

## TESTING EN LOS TIEMPOS DE DEVOPS

**Juan Alejandro Pérez Bermúdez, Sebastián Agudelo Morales, Luz Estela Valencia Ayala**

### **Resumen**

En la actualidad el desarrollo de aplicaciones móviles o de escritorio están presentes en cada aspecto de la vida humana y empresarial. Estas aplicaciones que se pueden usar en diferentes dispositivos requieren un desarrollo rápido y confiable, que responda a las necesidades del cliente además de contar con atributos como simplicidad, seguridad y rendimiento.

Con este propósito la industria del software realiza procesos para asegurar la calidad del software en cada etapa del ciclo de vida, las pruebas de software es el proceso por el cual se evalúan procesos internos de ejecución de un programa informático en distintos niveles encontrando y solucionando defectos que se introducen durante el desarrollo. Pero en las empresas pequeñas o medianas esto se realiza de manera parcial o inadecuada, o en el peor de los escenarios, no se realiza. Esto trae como consecuencia reprocesos, pérdida de competitividad y pérdidas monetarias en la fase productiva.

Cuando se conciben y practican pruebas de software nace la necesidad de agilizar procesos repetitivos por lo que la automatización de pruebas de software ofrece la posibilidad de revisar actividades rutinarias mientras el esfuerzo de los probadores se centra en pruebas más complejas . Sin embargo, el uso de pruebas automáticas de software es naturalmente menor en el panorama general, por lo que aplicar procesos de pruebas de software sin automatización puede terminar en sobre costos y dilataciones de tiempo.

**Palabras claves:** DevOps, SCRUM, QA, Metodologías ágiles.

### **1. Planteamiento del problema**

¿Es posible para las empresas trabajar en integración y despliegue continuos con pruebas manuales?

La industria del software presenta una creciente demanda en la economía en el ámbito nacional e internacional, dado que constituye una estrategia para la transformación productiva y competitiva, para las empresas. La importancia de esta industria a nivel económico radica en el respaldo de la operatividad y estabilidad que otorga a otros sectores de la economía nacional. Pero esta demanda creciente exige alta velocidad en el desarrollo y operatividad del software, lo que ha llevado a las empresas de desarrollo de software a implementar la cultura DevOps.

Esta adopción demanda mejoras en el aseguramiento de calidad (QA). El aseguramiento de calidad es un proceso implementado en las grandes empresas de desarrollo de software para optimizar procesos de desarrollo, (Conasa, s.f.) Sin embargo, para las empresas pequeñas y medianas de Colombia estos términos siguen siendo difusos, por lo que el desarrollo software se puede dar bajo métodos artesanales o poco óptimos trayendo como consecuencias defectos a la hora de entregar un producto a un cliente o usuario. Las pruebas de software hacen parte de los protocolos de aseguramiento de software, una práctica importante en el desarrollo de sistemas informáticos independiente del nivel en el que se desempeñen; el proceso de pruebas de software consume en promedio el 30% de recursos que toma la construcción del mismo (Automatización acorta hasta en 70 % pruebas de software en empresas, 2019), sin embargo, en el momento en el que se omiten, se practican incorrectamente o se obvia su necesidad (Rojas-Montes et al.,2015), se presentan defectos que mínimamente se reflejan en costos monetarios, ya que pueden escalar a tal punto de poner en riesgo la integridad de los datos tratados o incluso la vida humana. El aseguramiento en cada etapa de desarrollo reduce la cantidad de reprocesos a futuro, además de reducir conflictos con los clientes a la hora de realizar mantenimiento y versionado.

La demanda de productos y servicios de software ha acelerado los tiempos de desarrollo de cada sprint, han pasado de semanas a días, y por ello se requiere acelerar los procesos de QA, usando metodologías mundialmente reconocidas, que validen constantemente los requisitos en tiempos más cortos y de forma automatizada.

Las pruebas manuales consumen mucho tiempo no sólo en la preparación sino en la ejecución, es la principal razón por lo que es necesario introducir herramientas de prueba que incorporen automatización para reducir los tiempos. Es importante resaltar que la depuración es crucial para

el ciclo de vida del desarrollo. ¿Es posible desarrollar una propuesta que aproveche las herramientas automatizadas que facilite a los equipos encontrar problemas potenciales con mayor eficiencia y en puntos más tempranos del ciclo de desarrollo?

