GUÍA DE PRÁCTICAS DE SEGURIDAD PARA UN MARCO ÁGIL DE DESARROLLO.

LUIS GABRIEL CADAVID GÓMEZ

UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERÍA
MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN
MEDELLÍN
2017

GUÍA DE PRÁCTICAS DE SEGURIDAD PARA UN MARCO ÁGIL DE DESARROLLO.

LUIS GABRIEL CADAVID GÓMEZ

Trabajo de grado para optar al título de:
MAGISTER EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Director
ANDRÉS FELIPE MUÑETON
Magister en Ingeniería de Sistemas

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
FACULTAD DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN
MAESTRÍA EN TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN
MEDELLÍN
2017

DECLARACIÓN ORIGINALIDAD

"El abajo firmante, declara que esta tesis (o trabajo de grado) no ha sido presentada para optar a un título, ya sea en igual forma o con variaciones, en esta o cualquier otra universidad". Art. 82 Régimen Discente de Formación Avanzada, Universidad Pontificia Bolivariana.

FIRMA AUTOR _____

LUIS GABRIEL CADAVID GÓMEZ
CC 71363769

AGRADECIMIENTOS

A Dios, por darme las fuerzas y la paciencia necesaria para seguir adelante con el trabajo, por animarme a seguir en los momentos difíciles

A mi esposa quien me apoyó desde el principio para realizar mis estudios, se sacrificó he hizo todo lo posible para que yo pudiera estudiar brindándome siempre aliento para seguir adelante.

A mi hija María Fernanda por entender que su papá no podía dedicarle todo el tiempo que ella merece., por apoyarme y brindarme su amor incondicional.

A mi Papá, a mi Mamá y a mi tía Rogeria, que me enseñaron lo importante que es el estudio y que me brindaron todas las posibilidades para poder alcanzar las metas que me he propuesto en la vida.

A mis hermanas por su amor y apoyo en los momentos en las que las necesité en durante mis estudios.

A Andrés Felipe Muñeton una persona que me brindo su conocimiento, su tiempo y siempre estuvo pendiente de que pudiera sacar este trabajo adelante.

A mis jefes María Eugenia Betancur, Antonio José Pérez Tobón quienes me brindaron la oportunidad de estudiar y vieron en mi potencial para aprender de este particular mundo de la seguridad informática.

A Lina Maria Garcia por permitirme trabajar con recursos de su dirección, al equipo colibrí y el equipo de devops por permitirme aprender de ellos y tener el tiempo de ver mi trabajo y realimentarlo con su valiosa experiencia.

A mis compañeros de trabajo en especial a Rubén Darío Sepúlveda y Gustavo Pérez Vera, personas que me apoyaron con su conocimiento durante todo este proceso.

A la Universidad Pontificia Bolivariana, por todos los conocimientos brindados durante mi formación como Magister.

Y a todos los que de alguna forma contribuyeron con este trabajo. Gracias.

RESUMEN

Con el rápido crecimiento y las ventajas ofrecidas por las metodologías ágiles, Empresas Públicas de Medellín (EPM) empezó a adoptarlas para el desarrollo de software en la empresa, basándose principalmente en Scrum. A su vez EPM tenía definidas políticas y lineamientos para el desarrollo seguro de software, con el fin de tener aplicaciones seguras y minimizar de manera temprana la inyección de posibles vulnerabilidades en el software desarrollado.

Buscando brindar entornos seguros a sus usuarios EPM implementó mecanismos de control de seguridad a varios de sus procesos, uno de estos controles son las pruebas de seguridad automatizadas antes de cada despliegue a producción. Los resultados de estas pruebas han mostrado que a pesar de las definiciones de seguridad los desarrollos que están a punto de pasar a producción presentan vulnerabilidades de criticidad media y alta.

Las consecuencias de este tipo de vulnerabilidades es que se debe impedir el despliegue en producción de los desarrollos, generando impactos financieros (Correcciones al final del ciclo) para los distintos negocios que los han solicitado, también impidiendo que se pueda contar con los desarrollos en las fechas esperadas.

Debido a que los lineamientos y políticas de desarrollo seguro fueron diseñadas pensando en metodologías de desarrollo tradicionales y no tuvieron en cuenta las metodologías ágiles. En este trabajo se plantea una guía que incluye prácticas de seguridad que pueden ser utilizadas dentro del marco ágil de desarrollo de EPM y que aportan al cumplimento de los lineamientos y políticas de seguridad definidos.

Para la realización de la guía se consultaron las distintas políticas y lineamientos de seguridad de EPM para extraer las prácticas que aplicaban para el desarrollo de software seguro. Luego a través de encuestas se identificaron los impedimentos para que estos lineamientos y políticas se aplicaran en EPM. Con los impedimentos claros se realizó una revisión bibliográfica de otros posibles impedimentos que se pudieran identificar, así mismo se buscaron prácticas de seguridad para metodologías ágiles que ayudaran a remover los impedimentos encontrados.

Finalmente se evaluaron las prácticas identificadas buscando las que más se ajustaran a las necesidades y realidades de EPM, generando como resultado una guía para la utilización de estas prácticas en la metodología de desarrollo ágil de EPM.

Palabras Clave: Scrum, Seguridad, Desarrollo seguro de software, Metodologías ágiles de desarrollo

ABSTRACT

With the rapid growth and advantages offered by ágile methodologies, Empresas Públicas de Medellín (EPM) began to adopt Scrum for its software development process. Besides, the company had defined policies and guidelines to secure the software development, looking to have secure applications and to minimize the injection of possible vulnerabilities in early stages.

Looking to provide secure environments to its users EPM implemented security control mechanisms in some of its processes, one of these controls is the automated security tests before the deployment to the production enviroments. The results of these tests have been showing that despite the security definitions the developments that are about to be deployed to production enviroments have medium and high criticality vulnerabilities.

The consequences of finding this type of vulnerabilities are: the deployment has to be prevented, generating financial impacts (Corrections at the end of the cycle) and generating delay in the expected dates.

Because the guidelines and policies for secure software development were designed thinking about traditional development methodologies and did not consider ágile methodologies. This work intends to define a guide that includes security practices that can be used within the EPM's ágile framework for software development and contribute to comply with defined security guidelines and policies.

For the implementation of the guide, EPM´s security policies and guidelines were consulted to extract only the related with secure software development. Then through surveys it was possible to extract impediments to comply with the stablished guidelines and policies. With these impediments identified, a bibliographic review was conducted looking to recognize other impediments, as well as security practices that helped to remove the impediments found.

Finally, the identified practices were evaluated, looking for those that best fit EPM's needs and realities, generating as a result a guide with security practices for the agile methodology used in EPM.

Key Words: Scrum, Security, Secure software development, Agile development methodologies.

CONTENIDO

LISTA DE TABLAS

LISTA DE ILUSTRACIONES

INTRODUCCIÓN

En la actualidad Empresas Públicas de Medellín (EPM) viene cambiando la forma como trabaja en sus proyectos de desarrollo de software, para esto viene asesorándose e implementando metodologías ágiles de desarrollo, entre las cuales destaca el uso de Scrum como su metodología base, debido a que ofrece flexibilidad en el proceso de desarrollo generando la capacidad de responder a rápidamente a cambios[1]. A la par también el Gobierno Colombiano junto con todas las empresas públicas vienen trabajando fuertemente en un proyecto denominado gobierno en línea[2][3] que pretende que el ciudadano pueda acceder a muchos de los servicios ofrecidos por las empresas del estado a través de internet. Para ser exitoso, este proyecto requiere que las empresas del estado ofrezcan un alto nivel de seguridad en las aplicaciones que ponen a disposición de la ciudadanía asegurándose de dar un correcto manejo y protección de los datos del ciudadano[2].

Teniendo en cuenta esta situación EPM ha venido incrementando las exigencias realizadas en términos de seguridad con respecto a las aplicaciones que se despliegan en ambientes productivos, debido a estas exigencias se han implementado controles de seguridad, como las pruebas de seguridad. Dentro de los resultados que arrojan estas pruebas se ha observado que las aplicaciones desarrolladas internamente vienen presentando vulnerabilidades de tipo medio o alto justo antes de salir a producción. Dentro de las causas analizadas para esta situación se encuentra que, aunque la empresa tiene definidas políticas y lineamientos para el desarrollo seguro de software, estos no fueron definidos pensando en metodologías ágiles de desarrollo. Lo que tiene como consecuencia que las prácticas de seguridad derivadas no necesariamente puedan ser aplicadas de manera directa durante la construcción del software.

Para abordar este problema algunos autores hacen una revisión de los retos que puede representar para una metodología ágil al incorporar las prácticas de seguridad más conocidas y utilizadas [4][5]. Varios trabajos se enfocan en definir prácticas de seguridad que pueden ayudar a agregar seguridad a los desarrollos realizados bajo el modelo de las metodologías ágiles [6][7][8] [9][10] mientras que otros se enfocan en proponer variaciones a metodologías como Scrum con prácticas de seguridad explicitas e integradas [11][12][13][14].

En este trabajo para mitigar la problemática mencionada se genera una guía de desarrollo seguro de software utilizando prácticas de seguridad que fueron concebidas o adaptadas para funcionar dentro de un marco ágil de desarrollo de software. Esta guía se obtiene mediante el análisis de las

principales causas por las cuales no se sigue lo que hoy está definido en la compañía. Realizando una revisión bibliográfica para identificar que prácticas ayudan a remover las causas de incumplimiento identificadas y seleccionar las que mejor se adaptan al contexto de la compañía.

El trabajo se presenta de la siguiente manera. En el capítulo uno y dos se plantea la problemática que se describió brevemente en esta introducción junto con su justificación, en el capítulo tres se plantean los objetivos de este proyecto, en el capítulo cuatro se presenta el marco referencial del trabajo para más adelante, en el capítulo cinco explicar la metodología que se siguió para llevar a cabo el trabajo.

