento.

Con respecto a las técnicas aplicadas en las pruebas de regresión que se asocian a la siguiente pregunta (RQI₇), se observa que el 19% aplica técnicas de minimización, el 81% de los encuestados dice aplicar técnicas de priorización, mientras que el 19% menciona usar técnicas de selección.

En cuanto a la pregunta RQI₈, que recopila información referente a la métrica más considerada al momento de determinar la bondad de una prueba de regresión, se observa que el tiempo de

ejecución de los casos de prueba es la métrica que es mayoritariamente considerada por la industria, según se muestra en la Tabla 10.

Tabla 10. Métricas utilizadas en las pruebas de regresión de software en la industria ecuatoriana

Métricas utilizadas por la industria en las pruebas de regresión de software	
Métrica	Respuestas
Tiempo de ejecución del caso de prueba	50%
Precisión	44%
Cobertura de código del caso de prueba	38%
Similitud de errores, defectos	25%
Costo de ejecución del caso de prueba	19%
Capacidad de detección de errores	13%

Fuente: Autores

Con respecto a la pregunta RQI₉, los encuestados mencionan que el 81% no tienen automatizado el proceso de regresión de software, mientras que el 19% mencionan sí tenerlo. Finalmente, con respecto a la RQI₁₀, sobre que, si las empresas que tienen automatizado las pruebas de regresión de software lo tienen licenciado, se indica que, el 67% usan herramientas de software con licencia mientras que el 33% utiliza software libre.

Discusión

En esta sección se discuten los resultados encontrados en relación con las preguntas de investigación planteadas.

Sector académico

Con respecto a la pregunta RQA₁ observamos que mayoritariamente el tema de pruebas de regresión se analiza en el contexto curricular de carreras afines a la Ingeniería de software. Referente a la pregunta de investigación RQA₂ la temática de las pruebas de regresión, así como la calidad se encuentra presente en algunas asignaturas principales. Con respecto a las técnicas de regresión que se enseñan, la información de la pregunta RQA₃, sugiere que la técnica de priorización, selección y optimización son las consideradas, así como las métricas orientadas a cobertura de código y eficiencia en sus planes analíticos. Así mismo, se observa el uso de libros,

artículos científicos y también código fuente de software como herramientas para aplicar el proceso de enseñanza aprendizaje (RQA₄). En cuanto al modelo de enseñanza, se menciona el basado en problemas, así como clase invertida (RQA₅). Finalmente se menciona, en la sección de observaciones del cuestionario que existe gran cantidad de técnicas y herramientas asociada al tema de pruebas de regresión de software, que constituye una limitante para analizar a profundidad la temática, por lo que se realiza de manera selectiva temas como la automatización del proceso de pruebas en general y de pruebas de regresión de manera particular.

En base a los resultados anteriores del sector académico ecuatoriano, contrasta con lo reportado por los hallazgos reportados en [12], donde se indica que las técnicas de regresión de software comúnmente investigadas son la de optimización con métricas de eficiencia en la detección de fallos versus de las de priorización y eficiencia en la ejecución de casos de prueba.

Sector Industrial

En este apartado consideramos las respuestas al cuestionario aplicado el sector industrial de software ecuatoriano. Referente a los dominios de aplicación de software (RQI₁), se obtiene que mayoritariamente las aplicaciones que se desarrollan en la industria ecuatoriana de software están en el dominio de aplicaciones de procesamiento transaccional, en un segundo lugar se considera las aplicaciones web, como, por ejemplo, el carro de compras; también se observa aplicaciones en tiempo real y aplicaciones de corte industrial. Con respecto a la metodología de desarrollo (RQI₂), que las EDS mencionan seguir es el enfoque iterativo incremental, seguido por las metodologías en cascada y Scrum.

En relación con la tercera pregunta de investigación RQI₃, la información recopilada muestra que las pruebas de aceptación, de sistema, de integración y unitarias son aplicadas, y que las pruebas de regresión son escasamente aplicadas, guardando relación los planes de pruebas que se construyen, como se describe en la pregunta de investigación RQI₄.

Como respuesta a la pregunta RQI₅ se obtiene que las EDS aplican técnicas de caja blanca y caja negra en su proceso de desarrollo. La respuesta a la pregunta RQI₆, recopila información respecto a la aplicación de las pruebas de regresión de software e indica que se aplican con un enfoque correctivo en fase de mantenimiento y con un enfoque progresivo en menor escala (durante desarrollo).

Con respecto a las técnicas que se aplican en las pruebas de regresión de software (RQI₇), las EDS informan que la técnica más aplicada es la priorización de casos de prueba, seguidamente por minimización y selección de casos de prueba, resaltando el hecho que no se ha recopilado información referente a la técnica de optimización. Por otro lado, la pregunta de investigación RQI₈, reporta que las métricas más utilizadas para determinar los casos de prueba en una prueba de regresión de software son: el tiempo de ejecución, la precisión, la cobertura de código, así como de similitud de errores y de defectos.

