Revista Peruana de Computación y Sistemas 2024 6(1):47-60
https://doi.org/10.15381/rpcs.v6i1.27402

ISSN Online: 2617-2003
Facultad de Ingeniería de Sistemas e Informática
Universidad Nacional Mayor de San Marcos

Artículo de Contribución

Un Marco de Trabajo para el Desarrollo de Software Web Seguro con Metodologías Ágiles

A Framework for the Development of Secure Web Software with Agile Methodologies

Gilmer Glicerio Valderrama Herrera[1,a], José Alfredo Herrera Quispe[2,b]

[1] Universidad Nacional Mayor de Marcos, Facultad de Ingeniería de Sistemas e Informática. Unidad de Posgrado.Lima, Perú

[2] Universidad Nacional Mayor de Marcos, Facultad de Ingeniería de Sistemas e Informática. Lima, Perú

[a] Autor de correspondencia: gilmer.valderrama@unmsm.edu.pe, ORCID: https://orcid.org/0009-0002-7065-9081

[b] E-mail: jherreraqu@unmsm.edu.pe, ORCID: https://orcid.org/0000-0002-8207-9714

Resumen

El desarrollo de software bajo el enfoque ágil con SCRUM se presenta como una alternativa a las metodologías tradicionales, ofreciendo características incrementales, iterativas y de entrega continua de código que aseguran un producto adaptable a los cambios. No obstante, este enfoque enfrenta desafíos en términos de desarrollo seguro, ya que las amenazas y vulnerabilidades pueden no ser abordadas adecuadamente debido a plazos ajustados y cambios frecuentes en los requisitos. En este contexto, se propone un marco de trabajo para el desarrollo de software web seguro utilizando Metodologías Ágiles (SCRUM). Basado en la literatura, para este propósito, se han seleccionado ocho actividades de seguridad que se integran durante el proceso de desarrollo, considerando su grado de agilidad y costo-beneficio. La validez de esta propuesta se confirmó mediante un instrumento que recopiló la opinión de 45 expertos en la industria de desarrollo de software en Perú.

Palabras clave: Software Seguro, Desarrollo Seguro, Desarrollo Agile Seguro.

Abstract

Software development using the agile approach with SCRUM presents itself as an alternative to traditional methodologies, offering incremental, iterative features and continuous code delivery that ensure a product adaptable to changes. However, this approach faces challenges in terms of secure development, as threats and vulnerabilities may not be adequately addressed due to tight deadlines and frequent changes in requirements. In this context, a framework for secure web software development using Agile Methodologies (SCRUM) is proposed. Based on the literature, eight security activities have been selected for this purpose and integrated into the development process, considering their level of agility and cost-benefit. The validity of this proposal was confirmed through an instrument that gathered the opinions of 45 experts in the software development industry in Peru.

Keywords: Secure software, Secure Development, Secure Agile Development.

Recibido: 15/02/2024 - Aceptado: 24/06/2024 - Publicado: 30/06/2024

Citar como:
Valderrama Herrera, G. & Herrera Quispe, J. (2024). Un Marco de Trabajo para el Desarrollo de Software Web Seguro con Metodologías Ágiles. Revista Peruana de Computación y Sistemas, 6(1):47-60. https://doi.org/10.15381/rpcs.v6i1.27402

1. Introducción

La mayoría de las organizaciones gestionan sus operaciones a través de internet (aplicaciones web, servicios web y aplicaciones móviles). Un pequeño fallo de seguridad en el software puede ser explotado, comprometiendo información confidencial y afectando tanto las operaciones técnicas como la imagen, la reputación, y los aspectos económicos y financieros de la organización.

En lo que va del año 2023 [28], los ataques dirigidos al software web han aumentado en comparación con el año anterior, poniendo en riesgo las operaciones e imagen de las organizaciones, según el reporte de vulnerabilidades y exposiciones comunes (CVE) del sitio web CVEDETAILS (ver Fig. 1 y Fig. 2).

2. Marco Teórico

2.1. Software Seguro

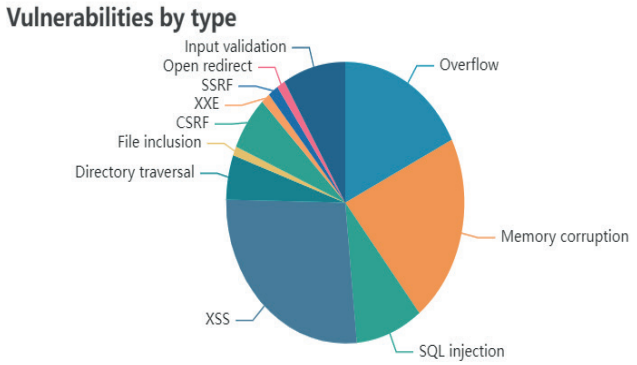
Según [1], el software seguro es aquel diseñado, construido y mantenido para proteger la integridad, confidencialidad y disponibilidad de la información que procesa. Este tipo de software integra consideraciones de seguridad en todas las fases de su ciclo de vida, desde el diseño y desarrollo hasta la implementación y el mantenimiento. Su enfoque está en prevenir, detectar y responder a amenazas y vulnerabilidades, minimizando así los riesgos de seguridad para los usuarios y la infraestructura asociada.

Figura 1

Reporte de ataques a software web, 2023 [28].

Figura 2

Reporte de tipos de ataques a software web más frecuentes, 2023 [28]



2.2. Ingeniería de Seguridad

La Ingeniería de Seguridad en el desarrollo de software, según el modelo SSE-CMM (Systems Security Engineering Capability Maturity Model), es una disciplina que integra principios y prácticas de seguridad en todas las fases del desarrollo de software. Su objetivo es construir sistemas de información y software inherentemente seguros. Esto incluye la identificación y gestión de requisitos de seguridad, el diseño de arquitecturas robustas, la implementación de controles de seguridad y la realización de pruebas y validaciones de seguridad a lo largo del ciclo de vida del desarrollo de software [2].

2.3. Amenazas de seguridad en software web

Con la aparición de aplicaciones web, plataformas de redes sociales, computación en la nube e Internet de las cosas (IoT), la cantidad de vulnerabilidades de software está creciendo, aumentando así el riesgo al que se expone el software web. Por lo tanto, la seguridad se ha convertido en un gran desafío para las organizaciones que desarrollan software [3]. En este contexto, organismos como OWASP (Open Web Application Security Project), una organización sin fines de lucro, trabajan para mejorar la seguridad del software. A través de proyectos de software de código abierto liderados por la comunidad y cientos de capítulos locales en todo el mundo, OWASP proporciona documentación, herramientas y metodologías libres y de código abierto relacionadas con el desarrollo seguro y la seguridad en aplicaciones web. Uno de estos recursos es el proyecto OWASP Top 10, que publica y difunde las 10 amenazas más comunes:

- A01:2021 - Pérdida de control de acceso.
- A02:2021 - Fallas criptográficas.
- A03:2021 – Inyección.
- A04:2021 - Diseño inseguro.
- A05:2021 - Configuración de seguridad incorrecta.
- A06:2021 - Componentes vulnerables y desactualizados.
- A07:2021 - Fallas de identificación y autenticación.
- A08:2021 - Fallas en el software y en la integridad de los datos.
- A09:2021 - Fallas en el registro y monitoreo.
- A10:2021 - Falsificación de solicitudes del lado del servidor.

2.4. Actividades de seguridad en el desarrollo de software

Las actividades de seguridad en el desarrollo de software incluyen prácticas y procesos destinados a garantizar la creación de sistemas informáticos seguras, entre estas actividades tenemos:

2.4.1. Educación y sensibilización en materia de seguridad: La educación y sensibilización en materia de seguridad en el desarrollo seguro de software implican capacitar y concienciar a los desarrolladores y a todos los involucrados en el ciclo de vida del software sobre la importancia de integrar prácticas de seguridad desde las primeras fases de desarrollo.

2.4.2. Identificación de Roles: Este proceso asegura que el acceso al sistema sea otorgado de acuerdo con el principio de mínimo privilegio, limitando la exposición a ataques y reduciendo el riesgo de abuso de acceso interno. La correcta asignación de roles es crítica para la seguridad del software, ya que facilita la auditoría, el seguimiento y la gestión de acceso a los recursos del sistema [4].