El desarrollo del trabajo se realiza en los capítulos del seis al nueve en donde se inicia diagnosticando los impedimentos para el cumplimento de las políticas y lineamientos establecidos (capitulo seis). Más adelante (capitulo siete) se determinan las prácticas que hay en la literatura que ayudan a mitigar los impedimentos encontrados en el capítulo seis.

En el capítulo ocho se seleccionan las prácticas que más se acomodan al contexto organizacional de EPM y se realiza la guía producto de este trabajo. En el capítulo nueve se recibe realimentación acerca de la guía generada y se hacen los ajustes necesarios a la misma.

Finalmente, en los capítulos diez y once se realizan las conclusiones y se plantea el trabajo futuro.

1. DESCRIPCIÓN DEL PROBLEMA

En la actualidad el software desarrollado para Empresas Públicas de Medellín EPM antes de su despliegue en ambientes productivos debe ser sometido a pruebas automatizadas de seguridad con el fin de verificar que no existan vulnerabilidades críticas o medias que puedan ser explotadas cuando el software sea desplegado en producción.

Desde que se implementó este control se tiene registro que 20 de las 22 revisiones hechas arrojan que poseen vulnerabilidades críticas o medias basados en el marco de referencia común para vulnerabilidades NVD CVSS[15], Tabla 1, Ilustración 1. Estas vulnerabilidades se están encontrando solo al final del ciclo antes de entrar a producción esto hace más retrasar los tiempos de salida a producción y hace más costosa su corrección que si se hubiera detectado de manera temprana [16] [17]. De esta manera se genera insatisfacción en los clientes de la Gerencia de tecnologías de información, por los tiempos de espera y los sobrecostos que son trasladados a ellos.

Aplicación	*Fecha*	*Altas*	*Medias*	*Bajas*	*Informativas*
App1	17/08/2016	0	0	4	1
App2	4/04/2016	1	17	8	29
App3	14/04/2016	2	4	3	2
App4	23/08/2016	5	8	11	7
App5	2/03/2016	4	2	3	1
App6	16/05/2016	4	3	1	0
App7	18/03/2016	1	21	3	9
App8	28/07/2016	0	3	3	1
App9	26/05/2016	0	17	4	1
App10	27/07/2016	16	32	3	8
App11	22/07/2016	1	7	2	2
App12	23/05/2016	0	5	0	1
App13	9/01/2015	0	0	0	2
App14	8/01/2015	27	117	18	123
App15	27/02/2015	0	10	5	7
App16	12/02/2015	0	6	5	9
App17	14/01/2015	0	20	3	1
App18	10/02/2016	2	4	3	5
App19	27/03/2015	2	9	11	9

App20	26/05/2015	1	3	5	2
App21	26/05/2015	0	32	7	50
App22	6/05/2015	2	10	17	95

Tabla 1 Aplicaciones listas para producción y sus vulnerabilidades

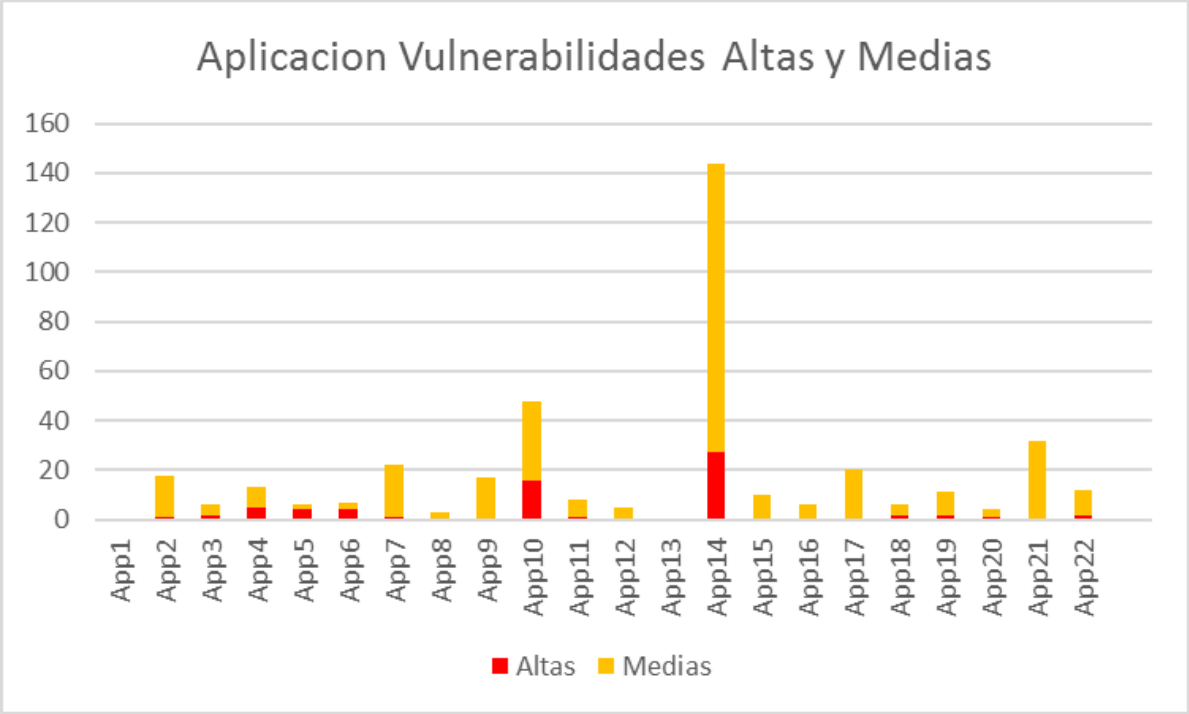


Ilustración 1 Aplicaciones con vulnerabilidades altas y medias

Con la implementación de este control se ha encontrado que de los últimos 22 desarrollos entregados a producción el 91% contiene vulnerabilidades de tipo medio o alto según el sistema de calificación común de vulnerabilidades CVSS[15], también se encontró que el 56% de las aplicaciones tienen de 1 a 27 vulnerabilidades medias y que en promedio cada aplicación llega para su despliegue en producción con 3 vulnerabilidades altas.

Las principales causas para que el software contenga este tipo de vulnerabilidades son que los equipos de desarrollo no hacen pruebas de seguridad, los analistas encargados de hacer requerimientos funcionales y técnicos no los hacen teniendo en cuenta la seguridad y que no hay suficiente análisis de los riesgos en las funcionalidades solicitadas. Esto se puede resumir en que, aunque en EPM existen políticas, lineamientos y procedimientos para el desarrollo de software seguro, esta serie de buenas prácticas organizacionales no son tenidas en cuenta a la hora de la construcción del software y tampoco son exigidas por los analistas de TI.

Finalmente, los lineamientos de seguridad establecidos para EPM fueron diseñados para metodologías de desarrollo tradicionales, sin embargo, EPM desarrolla la mayoría de su software a la medida utilizando metodologías ágiles. Los métodos para alcanzar la seguridad orientados a metodologías tradicionales de desarrollo de software como el caso de Clean Room, Construction by Correctness, CMMI- DEV, entre otros, utilizan conceptos rígidos y formales de planeación temprana y diseño detallado [8], [12] lo cual hace difícil su implementación en metodologías ágiles ya que estas son conocidas por reducir la documentación, las largas fases iniciales de diseño y obtención de requisitos para enfocarse en producir código ejecutable[8].

Esta última situación es la que se abordará como problemática del presente trabajo. La razón por la cual se selecciona esta problemática es su relación con las otras dos problemáticas expuestas; sí las políticas y lineamientos de seguridad no se pueden utilizar directamente en las metodologías ágiles entonces: (i) para los desarrolladores no será posible aplicarlas de manera adecuada y (ii) tampoco será posible que los analistas de la gerencia de TI puedan evaluar su adecuado cumplimiento.

2. JUSTIFICACIÓN

En el contexto de EPM cuando se encuentran vulnerabilidades altas o medias en una aplicación que va a ser desplegada en producción esta aplicación no se despliega y se devuelve a fases anteriores para su debida corrección. Posteriormente, se ejecutan nuevamente las pruebas verificando la remoción de las vulnerabilidades previamente identificadas y prestando atención en que no se hayan inyectado nuevas. Una vez se ha cumplido con esta condición, se autoriza su despliegue en ambiente productivo.

Estos frecuentes reprocesos conllevan impactos negativos para la gerencia de tecnologías de información y a su vez para todo EPM. Uno de los impactos más relevantes es el costo de corregir fallas que son detectadas solo hasta el final de ciclo de desarrollo, durante el despliegue; como se presenta en [5] y [6], dicho costo sería menor si las fallas fueran identificadas desde etapas tempranas. Adicionalmente, cuando se detiene la salida a producción de un desarrollo se genera insatisfacción del cliente y una mala imagen de la gerencia de tecnologías de información pues el cliente entiende que el desarrollo solicitado está terminado y listo para ser utilizado.

Debido al desconocimiento de las consecuencias de autorizar el despliegue de un desarrollo con vulnerabilidades el equipo de seguridad se empieza a percibir como un impedimento y no como un apoyo para brindar soluciones de calidad a los usuarios.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Generar una guía con prácticas de seguridad para el marco ágil de desarrollo de EPM, que contribuya a mejorar el cumplimiento de los lineamientos, políticas y procedimientos de seguridad.

3.2 OBJETIVOS ESPECÍFICOS

1. Determinar los impedimentos por los cuales no se cumple con los lineamientos, políticas y procedimientos de seguridad.
2. Identificar prácticas de seguridad aplicadas en marcos ágiles que contribuyan a remover los impedimentos para el cumplimiento de las políticas y lineamientos de seguridad.
3. Definir prácticas de seguridad a incluir en guía para el marco de desarrollo ágil de EPM.
4. Analizar viabilidad de la aplicación de la guía propuesta utilizando cuestionarios

4. MARCO REFERENCIAL

4.1.MARCO CONTEXTUAL

En la actualidad, EPM está cambiando la forma de implementar sus proyectos de desarrollo de software, reemplazando el uso de metodologías tipo cascada por metodologías ágiles, específicamente Scrum. Al ser una iniciativa reciente, las guías de desarrollo seguro, cuyo fin es garantizar el cumplimiento de las políticas de seguridad del software de la organización, aún no han sido adaptadas al nuevo esquema de trabajo ágil.