Finalmente, las respuestas a las preguntas de investigación RQI₉ y RQI₁₀ nos muestran la relación entre el tema de la automatización con la utilización de software licenciado y libre, e indican que las EDS mayoritariamente no tienen automatizado su proceso de pruebas de regresión, y que las que las tienen, utilizan software con licencia.

limitaciones

A continuación, se presenta cómo se minimizó alguna de las amenazas que suelen reportarse en este tipo de estudios. En primer lugar, el sesgo en la selección de los participantes es una posible amenaza por el tipo de muestreo utilizado, en total participaron 10 instituciones de educación superior (IES) de un total de 53 registradas. No obstante, el muestreo se aplicó a las IES geográficamente ubicadas en distintas regiones del país, dando la posibilidad de que la información reportada sirva como referente inicial. Por otra parte, participaron 20 empresas desarrolladoras de software de un total de 120 que están registradas en la Cámara de Innovación y Tecnología Ecuatoriana (CITEC), que de igual forma asocia a empresas del todo el país. para garantizar la confidencialidad de los participantes, los formularios fueron respondidos de manera anónima garantizando información sensible de los participantes. La validación del formulario se realizó a través de reuniones del tipo grupo focal en donde se refinaron algunas preguntas. Finalmente, el formulario diseñado y aplicado es similar a los reportados en otros estudios(Garousi et al., 2017), (Garousi & Felderer, 2017).

Conclusiones

En este trabajo, hemos reportado resultados referenciales iniciales acerca de una primera encuesta nacional sobre la enseñanza y la aplicación de las pruebas de regresión de software en la academia y en la industria ecuatoriana. los resultados determinados sirven como referente

inicial para un mejor entendimiento acerca de la efectividad de la enseñanza de las pruebas de regresión del software en la academia y su aplicabilidad en la industria del Ecuador. Estos resultados también pueden ser utilizados como soporte en la toma de decisiones para la actualización de contenidos en asignaturas afines con esta temática de estudio.

Cómo trabajo a futuro se puede indicar que los resultados de este trabajo pueden servir para la realización de un estudio longitudinal con la idea de hacer un seguimiento sobre la evolución en la enseñanza de las pruebas de regresión de software en la academia y su aplicabilidad en la industria ecuatoriana.

Referencias

1. Arcos, G., Aguirre, G. L., Hidalgo, B., Rosero, R. H., & Gómez, O. S. (2018). Current Trends of Teaching Computer Programming in Undergraduate CS Programs: A Survey from Ecuadorian Universities. *KnE Engineering*, 253-275. <https://doi.org/10.18502/keg.v1i2.1499>
2. CITEC – Cámara de Innovación y Tecnología Ecuatoriana. (s. f.). Recuperado 13 de noviembre de 2021, de <https://citec.com.ec/>
3. Ebert, C., & Paasivaara, M. (2017). Scaling Agile. *IEEE Software*, 34(6), 98-103. <https://doi.org/10.1109/MS.2017.4121226>
4. Engström, E., Runeson, P., & Skoglund, M. (2010). A systematic review on regression test selection techniques. *Information and Software Technology*, 52(1), 14-30. <https://doi.org/10.1016/j.infsof.2009.07.001>
5. Garousi, V., & Felderer, M. (2017). Worlds Apart: Industrial and Academic Focus Areas in Software Testing. *IEEE Software*, 34(5), 38-45. <https://doi.org/10.1109/MS.2017.3641116>
6. Garousi, V., Felderer, M., Kuhrmann, M., & Herkiloğlu, K. (2017). What industry wants from academia in software testing? Hearing practitioners' opinions. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 65-69. <https://doi.org/10.1145/3084226.3084264>
7. Gómez, O. S., Vegas, S., & Juristo, N. (2016). Impact of CS programs on the quality of test cases generation: An empirical study. *Proceedings of the 38th International*