2.4.3. Análisis de riesgos de seguridad (Identificación de ataques): La actividad de seguridad análisis de riesgos de seguridad o identificación de ataques es fundamental para el ciclo de vida del desarrollo de software, ya que permite a los desarrolladores implementar medidas de seguridad robustas durante las fases de diseño y codificación. Su objetivo es minimizar los riesgos de seguridad y proteger las aplicaciones de posibles ataques malintencionados [5].

2.4.4. Análisis de requisitos de seguridad: Esta actividad asegura que los controles de seguridad se integren eficazmente en el diseño del sistema, estableciendo un marco para el desarrollo de software que sea seguro por diseño y cumpla con las expectativas de los interesados en cuanto a seguridad se refiere [2].

2.4.5. Modelado de amenazas: Este proceso proactivo ayuda a los equipos a entender y mitigar riesgos antes de que el software sea desplegado, asegurando que las contramedidas (requisitos) de seguridad se integren eficazmente en la arquitectura y diseño del sistema [6].

2.4.6. Análisis de código estático: Se realiza en las etapas tempranas del ciclo de vida del desarrollo del software, permitiendo a los desarrolladores corregir problemas antes de que el software avance a producción. Se puede hacer uso de herramientas de análisis de código estático para identificar patrones de código problemáticos, flujos de datos inseguros y otros riesgos de seguridad que, si no se corrigen, podrían ser explotados por atacantes [7].

2.4.7. Pruebas de penetración

Las pruebas de penetración proporcionan información crítica sobre deficiencias reales en la seguridad y la efectividad de los controles existentes, permitiendo a los equipos de desarrollo y seguridad fortalecer la postura de seguridad de sus aplicaciones [8].

2.4.8. Planificación de respuesta a incidentes

Esta actividad implica la preparación, identificación, contención, erradicación y recuperación ante incidentes de seguridad. Un plan de respuesta a incidentes bien desarrollado permite a las organizaciones actuar rápidamente y eficientemente para minimizar los daños, restaurar las operaciones y comunicar de manera efectiva con todas las partes interesadas [9].

2.5. Modelos de categorización y clasificación de amenazas

2.5.1. STRIDE

Es un framework de modelado de amenazas desarrollado por Microsoft para identificar y categorizar amenazas de seguridad en sistemas de software. El acrónimo STRIDE representa los siguientes tipos de amenazas: Suplantación de identidad (Spoofing), Manipulación (Tampering), Repudio (Repudiation), Información (Information Disclosure), Denegación de servicio (Denial of Service) y Elevación de privilegios (Elevation of Privilege), [10].

2.5.2. DREAD

[11] DREAD es un framework de clasificación de riesgos utilizado para priorizar las amenazas y vulnerabilidades en la seguridad de la información y el desarrollo de software. El acrónimo DREAD representa cinco categorías: Daño (Damage potential), Reproducibilidad (Reproducibility), Explotabilidad (Exploitability), Afectados (Affected users), y Descubrimiento (Discoverability).

2.5.3. Modelo ágil de gestión de riesgo

La gestión de riesgos ágil se caracteriza por su enfoque colaborativo, priorización basada en la exposición al riesgo, y la promoción de la conciencia de seguridad en todos los niveles del proyecto. Este enfoque busca equilibrar la flexibilidad del desarrollo ágil con la necesidad de abordar las preocupaciones de seguridad de manera efectiva y oportuna [12].

2.6. Historias de abuso

Las historias de abuso en el desarrollo seguro de software son una técnica utilizada para identificar y prevenir posibles ataques malintencionados en aplicaciones y sistemas. A diferencia de las historias de usuario tradicionales, que se centran en las necesidades legítimas del usuario, las historias de abuso se enfocan en cómo un usuario malicioso podría explotar o atacar el sistema [13]. Por ejemplo, una historia de abuso podría describir cómo un atacante podría llevar a cabo una inyección SQL para acceder a la base de datos de un sistema informático web, o cómo podrían explotar una configuración débil de autenticación para obtener acceso no autorizado.

2.7. Historias de usuario relacionadas a la seguridad

Las historias de usuario relacionadas con la seguridad en el desarrollo de software son descripciones breves y funcionales que se centran en las características de seguridad necesarias para proteger a los usuarios y la aplicación de posibles amenazas. Estas historias ayudan a incorporar requisitos de seguridad de forma integrada en el proceso de desarrollo ágil, asegurando que las funcionalidades de seguridad se consideren desde el principio del ciclo de vida del desarrollo [14]. Un ejemplo

podría ser: "Como usuario de la plataforma bancaria en línea, quiero que se me solicite una verificación de dos factores al iniciar sesión, para asegurar que solo yo pueda acceder a mi cuenta" [14].

3. Trabajos Relacionados

El presente trabajo de investigación se basa en estudios científicos enfocados en incorporar actividades de seguridad en los procesos de desarrollo de software bajo enfoques ágiles. Para seleccionar las actividades de seguridad en función de su grado de agilidad, se tomó como referencia principal el trabajo de [15], que analiza el impacto de las prácticas de seguridad en la agilidad de los modelos de proceso, basándose en el estudio de [16]. Este trabajo incluye 11 prácticas de seguridad sometidas a una evaluación empírica para determinar su grado de agilidad, utilizando un cuestionario con una escala de 5 puntos [17], obteniendo resultados aplicables a metodologías ágiles como Programación Extrema (XP) y SCRUM.

Otro estudio fundamental para la selección de las actividades de seguridad en este marco de trabajo es el realizado por [18], donde se utilizaron métodos empíricos como AHP y PROMETHEE para identificar las actividades de seguridad que mejor se integran en procesos de desarrollo ágil, considerando su bajo costo (grado de dificultad) y mayor beneficio, dentro de metodologías de desarrollo seguro como CLASP, Common Criteria, Cigital Touchpoints y Microsoft SDL.

Un tercer trabajo de referencia es el de [19], que propone un proceso de Scrum seguro que integra prácticas de desarrollo seguro en el proceso ágil Scrum. Este estudio analiza enfoques de desarrollo seguro y adapta las actividades de seguridad para que se ajusten al desarrollo ágil. Se utilizan los procesos descritos en SSE-CMM [2] como línea base para seleccionar e integrar un subconjunto de actividades de seguridad en Scrum. SSE-CMM es un estándar ISO/IEC para el desarrollo seguro de software.

4. Marco de Trabajo para el Desarrollo de Software Web Seguro con Metodologías Ágiles (SCRUM)

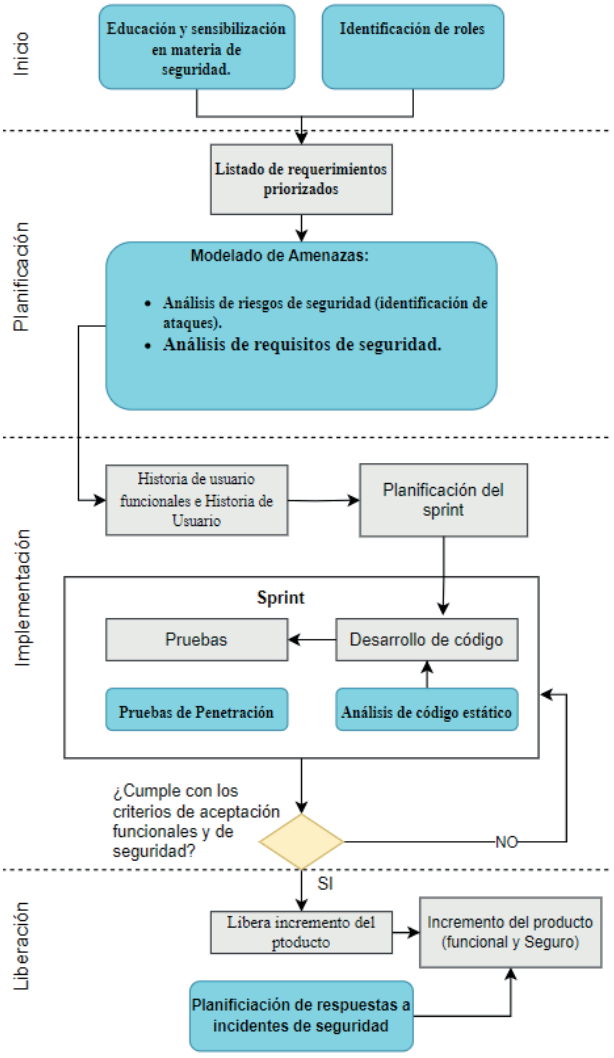
Basado en las investigaciones [15] y [18], se han seleccionado ocho actividades de seguridad que forman parte del marco de trabajo propuesto, distribuidas en las cuatro fases del marco de trabajo SCRUM: Inicio, Planificación, Implementación y Liberación:

- Inicio
  - Educación y sensibilización en materia de seguridad.
  - Identificación de roles.
- Planificación
  - Modelado de amenazas
  - Análisis de riesgos de seguridad (identificación de ataques).
  - Análisis de requisitos de seguridad.
- Implementación
  - Análisis de código estático.
  - Pruebas de penetración
- Liberación
  - Planificación de respuesta a incidentes

En la Fig. 3 se muestra de manera gráfica el marco propuesto:

Figura 3

Marco de trabajo propuesto (elaboración propia)



El marco de trabajo propuesto incluye actividades de seguridad a lo largo de todo el proceso de desarrollo de software utilizando metodologías ágiles (SCRUM). A continuación, se describen cada una de estas actividades de seguridad.

