Propuesta de Modelo Seguro en Etapas Tempranas del Desarrollo de Software

Malena Páez [1] [0009-0004-1479-4224], Flavia Millan[1] [0009-0004-7457-3837] Laura N. 1Aballay[2] [0000-0001-5462-7683], Alex A. Torres[3] [0000-0003-3830-3982]

[1] Departamento de Informática, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de San Juan, Argentina

[2] Instituto de Informática, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de San Juan, Argentina

[3] Corporación Universitaria Comfacauca
UNICOMFACAUCA, Popayán, Colombia
malenapaez27@gmail.com, flavia.millan@gmail.com
lnaballay@gmail.com,atorres@unicomfacauca.edu.co

Abstract. La presente propuesta de trabajo final de la Licenciatura en Ciencias de la Computación, involucra un proceso de revisión de autores, trabajos y bibliografía referida al abordaje de una metodología para el desarrollo seguro de software. Es una imperiosa necesidad de las organizaciones el querer diseñar aplicaciones o servicios innovadores para resolver problemas de negocio. Por otro lado, éstas empresas u organizaciones, anhelan que sus desarrollos, sean productos seguros, por ende, desean que el proceso de desarrollo cumpla con estándares de seguridad. Ahora bien, el desafío consiste en poder llevar adelante estos deseos y, además, aplicar metodologías ágiles en el desarrollo. Las mismas, son las empleadas en la actualidad por las empresas desarrolladoras. Lo que implica implementar una metodología de desarrollo de software seguro desde el inicio del proceso, enfocada principalmente en la gestión de los recursos, la que se caracteriza por ayudar a identificar problemas de seguridad al comienzo del proceso de desarrollo y no después de lanzado el producto de software. Esto es con la finalidad de reducir costos y aumentar la velocidad de recuperación frente a un incidente de seguridad, entre otros. El objetivo de este trabajo es abordar la integración de metodología, herramientas y prácticas, en las etapas tempranas del desarrollo de software, de manera que la seguridad sea incorporada desde el comienzo, para lograr un producto dirigido al usuario lo más seguro, confiable y en el menor tiempo posible.

Keywords: Proceso de Desarrollo de Software, Desarrollo Seguro de Software.

1 Justificación

Las empresas desarrolladoras de software enfrentan problemas referidos a la rapidez en la entrega, la confiabilidad y la seguridad de sus productos. Son muchas las empresas que tienen complicaciones con los proyectos de software, y sus fracasos suelen estar relacionados con problemas en el desarrollo.[8]. Además, los temas

relacionados con seguridad, se abordan en una etapa demasiado avanzada del proceso: la de pruebas, después de haber completado las tareas más importantes de diseño e implementación. En muchos casos, los controles de seguridad que se ejecutan en esa etapa son muy leves y superficiales, es decir, se limitan al análisis y las pruebas de intrusión. Por eso, es posible, que se pasen por alto problemas de seguridad más complejos que, de detectarse, podrían retrasar la llegada del sistema a la producción. Además, la resolución de los problemas de seguridad en etapas avanzadas del diseño, lleva mucho tiempo y es más costosa, ya que puede requerir que se vuelva a desarrollar y probar todo el software y en ciertas oportunidades, revisar desde los requerimientos del usuario. El aplicar este tipo de prácticas, herramientas y metodología en las etapas de desarrollo de software, puede implicar [1]:

- Mejorar la seguridad y disminuir los riesgos gracias a la eliminación de más puntos vulnerables desde el inicio del ciclo de vida de la infraestructura y el desarrollo de las aplicaciones, lo cual reduce los problemas potenciales en la etapa de producción.
- Aumentar la eficiencia y la velocidad de los ciclos de lanzamiento de DevOps con la eliminación de las prácticas y las herramientas de seguridad heredadas. El uso de la automatización; la adopción de una cadena de herramientas de manera estandarizada; y la implementación de la infraestructura, la seguridad y el cumplimiento normativo como código para mejorar la capacidad de repetición y la uniformidad pueden optimizar el proceso de desarrollo.
- Disminuir los riesgos y aportar claridad mediante la implementación de controles de seguridad desde el comienzo del ciclo de vida de la infraestructura y el desarrollo de las aplicaciones, lo cual reduce la probabilidad de que se cometan errores humanos y mejora la seguridad, el cumplimiento normativo y la capacidad para anticipar los inconvenientes y repetir los procesos que permitan solucionarlos, y disminuye los problemas de auditoría.

Otra problemática, es desconocer el procedimiento de implementar metodologías ágiles, además con la condición de que sea un software seguro y confiable y ponerlo a disposición del usuario en el menor tiempo posible. Por lo detallado anteriormente, es que propone este Trabajo Final, donde se logre un abordaje integral para conocer y orientar en la implementación de metodologías ágiles en el desarrollo de software, sin descuidar la seguridad de este producto desde los inicios de su desarrollo.