## **2. Justificación**

La Actualmente, la velocidad con la que se desea realizar el despliegue de las aplicaciones es cada vez mayor debido al aumento de aplicaciones que forman parte de la vida diaria de las personas, en áreas como la salud y la movilidad. En este contexto, DevOps se presenta como una cultura más que como una tecnología, y el proceso de pruebas debe adaptarse a esta nueva "cultura". La necesidad de la automatización de las pruebas y la integración de herramientas para este propósito son fundamentales para asegurar una entrega rápida y efectiva de aplicaciones.

Al involucrar la metodología SCRUM y DevOps en el desarrollo de software, se pueden obtener beneficios y ventajas como una mayor calidad del software, una reducción en los costos y tiempos de pruebas, una mejora en la colaboración entre equipos y una mayor satisfacción del cliente. Además, el uso de la metodología DevOps permite a los desarrolladores y testers trabajar en conjunto de manera más efectiva y lograr una entrega más rápida y eficiente de aplicaciones.

## **Objetivos**

### **2.1. General**

- Desarrollar una propuesta metodológica que integre prácticas de Quality Assurance (QA) que se dan en el uso de la metodología SCRUM y su implementación en el ambiente DevOps, con el fin de garantizar una mayor eficiencia y calidad en los desarrollos de software en las empresas de la región.

### **2.2. Específicos**

- Identificar las prácticas de QA usadas en las Metodologías Ágiles como SCRUM.
- Identificar como se desempeña el QA en la metodología aplicada dentro de la cultura DevOps.
- Diseñar la propuesta metodológica.

## **3. Referente teórico**

### **3.1. Metodologías Ágiles**

Las metodologías ágiles de software representan un enfoque innovador en la gestión de proyectos de software, centrado en la toma de decisiones adaptativas y la evolución continua de requisitos y soluciones. A diferencia de los métodos tradicionales de ingeniería del software, que siguen un proceso lineal y predecible, las metodologías ágiles abrazan la incertidumbre y el cambio, empleando un proceso iterativo e incremental.

Estas metodologías se basan en ciclos cortos de trabajo, permitiendo a los equipos evaluar y ajustar el producto y sus requisitos a lo largo del proyecto, en lugar de hacerlo solo al final. Esta adaptabilidad proporciona una mayor flexibilidad para responder a las cambiantes necesidades del cliente y del mercado, lo que resulta en productos más competitivos y relevantes.

Las metodologías ágiles promueven la colaboración entre los miembros del equipo y las partes interesadas, con énfasis en la comunicación cara a cara y la retroalimentación constante. Al priorizar la entrega de valor y el aprendizaje continuo, las metodologías ágiles buscan optimizar la calidad, la eficiencia y la satisfacción del cliente en la creación de software.

### **3.2. SCRUM.**

SCRUM es un marco de trabajo ágil en el que se desarrollan piezas del software en ciclos de trabajo llamados "sprints" (Schwaber & Sutherland, 2020). Se inicia un sprint con una reunión de planificación en la que se define qué se abordará y cómo se llevará a cabo. Basándose en lo acordado, se crea una lista de tareas a realizar para cumplir con los objetivos establecidos durante la reunión. Un sprint suele durar un máximo de cuatro semanas (Cohn, 2009).

Al finalizar el sprint, se lleva a cabo una reunión de revisión para discutir los logros obtenidos y evaluar el progreso del proyecto. Posteriormente, se realiza una reunión de retroalimentación, conocida como "reunión de retrospectiva", con el objetivo de identificar áreas de mejora y ajustar el proceso de trabajo para futuros sprints (Derby & Larsen, 2006).

Los roles clave en SCRUM incluyen el Product Owner, el Scrum Master y el Equipo de Desarrollo (Schwaber & Sutherland, 2020). El Product Owner representa las necesidades del cliente y prioriza las características del producto a desarrollar. El Scrum Master actúa como facilitador y líder del equipo, asegurando que se sigan las prácticas y principios de SCRUM. El Equipo de Desarrollo trabaja de manera colaborativa para entregar incrementos de software de alta calidad en cada sprint.

Adicionalmente, SCRUM incorpora reuniones diarias conocidas como "Daily Stand-ups" o "Daily SCRUM" (Schwaber & Sutherland, 2020). Estas reuniones breves y estructuradas permiten que los miembros del equipo compartan información sobre el progreso, las dificultades encontradas y los planes para el día, ayudando a identificar rápidamente los obstáculos y mantener a todos informados sobre el estado del proyecto.