4.1. Inicio

En esta fase se incorpora las actividades de seguridad como educación y sensibilización en materia de seguridad en el desarrollo de software y socialización de las políticas de seguridad.

4.1.1. Educación y sensibilización en materia de seguridad

Definidos los requerimientos funcionales (product backlog), y las políticas de desarrollo seguro se llevan a cabo las siguientes actividades:

El especialista de desarrollo seguro de software realiza la sensibilización sobre las actividades de seguridad

que se realizarán durante el proceso de desarrollo a cada uno de los miembros del equipo dependiendo de su rol.

El especialista de desarrollo seguro de software junto al equipo y todos los interesados del proyecto identifican los requisitos de seguridad de alto nivel que formaran parte de los criterios de aceptación de cada una de las historias de usuarios funcionales.

4.1.2. Identificación de roles

En base a los requerimientos funcionales el especialista de desarrollo seguro de software junto al equipo y todos los interesados del proyecto identifican los perfiles (niveles de confianza), para todos los usuarios que tendrán acceso al aplicativo informático (software web).

Durante esta actividad también se identifican los perfiles de los posibles atacantes que se han de tener en cuenta al momento de realizar las pruebas en la fase de planeación e implementación.

4.2. Planificación

En esta fase, el equipo de desarrollo, consciente de las políticas y prácticas de desarrollo seguro de software, se basa en los requisitos de alto nivel y en las 10 amenazas más comunes identificadas por OWASP [20] para llevar a cabo las siguientes actividades de seguridad:

- Modelado de amenazas
- Descomposición de la propuesta de solución en términos de seguridad.
- Análisis de riesgos de seguridad (identificación de ataques).
- Análisis de requisitos de seguridad.

4.2.1. Modelado de Amenazas

El objetivo de esta actividad de seguridad es identificar posibles ataques (amenazas de seguridad) a los que un sistema informático (software web) puede estar expuesto y definir un conjunto de requisitos de seguridad para mitigarlos (ver Fig. 4).

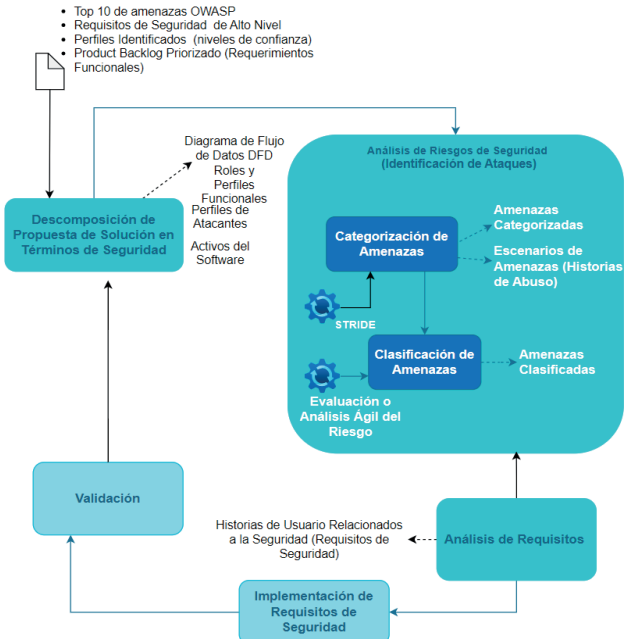
Esta actividad se enmarca en el proceso de modelado de amenazas (Threat Modeling Process) definido por OWASP [11], que incluye dos actividades clave del modelo de desarrollo seguro propuesto (el análisis de riesgos de seguridad y el análisis de requisitos de seguridad):

a. Descomposición de la propuesta de solución en términos de seguridad.
b. Análisis de riesgos de seguridad (identificación de ataques).
c. Análisis de requisitos de seguridad.
d. Mitigación de riesgos/implementación de requisitos de seguridad
e. Validación.

a. <u>Descomposición de la propuesta de solución en términos de seguridad</u>: Guiados por el especialista en desarrollo seguro el equipo de desarrollo identifica los posibles objetivos de amenazas, desde

Figura 4

Proceso Modelado de Amenazas (elaboración propia)



la perspectiva de un atacante, representándolos visualmente a través de un diagrama de flujo de datos (DFD), [21]. El diagrama de flujo de datos permite al equipo de desarrollo identificar los denominados "puntos de entrada" (vulnerabilidades de acceso), por donde un atacante potencial podría acceder al sistema informático (software web) y causar daño. El resultado de la descomposición de la propuesta de solución, en términos de seguridad es:

- o El diagrama de Flujo de Datos (DFD).
- o La identificación de posibles puntos vulnerables donde un atacante potencialmente podría interactuar maliciosamente con el software producido.
- o La identificación de los activos, es decir, elementos o áreas en las que el atacante potencialmente estaría interesado.
- o Roles y/o perfiles funcionales y perfiles de atacantes.

b. <u>Análisis de riesgos de seguridad (identificación de ataques)</u>: El equipo de desarrollo incluido el product owner, guiados por el especialista de desarrollo seguro, identifican las exposiciones o ataques a las que puede estar expuesto el software web (amenazas), vulnerabilidades e impacto. Las amenazas identificadas son traducidas en un lenguaje de historias de abuso [13], estas historias son una variante de las historias de usuario y fueron propuestas por los autores [22]. Identificadas las amenazas, vulnerabilidades e impacto se procede a su categorización para ello se hace uso del modelo de categorización de amenazas STRIDE (Spoofing, Tempe ring, Repudio, Information disclosure, Denal of service, Elevation of privileges), Categorizadas las amenazas y definidos las historias de abuso, el equipo de desarrollo procede a

clasificarlos desde la perspectiva de factores de riesgo (probabilidad e impacto de ocurrencia), en el presente trabajo se le define como evaluación ágil de riesgos y está basado en el modelo de clasificación de riesgos definido por OWASP.

c. Análisis de requisitos de seguridad: El equipo de desarrollo junto al especialista en desarrollo seguro de software identifica y definen las nuevas funciones del software (requisitos de seguridad), que permitan resolver uno o más problemas de seguridad en específico, eliminando una o más vulnerabilidades potenciales y evitar que una amenaza se materialice. Los requisitos de seguridad identificados se formulan y describen como historias de usuario relacionadas a la seguridad [14], un ejemplo de historia de usuario relacionado a la seguridad, puede ser: "Yo como cliente deseo que mis datos sean protegidos para no ser divulgados a otros clientes o personas desconocidas". Los requisitos de seguridad forman parte de la definición de hecho (Definition of Done) de cada sprint en el presente marco propuesto, una vez que estos hayan completado se da por concluido todas las actividades del sprint.

d. Implementación de requisitos de seguridad (Mitigación del Riesgo: En lo que respecta a la mitigación de riesgos o implementación de requisitos de seguridad estos se llevan a cabo durante la implementación del sprint (fase de implementación), es decir mediante el desarrollo o implementación de cada uno de las historias de usuario relacionadas a la seguridad correspondiente a cada uno los requisitos de seguridad identificados.

e. Validación de la implementación de los requisitos de seguridad: La validación de la implementación de los requisitos de seguridad (historias de usuario relacionados a la seguridad), se realiza al final de la fase de implementación durante la revisión del sprint, y validando si se ha cumplido con lo establecido en las definiciones de terminado (Definitions of Done).