Los lineamientos y procedimientos que actualmente tiene EPM para desarrollo seguro fueron creados pensando en metodologías de desarrollo tradicionales, lo cual hace difícil que puedan ser tomados directamente y utilizados con las metodologías actuales, esto principalmente se debe a que muchos de ellos deben ser ajustados para engranar dentro del contexto de los principios y

valores ágiles. Esta dificultad es evidenciada en trabajos como [4] en donde se presenta una tabla comparativa con prácticas de seguridad y las dificultades que puede representar utilizarlas en una metodología ágil. En el trabajo se concluye que más de la mitad de las prácticas de seguridad se encuentran en las categorías que presentan mayor dificultad para su adaptación a metodologías ágiles. También, en [5] los autores realizan un trabajo similar donde extraen de una búsqueda bibliográfica los desafíos planteados al utilizar prácticas de seguridad en metodologías ágiles; para esto toman los valores ágiles, las prácticas ágiles y algunas prácticas de seguridad. De esta evaluación obtienen que de un total de 20 desafíos identificados 14 de ellos son válidos y 6 inválidos. Como conclusión de este trabajo los autores afirman que desarrollar software seguro utilizando metodologías ágiles es desafiante.

Paralelamente a la incursión de EPM en las metodologías ágiles, el gobierno colombiano ha venido exigiendo a sus instituciones la implementación de la estrategia de gobierno en línea [2] que se basa en cuatro componentes: TIC para servicios, cuyo objetivos es proveer trámites y servicios a través de medios electrónicos; TIC para el gobierno abierto, su principal objetivo es apoyarse en las tecnologías de la información para hacer un gobierno más abierto y transparente; TIC para la gestión que comprende la gestión y la mejora de procesos internos e intercambio de información así como el aprovechamiento de la información para la toma de decisiones y finalmente seguridad y privacidad de la información componente que pretende proteger la información y los sistemas que la contienen del acceso, uso, divulgación, interrupción o destrucción no autorizada [2].

Una consecuencia de Gobierno en Línea es que EPM deberá exponer varios de sus sistemas de información en Internet donde puedan ser utilizados públicamente. Esto supone que las aplicaciones deben tener un alto componente de seguridad para garantizar la privacidad de la información de los usuarios y de EPM.

Para este tipo de exigencias, EPM definió una serie de prácticas de seguridad para el desarrollo de software las cuales fueron dadas a conocer a la organización y reglamentadas por medio de su definición dentro de las políticas y lineamientos para el desarrollo de software de la gerencia de TI.

Estas prácticas de seguridad fueron definidas en una época en la que todos los proyectos de desarrollo de software se realizaban utilizando metodologías en cascada y fueron especificadas pensando en las diferentes fases de esta metodología: requisitos, diseño, implementación, verificación y mantenimiento. Dentro de las prácticas se encuentran lineamientos para

codificación segura, técnicas de elicitación de requisitos de seguridad, prácticas para el diseño de aplicaciones seguras, técnicas para revisiones de seguridad entre otras.

Adicionalmente, hace más de un año se implementó un control por medio del cual se pretende detectar vulnerabilidades en el software que se va a desplegar en ambientes de producción, control que evidenció que las aplicaciones que se pretenden desplegar en ambientes productivos no llegan con la mejor calidad en términos de seguridad.

4.2.MARCO CONCEPTUAL

4.2.1. METODOLOGÍAS ÁGILES

El movimiento ágil en la industria del software vio la luz con el manifiesto de desarrollo ágil de software publicado por un grupo de practicantes y consultores de software en 2001, su objetivo era dar a conocer los valores que debería tenerse en cuenta cuando un equipo de trabajo requiriese desarrollar software de manera ágil respondiendo de manera oportuna a los diferentes cambios planteados por el cliente durante el ciclo de vida del proyecto.

Los valores en los que se enfocó este movimiento fueron:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

Definiciones recientes de metodologías ágiles de desarrollo de software dicen que la agilidad en el desarrollo de software es la capacidad de un equipo de responder e incorporar cambios en los requerimientos de usuario de manera eficiente y efectiva durante el ciclo de vida del proyecto[18]. También en definiciones más específicas se dice que son un conjunto de métodos evolutivos basados en el mejoramiento iterativo y en procesos de desarrollo oportunos donde cada iteración son mini proyectos auto contenidos con actividades que comprende el análisis de requerimientos, diseño, implementación y pruebas. [19].

4.2.2. SCRUM

El término de Scrum se deriva del rugby, es una formación en la que con trabajo en equipo se logra obtener la posesión de la pelota. Scrum fue desarrollado para gestionar el proceso de desarrollo de los sistemas introduciendo ideas de flexibilidad, adaptabilidad y productividad. Scrum no define ninguna técnica específica de desarrollo de software para la fase de implementación. Se concentra en cómo deben funcionar los miembros de un equipo para conseguir flexibilidad en los sistemas cuando se trata de un ambiente en constante cambio. La idea principal de Scrum es que en el desarrollo de sistemas generalmente hay variables cambiantes durante el proceso lo que hace que su desarrollo sea impredecible y complejo. Haciendo requerido que haya flexibilidad en el proceso de desarrollo de tal manera que sea capaz de responder a cambios [1].

Scrum funciona de la siguiente manera, al principio de la iteración el equipo revisa que se debe hacer, luego selecciona lo que creen que puede ser una funcionalidad incremental y potencialmente entregable para el final de la iteración. Luego el equipo hace su mejor esfuerzo en el resto de la iteración para al final presentar una funcionalidad incremental que los interesados puedan inspeccionar.

El corazón de Scrum es la iteración, el equipo mira los requerimientos, toma en consideración la tecnología disponible y evalúa sus propias destrezas y capacidades. Luego colectivamente determina como construir la funcionalidad, modificando su enfoque diariamente a medida que se encuentran con nuevas dificultades y sorpresas. El equipo se da cuenta que actividades se deben realizar y selecciona la mejor manera de hacerlo. Este proceso creativo es el corazón de la productividad de Scrum [20].

Scrum tiene tres roles, uno es el dueño del producto o “Product Owner” es responsable del éxito del proyecto desde el punto de vista de los interesados, determina la visión del producto, gestiona las expectativas de los interesados, maximiza la rentabilidad del producto determinando prioridades de las características según avanza el proyecto. El equipo de desarrollo es el responsable de la construcción y calidad del producto. El equipo es auto organizado y su objetivo a cumplir es transformar las funcionalidades comprometidas en software funcional y calidad productiva. Finalmente, el Scrum Master es el responsable por el proceso de Scrum debe asegurar que todos sigan sus reglas y prácticas. Es el responsable de enseñar Scrum a todos los involucrados en el proyecto así como de implementar Scrum de tal manera que encaje en la cultura de una organización entregando los resultados esperados[20], [21].

Scrum tiene tres elementos principales, el Product Backlog que es un listado de ítems o características del producto a construir, mantenido y priorizado por el Product Owner. un Sprint Backlog que son el conjunto de ítems del Product Backlog que se construirán en la iteración o Sprint y finalmente el incremento funcional potencialmente entregable que es el entregable del Sprint[21].

Las principales características de Scrum son que el desarrollo de software se realiza en iteraciones que no duran más de 30 días llamadas Sprint, además el resultado de cada Sprint es un entregable que agrega funcionalidad y es mostrado al cliente. La segunda característica es que durante las fases de los proyectos se realizan reuniones en donde destaca la reunión que realiza diariamente el equipo de desarrollo para coordinar e integrar [22].

4.2.3. DESARROLLO DE SOFTWARE SEGURO Y ASEGURAMIENTO DE SOFTWARE

El aseguramiento de software en palabras simples es la confianza que un sistema cumple con todos sus requerimientos de seguridad. Mayormente aquellos de interés de los usuarios y dueños del producto, esta confianza se basa en evidencia específica recolectada y evaluada a través de técnicas de aseguramiento. [4]

Software seguro según Gary McGraw referenciado por Oueslati et al. [5] es el software que continua funcionando correctamente aún bajo ataques malicioso intencionados.

Goertzel en [23] define software seguro como software que no puede ser intencionalmente alterado o forzado a fallar. En definitiva, un software que sigue siendo fiable (adecuado y predecible) a pesar de esfuerzos intencionales para comprometer su fiabilidad. Adicionalmente Goertzel, nos muestra varias definiciones de aseguramiento de software entre la que destaca la del departamento de defensa de Estados Unidos que dice que el aseguramiento de software es el nivel de confianza en que las funciones del software se encuentran según lo previsto y que se encuentran libres de vulnerabilidades diseñadas intencionalmente o no y libre de vulnerabilidades insertadas como parte del software.

4.3.ESTADO DEL ARTE

Debido a que EPM viene trabajando con Scrum como la metodología base de su marco de desarrollo, la revisión de literatura tiene en cuenta los diferentes trabajos para involucrar

seguridad en Metodologías ágiles, sin embargo, se presta especial atención a los estudios centrados en Scrum pues es la metodología objetivo.

De acuerdo con [24] los mecanismos establecidos para la seguridad de la información y la seguridad organizacional se consideran rígidos para responder a los cambios y avances que se presentan en el cambiante entorno de la seguridad, en donde existe la necesidad de un método más ágil que para hacer frente a las nuevas amenazas y vulnerabilidades. De este modo el autor sugiere que los mecanismos de seguridad tradicionalmente establecidos ya no son eficaces cuando se utilizan con metodologías de desarrollo de software adaptadas para las necesidades del entorno actual como las metodologías ágiles.