2 Marco Teórico

Actualmente las metodologías de ingeniería de software pueden considerarse como una base necesaria para la ejecución de cualquier proyecto de desarrollo de software que se considere serio, y que necesite sustentarse en algo más que la experiencia y capacidades de sus programadores y equipo. Estas metodologías son necesarias para poder realizar un proyecto profesional, tanto para poder desarrollar efectiva y eficientemente el software, como para que sirvan de documentación y se puedan rendir cuentas de los resultados obtenidos [2].

Frente a las metodologías tradicionales en el desarrollo de software, se presentan las metodologías ágiles. Este enfoque nace como respuesta a los problemas que puedan ocasionar las metodologías tradicionales y se basa en dos aspectos fundamentales, retrasar las decisiones y la planificación adaptativa. Basan su fundamento en la adaptabilidad de los procesos de desarrollo. Un modelo de desarrollo ágil, generalmente es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento) [2].

Los términos DevOps y Agile forman parte del mundo del desarrollo de software, lo que hace que a menudo las personas que acaban de empezar en el sector los confundan. A pesar de que ambos se refieren a metodologías de desarrollo de software, son dos conceptos distintos que acaban confluyendo a la hora de ponerlos en práctica de manera conjunta.

Por un lado, DevOps se refiere a la metodología que ha conseguido unir los equipos de desarrollo con los de sistemas, que históricamente han estado divididos por una línea infranqueable que impedía tener una comunicación fluida entre ellos.

Por otro lado, Agile consiste más bien en una filosofía de trabajo en la que se apoyan los programadores de aplicaciones. Tras definir el manifiesto Agile en 2001, se crearon varias metodologías de trabajo como Scrum o Kanban que parten de la idea de este. Estas metodologías ágiles permiten desarrollar software de una manera más rápida y productiva, por lo que el producto final será mucho mejor. [6]

Ambas herramientas pueden confluir e incluso trascender al desarrollo de cualquier proyecto de software, producto o servicio, y aunque son fundamentalmente distintas, también tienen sus similitudes. Además de que DevOps se nutre de las metodologías ágiles, ambas tienen objetivos similares: acelerar los procesos de desarrollo y distribución de software. La adopción de DevOps extiende los beneficios de Agile más allá del equipo de desarrollo. Adaptarse al ritmo de trabajo de los desarrolladores y trabajar en fragmentos más pequeños facilita la detección y el aislamiento de los problemas [6].

El desarrollo de software se vuelve cada vez más democratizado y descentralizado, lo que facilita en gran medida las labores del desarrollador y la distribución de conocimiento en la comunidad de software. Pero pocos se han detenido a pensar que a medida que aumenta la cantidad de desarrollos de software y, también lo hacen las vulnerabilidades de seguridad [3]. Por tal situación, se presenta DevSecOps como una filosofía que se basa en la integración de mecanismos y elementos de seguridad desde el principio del proceso DevOps. DevSecOps es la abreviatura de «Development, Security and Operations» (Desarrollo, Seguridad y Operaciones, en español). El modelo DevSecOps integra la capa de seguridad desde el principio dentro de la metodología DevOps. Establece que tanto los desarrolladores como el personal de operaciones debe ocuparse de la seguridad desde el primer día. Porque, además de los beneficios de seguir la metodología DevOps, al tener en cuenta la seguridad en cada etapa del desarrollo se puede [4]:

- Reducir la cantidad de revisiones y correcciones de la aplicación una vez lanzada.

- Evitar fallos y errores.
- Mejorar la rentabilidad a la hora de enfrentarse a problemas de seguridad.
- Aumentar la fidelidad y la satisfacción de los clientes.

La metodología DevSecOps, como DevOps, se apoya considerablemente en la automatización. De modo que, todos los procesos de seguridad que se puedan automatizar, esto hace que las entregas de los productos de software, sean mucho más rápidas y confiables. DevSecOps implica pensar desde el inicio en la seguridad de las aplicaciones desarrolladas y de la infraestructura, y también en la automatización de aquellos elementos de seguridad que pueden impedir que se ralentice el flujo de trabajo (workflow) de DevOps. A fin de cumplir con estos objetivos, es necesario seleccionar las herramientas adecuadas para integrar la seguridad de manera permanente [5]. Esta integración no sólo requiere emplear nuevas herramientas sino implementar un enfoque organizativo distinto, incorporando la seguridad como un componente más del desarrollo, en lugar de algo exclusivo del departamento de seguridad [3]. El principal beneficio de DevSecOps es la creación de un producto final más seguro, en lugar de un obstáculo final que se deba superar. No solo garantiza que el producto sea sólido y ágil, sino también seguro y compatible [7].