### **3.3. DevOps**

El término "DevOps" surge de la unión de las palabras "development" (desarrollo) y "operations" (operaciones), pero actualmente su definición va más allá de una simple metodología de trabajo. El objetivo de DevOps es asegurar que los equipos de desarrollo y operaciones trabajen juntos de manera colaborativa en cada etapa del proceso de desarrollo de software para entregar software de calidad con una mejor experiencia de usuario, mayor seguridad y menos errores (El concepto de DevOps, s. f.-b).

DevOps busca aumentar la flexibilidad, la adaptabilidad y la eficiencia en el ciclo de vida del software mediante la creación de un ambiente de trabajo colaborativo que incluye la participación temprana y constante del usuario en el proceso de desarrollo. La cultura DevOps es importante para el éxito del desarrollo de software de una empresa y puede ser implementada mediante la adopción de prácticas modernas de desarrollo, automatización y colaboración.

### **3.4. Quality Assurance (QA)**

En el desarrollo de software, el aseguramiento de calidad se lleva a cabo a lo largo de todas las etapas del ciclo de vida del software. Este proceso implica realizar pruebas exhaustivas e implementar controles de calidad en cada fase del desarrollo para asegurar que el software sea robusto, seguro, confiable y cumpla con los requisitos establecidos. En general, el objetivo es minimizar los errores y fallas del software para garantizar una experiencia de usuario óptima y la satisfacción del cliente (Gillis, 2019).

### **3.5. Pruebas Manuales**

Las pruebas manuales son un tipo de prueba de software en la que los analistas ejecutan casos de prueba manualmente sin herramientas o scripts de automatización. Estas pruebas verifican el diseño, la ortografía, la funcionalidad y el rendimiento de la interfaz del sistema, y se realizan antes de pasar a la automatización (Tipos de pruebas y dónde aplicarlas, 2020).

### 3.6. Pruebas Automáticas

Las pruebas automatizadas no requieren intervención humana en su ejecución y se centran en funcionalidades críticas del sistema (Tipos de pruebas y dónde aplicarlas, 2020). Estas pruebas se ejecutan varias veces para temas de regresión y generan reportes fáciles de entender. Algunas herramientas populares incluyen Selenium, Serenity, Cucumber, Katalon y Appium Studio. Los patrones conocidos abarcan Automation Native, PageObjectModel (POM) y Serenity BDD.

Las pruebas automatizadas son apropiadas para aplicaciones con una cantidad considerable de flujos críticos y repetitivos, elementos con identificadores únicos y exactos, y cuando la automatización se ejecutará en el futuro. No se recomienda automatizar sistemas centrados únicamente en contenido o diseño.

### 3.7. ISTQB

Establecido en 1998, el International Software Testing Qualifications Board (ISTQB) se ha consolidado como el esquema de certificación en pruebas de software más reconocido a nivel mundial. Esta organización sin fines de lucro desempeña un papel fundamental al establecer estándares y directrices en el ámbito de las pruebas y el aseguramiento de la calidad del software (ISTQB, 2022).

La certificación ISTQB se fundamenta en un marco que abarca distintos niveles y módulos, desde los fundamentos del testing hasta niveles avanzados y expertos. Existen diferentes flujos que contienen módulos específicos. Por ejemplo, el flujo "Core" incluye módulos enfocados en tecnologías, metodologías y aplicaciones, basados en los fundamentos del testing. Por otro lado, el flujo "Agile" se centra en metodologías ágiles, mientras que el flujo "Specialist" ofrece un enfoque más profundo en áreas específicas de la calidad del software (ISTQB, 2022).

## 4. Metodología

### 4.1. Tipo de investigación

Para el desarrollo del proyecto se hará uso de la Investigación aplicada.

### 4.2. Enfoque

- Identificar las Prácticas de Q.A. usadas en las Metodologías Ágiles como SCRUM.

Este estudio busca caracterizar regionalmente a empresas desarrolladoras de software y analizar sus prácticas relacionadas con QA. Para lograrlo, se recopilarán datos de diversas empresas de la región y se identificarán los factores que influyen en la calidad del software, con el objetivo de mejorarla.