Uno de los retos que presenta el introducir prácticas de desarrollo seguro es que entran en conflicto con las prácticas de desarrollo ágil, en [4] los autores examinan como las prácticas de seguridad convencional se adaptan a las metodologías ágiles de desarrollo de software.
Para esto presentan una tabla en donde clasifican las prácticas de seguridad según su compatibilidad con las metodologías ágiles, para lo cual definieron cuatro categorías de dificultad "match" en donde la práctica encaja de manera natural, "independent" donde las prácticas son independientes de la metodología de desarrollo, "semi-automated" que son las prácticas que tal vez puedan automatizarse, aunque de todos modos requerirán del acompañamiento de un experto en seguridad. Finalmente, mismatch en donde la práctica de seguridad choca con los principios y prácticas del desarrollo ágil. En el artículo dividen las prácticas en dos grandes grupos (match, independent) y (semi automated, mis-match), donde las primeras se pueden integrar con metodologías ágiles fácilmente y las segundas requieren ajustes para poderlo hacer (Tabla 2).

Método o técnica de seguridad		Ajuste (Match)	Independiente (Independent)	Semi - automatizada (semi - automated)	Sin ajuste (MIsmatch)
Requerimientos	Guías		x		
	Análisis de especificaciones				x
	Revisión				x
Diseño	Aplicación de enfoques arquitectónicos específicos		x		
	Uso de principios de diseño seguro		x		
	Validación formal				x
	Validación informal				x
	Revisión interna	x			
	Revisión externa				x
Implementaciones	Análisis informal de correspondencia				x
	Pruebas de requerimientos			x	
	Validación informal				x
	Validación formal				x
	Pruebas de seguridad			x	
	Pruebas de vulnerabilidad y penetración			x	
	Pruebas de análisis en profundidad				x
	Análisis de seguridad estáticos			x	
	Lenguajes de programación y herramientas de alto nivel		x		
	Adherencia a estándares de implementación		x		
	Uso de control de versiones y seguimiento de cambios		x		
	Autorización de cambios				x
	Integración de procedimientos		x		
	Uso de herramientas de generación de productos		x		
	Revisión interna	x			
	Revisión externa				x
	Evaluación de seguridad				x

Tabla 2 prácticas de seguridad vs ágiles [4]

Para tratar de minimizar estas discrepancias entre las técnicas del segundo grupo y las metodologías ágiles los autores afirman que las técnicas clasificadas como “semi-automated” pueden de alguna forma acomodarse dentro de las metodologías ágiles, sin representar altos costos, por ejemplo, pruebas estáticas de seguridad automatizadas.

Para las clasificadas como mismatch, los autores, dicen que estas técnicas difícilmente se pueden implementar durante todas las iteraciones del proyecto, debido a que se soportan en documentación extensiva, la participación de terceros y la gran cantidad de tiempo que puede agregar al proyecto.

Los autores ofrecen dos alternativas de solución a sus hallazgos, la primera consiste en inventar una nueva metodología que sea ágil y amigable con la seguridad, la otra alternativa seria hacer estas actividades al menos dos veces en todo el proyecto, una en la iteración cerca de la mitad y otra en una iteración casi al final.

Como conclusión los autores reconocen que hay conflictos entre las metodologías ágiles y las prácticas de seguridad y proponen una posible solución, sin embargo, plantean la pregunta de qué se necesita hacer para que las prácticas de seguridad se integren efectivamente a las

metodologías ágiles. Los autores se preguntan si será necesario que se replanteen los pesados procesos de aseguramiento de software.

Por otro lado según [5] aplicar prácticas de seguridad en metodologías ágiles presenta desafíos debido a que las metodologías ágiles apoyan los cambios de requisitos, prefieren entregas frecuentes, y sus prácticas no incluyen actividades de ingeniería de seguridad. Por esto los autores hacen una revisión a la literatura acerca de los retos que puede representar el desarrollo de software seguro, cuando se utiliza una metodología ágil.
En este trabajo se plantea resolver las siguientes preguntas: ¿Cuáles son los desafíos del desarrollo de software seguro usando los métodos ágiles propuestos en la literatura? ¿Son desafíos válidos?
Para contestar las preguntas anteriormente planteadas los autores identifican los valores y principios ágiles (Ilustración 2) (Ilustración 3Ilustración 3 Principios para el desarrollo ágil de software)

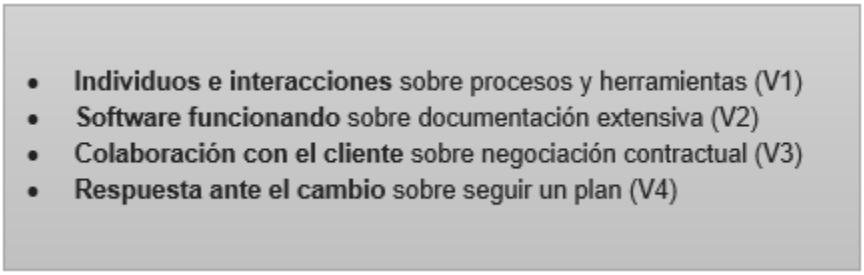


Ilustración 2 valores ágiles [5]

P1	Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software con valor.
P2	Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
P3	Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
P4	Los responsables del negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
P5	Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
P6	El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
P7	El software funcionando es la medida principal de progreso.
P8	Los procesos ágiles promueven el desarrollo sostenido. Los promotores, desarrolladores y usuarios debemos mantener un ritmo constante de forma indefinida.
P9	La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
P10	La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
P11	Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
P12	A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo para, a continuación, ajustar y perfeccionar su comportamiento en consecuencia.

Ilustración 3 Principios para el desarrollo ágil de software [5]

Los autores también propusieron el uso de cinco prácticas de seguridad (Tabla 3Tabla 3 Riesgos de seguridad y prácticas de aseguramiento) como criterios para la evaluación de los desafíos para el desarrollo de software seguro.

Código	Práctica de aseguramiento	Descripción
S1	Especificar políticas de seguridad	Identificar las necesidades de seguridad para cumplir con términos legales, políticas y requerimientos organizacionales considerando los objetivos de seguridad. Las políticas se convierten en requerimientos de seguridad.
S2	Entrenamiento en seguridad	Entrenar a los desarrolladores de software en el uso de librerías de criptografía y técnicas para el desarrollo de características de seguridad, estándares de codificación segura, realizar análisis de código utilizando herramientas estáticas y dinámicas y ejecutando pruebas de seguridad.
S3	Identificar vulnerabilidades	Identificar y categorizar los defectos del software. Estos defectos son (1) vulnerabilidades en el código fuente como buffer overflows, y entradas no validadas o (2) debilidades relacionadas con la arquitectura del software como el envío de daros no protegidos a una red pública
S4	Evaluar riesgos de seguridad	Identificar las amenazas de seguridad del software y evaluar la probabilidad de ocurrencia y sus impactos.
S5	Verificar y validar la seguridad	Realizar revisión de seguridad, análisis estático y dinámico de código fuente, revisiones de arquitectura, pruebas funcionales de las funcionalidades de seguridad y pruebas de intrusión. Desarrollar los argumentos de aseguramiento para evidenciar que las necesidades de seguridad del cliente se están cumpliendo. Un argumento de aseguramiento es un conjunto de objetivos de aseguramiento declarados que son respaldados por evidencia de aseguramiento derivada de múltiples fuentes, tales como pruebas de seguridad y revisión de seguridad del código fuente.

Tabla 3 Riesgos de seguridad y prácticas de aseguramiento [5]

Luego los autores hicieron una investigación en diferentes artículos para encontrar los desafíos de la implementación de prácticas de seguridad en metodologías ágiles. Para esto a través de referencias bibliográficas encontraron artículos relevantes de la literatura del tema y desecharon los artículos donde se planteaban desafíos cuya argumentación no era lo suficientemente fuerte, en esta revisión encontraron 20 desafíos. Como segundo paso para determinar si los desafíos identificados eran válidos los autores evaluaron los desafíos que encontraron con respecto a 4 valores ágiles, 12 principios ágiles y con respecto a 5 prácticas de seguridad a implementar.
En donde un desafío se clasifica como válido si es causado por uno de los valores y o principios ágiles o si es causado por alguna de las prácticas de seguridad.

Código	Desafío	Valor ágil	Principio ágil	Práctica de seguridad
Desafíos en el ciclo de vida del desarrollo.				
CH1.1	La actividad de obtención de requisitos no está incluida en las metodologías de desarrollo ágil.			
CH1.2	La actividad de evaluación de riesgo no está incluida en las metodologías de desarrollo ágil.			
CH1.3	Las actividades relacionadas con la seguridad necesitan ser aplicadas para cada iteración del desarrollo.		P1, P3	S1, S3, S4, S5
CH1.4	El tiempo de una iteración es limitado se ajusta a actividades de seguridad que consumen mucho tiempo.		P3	S1, S3, S4, S5
Desafíos en el desarrollo incremental				
CH2.1	La práctica de refactorización rompe las restricciones de seguridad.	V4	P9	S3, S5
CH2.2	Los cambios en los requerimientos y el diseño rompen los requerimientos de seguridad de sistema.	V4	P2	S1, S5
CH2.3	Los cambios continuos en el código hacen difícil completar las actividades de aseguramiento.	V4	P2	S3, S5
CH2.4	Los cambios en los requerimientos hacen que la trazabilidad se difícil para los objetivos de la seguridad.	V4	P2	S5
Desafío en aseguramiento				
CH3.1	Las evaluaciones de seguridad se favorecen de la documentación detallada.	V2	P7, P10	S4, S5
CH3.2	Las pruebas son, en general insuficientes para asegurar la implementación de los requerimientos de seguridad.		P7, P10	S5
CH3.3	Las pruebas en general no cubren todos los casos de vulnerabilidad.		P7, P10	S5
CH3.4	Las pruebas de seguridad en general son difíciles de automatizar.			
CH3.5	El cambio continuo de los procesos de desarrollo entra en conflicto con las necesidades de auditoría de procesos estables y uniformes.	V4	P11, P12	S5
Desafíos en conciencia y colaboración				
CH4.1	A menudo se descuidan los requerimientos de seguridad.		P7, P10	
CH4.2	Falta experiencia de los desarrolladores en software seguro.			
CH4.3	Al cliente le falta conciencia en seguridad.			
CH4.4	El rol de desarrollador debe estar separado del rol que revisa la seguridad para tener resultados objetivos.	V1	P4, P6	S5
Desafíos gestión de la seguridad				
CH5.1	Las actividades de seguridad incrementan el costo del software.			S1, S2, S3, S4, S5
CH5.2	No hay incentivos para que se desarrollen funcionalidades de seguridad en los incrementos iniciales.			
CH5.3	Las organizaciones comprometen las actividades de seguridad para acomodarse a cronogramas de entregas aceleradas.	V2	P3	S1, S3, S4, S5

Tabla 4 causas de los retos para desarrollar software seguro usando metodologías ágiles [5]

Los autores, realizan una tabla (Tabla 4) en donde enumeran cada uno de los desafíos que identificaron en su búsqueda bibliográfica, cada uno de los desafíos es enumerado con un código que empieza por CH y son agrupados en 5 categorías:

1. Software development life-cycle challenges. (Desafíos del ciclo de vida del desarrollo).
2. Incremental development challenges. (Desafíos del desarrollo incremental).
3. Security assurance challenges (Desafíos del aseguramiento).
4. Awareness and collaboration challenges. (Desafíos de concienciación y colaboración).
5. Security management challenges. (Desafíos de la gestión de la seguridad)

Al frente de cada uno de los desafíos hay tres columnas en donde se listan los valores ágiles (Ilustración 2), los principios ágiles (Ilustración 3) y las prácticas de seguridad (Tabla 3) que causan el desafío.