4.3. Implementación.

4.3.1. Análisis de código estático: Bajo la orientación del especialista en desarrollo seguro de software el equipo de desarrollo planifica, diseña y ejecuta las pruebas de código estático. Para está actividad el equipo de desarrollo puede hacer uso de herramientas automatizadas de análisis de código estático conocidas como Static Application Security Testing (SAST), algunas herramientas ya se introducen en los entornos de programación (IDE) facilitando aún más el análisis de código estático. OWASP también proporciona una lista de herramientas automatizadas para realizar el análisis de código estático, y se puede acceder a esta desde el siguiente enlace: https://owasp.org/www-community/Source_Code_Analysis_Tools.

4.3.2. Pruebas de penetración (pentesting): El equipo de desarrollo debe realizar las pruebas de penetración al finalizar cada sprint antes de su liberación o pase a producción, actividad que debe realizarse poniendo en práctica las pruebas de penetración por pares. Con la orientación del especialista en desarrollo seguro de software realiza las pruebas de penetración, para simular acciones de un atacante informático con el objetivo de descubrir posibles vulnerabilidades resultantes de errores de codificación, configuración u otras debilidades.

El equipo de desarrollo se debe apoyar en el uso de herramientas automatizadas para escanear de manera externa (desde la perspectiva de un atacante), el sistema informático en búsqueda de vulnerabilidades de seguridad, estas herramientas son conocidas como pruebas dinámicas de seguridad de aplicaciones (Dynamic Application Security Testing-DAST).

OWASP proporciona una lista de herramientas para realizar las pruebas de penetración accesible desde el siguiente enlace web: https://owasp.org/www-community/Free_for_Open_Source_Application_Security_Tools. De esta lista se sugiere usar la herramienta para prueba dinámica conocida como Zed Attack Proxy (ZAP) y la puede descargar desde el siguiente enlace web: https://owasp.org/www-project-zap/.

4.4. Liberación

En esta fase del proceso se realizan todas las actividades para la puesta en producción (implantación), del entregable (incremento funcional y seguro), adicional a ello el equipo de desarrollo bajo la orientación del especialista en desarrollo seguro de software realizan la planificación de respuesta a incidentes.

4.4.1. Planificación de respuesta a incidentes: El objetivo de contar con un plan de respuesta ante incidentes de seguridad es responder ante los incidentes de seguridad de manera sistemática (siguiendo una metodología coherente de gestión de incidentes), para que se adopten las medidas adecuadas a fin de reducir al mínimo su impacto y como recuperarse producido el incidente. El plan debe contener como mínimo:

- Estrategias y objetivos.
- Aprobación de la alta dirección.
- Metodología (que define el procedimiento de respuesta ante incidentes de seguridad).
- Roles y responsabilidades.
- Estrategia de comunicaciones para comunicarse con el resto de la organización y posiblemente con otras organizaciones.
- Métricas para medir la capacidad de respuesta ante incidentes y su eficacia.
- Registro de las actividades de respuesta (bitácora de respuestas ante incidentes de seguridad).
- Actualización del plan basado en lecciones aprendidas.
- Cronograma y resultado de las pruebas del plan

5. Evaluación

En el presente estudio se aplicó el tipo de investigación aplicada, con un enfoque "cualitativo" y "cuantitativo" (dado que se analizarán los datos

empleando métodos estadísticos), para lograr determinar el impacto de las actividades y/o prácticas de seguridad al ser incorporadas en el proceso de desarrollo de software web, tomando en cuenta las dimensiones de grado de agilidad, costo-beneficio y tiempo en el desarrollo de software con metodologías ágiles (SCRUM).

5.1. Encuesta de Investigación

El diseño de la presente investigación es "no experimental", dado que se analizan los datos a través de encuestas de opinión que según [23] y [24] están consideradas entre las investigaciones no experimentales, con la cual se obtuvo la percepción de los expertos sobre: causas, frecuencia y consecuencias de ataques a un software web, experiencia en desarrollo de software usando SCRUM, la incorporación de un especialista en desarrollo seguro de software usando SCRUM, el costo-beneficio de la actividad de seguridad "modelado de amenazas", y el tiempo adicional requerido al incorporar las actividades de seguridad, en el proceso de desarrollo de software web usando SCRUM.

La unidad de análisis del presente estudio lo conforman los gerentes/directores, analistas funcionales, programadores, analistas de calidad de software, scrum masters, gestores de proyectos de software, se determinó un tamaño de muestra de 45 especialistas y/o profesionales quienes se les aplico la encuesta, con un límite de error de 5% (0.05), para escoger esta muestra se practicó un muestreo aleatorio y probabilístico por conveniencia [25], donde el investigador puede seleccionar el conjunto de la población bajo criterios propios y de interés. Siguiendo esa premisa, el criterio considerado para escoger la muestra se basó en la experiencia que poseen los expertos en el desarrollo de software web usando metodologías ágiles (SCRUM), se enviaron e-mails (adjuntando el enlace web de la encuesta en online) a expertos (gerentes/directores, analistas funcionales, programadores, analistas de calidad de software, scrum masters, gestores de proyectos de software), en desarrollo de software web usando metodologías ágiles (SCRUM), de instituciones tanto públicas y/o privadas, y se aplicó la técnica de muestreo bola de nieve [26], solicitando a cada experto que reciba la encuesta, lo remita a otras personas que tengan el mismo perfil. Véase el apéndice para la tabla de preguntas, tipo de preguntas y escala respectiva.

5.2. Resultados

Tras recopilar los datos a través del cuestionario, se realizaron los siguientes análisis estadísticos utilizando la herramienta IBM SPSS Statistics Versión 29.0.0.0: prueba de fiabilidad de datos mediante el Alfa de Cronbach, análisis descriptivo de la muestra, y análisis de aspectos demográficos de la encuesta. Estos aspectos incluyen la causa, frecuencia y consecuencias (impacto) de los ataques a software web, años de experiencia en desarrollo web usando SCRUM, incorporación de un especialista en desarrollo seguro de software, costo y beneficio de incorporar actividades de seguridad, y el tiempo adicional requerido para estas actividades durante el proceso de desarrollo. Se realizó un análisis de modelado de amenazas basado en datos no paramétricos para comparar las muestras relacionadas (costo/beneficio) utilizando la prueba de rango de Wilcoxon [27], y un análisis de la dimensión tiempo basado en la media aritmética de datos agrupados. A continuación, se presentan los resultados cualitativos de las preguntas base para evaluar el marco de trabajo propuesto.

5.2.1. Fiabilidad de los Datos (Alpha Cronbach)

Según el trabajo de investigación de los autores [29] un valor aceptable para el coeficiente alfa de Cronbach es 0.7. Obtener un valor superior a este indica una fuerte relación entre las preguntas, lo que valida el instrumento. En el presente trabajo de investigación, utilizando la herramienta IBM SPSS Statistics, se obtuvo un alfa de Cronbach de 0.737 (ver Fig. 5), lo que confirma que la validez del instrumento es aceptable.

Figura 5

Resultado de fiabilidad de datos del cuestionario aplicado en el presente trabajo de investigación

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,737	15

5.2.2. Análisis Descriptivo de los Resultados de la Muestra

A continuación, se presentan los datos obtenidos mediante un instrumento de recolección diseñado para este estudio. Este instrumento, compuesto por 15 ítems, fue aplicado a 45 profesionales y expertos en desarrollo de software web con SCRUM. El análisis descriptivo de los datos se realizó utilizando la herramienta IBM SPSS Statistics.

a. Experiencia en el desarrollo de software usando Scrum

Para determinar la experiencia de los profesionales encuestados y asegurar que sus respuestas estén basadas en esta, se les formuló la siguiente pregunta: '¿Cuántos años de experiencia tiene en el desarrollo de software usando SCRUM?'. Los resultados obtenidos del análisis estadístico fueron los siguientes:

- Media: 2.6
- Mediana: 2.0
- Moda: 2
- Desviación estándar: 1.25
- Varianza: 1.56
- Cantidad de respuestas: 45

Los resultados indican que la mayoría de los encuestados tienen entre 3 y 5 años de experiencia en

el desarrollo de software usando SCRUM. Sin embargo, hay una variabilidad considerable, con algunos encuestados que tienen menos experiencia y otros que tienen más de 10 años de experiencia. Esto sugiere que los encuestados tienen un rango amplio de experiencia, lo cual es importante para obtener una perspectiva diversa sobre las prácticas y desafíos en el desarrollo seguro de software con SCRUM.