- Identificar cuál es la metodología aplicada dentro de la cultura DevOps.

En DevOps, las herramientas de QA se utilizan en cada una de las fases del ciclo de desarrollo, que incluyen planificación, codificación, integración, implementación y monitoreo. En la fase de planificación, las herramientas de QA se enfocan en la gestión del proyecto y los requisitos. En la fase de codificación, se enfocan en garantizar que el código se adhiera a los estándares de codificación. En la fase de integración, se enfocan en la integración correcta del software. En la fase de implementación, se enfocan en la implementación adecuada y la prueba del software. En la fase de monitoreo, se enfocan en la identificación y solución de problemas.

Contemplando lo mencionado anteriormente, se realizará un comparativo de las herramientas de QA más usadas en cada fase.

- Diseñar la propuesta metodológica.

Se tiene esperado recopilar datos de las empresas de la región para obtener información de cómo son sus metodologías, procesos, herramientas y capacidad en talento humano relacionado a QA, una vez recopilados todos los datos previamente descritos, el proyecto se consolidará en el diseño de una propuesta que permita el cumplimiento de una buena calidad del software en sus desarrollos siguiendo los pasos que recomiendan seguir SCRUM y DevOps.

Se tiene como objetivo recopilar información de las empresas de la región sobre sus metodologías, procesos, herramientas y capacidad en talento humano de QA, para posteriormente diseñar una propuesta que permita mejorar la calidad del software en sus desarrollos, siguiendo los pasos de SCRUM y DevOps. Para ello, se llevará a cabo un proceso de recolección de datos y se analizarán las prácticas en esta área.

La propuesta se centrará en la mejora continua de la calidad del software y su implementación a largo plazo en las empresas de la región, lo que permitirá una mayor eficiencia en su gestión y un mejor aprovechamiento de sus recursos en el desarrollo de proyectos de software.

## **5. Resultados esperados**

### **5.1. Prácticas de QA en SCRUM**

En el caso de QA en la metodología SCRUM este se tiene en cuenta desde el comienzo del proyecto con la función de entender bien lo que quiere el cliente, su negocio, con esto se conoce al tipo usuario final del proyecto para así hacer diseñar y ejecutar las pruebas de manera más adecuada, en esta etapa también se hacen sugerencias y se alinea el equipo de desarrollo con el proyecto.

Cuando se realizan las reuniones de planificación y se decide lo que se va a realizar los QA ya tienen las historias de usuario necesarias para la creación de los casos de prueba que se desarrollaran en ese sprint, en caso de SCRUM se maneja la integración continua por lo cual se tiene que revisar constantemente el software haciendo uso de pruebas de regresión ya que el tamaño de estas pruebas crece tras cada iteración se deben automatizar.

En el caso de las pruebas unitarias se crean un conjunto de pruebas automatizadas y se establece que lo desarrollado debe pasar por todo este conjunto de pruebas para que sea aceptado. En SCRUM no se define como se deben desarrollar las pruebas o que se debe tener un equipo de QA separado si no que puede ser el mismo equipo de desarrollo que realizan las pruebas (James, 2018).

### **5.2. Prácticas de QA en DevOps**

En otras metodologías ágiles, el desarrollador se encarga de codificar y crear funcionalidades, así como de escribir y ejecutar pruebas unitarias, mientras que los testers llevan a cabo pruebas automatizadas y manuales en dicha funcionalidad. En contraste, DevOps difumina intencionalmente las fronteras entre estas responsabilidades y fomenta la colaboración. Los desarrolladores adoptan una mentalidad de verificación constante de errores en su código, mientras que los testers expanden sus responsabilidades más allá de la validación de la aplicación, asegurando que esté siempre desplegable e incluso corrigiendo el código cuando sea necesario.



Todas estas partes trabajan en conjunto para garantizar una entrega rápida de características, y un enfoque centrado en el cliente impulsa el trabajo de todo el equipo.

En DevOps, la perspectiva de las pruebas cambia de "pruebas completadas en este módulo" a "¿Qué riesgos empresariales se han mitigado en la versión candidata a lanzamiento?". La metodología se centra en despliegues escalables, la automatización desde el inicio del proceso de desarrollo, la estandarización de entornos de control de calidad, la alineación de tareas de control de calidad con pruebas continuas y tuberías de CI/CD.