De la evaluación los autores obtuvieron que de 20 desafíos identificados 14 de ellos son válidos y 6 inválidos según el criterio explicado anteriormente.

Estos desafíos fueron clasificados en:

- Causados por los valores ágiles o principios ágiles y por prácticas de aseguramiento de software. (12 validos)
- Causados por los valores ágiles o principios ágiles. (1 validos)
- Causados por las prácticas de aseguramiento de software. (1 validos)
- No causados por los valores ágiles o principios ágiles o prácticas de aseguramiento de software. (6 Inválidos).

Como resultado de su trabajo los autores evidencian que desarrollar software usando metodologias ágiles representa un desafio y que su trabajo contribuye a adaptar metodos de desarrollo ágil y prácticas de seguridad para habilitar el desarrollo de software seguro utilizando metodos ágiles sin problemas.

En [25] el autor hace una revisión hasta el año 2011 de la literatura para implementar actividades de seguridad en metodologías ágiles. De la literatura revisada el autor pretende responder a la pregunta: "C*uan fuerte es la evidencia que las prácticas de software seguro pueden ser implementadas exitosamente en el ciclo de desarrollo de software ágil?*" del estudio se expone que algunas prácticas son preferidas y de más fácil implementación cuando se comparan con otras. En el trabajo se enfatiza en tres prácticas: las historias de abuso (historias de usuario seguras, misuse cases), pruebas unitarias de seguridad automatizadas y el uso de estándares de codificación segura. Con base a esto el autor concluye que hay evidencia de que algunas

prácticas de seguridad pueden ser implementadas en el ciclo de vida del desarrollo ágil, aunque no tiene evidencia empírica que la seguridad del software mejora cuando se aplica su método.

Otra propuesta para introducir seguridad en las metodologías ágiles es la de [9] allí los autores identifican como problemática que las metodologías de desarrollo ágil en algunas ocasiones adolecen de características específicas de seguridad. Para remediar esta problemática los autores ofrecen un ejemplo donde se aplican técnicas clave de seguridad a una metodología de desarrollo ágil.

Los autores proponen agregar elementos clave de seguridad en las diferentes fases del desarrollo de software (requerimientos, diseño, implementación y pruebas). Y para ilustrar su propuesta los autores toman una metodología ágil en específico FDD Feature Driven Development en donde integraron su solución. Dentro de sus conclusiones los autores afirman que, aunque se necesita más investigación y experiencias prácticas, su enfoque de solución es prometedor.

Según [10] existe la percepción que las metodologías ágiles de desarrollo no crean código seguro y que esto se debe a la poca experiencia que hay de "secure agile" en el mercado. En el documento mencionan que Microsoft posee un proceso que de desarrollo de software llamado Security Development lifecycle (SDL) que ha mostrado su efectividad en reducir el número de vulnerabilidades en el software, sin embargo, desde el punto de vista del desarrollo de software ágil este proceso es bastante pesado pues fue pensado para productos muy grandes.
Como respuesta a esta problemática en [10] Microsoft propone un proceso denominado SDL-agile para fusionar prácticas de SDL a las metodologías ágiles.
En el documento proponen que se deben hacer dos cambios. El primero, que las prácticas de SDL que se adopten a los procesos ágiles deben ser ligeras, lo que significa que el equipo de trabajo solo hace lo necesario de SDL antes de pasar a otra funcionalidad. El otro cambio consiste modificar las fases que se basan en el desarrollo en cascada para utilizar un formato más amigable con las metodologías ágiles. Este documento propone integrar los requerimientos de SDL dentro de las iteraciones (sprints) de las metodologías ágiles a modo de tareas en el backlog de producto y los diferentes backlogs de sprint. También recomienda usar al menos una vez en el sprint herramientas automatizadas que realicen tareas de seguridad. Adicionalmente el documento motiva a la implementación de un modelamiento de amenazas como parte de trabajo de diseño de un sprint.

Otro trabajo que trata el tema de la utilización prácticas de seguridad en las metodologías ágiles es [6], en este trabajo abordan como problemática el hecho que los sprints o ciclos son

demasiado cortos haciendo difícil el cumplimento de las largas listas de tareas de aseguramiento de software las cuales pueden terminar siendo saltadas. Lo que se propone en [6] para atacar el problema es proveer una lista de “historias de seguridad”(similares a historias de usuario) y una lista tareas de seguridad en un lenguaje y formato prácticos que permitan su utilización como plantilla que puede ser utilizada por cualquier proyecto permitiendo ahorrar tiempo. [6] es un artículo por lo que no realiza conclusiones acerca de su propuesta.

En [26] el autor brinda diferentes prácticas que se recomiendan utilizar dentro del flujo de trabajo de Scrum. En el artículo se reconoce la importancia que tiene un Product Owner en Scrum pues es el quien hace gestión sobre los requerimientos que el usuario está buscando, el autor hace visible que, a no ser que se hagan responsables a los Product Owners de los riesgos de seguridad que se deciden asumir en el proyecto, el Product Owner no tiene ninguna motivación de adicionar requerimientos de seguridad.

Para que el Product Owner tenga elementos con los cuales pueda decidir sobre requerimientos de seguridad, se propone realizar una identificación de riesgos basados en un modelamiento de amenazas al realizar las historias de usuarios y el diseño. Los controles que se identifiquen para estas amenazas muy probablemente serán requerimientos de seguridad.

Para facilitar esta tarea, es deseable un Product Owner con entrenamiento en seguridad, adicionalmente se puede brindar una plantilla de historias de usuario de seguridad tales como las presentadas en [6], esto le puede servir al Product Owner para agregarlos antes interpretándolos en el contexto de su proyecto.

También para facilitar la producción de requerimientos de seguridad los autores recomiendan que en la metodología se permita el uso de historias de usuario negativas en donde se describe que comportamientos no se quieren del sistema, esta práctica también se conoce como “misuse cases”. Como conclusión, el autor remarca que es muy importante que todos los riesgos de seguridad sean mitigados o aceptados, pero nunca ignorados.

En [27] se discute la problemática de como las actividades de seguridad que se deben realizar dentro de Scrum afectan al framework. Para esto el autor selecciona los puntos de control más importantes que pueden ser personas o actividades (el Product Owner, el sprint planning y el sprint review) para allí incluir prácticas de seguridad como: plantillas de seguridad, modelamiento de amenazas, seguimiento a funcionalidades y aprobación del riesgo no mitigado.

El autor evalúa la implementación de estas técnicas de seguridad en los puntos de control propuestos utilizando una técnica de cálculo del nivel de agilidad de cada actividad propuesto por [28]. Como conclusión el autor encuentra que es seguro decir que las actividades de seguridad propuestas en los puntos de controles aumentan la seguridad del producto.

En [23] se aborda la necesidad de asegurar el software producido con metodologías ágiles, los autores se enfocan en la problemática que surge al querer agregar prácticas de seguridad en metodologías ágiles y el impacto que estas prácticas podrían causar en la agilidad de la metodología.

Para mitigar esta problemática los autores proponen la introducción de prácticas de seguridad con el uso de un parámetro ajustable que puede controlar las características ágiles del proceso mediante el cálculo del grado de agilidad de un de una actividad y del proceso completo. Como conclusión los autores afirman que utilizando su método se puede agregar actividades de seguridad a metodologías ágiles y según la parametrización utilizada en el método lograr un balance entre los costos de un nivel de agilidad menor pero una seguridad mayor.

En [8] se expresa preocupación porque según los autores, Scrum se centra en entregar funcionalidad reduciendo las etapas de análisis lo que para la seguridad significa que no se realiza una arquitectura de seguridad ni instrucciones detalladas al principio de un proyecto. Como propuesta presentan Secure Scrum definido por sus autores como una extensión del framework Scrum, una versión enriquecida con funcionalidades específicas para la seguridad que consiste en cuatro componentes: Identificación, el cual es usado para identificar problemas de seguridad que se puedan presentar, se realiza a través marcas en el Product Backlog. Componente de Implementación el cual incrementa la notoriedad de las actividades de seguridad induciéndolas en las actividades del sprint backlog. El componente de verificación en donde el equipo debe estar en capacidad de probar el software con enfoque en la seguridad. Finalmente, el componente de la definición de completado que es donde se habilita a los miembros del equipo en dar por terminado un requerimiento de seguridad. Como conclusión esta investigación demuestra mediante una pequeña evaluación que un aporta a Scrum dotándolo de herramientas que apoyan la seguridad del software como el uso de las “S-Tags”, una manera de marcar elementos del backlog, con información relacionada a la seguridad.