3 Propuesta

Se utilizará la metodología de trabajo de tipo cuantitativa aplicando los conceptos del método experimental. Esta metodología permite: la sistematización de la investigación; elaborar un plan que servirá de guía durante el desarrollo del trabajo; conlleva a mantener una secuencia en el estudio; revisar la bibliografía; realizar mediciones y predicciones exactas. Para ello, se utilizará una revisión sistemática (artículos científicos, revistas indexadas, libros de metodología de la investigación de editoriales internacionales reconocidas, sitios web científicos y/o reconocidos por la comunidad informática, de la temática abordada) como técnica exploratoria y analítica para la recolección de información relevante sobre los principios de la metodología ágil, las herramientas y prácticas utilizadas más actuales en la incorporación de la seguridad, en el proceso de desarrollo, para lograr un producto seguro, confiable y disponible al usuario en el menor tiempo.
El desarrollo de este trabajo de licenciatura, tendrá las siguientes fases principales:

1. Estudiar las metodologías ágiles y sus principios, aplicados a proyectos de desarrollo de software.
2. Investigar sobre las herramientas y técnicas de la filosofía de DevSecOps.
3. Analizar las actividades requeridas para brindar seguridad al proceso de desarrollo de software desde las etapas tempranas del proceso.
4. Determinar los pasos para implementar una metodología de desarrollo seguro de software.
5. Redactar conclusiones y trabajos futuros.
6. Documentación y armado del trabajo final.

4 Resultados Esperados

Algunos de los resultados esperados de este trabajo son:

- Integrar al proceso de desarrollo de software: metodología, herramientas y prácticas, en las etapas tempranas del desarrollo de software, de manera que la seguridad sea incorporada desde el comienzo, y conseguir un producto de software, dirigido al usuario lo más seguro y confiable posible.
- Gestionar adecuadamente los recursos: tiempo y dinero. Esto se caracteriza por ayudar a identificar problemas de seguridad al comienzo del proceso de desarrollo y no después de lanzado el producto. Esto es con la finalidad de reducir costos asociados con la corrección de vulnerabilidades en etapas posteriores.
- Detectar desde etapas tempranas del proceso de desarrollo, las vulnerabilidades o posibles problemas de seguridad del producto.
- Aportar al desarrollo seguro de software de calidad, un conjunto de estrategias, herramientas y prácticas para obtener desarrollos seguros y robustos.
- Fomentar la incorporación de la cultura de la seguridad de software a todos los integrantes del proceso de desarrollo de software y durante todo el proceso, no sólo en las últimas etapas.

Agradecimientos

Este trabajo es un resultado intermedio y ha sido financiado por el proyecto de investigación titulado **“Propuesta de Evaluación de Experiencia de Usuario en Sistemas Interactivos usando reconocimiento de emociones”** aprobado en la convocatoria interna de proyectos, con código VRIE2024-04G, por la Corporación Universitaria Comfacauca - **UNICOMFACAUCA.**

Referencias

1. RedHat. “Seguridad en el ciclo de vida de desarrollo del software”. Motivos por los que conviene elegir Red Hat para DevSecOps. Disponible en: https://www.redhat.com/es/topics/security/software-development-lifecycle-security . 2022.
2. Maida, EG, Pacienzia, J. “Metodologías de desarrollo de software”. Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería “Fray Rogelio Bacon”. Universidad Católica Argentina. Disponible en: http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf. 2015
3. C. Correa. “DevSecOps: seguridad en el desarrollo de Software”. Pragma. Desarrollo de software. Disponible en: https://www.pragma.co/es/blog/devsecops-seguridad-en-el-desarrollo-de-software .2020.
4. Stackscale. “DevOps y DevSecOps: desarrollo de software Agile”. *Grupo Aire*. Disponible en: https://www.stackscale.com/es/blog/devops-devsecops/ .2023.
5. RedHat. “¿Qué es DevSecOps?” Disponible en: https://www.redhat.com/es/topics/devops/what-is-devsecops. 2023.

6.JetBrains. “¿Qué es CI/CD en DevOps ?”. Cómo Agile cambió las reglas del juego. Disponible en: https://www.jetbrains.com/es-es/teamcity/ci-cd-guide/devops-ci-cd/.
7.H. Bell. “DevSecOps: Integración de la seguridad en el ciclo de vida de DevOps”. DevOps.Com. Group Techstrong. Disponible en: https://devops.com/devsecops-integrating-security-into-the-devops-lifecycle/. 2024.
8 Sanjeev Sharma, & Bernie Coyne. (2015). *DevOps para Dummies* (Inc. John Wiley & Sons, Ed.; Limitada, Vol. 2).