Un alto nivel de colaboración entre desarrolladores, testers y el equipo de operaciones permite al personal de testing monitorear la aplicación en producción y utilizar los datos para ayudar a los interesados a tomar decisiones informadas (Gurock, 2022).

Las pruebas automatizadas son un componente esencial del proceso de desarrollo de software moderno. En una cultura como DevOps, donde la entrega rápida y continua de nuevas características y actualizaciones es la norma, las pruebas manuales pueden resultar lentas y no siempre son adecuadas para detectar todos los errores. Por otro lado, las pruebas automatizadas pueden diseñarse y ejecutarse en gran cantidad y de manera repetitiva, lo que las hace más efectivas para encontrar errores y garantizar la calidad del software.

Además, en DevOps, la automatización es un aspecto clave para asegurar que el proceso de desarrollo, pruebas y despliegue sea rápido y escalable. También existen otros factores por los cuales es beneficioso automatizar, como: reducción de errores humanos, ahorro de tiempo, mayor cobertura de pruebas, detección temprana de errores y escalabilidad. La implementación de estas prácticas ayuda a mejorar la calidad del software y a satisfacer las necesidades del cliente de manera más eficiente y efectiva.

### **5.3. Propuesta Metodológica**

Tanto en SCRUM como en DevOps, el enfoque en la automatización es crucial para aumentar la eficiencia y reducir el tiempo y los errores en el ciclo de vida del desarrollo de software. Las herramientas de automatización pueden ayudar a los equipos a integrar, implementar, probar y

monitorear continuamente el software, lo que permite una entrega rápida y confiable de productos de alta calidad. Por lo tanto, es fundamental que la propuesta resultante se enfoque en las herramientas de automatización y el cómo utilizarlas de manera efectiva para obtener los mejores resultados.

## BIBLIOGRAFIA

- ACADEMIA *pragma*. (23 de Noviembre de 2020). Recuperado el 18 de Marzo de 2023, de <https://www.pragma.com.co/academia/lecciones/tipos-de-pruebas-y-donde-aplicarlas>
- Conasa. (s.f.). *Conasa*. Obtenido de <https://conasa.grupocibernos.com/blog/la-importancia-del-cmmi-en-el-desarrollo-de-software>
- EL TIEMPO. (22 de Agosto de 2019). Obtenido de <https://www.eltiempo.com/economia/sectores/empresas-pueden-ahorrar-hasta-70-en-pruebas-de-software-si-las-automatizan-403514>
- Gillis, A. (2019). *Software Quality*. Recuperado el 18 de Marzo de 2023, de <https://www.techtarget.com/searchsoftwarequality/definition/quality-assurance>
- Gonzáles Castro, A. F., & Narváez Arteaga, J. P. (2023). Caracterización De Empresas De Software En Risaralda. Pereira.
- ISTQB, n.-f.-p. a. (2022). *ISTQB*. Recuperado el 18 de Marzo de 2023, de <https://www.istqb.org/about-us/what-we-do>
- James, A. (4 de Noviembre de 2018). *ISTQB - Certified Tester Foundation Level Extension Syllabus / Agile Tester Foundation Level: Independently certified training material (English Edition)*.
- Cohn, M. (2009). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional.
- Derby, E., & Larsen, D. (2006). *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org. Recuperado de <https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- El concepto de DevOps*. (s. f.-b). <https://www.redhat.com/es/topics/devops>
- C. (s. f.). La importancia del CMMI en el desarrollo de software. <https://conasa.grupocibernos.com/blog/la-importancia-del-cmmi-en-el-desarrollo-de-software>

- Álvarez, I., Martínez, J., Pérez, L. D., Figueroa, F., De Armas Mestre, J., & Llop, M. (2017). Contaminación ambiental por ruido. *Revista Médica Electrónica*, 39(3), 640-649.  
[http://www.revmedicaelectronica.sld.cu/index.php/rme/article/download/2305/pdf\\_269](http://www.revmedicaelectronica.sld.cu/index.php/rme/article/download/2305/pdf_269)
- THERAC-25. (2012, 20 mayo). fracasosdesoftware.  
<https://fracasosdesoftware.wordpress.com/2012/05/18/therac-25/>
- Son, H. (2022). *What is the Role of QA in DevOps?* TestRail Blog. <https://blog.gurock.com/qa-devops-role/>