En Finlandia, el gobierno en busca de garantizar ciertos aspectos como la seguridad en sus sistemas de información realizó un conjunto de requerimientos de seguridad denominados VAHTI, este conjunto de requerimientos vienen acompañados de un conjunto de instrucciones

para aplicarlos en el desarrollo de software, la problemática que identifican los autores en [11] es que estas instrucciones no necesariamente son aplicables en las metodologías ágiles, por esto presentan una adaptación de Scrum para cumplir con VAHTI.
Las principales modificaciones propuestas a Scrum por este trabajo son:

- La creación de documentación relacionada con la seguridad.
- Entrenamiento en seguridad concerniente a las plataformas y tecnologías.
- Codificación, casos de prueba y revisiones de interfaces.
- Tests de seguridad.
- Auditorias de seguridad.

Otras de las modificaciones propuestas en este trabajo son que el Product Owner califique el proyecto en uno de tres niveles "basic", "heightened" y "high" y dependiendo de esta calificación la metodología agrega más o menos tareas.

Otro aspecto importante de este estudio es que en el proyecto debe haber una persona dedicada al desarrollo de la seguridad y que también se debe nominar a una persona responsable de la seguridad.
Este trabajo es una muestra que Scrum puede ser adaptado para el cumplimiento normas de seguridad como VAHTI las cuales están diseñadas para ser trabajadas con métodos estandarizados de desarrollo seguro muy parecidos a los realizados en los métodos de cascada.

Al igual que otros estudios en [14] que se propone una extensión derivada de Scrum llamada US-Scrum que pretende aportar a las metodologías ágiles en los campos de la seguridad y la usabilidad. Ya que, para los autores, las metodologías ágiles descuidan un poco estos atributos de calidad. Para esto proponen combinar fases de Feature driven development (FDD) y el modelo de Scrum, sus principales características son las de construir un Product Backlog dividido en tres secciones: funcional, seguridad y usabilidad. Para esto también proponen la creación de otros roles adicionales a los de Scrum como el "Functional Master", "Security Master", "Usability Master", "Functional team", "Functional Team", y "Security Team", en este caso el Product Owner actuara como un gerente de proyectos que representa al cliente y mantiene una constante comunicación con los otros masters. La idea es que cada uno de los equipos y sus masters se encarguen de las divisiones del Product Backlog (seguridad, funcionalidad, usabilidad) según su especialidad. Los autores dicen que los principales beneficios del método propuesto son que se incrementa el enfoque en el usuario generando una mayor satisfacción de los mismos, que se validan los requerimientos debido a procesos de verificación más fuertes y que se hace una mejor gestión de proyectos presionando menos a los desarrolladores.

En [12], los autores reconocen la creciente sofisticación de los ataques informáticos que se producen contra sistemas de información alrededor del mundo, por esto proponen el uso de ScrumS, un modelo que agrega técnicas de análisis de riesgos al ciclo de vida de Scrum con el fin de hacerlo más seguro. Este modelo propone un sprint cero en el cual se seleccionan herramientas y procesos que tienen que ver con el desarrollo y pruebas del software. Luego de este sprint los autores nos proponen la realización de un análisis de riesgo como se puede visualizar en la (Ilustración 4), involucrando herramientas como la extracción de activos, el uso de historias de usuario de seguridad, análisis de vulnerabilidad, inventario de riesgos, controles de seguridad y pruebas de seguridad.

Ilustración 4 Scrums extracción de tareas de seguridad [12]

Este modelo fue probado por los autores mediante un caso de estudio con un equipo de trabajo sin experiencia en donde pudieron hacer un entregable seguro. Como experiencia encontraron que la primera vez que se aplica este método puede ser un tanto difícil y que por esta razón se debe construir un repositorio de seguridad donde pueden ir pruebas automatizadas para ser reusadas.

En [29] reconocen que Scrum es una de las metodologías más populares debido a un proceso simplificado y su énfasis en el trabajo en equipo, no obstante, los autores expresan su preocupación pues el objetivo de Scrum es hacer entregas constantes y funcionales al usuario en donde los requerimientos de seguridad comúnmente tienen baja prioridad frente a otros requerimientos funcionales. Para los autores esto puede provocar que se desarrollen soluciones que cumplen con los requerimientos de negocio pero que no son muy robustas en términos de seguridad.

Para mitigar esta situación los autores proponen integrar prácticas de seguridad en Scrum y aprovechar el carácter iterativo de Scrum para así introducir pruebas de penetración y descubrir vulnerabilidades de manera temprana y constante.

Los autores hablan de dos tipos de pruebas de penetración, las automáticas y las manuales. Para aplicar las pruebas automáticas los autores proponen la selección de requerimientos que representen más riesgos o agreguen más valor para probarlos de manera regular y automatizada. También proponen el uso de pruebas manuales realizadas por un hacker ético cada que el Product Owner lo considere necesario o cada que se haga una entrega a producción, teniendo en cuenta que si la entrega en producción solo incluye cambios pequeños no es necesario probar nuevamente.

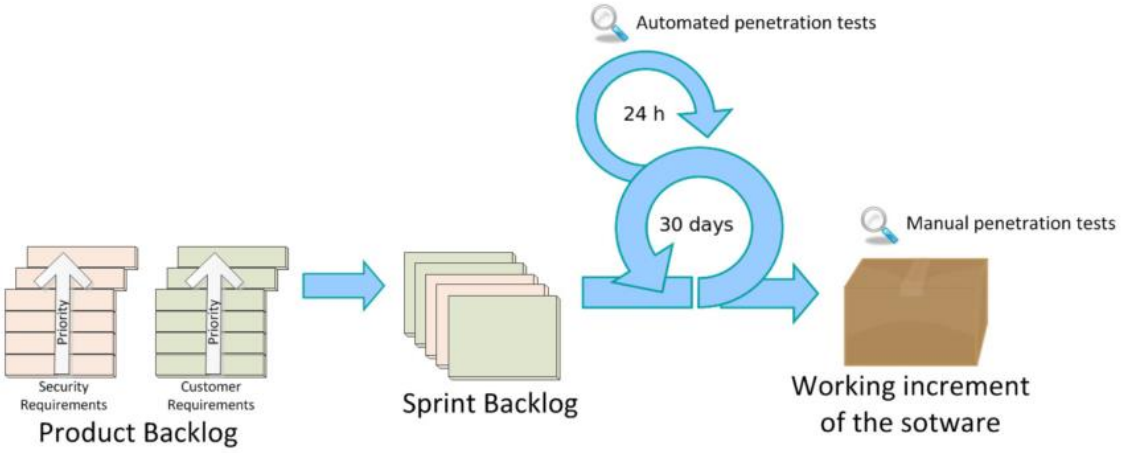


Ilustración 5 pruebas de seguridad en Scrum [29]

Como principales ventajas de este trabajo los autores resaltan que se aplican pruebas de penetración de manera más frecuente, que se detectan y corrigen vulnerabilidades en fases tempranas del desarrollo y que se pueden obtener métricas de defectos de seguridad por iteración. Como desventajas los autores señalan que la preparación de las pruebas en los primeros sprints pueden ser demandantes en esfuerzo, también que puede haber sobrecostos por la utilización de software especializado y por el uso de expertos en seguridad para hacer pruebas de penetración dinámicas.

En la misma línea de la implementación de pruebas de seguridad en metodologías ágiles se encuentra la tesis [7] cuyo objetivo es hacer el proceso de desarrollo ágil más seguro, a través de la automatización de pruebas de seguridad en metodologías ágiles con el fin de reducir vulnerabilidades en ambientes productivos. Su mayor aporte se centra en probar que hay varias herramientas de seguridad que se pueden usar en procesos de integración continua los cuales a su vez son ampliamente usados en metodologías ágiles.

En la misma línea de los dos trabajos anteriormente presentados está [30] cuya problemática a atacar es muy similar a otras problemáticas presentadas anteriormente en donde los autores piensan que Scrum podría fallar en producir software con buena seguridad por no especificar estándares claros de seguridad. [30] es una propuesta bastante similar a [29] pues se asemejan en querer introducir pruebas de seguridad en Scrum, además comparten la idea de realizar pruebas de seguridad de la funcionalidad del software antes de realizar un incremento del producto, se diferencian en que [30] propone una segunda prueba justo después de entregar el incremento en donde el objetivo es probar las vulnerabilidades en la infraestructura donde se despliega la aplicación y no la funcionalidad del software. Este trabajo aporta como novedad que no solo se centra en las pruebas funcionales, sino que también tiene en cuenta la infraestructura en donde se despliega la solución, dando relevancia a la aplicación y a la plataforma.

5. METODOLOGÍA

5.1.DIAGNÓSTICO

Dentro del diagnóstico se debe identificar cuáles son políticas, lineamientos y procedimientos de seguridad que hay en EPM con el fin de establecer claramente cuáles están relacionadas con el desarrollo de software. Esta información será importante debido a que la guía que se entregará

como producto final de este trabajo apunta a que este conjunto de normativas de EPM sean tomadas en cuenta para lograr software más seguro. Para realizar este objetivo es necesario obtener toda la información relacionada y hacer un inventario de todo lineamiento, política o procedimiento orientado al aseguramiento de software.

Después es necesario diagnosticar cuales son los impedimentos por los cuales los equipos de trabajo no aplican los lineamientos y políticas de seguridad establecidos por EPM. Para lograrlo se diseñan y aplican cuestionarios que sirvan para identificar los impedimentos que tienen los equipos de trabajo ágiles para cumplir con lo establecido por EPM.

5.2.ESTADO DEL ARTE

Partiendo del estudio realizado en el estado del arte, se profundiza en él, enfocándose en encontrar cuál es el trabajo previo orientado a la implementación de prácticas de seguridad en marcos de desarrollo ágil que ayuden a remover los impedimentos identificados en la fase de diagnóstico.

Una vez realizado este estudio se procede a extraer cuáles prácticas de las encontradas aportan de manera directa a la remoción de los impedimentos anteriormente identificados. Como resultado final de esta fase se realiza una lista cruzada de las causas encontradas y las prácticas relacionadas para su remoción.