b. Ausencia de actividades de desarrollo seguro como causa principal de ataques

Para evaluar la percepción de los profesionales encuestados sobre la ausencia de actividades de desarrollo seguro como causa principal de ataques, se les formuló la siguiente pregunta: '¿Considera usted que una de las principales causas de ataques a un software web es la falta de incorporación de actividades de desarrollo seguro durante su proceso de desarrollo?'. Los resultados obtenidos del análisis estadístico fueron los siguientes:

- Media: 3.96
- Mediana: 4.0
- Moda: 5
- Desviación estándar: 1.33
- Varianza: 1.77
- Cantidad de respuestas: 45

Los resultados muestran que la mayoría de los profesionales encuestados están de acuerdo con la afirmación, con una media cercana a 'De Acuerdo' y una moda en 'Totalmente de Acuerdo'. Aunque la desviación estándar indica cierta variabilidad en las respuestas, la tendencia general es positiva hacia la idea de que la ausencia de actividades de desarrollo seguro se considera una causa principal de los ataques a un software web.

c. Frecuencia de ataques a un software web si durante su desarrollo no se incorporaron actividades de desarrollo seguro

Para evaluar la percepción de los profesionales encuestados sobre la frecuencia de ataques a un software web sin actividades de desarrollo seguro, se les formuló la siguiente pregunta: '¿Con qué frecuencia sería atacado un software web si durante el desarrollo no se incorporaron actividades de desarrollo seguro?'. Los resultados obtenidos del análisis estadístico fueron los siguientes:

- Media: 4.09
- Mediana: 4.0
- Moda: 4
- Desviación estándar: 0.63
- Varianza: 0.40
- Cantidad de respuestas: 45

Los resultados indican que la mayoría de los encuestados creen que, si no se incorporan actividades de desarrollo seguro durante el proceso de desarrollo de un software web, es muy probable que el software sea atacado con frecuencia. Esto subraya la importancia de implementar prácticas de seguridad durante el desarrollo para mitigar el riesgo de ataques.

d. Impacto luego de un ataque a un sistema informático(software)

Para evaluar la percepción de los profesionales encuestados sobre el nivel de impacto de un ataque a un sistema informático (software), se les formuló la siguiente pregunta: '¿Cuál es el nivel de impacto que se da en una organización luego de producirse un ataque a su sistema informático?'. Los resultados obtenidos del análisis estadístico fueron los siguientes:

- Media: 1.58
- Mediana: 1.0
- Moda: 1
- Desviación estándar: 0.81
- Varianza: 0.66
- Cantidad de respuestas: 45

Los resultados indican que la mayoría de los encuestados creen que el impacto de un ataque a un sistema informático (web), en una organización es extremadamente negativo. Esto destaca la importancia de la seguridad informática y la necesidad de medidas preventivas para proteger los sistemas y mitigar los riesgos de ataques que puedan tener consecuencias graves para la organización.

e. Incorporación de un especialista en desarrollo seguro de software usando Scrum

Para evaluar la percepción de los profesionales encuestados sobre la incorporación de un especialista en desarrollo seguro de software usando SCRUM, durante el proceso de desarrollo se les formuló la siguiente pregunta: '¿Cuál es su posición respecto a la incorporación de un especialista en desarrollo seguro de software usando SCRUM?'. Los resultados obtenidos del análisis estadístico fueron los siguientes:

- Media: 4.53
- Mediana: 5.0
- Moda: 5
- Desviación estándar: 0.73
- Varianza: 0.53
- Cantidad de respuestas: 45

Estos resultados indican que la mayoría de los encuestados están fuertemente de acuerdo con la afirmación, con una media y una mediana cercanas a "Totalmente de Acuerdo" y una moda en "Totalmente de Acuerdo". La baja desviación estándar sugiere que hay poca variabilidad en las respuestas, lo que refuerza el consenso positivo hacia la incorporación de un especialista en desarrollo seguro en equipos que usan SCRUM.

f. Costo de incorporar la actividad de seguridad "Modelado de Amenazas" durante el desarrollo

Para evaluar la percepción de los profesionales encuestados respecto al costo que demanda el incorporar la actividad de seguridad "Modelado de Amenazas" durante el proceso de desarrollo, se les formuló la siguiente pregunta: '¿Qué tan costoso considera usted que significaría incorporar la actividad de seguridad "Modelado de Amenazas" durante el proceso de

desarrollo de software usando SCRUM?'. Los resultados obtenidos del análisis estadístico fueron los siguientes:

- Media: 2.98
- Mediana: 3.0
- Moda: 3
- Desviación estándar: 0.92
- Varianza: 0.84
- Cantidad de respuestas: 45

Los resultados indican que, en general, los encuestados tienen una percepción neutral sobre el costo de incorporar el modelado de amenazas durante el desarrollo de software usando SCRUM. Esto sugiere que, aunque algunos pueden considerar que es una inversión significativa, otros pueden verlo como un costo manejable o necesario para mejorar la seguridad del software. La opinión está equilibrada, lo que implica que la percepción del costo puede variar dependiendo del contexto y de los recursos de cada organización.

g. Beneficio de incorporar "Modelado de Amenazas" como actividad de seguridad durante el desarrollo

Para evaluar la percepción de los profesionales encuestados respecto al beneficio de incorporar la actividad de seguridad "Modelado de Amenazas" durante el desarrollo, se les formuló la siguiente pregunta: '¿Qué tan beneficioso considera usted que significaría incorporar la actividad de seguridad "Modelado de Amenazas" durante el proceso de desarrollo de software usando SCRUM?'. Los resultados obtenidos del análisis estadístico fueron los siguientes:

- Media: 4.6
- Mediana: 5.0
- Moda: 5
- Desviación estándar: 0.58
- Varianza: 0.34
- Cantidad de respuestas: 45

Los resultados indican que la mayoría de los encuestados creen que la incorporación de la actividad de seguridad "Modelado de Amenazas" durante el proceso de desarrollo de software usando SCRUM es altamente beneficiosa. Esto sugiere que los encuestados valoran mucho la inclusión de esta práctica de seguridad y reconocen su importancia para mejorar la calidad y la seguridad del software desarrollado.

Tiempo adicional que demanda incorporar actividades de seguridad en el desarrollo con SCRUM

5.2.3. Análisis del Modelado de Amenazas (Costo-Beneficio)

Para comparar el costo-beneficio de incorporar la actividad de seguridad "Modelado de Amenazas" en el desarrollo de software basado en los resultados de la encuesta aplicada en el presente trabajo de investigación (literales f y g del numeral 5.2.2.), se utilizó la prueba de rango de Wilcoxon [27] (prueba de hipótesis estadística no paramétrica para comparar dos muestras relacionadas).

Hipótesis:

H0: No hay diferencias entre costo y beneficio al aplicar la actividad de seguridad de modelado de amenazas en el marco de trabajo propuesto.

H1: Sí hay diferencias entre costo y beneficio al aplicar la actividad de seguridad de modelado de amenazas en el marco de trabajo propuesto.

Utilizando la herramienta IBM SPSS Statistics Versión 29.0.0.0 y con un nivel de significancia de 0.05 se encontraron los siguientes resultados:

Estadísticos descriptivos: En base a valores de la escala de Likert obtenidos en la encuesta referidos al costo y beneficio de incorporar la actividad de seguridad "Modelado de Amenazas" durante el proceso de desarrollo de software, se calcularon los estadísticos descriptivos (media, desviación estándar, valores mínimo y máximo), para las 45 respuestas. Los resultados se presentan en la Tabla 1.