- Conference on Software Engineering Companion, 374-383.
<https://doi.org/10.1145/2889160.2889190>
8. Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation. 2011 16th IEEE International Conference on Engineering of Complex Computer Systems, 305-314. <https://doi.org/10.1109/ICECCS.2011.37>
9. Khalid, A., Butt, S. A., Jamal, T., & Gochhait, S. (2020). Agile Scrum Issues at Large-Scale Distributed Projects: Scrum Project Development At Large. International Journal of Software Innovation, 8(2), 85-94. <https://doi.org/10.4018/IJSI.2020040106>
10. Khatibsyarbini, M., Isa, M. A., Jawawi, D. N. A., & Tumeng, R. (2018). Test case prioritization approaches in regression testing: A systematic literature review. Information and Software Technology, 93, 74-93.
<https://doi.org/10.1016/j.infsof.2017.08.014>
11. Kumar, M., Sharma, A., & Kumar, R. (2014). Fuzzy entropy - based framework for multi - faceted test case classification and selection: An empirical study. IET Software, 8(3), 103-112. <https://doi.org/10.1049/iet-sen.2012.0198>
12. Manifesto for Agile Software Development. (s. f.). Recuperado 13 de noviembre de 2021, de <https://agilemanifesto.org/>
13. Martin, W., Sarro, F., Jia, Y., Zhang, Y., & Harman, M. (2017). A Survey of App Store Analysis for Software Engineering. IEEE Transactions on Software Engineering, 43(9), 817-847. <https://doi.org/10.1109/TSE.2016.2630689>
14. Mina, M. A. E., & Barzola, D. del P. G. (2017). La industria del software en Ecuador: Evolución y situación actual. Espacios, 38(57), 25.
15. Mohanty, R., Ravi, V., & Patra, M. (2010). The application of intelligent and soft-computing techniques to software engineering problems: A review. Int. J. Inf. Decis. Sci. <https://doi.org/10.1504/IJIDS.2010.033450>
16. Molléri, J. S., Petersen, K., & Mendes, E. (2016). Survey Guidelines in Software Engineering: An Annotated Review. Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 1-6.
<https://doi.org/10.1145/2961111.2962619>

17. Morales, D. Á., & Aguilar Vera, R. A. (2021). Research in the area of Software Quality: A Mapping Study of the Last Decade. 2021 Mexican International Conference on Computer Science (ENC), 1-6. <https://doi.org/10.1109/ENC53357.2021.9534818>
18. Panigrahi, C. R., & Mall, R. (2010). Model-Based Regression Test Case Prioritization. En S. K. Prasad, H. M. Vin, S. Sahni, M. P. Jaiswal, & B. Thipakorn (Eds.), Information Systems, Technology and Management (pp. 380-385). Springer. https://doi.org/10.1007/978-3-642-12035-0_39
19. Parsa, S., & Khalilian, A. (2009). A Bi-objective Model Inspired Greedy Algorithm for Test Suite Minimization. En Y. Lee, T. Kim, W. Fang, & D. Ślęzak (Eds.), Future Generation Information Technology (Vol. 5899, pp. 208-215). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-10509-8_24
20. Rosero, R. H., Gómez, O. S., & Rodríguez, G. (2016). 15 Years of Software Regression Testing Techniques—A Survey. International Journal of Software Engineering and Knowledge Engineering, 26(05), 675-689. <https://doi.org/10.1142/S0218194016300013>
21. Rosero, R. H., Gomez, O. S., & Rodriguez, G. (2017). Regression Testing of Database Applications Under an Incremental Software Development Setting. IEEE Access, 5, 18419-18428. <https://doi.org/10.1109/ACCESS.2017.2749502>
22. Rosero, R. H., Gómez, O. S., & Rodríguez, G. (2017). Regression Testing of Database Applications Under an Incremental Software Development Setting. IEEE Access, 5, 18419-18428. <https://doi.org/10.1109/ACCESS.2017.2749502>
23. Rothermel, G., & Harrold, M. J. (1996). Analyzing Regression Test Selection Techniques. CSE Journal Articles. <https://digitalcommons.unl.edu/csearticles/13>
24. Rothermel, G., & Harrold, M. J. (1994). A framework for evaluating regression test selection techniques. Proceedings of 16th International Conference on Software Engineering, 201-210. <https://doi.org/10.1109/ICSE.1994.296779>
25. Rothermel, G., & Harrold, M. J. (1993). A safe, efficient algorithm for regression test selection. 1993 Conference on Software Maintenance, 358-367. <https://doi.org/10.1109/ICSM.1993.366926>

26. SENESCYT - Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación. (s. f.). Recuperado 13 de noviembre de 2021, de <https://www.senescyt.gob.ec/web/guest/consultas>
27. Taibi, D., Lenarduzzi, V., Ahmad, M. O., & Liukkunen, K. (2017). Comparing Communication Effort within the Scrum, Scrum with Kanban, XP, and Banana Development Processes. Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 258-263. <https://doi.org/10.1145/3084226.3084270>
28. Wong, W. E., Horgan, J. R., London, S., & Mathur, A. (1995). Effect of Test Set Minimization on Fault Detection Effectiveness. 1995 17th International Conference on Software Engineering. <https://doi.org/10.1145/225014.225018>
29. Yoo, S., & Harman, M. (2010). Regression testing minimization, selection and prioritization: A survey. Software Testing, Verification and Reliability, n/a-n/a. <https://doi.org/10.1002/stvr.430>