5.3.DESARROLLO

Tomando en cuenta el marco de desarrollo de EPM, la lista cruzada de impedimentos y prácticas de seguridad, se procede a desarrollar una guía para implementar durante los procesos de desarrollo de EPM, teniendo en cuenta las particularidades y exigencias de la empresa. Para esto se analizan las prácticas de seguridad asociadas a cada uno de los impedimentos diagnosticados y se seleccionan las que serán implementadas. Para esto se debe tener en cuenta la factibilidad de su implementación dentro de EPM, el marco de desarrollo ágil utilizado, su aporte directo al cumplimiento de la normatividad empresarial y su alineación con lo establecido en la normatividad (lineamientos, políticas y procedimientos.)

5.4.VIABILIDAD

Una vez se tiene una alternativa de solución se procede a evaluar la viabilidad de que los equipos de desarrollo puedan seguir la guía propuesta, verificando que no haya impedimentos en cada una de las prácticas propuestas.

Esto se realiza aplicando cuestionarios a los equipos de desarrollo ágil. Con la información recolectada se realiza un análisis de la viabilidad de la alternativa de solución propuesta y su aporte al cumplimiento de los lineamientos, políticas y procedimientos establecidos. Si es necesario se pueden realizar ajustes a la propuesta de manera iterativa hasta encontrar que las prácticas propuestas no presentan impedimentos en su implementación.

6. DIAGNÓSTICO DE IMPEDIMENTOS PARA EL CUMPLIMIENTO DE LINEAMIENTOS POLÍTICAS Y PROCEDIMIENTOS DE SEGURIDAD EN EPM.

El objetivo de esta sección es diagnosticar cuáles son los impedimentos para ejecutar las prácticas de seguridad definidas en EPM cuando se está utilizando Scrum como metodología ágil de desarrollo.

Por consiguiente, se identifican las políticas y lineamientos de seguridad y a partir de estas se recopilan las prácticas de seguridad en el desarrollo de software allí consignadas. Una vez claras las prácticas de seguridad se diagnostican los posibles impedimentos para su ejecución.

6.1.IDENTIFICACIÓN DE POLÍTICAS Y LINEAMIENTOS DE SEGURIDAD ORIENTADOS AL DESARROLLO DE SOFTWARE

Esta sub sección se centra en identificar los lineamientos, políticas y procedimientos establecidos en EPM para la seguridad en el desarrollo de software extrayéndolos de toda la documentación existente, utilizando como formato un resumen con los detalles relevantes para secciones posteriores.

La documentación consultada está disponible en la intranet de EPM para referencia de todos los empleados y contratistas. De esta documentación se encuentra que en los siguientes documentos hay prácticas para el desarrollo software seguro:

- *"Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte"*.
- *"Guía para la definición de requerimientos de seguridad"*.
- *"Lineamientos para la seguridad para lenguajes de programación"*.
- *"Lineamientos Seguridad Correcto Procesamiento Aplicaciones"*.
- *"Lineamientos Seguridad Archivos Sistema"*.

Del documento *"Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte"*. Se identificó la existencia de lineamientos específicos a seguir en cada una de las fases del desarrollo de software que EPM tiene institucionalizadas para su modelo tradicional basado en cascada.

El documento *"Guía para la definición de requerimientos de seguridad"*, define los aspectos de seguridad que debe tener en cuenta un analista para obtener requisitos de seguridad o para realizar una implementación segura. El documento contiene información específica para definir requerimientos de seguridad.

En los documentos *"Lineamientos para la seguridad para lenguajes de programación"*, *"Lineamientos Seguridad Correcto Procesamiento Aplicaciones", "Lineamientos Seguridad Archivos Sistema"*, se refieren a las prácticas a tener en cuenta durante las fases de implementación mediante la recomendación de prácticas de codificación segura.

6.2.RECOPILACIÓN PRÁCTICAS IDENTIFICADAS

A continuación, se realiza una tabla resumen que recopila todas las prácticas de seguridad identificadas en la documentación (Tabla 5). La tabla contiene las siguientes columnas:

- **Práctica de seguridad:** en esta columna se especifica el nombre de la práctica de seguridad encontrada en la documentación.

- **Fase de desarrollo**: debido a que los lineamientos y prácticas fueron desarrollados pensando en la metodología de cascada las prácticas de seguridad están ubicadas en la fase que les corresponde.

- **Descripción:** una breve descripción de la práctica de seguridad.

- **Roles:** los roles definidos que intervienen en la aplicación de la práctica de seguridad.

- **Entregables:** documentos u otros artefactos esperados como resultado de la ejecución de la práctica de seguridad.

- **Fuente:** documento de donde se obtiene la práctica, de manera literal o tácita.

Práctica de seguridad	Fase de desarrollo	Descripción	Roles	Entregables	Fuente
Levantamiento de Requerimientos de seguridad de la información.	Ingeniería de requisitos	Levantar los requerimientos de Seguridad de la Información al mismo tiempo como los requerimientos funcionales de la aplicación. Estos requerimientos deben avalarse con los Clientes, con el Arquitecto de Dominio, de Aplicación y el Analista Funcional	Analista de Seguridad de la de aplicaciones	Documentación con requerimientos de seguridad. Aprobación de: Arquitecto de dominio. Arquitecto de aplicación. Analista funcional.	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte
Plantilla de requerimientos de seguridad	Ingeniería de requisitos	Brindan una lista de chequeo con los requerimientos de seguridad que se deben cuenta cuando se levantan requisitos	No especificado	Requerimientos de seguridad	Guía para la definición de requerimientos de seguridad
Análisis del Ambiente de Implementación de la solución.	Análisis de la solución	El análisis del ambiente debe incluir los siguientes aspectos: El número de usuarios, su	Analista de Seguridad de la de aplicaciones.	Documentación con el resultado de los análisis de los ambientes de la solución.	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte

		procedencia, la clasificación de la información a exponer, las plataformas involucradas, las tecnologías a utilizar y la interacción con otros activos de información.			
Análisis de Riesgos de las nuevas funcionalidades.	Análisis de la solución	Identificar las posibles vulnerabilidades y amenazas que pueden están presentes para el nuevo sistema de información o en el desarrollo de nuevas funcionalidades.	Analista de Seguridad de la de aplicaciones. Arquitecto de la Aplicación. Arquitecto de Dominio.	Análisis de riesgos. Controles de mitigación de los riesgos detectados.	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte
Desarrollo de casos de abuso	Análisis de la solución	Después de tener en cuenta los Requisitos Funcionales y el Análisis de Riesgos, se debe crear un documento de casos de abuso. A partir de los requisitos funcionales, crear unos anti-requerimientos, es decir aquellas funcionalidades que no debería realizar el sistema de información. Estos anti-requerimientos deben tener en cuenta los estados normales de ejecución e invertirlos para generar casos de	Analista de Seguridad de la de aplicaciones.	Documento con anti-requerimientos, escenarios de pruebas.	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte

		error. A partir de los anti-requerimientos, se deben crear los escenarios de prueba de las funcionalidades a implementar.			
Diseño de la seguridad del sistema de información.	Diseño y construcción de la solución.	Tomando como base el análisis de riesgos y el análisis de la solución de la aplicación, se debe realizar el diseño de los controles de seguridad, tanto a nivel de codificación, como a nivel de plataforma. Para lograr esto se deben identificar la criticidad de la información que manejan los componentes del diseño, implementar controles para asegurar la confidencialidad, integridad y disponibilidad.	Arquitecto de Diseño Analista de Seguridad	Diseño de seguridad en UMLsec	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte
Codificación e Implementación de la Solución.	Diseño y construcción de la solución.	Codificar siguiendo lineamientos de codificación segura. Evitar la inyección de posibles vulnerabilidades en la codificación de las soluciones, asegurar prácticas de	Analista de desarrollo	Código fuente	- Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte - Lineamientos Seguridad Correcto Procesamiento Aplicaciones - Lineamientos Seguridad

		cifrado de datos y correcto manejo de los archivos generados			Archivos Sistema
Revisión de Código	Diseño y construcción de la solución.	Después de la codificación, se debe realizar la revisión del código en busca de vulnerabilidades estáticas.	Analista de Seguridad de Aplicaciones:	Código revisado, hallazgos	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte
Ejecución de casos de abuso	Pruebas	Codificar y ejecutar las pruebas de abuso sobre la aplicación. La aplicación debería comportarse de una manera adecuada	Analista de seguridad de aplicaciones. Analista de desarrollo	Ejecución de las pruebas y hallazgos corregidos	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte
Pruebas de intrusión	Pruebas	Ejecutar pruebas de intrusión sobre la aplicación, con el fin de encontrar fallas de seguridad usando la experiencia de ataques comunes sobre aplicaciones	Analista de seguridad de aplicaciones. Analista de desarrollo	Ejecución de las pruebas y hallazgos corregidos	Lineamientos para la Seguridad en los Procesos de Desarrollo y Soporte

Tabla 5 Recopilación de prácticas de seguridad encontradas

6.3.DIAGNÓSTICO DE IMPEDIMENTOS PARA EL CUMPLIMIENTO DE PRÁCTICAS DE SEGURIDAD IDENTIFICADAS

Para determinar cuáles son los posibles impedimentos para llevar a cabo las prácticas de seguridad establecidas por EPM y recopiladas en la tabla 5, se llevan a cabo encuestas en distintos equipos ágiles de la compañía. Esto implicó el desarrollo de formularios con preguntas que permitieron identificar varios comportamientos en los equipos de desarrollo ágil en EPM.

6.3.1. Metodología para realizar encuestas

Las encuestas se aplicaron a tres equipos de trabajo (número de equipos en los que se permitió realizar el trabajo) que vienen trabajando en desarrollo de aplicaciones utilizando Scrum y que están conformados por distintas cantidades de integrantes, siete, seis y nueve personas.
Las encuestas fueron realizadas de manera grupal y en periodos de tiempo de treinta minutos, tiempo concedido para realizar este trabajo.