Tabla 1

Estadísticos descriptivos de costo y beneficio al incorporar en el proceso la actividad de "Modelado de Amenazas"

	N	Media	Desviación Estándar	Mínimo	Máximo
Costo	45	2,9778	,91674	1,0	5,0
Beneficio	45	4,6000	,57997	3,0	5,0

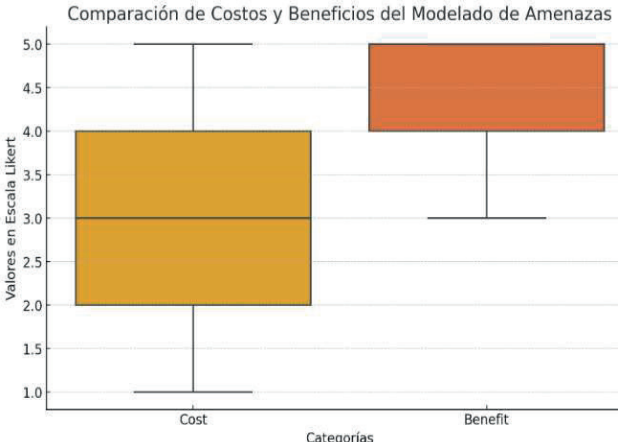
Resumen de la Prueba de Rangos de Wilcoxon:

- N total: 45
- Estadístico de prueba: 0.0
- Error estándar: 0.137
- Estadístico de prueba estandarizado: 11.871
- Significación asintótica (prueba bilateral): 7.58e-08

Dado que el p-valor es extremadamente bajo (7.58e-08), rechazamos la hipótesis nula (H0) con un nivel de significancia de 0.05. Esto indica que hay diferencias significativas entre las observaciones de costo y beneficio al aplicar la actividad de seguridad de "Modelado de Amenazas" en el proceso de desarrollo de software.

Figura 6

Gráfico de cajas de la Comparación de Costos y Beneficios del Modelado de Amenazas



El gráfico de cajas (ver Fig. 6) compara las distribuciones de los valores de costo y beneficio para la actividad de seguridad de "Modelado de Amenazas". Este gráfico destaca las diferencias entre las dos categorías, con el beneficio mostrando valores más altos en la escala Likert en comparación con el costo.

En conclusión, los resultados del análisis estadístico y la visualización (gráfico de cajas), indican una diferencia significativa entre las percepciones de costo y beneficio. Esta diferencia sugiere que los encuestados perciben a la actividad de seguridad de "Modelado de Amenazas" como una actividad Muy Beneficiosa a pesar de ser Costoso.

5.2.4. Análisis de la Dimensión Tiempo

La Tabla 2 muestra las frecuencias de las respuestas a la pregunta sobre el tiempo adicional en horas que se requeriría para implementar cada actividad de seguridad propuesta en el presente marco de trabajo, durante el proceso de desarrollo de software con SCRUM.

Para poder calcular la media en base a los resultados obtenidos de la encuesta, se ha tomado en cuenta la media estadística de datos agrupados para los rangos de 0-2, de 2-5, de 5-8, de 8-10 y mayor a 10 horas. Por ejemplo, para la actividad de seguridad "Educación y sensibilización en materia de seguridad (entrenamiento en políticas y actividades de desarrollo seguro)", obtenemos su media a partir de los resultados obtenidos de la encuesta que se muestran en la Tabla 2.

De acuerdo al resultado obtenido para la actividad de seguridad 'Educación y sensibilización en materia de seguridad', ver Tabla 3, se puede afirmar que los expertos y profesionales consultados estimaron que incorporar esta actividad de seguridad demanda un tiempo adicional de 6.61 horas. Aplicamos el mismo procedimiento para obtener la media de las todas las actividades de seguridad que forman parte del marco de trabajo propuesto, los resultados se muestran en la Tabla 4.

Tabla 2

Resultado de tiempo adicional que demanda incorporar las actividades de seguridad

Actividad de Seguridad	Respuestas (rangos de horas)				
	0-2	2-5	5-8	8-10	>10
Educación y Sensibilización en Materia de Seguridad	1	13	15	9	6
Identificación de Roles	8	23	8	5	1
Modelado de Amenazas	3	11	13	9	9
Análisis de Riesgos de Seguridad (Identificación de Ataques)	4	12	17	7	5
Análisis de Requisitos de Seguridad	5	11	16	9	4
Análisis de Código Estático	3	17	7	11	7
Pruebas de Penetración	4	6	13	12	10
Planificación de Respuesta a Incidentes	4	13	14	11	3

Tabla 3

Valores para el cálculo de media de datos agrupados

Rango de Tiempo Adicional (Horas)	x_i	f_i	$x_i f_i$
Mayor a 10	11	6	66
(8-10)	9	9	81
(5-8)	6.5	15	97.5
(2-5)	3.5	13	45.5
(0-2)	1	1	1
Sumas		44	291

Donde:
x_i = Valor representativo del intervalo (marca de clase)
f_i = Frecuencia absoluta.

El cálculo de la media datos agrupados se obtiene aplicando la fórmula (1)

$$\bar{x} = \sum \frac{x_i * f_i}{n} \quad (1)$$

Reemplazando en (1):

$\bar{x}$ = (291)/44
$\bar{x}$ = 6.61

Tabla 4

Resultado (frecuencias) de tiempo adicional que demanda incorporar las actividades de seguridad

N°	Actividades de Seguridad	Tiempo (horas)
1	Educación y Sensibilización en Materia de Seguridad	6.61
2	Identificación de Roles	5.04
3	Modelado de amenazas	6.80
4	Identificación de ataques	6.10
5	Análisis de requisitos de seguridad	6.06
6	Análisis de código estático	7.47
7	Pruebas de penetración	7.28
8	Planificación de respuesta a incidentes	6.06
	Total (horas adicionales por sprint)	51.42

De acuerdo a los resultados se tiene que incorporar las 8 actividades de seguridad en el proceso de desarrollo de software usando SCRUM (marco de trabajo propuesto), tomaría 51.42 horas adicionales, considerando que un día consta de 8 horas de trabajo ideales, en días da como resultado 6.43 días adicionales por sprint.

6. Evaluación de Agilidad del Marco de Trabajo Propuesto

6.1. Grado de Agilidad al Aplicar el Marco de Trabajo Propuesto

Para validar en términos de agilidad el marco de trabajo propuesto se ha tomado en cuenta los siguientes factores de ponderación: años de experiencia en el desarrollo de soltare usando SCRUM, incorporación de un especialista en desarrollo seguro de software y tiempo adicional (dimensión tiempo).

De acuerdo al trabajo de investigación de los autores [15] el grado de agilidad luego de aplicar las actividades de seguridad a un proceso de desarrollo de

software usando SCRUM se calcula de acuerdo a la siguiente fórmula:

AAAS= [(ART1+ART2+…+ARTn) ÷ n]*AOM (2)

Donde por sus siglas en inglés:

AAAS: Agility After Application of Security (agilidad después de aplicar actividades de seguridad)

ART: Agility Reduction Tolerance (tolerancia a la reducción de agilidad, calculado como grado de agilidad de cada actividad de seguridad)

AOM: Actual Agility of Model (agilidad actual de SCRUM 0.7).

El cálculo de la agilidad después de aplicar actividades de seguridad (AAAS), se obtiene aplicado la fórmula (2).

Reemplazamos en la fórmula (2) los valores de ART de las ocho actividades de seguridad tomadas en cuenta en el marco de trabajo propuesto.

AAAS = [(0.75+0.89+0.87+0.83+0.88+0.85+0.86+0.81) ÷8] * 0.7

AAAS = (6.74÷8) * 0.7

AAAS = 0.59

El valor del grado de agilidad después de aplicar actividades de seguridad (AAAS), en el trabajo de investigación de los autores [15] es de 0.59.

6.1.1. Cálculo del factor de ponderación de incorporación de un especialista en desarrollo seguro de software respecto al grado de agilidad al aplicar el marco de trabajo propuesto: Para el caso se toma se toma como base el resultado de la encuesta aplicada en el presente trabajo de investigación, y la fórmula aplicada en el trabajo de investigación de los autores [15], usando una ponderación del 1 al 5.

La ponderación 5 se asigna a la categoría A que corresponde a los profesionales o especialistas con más de 10 años de experiencia, la ponderación 4 a la categoría B que corresponde a los profesionales o especialistas con más de 7 y menos de 10 años de experiencia, la ponderación 3 a la categoría C que corresponde a los profesionales o especialistas con más de 5 y menos de 7 años de experiencia, la ponderación 2 a la categoría D que corresponde a los profesionales o especialistas con más de 3 y menos de 5 años de experiencia, la ponderación 1 a la categoría E que corresponde a los profesionales o especialistas con menos de 3 años de experiencia.

Formula basada en el trabajo de investigación de [15]:

FIEDS= {[(Suma de valores de categoría A*5 + Suma de valores de categoría B*4 + Suma de valores de categoría C*3 + Suma de valores de categoría D*2 + Suma de valores de categoría E*1) ÷ (Frecuencia de categoría A*5 + Frecuencia de categoría B*4 + Frecuencia de categoría C*3 + Frecuencia de categoría D*2 + Frecuencia de categoría E*1)] ÷ 8} * AAAS (3)

Donde:

FIEDS=Factor ponderación Incorporación de Especialista en Desarrollo Seguro.

1. Del resultado de la encuesta se tiene:
2. Frecuencia de categoría A=5
3. Frecuencia de categoría B=6
4. Frecuencia de categoría C=8
5. Frecuencia de categoría D=18
6. Frecuencia de categoría E=8

La suma de valores (para la incorporación de un especialista en desarrollo seguro de software), marcados por los expertos de la categoría A=23, la suma de valores marcados por los expertos de la categoría B=29, la suma de valores marcados por los expertos de la categoría C=39, la suma de valores marcados por los expertos de la categoría D=77, la suma de valores marcados por los expertos de la categoría E=36.

El cálculo del factor de ponderación de incorporación de especialista en desarrollo seguro (FIEDS), se obtiene aplicando la formula (3).

Reemplazando en la formula (3):

FIEDS = { [(23 * 5 + 29 * 4 + 39 * 3 + 77 * 2 + 36 * 1) ÷ (5*5+6*4+8*3+18*2+8*1)] ÷8} * AAAS

FIEDS = [(538÷117) ÷ 8]*AAAS

FIEDS = (0.575) (0.59)

FIEDS = 0.34

Por tanto, el valor obtenido para el factor de incorporación de especialista en desarrollo seguro (FIEDS), en el marco de trabajo propuesto es igual a 0.34.

6.1.2. Cálculo del factor de ponderación de la dimensión tiempo: El factor de ponderación en función al tiempo adicional por sprint al aplicar el marco de trabajo propuesto se obtiene tomando en cuenta la dimensión tiempo (expresado en días), luego de incorporar las actividades de seguridad durante el proceso de desarrollo, y aplicando la siguiente fórmula:

FDT= (DT ÷ Días ideales por sprint) * AAAS (4)

Donde:

FDT= Factor de ponderación de la dimensión tiempo

DT= Dimensión tiempo (ítem 5.2.4)

El cálculo del factor de ponderación de la dimensión tiempo (FDT), se obtiene aplicando la formula (4).

Reemplazando los valores en la fórmula (4):

FDT = (6.43 ÷ 30) * 0.59

FDT= 0.214 *0.59

FDT = 0.13

Por tanto, luego de aplicar la fórmula se obtiene que el valor del factor de ponderación basado en la dimensión tiempo es de 0.13.

6.1.3. Cálculo del grado de agilidad aplicando los factores de ponderación: El grado de agilidad luego de aplicar el marco de trabajo propuesto (considerando los factores de ponderación), se obtiene sumando al grado de agilidad obtenido en el trabajo de investigación

[15], de donde AAAS=0.59, el factor de ponderación de incorporar un especialista de desarrollo seguro al proceso y el factor de ponderación de la dimensión tiempo, tomando en cuenta que al incorporar un especialista de desarrollo seguro la agilidad del proceso se ve impactada positivamente en tanto que incorporar las actividades de seguridad demandan de un tiempo adicional por tanto impacta negativamente respecto a la agilidad (ver Tabla 5).

Tabla 5

Datos para determinar el grado de agilidad

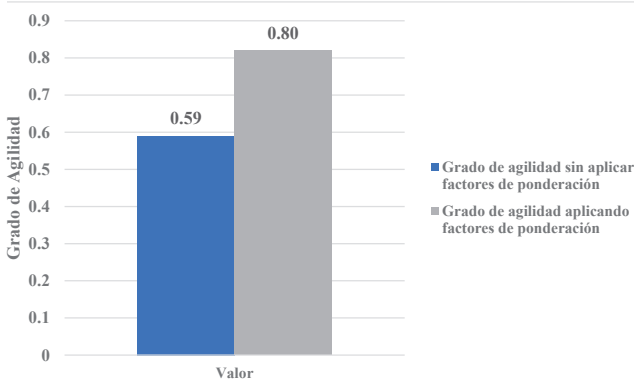
AAAS	FIEDS (+)	FDT (-)
0.59	0.34	0.13

Por tanto, por lo descrito en el párrafo anterior el grado de agilidad luego de aplicar el marco de trabajo propuesto (AAAS') se obtiene aplicando la siguiente fórmula:

AAAS'=AAAS+FIEDS-FDT (5)

Figura 7

Grado de agilidad al aplicar el marco de trabajo propuesto (aplicando factores de ponderación)



El cálculo del grado de agilidad aplicando los factores de ponderación (AAAS'), se obtiene aplicando la formula (5).

Reemplazando los datos en la fórmula (5)

AAAS' = 0.59+0.34-0.13

AAAS'= 0.80

Luego de reemplazar los valores en la fórmula (5) el grado de agilidad aplicando el marco de trabajo propuesto (AAAS´) es 0.80.

Para determinar qué tan ágil es el marco de trabajo propuesto considerando una escala de 0 a 1 se tiene:

0.9< AAAS'<1.0=Muy Ágil

0.7< AAAS'<0.9= Ágil

0.5< AAAS'<0.7= Ligeramente Ágil

0.0< AAAS'<0.5= Nada Ágil

Según la escala, y dado que AAAS'=0.80, se observa que el marco de trabajo propuesto se mantiene como Ágil.

7. Conclusiones y Trabajos Futuros

7.1. Conclusiones

- Un factor determinante en el éxito de la aplicación del marco propuesto es la incorporación de un especialista en desarrollo de software seguro y que adicional a ello cuente con experiencia en procesos de desarrollo como SCRUM, de los resultados de la encuesta aplicada en el presente trabajo de investigación donde el 60% de encuestados está totalmente de acuerdo con su incorporación, de estos resultados se ha podido estimar que el valor del fator de ponderación de incorporación de un especialista en el desarrollo seguro de software en 0.34.
- Un aporte y también factor determinante para obtener el grado de agilidad en la aplicación del marco de trabajo propuesto y en la presente investigación es el factor tiempo adicional que demanda la incorporación de actividades de seguridad durante el proceso de desarrollo de software usando SCRUM, de los resultados de la encuesta el valor del factor de ponderación de la dimensión tiempo es -0.13 (con valor negativo toda vez que impacta de manera negativa al proceso ágil de desarrollo con SCRUM).
- Tomando en cuenta el costo-beneficio al aplicar el presente marco de trabajo propuesto se concluye que si bien es cierto incorporar actividades de seguridad "modelado de Amenazas" durante el proceso de desarrollo demandan de un costo en esfuerzo, de la literatura y resultados obtenidos se concluye que su incorporación es Muy Beneficioso, lo que significa que aplicando el marco de trabajo propuesto se logra un producto de software web más seguro.
- Los resultados obtenidos en este trabajo de investigación (ver Fig. 7) demuestran que es factible desarrollar software web utilizando el marco de trabajo propuesto sin afectar drásticamente el grado de agilidad de las actividades de seguridad. Considerando la incorporación de un especialista en desarrollo seguro de software y el tiempo adicional requerido para implementar las actividades de seguridad, el grado de agilidad resultante es de 0.80, en comparación con el 0.59 obtenido en el estudio de los autores [15]. Esto muestra una diferencia significativa de 0.21 en el grado de agilidad.

7.2. Trabajos Futuros

- El marco de trabajo propuesto en el presente trabajo de investigación es un referente en el

desarrollo de software seguro y puede ser la base para el desarrollo futuro de otros marcos de trabajo usando otras dimensiones, factores y variables que en este trabajo de investigación no se hayan contemplado, como por ejemplo considerar la dimensión costo en términos monetarios de incorporar actividades de seguridad en procesos de desarrollo ágil.

- El marco de trabajo propuesto puede servir como base para integrar la inteligencia artificial (IA) en los procesos de desarrollo de software seguro bajo un enfoque ágil. Esto incluye el uso de IA para analizar reportes de las principales amenazas, vulnerabilidades y malas prácticas en el desarrollo de software, así como para realizar análisis de riesgos. El objetivo final es lograr resultados más precisos en el desarrollo de software seguro.

Referencias

[1] Allen, J. 'Software, Security Engineering: A Key Discipline for Project Managers'. the IEEE Reliability Society 2008 Annual Technology Report, 2008

[2] ISO/IEC 21827:2008(E) 'Systems engineering-Systems security engineering-Capability maturity model (SSE-CMM)'. International Organization for Standardization. https://www.iso.org/standard/45135.html, 2008

[3] Alenezi, M., Abdul Basit, H., Anwar Beg, M., Saad Shaukat, M.: 'Synthesizing secure software development activities for linear and agile lifecycle models'. Software Practice and Experience 52(8), 2022.

[4] European Union Agency for Cybersecurity (ENISA). 'Good practices for security of Internet of Things in the context of smart manufacturing'. https://www.enisa.europa.eu/publications/good-practices-for-security-of-iot-1, 2019.

[5] Smith, J. 'Advanced Security in Software Development'. Editorial TechPress, 2021.

[6] OWASP Foundation. 'Threat Modeling'. https://owasp.org/www-community/Threat_Modeling, 2021.

[7] Chess, B., & West, J. 'Secure Programming with Static Analysis'. O'Reilly Media, 2021.

[8] Weidman, G. 'Penetration Testing: A Hands-On Introduction to Hacking'. No Starch Press, 2020.

[9] Kouns, J., & Minoli, D. 'Information Technology Risk Management in Enterprise Environments: A Review of Industry Practices and a Practical Guide to Risk Management Teams'. Wiley, 2021.

[10] Microsoft. Threat modeling: Uncover security design flaws using the STRIDE approach. https://docs.microsoft.com/en-us/security/engineering/threat-modeling-tool-getting-started, 2020.

[11] Conklin, L., Drake, V., Strittmater, S. (s.f.). Threat Modeling Process. https://owasp.org/www-community/Threat_Modeling_Process, accedido 02 enero 2023.

[12] De Haan, J., Barendsen, E., & Gieles, P. Agile Risk Management. En S. W. Ambler & M. Lines, Disciplined Agile Delivery: 'A Practitioner's Guide to Agile Software Delivery in the Enterprise' (pp. 195-210). IBM Press, 2021.

[13] Kohnfelder, L., & Garg, P. Threat Modeling: 12 Available Methods. Microsoft Security Development Lifecycle. https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling, 2021.

[14] McGraw, G. 'Software Security: Building Security In'. Addison-Wesley Professional, 2020.

[15] Maqsood, M., Bondavalli, A. 'Agility of Security Practices and Agile Process Models: An Evaluation of Cost for Incorporating Security in Agile Process Models'. ENASE 2020 - 15th International Conference on Evaluation of Novel Approaches to Software Engineering, 2020.

[16] keramati, H., Hosseinabadi, M. 'Integrating Software Development Security Activities with Agile Methodologies'. International Conference on Computer Systems and Applications. Doha, Qatar: IEEE, 2008.

[17] Likert. R. 'A Technique for the Measurement of Attitudes'. Archives of Psychology, Vol. 22, No. 140, pp. 1-55, 1932.

[18] Sharma, A., Bawa, R. 'Identification and integration of security activities for secure agile development'. International Journal of Information Technology volume 14, pages1117–1130, 2020.

[19] Maier, P., Zhendong, M., Bloem, R. 'Towards a secure scrum process for agile web application development'. Proceedings of the 12th International Conference on Availability, Reliability and Security;1-8, 2017.

[20] Van der Stock, A., Glas, B., Smithline, N., Gigler, T. OWASP Top 10 – 2021. https://owasp.org/Top10/es/, 2020.

[21] Shostack, A. 'Threat Modeling: Designing for Security'. Wiley, 2014.

[22] Pohl, C., Hof, H. 'Secure Scrum: Development of Secure Software with Scrum'. MuSe - Munich IT Security Research Group Munich University of Applied Sciences, 2015.

[23] Creswell, J.W. 'Research design: Qualitative, quantitative, and mixed methods', 2003.

[24] Mertens, D.M. 'Research methods in education and psychology: Integrating diversity with quantitative and qualitative approaches. (2nd ed.) Thousand Oaks: Sage, 2005.

[25] Hernández, J., y Sarmiento, I. 'Prototipo de herramienta de cómputo portable para el desarrollo de modelos de negocios basado en Business Model Canvas. Ciencias Huasteca Boletín Científico de la Escuela Superior de Huejutla', 4(7), 1-12, 2016.

[26] De León, J., Pérez, J. V. & Boza, B. 'Introducción a las técnicas de muestreo', (9th ed.). Ediciones Pirámide, 2016.

[27] McKnight, P. E., & Najab, J. Mann-Whitney U Test. En N. J. Salkind (Ed.), 'Encyclopedia of Research Design' (pp. 715-718). SAGE Publications, Inc., 2010.

[28] CveDetails. Vulnerabilities By Types/Categories. CveDetails. https://www.cvedetails.com/vulnerabilities-by-types.php, 2023.

[29] Celina Oviedo, H., Campo Arias, A. 'Aproximación al uso del coeficiente alfa de Cronbach Revista Colombiana de Psiquiatría', vol. XXXIV, núm. 4, 2005, pp. 572-580.Asociación Colombiana de Psiquiatría Bogotá, D.C., Colombia.

Apéndice

En la siguiente Tabla se muestra todas las preguntas del cuestionario formulado como parte del presente trabajo de investigación, así como también el tipo de pregunta y su respectiva escala. Los resultados obtenidos de la encuesta han sido analizados desde un enfoque estadístico y son el sustento para validar el marco del trabajo propuesto

Preguntas	Tipo de Pregunta	Escala
¿Considera usted que una de las principales causas de ataques a un software web es que durante el proceso de su desarrollo no se incorporaron ACTIVIDADES DE DESARROLLO SEGURO?	Cerrada con escala de Likert	5 = Totalmente de Acuerdo 4 = De Acuerdo 3 = Neutral 2 = En Desacuerdo 1 = Totalmente en Desacuerdo
¿Con qué frecuencia sería atacado un software web si durante el desarrollo no se incorporaron ACTIVIDADES DE DESARROLLO SEGURO?	Cerrada con escala de Likert	5 = Muy Frecuente 4 = Frecuente 3 = Neutral 2 = Poco Frecuente 1 = Nada Frecuente
¿Cuál es el nivel de impacto que se da en una organización luego de producirse un ataque a su sistema informático?	Cerrada con escala de Likert	5 = Nada Negativo 4 = Poco Negativo 3 = Neutral 2 = Negativo 1 = Totalmente Negativo
¿Cuántos años de experiencia tiene en el desarrollo de software usando SCRUM?	Cerrada con escala ordinal	Más de 10 años Entre 7 y 10 años Entre 5 y 7 años Entre 3 y 5 años Menor a 3 años
¿Cuál es su posición respecto a la incorporación de un especialista en desarrollo seguro de software usando SCRUM?	Cerrada con escala de Likert	5 = Totalmente de Acuerdo 4 = De Acuerdo 3 = Neutral 2 = En Desacuerdo 1 = Totalmente en Desacuerdo
¿Qué tan costoso considera usted que significaría incorporar la actividad de seguridad "Modelado de Amenazas" durante el proceso de desarrollo de software usando SCRUM?	Cerrada con escala de Likert	5 = Nada Costoso 4 = Poco Costoso 3 = Neutral 2 = Costoso 1 = Muy Costoso
¿Qué tan beneficioso considera usted que significaría incorporar la actividad de seguridad "Modelado de Amenazas" durante el proceso de desarrollo de software usando SCRUM?	Cerrada con escala de Likert	5 = Muy Beneficioso 4 = Beneficioso 3 = Neutral 2 = Poco Beneficioso 1 = Nada Beneficioso
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Entrenamiento en Políticas y Actividades de Desarrollo Seguro" durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Identificación de Roles" durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Análisis de Riesgos de Seguridad (Identificación de Ataques)" durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Análisis de Requisitos de Seguridad durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Modelado de Amenazas" durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Análisis de Código Estático" durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Pruebas de Penetración" durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas
¿Qué tiempo adicional considera usted que demandaría incorporar la actividad de seguridad "Planificación de Respuesta a Incidentes" durante el proceso de desarrollo con SCRUM?	Cerrada con escala ordinal	Más de 10 horas Entre 8 y 10 horas Entre 5 y 8 horas Entre 2 y 5 horas Menos de 2 horas