La premisa para la realización de las encuestas fue no condicionar las respuestas ofrecidas por los participantes. Teniendo en cuenta esta premisa se minimizaron los posibles impactos de varios aspectos:

- Muchas personas de las que van a participar en las encuestas trabajan como contratistas de EPM, por lo que se podrían ver influenciados a contestar algo que a su contratante EPM le gustaría oír.
- Una respuesta colectiva puede causar que se pierdan opiniones valiosas pues algunas personas podrían verse influenciadas por las opiniones de la mayoría.
- Cuando se hacen las preguntas no se debe influenciar las respuestas.

El resultado que se quiere obtener es conocer cuáles serían las posibles razones para que no se lleve a cabo la práctica.

Dado lo anterior se plantea la siguiente metodología para aplicar la encuesta.

1. Se aclara que las respuestas se brindan de manera libre y anónima, que de ninguna manera se tomarán datos personales para que las personas se sientan libres de dar sus respuestas.
2. Se explica cada una de las prácticas de seguridad definidas por EPM para el desarrollo seguro.
3. Se entrega una hoja con todas las prácticas de seguridad y un espacio después de cada práctica para responder preguntas.
4. Se solicita que no se marquen las hojas.
5. Durante la explicación de cada práctica se solicita que se respondan las siguientes preguntas

a. En qué porcentaje cree usted que, en EPM bajo su marco ágil de desarrollo, se sigue esta práctica. Esta pregunta servirá para hacer consientes a las personas de que tanto aplican las prácticas de seguridad y que así en la segunda pregunta contesten sobre los impedimentos que pueden existir para la aplicación de las prácticas definidas.
b. Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

6. Para contestar por cada una de las 6 prácticas se otorga un tiempo de 5 minutos.
7. Se recogen las encuestas.

6.3.2. Cuestionarios

Basado en la metodología y restricciones anteriormente planteadas se diseña el siguiente cuestionario de diagnóstico, ver anexo 1.:

- En qué porcentaje cree usted que, en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

2. Plantilla de requerimientos de seguridad.

3. Desarrollo y ejecución de pruebas de casos de abuso.

4. Análisis de riesgos de las nuevas funcionalidades y diseño de seguridad.

5. Codificación segura y revisión de código.

6. Pruebas de intrusión.

En la sección 6.2 tabla 5 se identificaron diez prácticas de seguridad a seguir, no obstante, estas prácticas fueron agrupadas con el fin de ser más generales a la hora de aplicar las encuestas y no entrar en detalles técnicos. Debido a que esto podría llegar a abrumar posibles encuestados, quienes tal vez no tienen un conocimiento específico de seguridad como es el caso de los Product owner, Scrum Master y funcionarios de EPM en la Gerencia de Tecnologías de Información.

La agrupación realizada es la siguiente:

1. Requerimientos de seguridad de la información
2. Plantilla de requerimientos de seguridad
3. Desarrollo y ejecución de pruebas de casos de abuso (se refiere a las prácticas especificadas como Desarrollo de casos de abuso y Ejecución de casos de abuso).
4. Análisis de riesgos de las nuevas funcionalidades y diseño de seguridad (se refiere a las prácticas especificadas como análisis de riesgos de las nuevas funcionalidades y Diseño de la seguridad del sistema de información).
5. Codificación segura y revisión de código (agrupa las siguientes prácticas: Codificación e Implementación de la Solución y Revisión de Código).
6. Pruebas de intrusión.

6.3.3. Impedimentos encontrados en las encuestas realizadas.

A continuación, se hace un análisis general de los impedimentos encontrados resaltando los más relevantes. Una vez finalizado el análisis general, se detallan los impedimentos para cada una de las prácticas utilizadas en la encuesta.

Análisis general de resultados de la encuesta:

Dentro de los equipos conformados en EPM para trabajar en desarrollo de software utilizando metodologías ágiles no existe una persona que vele por el aseguramiento del software. Al no

existir estas personas por parte del cliente (EPM) y los distintos proveedores, los equipos de trabajo perciben que no existe nadie responsable por el tema y ocurren tres fenómenos: nadie asume esta responsabilidad, los integrantes del equipo no realizan actividades orientadas a asegurar su desarrollo y no hay exigencias con respecto a la seguridad.

Respuestas encontradas en las encuestas que evidencian la anterior conclusión:

"No hay persona de seguridad en equipos ágiles"
"No hay security owner similar al Product Owner para que sea el dueño de seguridad"
"No hay acompañamiento de seguridad"
"todas las personas relacionadas no tienen contexto relacionado con la seguridad"
"No hay rol responsable presente en los proyectos"
"El rol existe reactivamente"
"Cliente predisponen que ya está la seguridad implementada sin ponerlo en contexto"
"No existe rol responsable"
"Algunos criterios de aceptación corresponden a casos de abuso se podría considerar este tipo de criterios. Si existiera un responsable de seguridad en el equipo se podrían tener más visibles"
"No hay responsables de seguridad en el equipo"
"falta un rol que se encargue de estas pruebas"
"No siempre hay responsables de seguridad en el equipo"

Lo anteriormente descrito puede ser consecuencia de otro de los impedimentos identificados dentro de las encuestas: se evidencia desconocimiento en materia de seguridad por todos los integrantes de los equipos de desarrollo ágil que se conforman. No es un problema específicamente de los distintos proveedores que contrata EPM para su desarrollo si no un problema general, que también incluye a EPM, sus analistas asignados y los dueños de producto.

El desconocimiento que se puede identificar no solo va desde el punto de vista técnico, pues los equipos desconocen técnicas de aseguramiento, a nivel de modelamiento arquitectónico, de desarrollo, de las pruebas y del levantamiento de requerimientos. Si no también desde el punto de vista del desconocimiento de lineamientos y reglas de negocio para el desarrollo seguro.

Aunque mediante las encuestas se evidencia un desconocimiento generalizado de las políticas y lineamientos con respecto al desarrollo seguro. También se evidencia que se necesitaría un acompañamiento de personas con conocimientos en seguridad para que los equipos ágiles

puedan ejecutar las prácticas de seguridad para EPM. Los equipos solicitan capacitación o personas capacitadas en seguridad.

Respuestas encontradas en las encuestas que evidencian la anterior conclusión:

"Exige capacitación de prácticas"
"Falta capacitación en cuanto a temas de seguridad"
"No hay conocimientos suficientes en seguridad"
"No hay una correcta definición por parte de los marcos ágiles"
"El responsable del negocio no sabe de seguridad informática "
"No se conoce acerca de esta práctica"
"No conocemos la plantilla"
"Se desconoce la plantilla es importante socializar"
"Estas plantillas no son muy conocidas y al parecer poco socializadas en los equipos"
"No se conoce la plantilla de seguridad en los proyectos"
"Desconocimiento de la plantilla y su finalidad"
"No se conoce mucho acerca de esta práctica"
"No se tiene conocimientos en el tema"
"No hay personas con conocimientos en pruebas de seguridad"
"No se conoce acerca de esta práctica"
"No se maneja este concepto ni se conocen herramientas para hacerlo, falta capacitación en el equipo"
"Exige capacitación"
"No se conoce acerca de esta práctica"
"Requiere extra en pruebas y conocimiento de cómo realizarlas"
"Exige capacitación de prácticas"
"Se requiere capacitación en el tema"
"Las personas que lo podrían hacer no tienen el conocimiento"

Otro de los impedimentos para aplicar las prácticas de seguridad identificadas está relacionado con las personas que ejercen el rol de Product Owner, pues los equipos identifican que generalmente las personas que ejercen este rol tienen gran conocimiento del negocio, pero no están interesadas, no dan prioridad y desconocen de seguridad, por lo tanto, difícilmente hacen requerimientos orientados a tener un sistema más seguro.

Algunas de las encuestas realizadas identifican la necesidad de asociar alguien responsable por parte del negocio con la identificación de riesgos de los sistemas a desarrollar. Adicionalmente los equipos de desarrollos parecen temerosos de implementar requerimientos no funcionales por encima de los funcionales, porque son catalogados por los clientes como "costosos".

Respuestas encontradas en las encuestas que evidencian la anterior conclusión:

"Los requerimientos de seguridad no son prioridad para el negocio"
"Dentro de las prioridades del negocio está la funcionalidad y es poco visible la seguridad"
"El responsable del negocio no sabe de seguridad informática "
"Si se crea, el cliente debe conocer temas de TI y seguridad"
"No hay presupuesto para hacer pruebas exhaustivas"
"No tenemos una persona de parte del cliente que participe e identifique cuales son los riesgos de seguridad para EPM"
"Implica más tiempo y costo el usuario/negocio es quien paga los requerimientos, solo pagan las funcionalidades dicen que somos muy caros
"No se da importancia"

Otros de los impedimentos identificados es que las prácticas tal y como están definidas en este momento exigen documentación detallada, aprobaciones de muchas personas no involucradas en el proceso y que por consiguiente tomarían mucho tiempo en ser ejecutadas generando sobrecostos.

Respuestas encontradas en las encuestas que evidencian la anterior conclusión:

"Exige mucha documentación"
"Muchas aprobaciones necesarias"
"No se conoce la plantilla solo se diligencias historias de usuario si se requiere muchas aprobaciones no es ágil"
"Necesitaría muchas aprobaciones"
"Exige mucha documentación como prerrequisito"
"Tomaría mucho tiempo hacer esta actividad y no se haría una entrega oportuna"
"Se exige mucha documentación"
"Son muy extensas y costosas"
"Costo quien lo pagaría"
"Se requiere mucha documentación"

“Tiempo y presupuesto para preparar y ejecutar pruebas”

Impedimentos específicos identificados por cada una de las prácticas consultadas:

1. **Requerimientos de seguridad de la información**

   Impedimentos:

   - Responsable de seguridad: Dentro de las encuestas se puede evidenciar que los equipos se inclinan por asociar un responsable específico para la realización de esta actividad alguien diferente al responsable del negocio, identifican que esta práctica la debería hacer un responsable del negocio, pero estos no tienen el suficiente conocimiento en seguridad informática.
   - Desconocimiento de seguridad (equipo): Dentro de los impedimentos que los equipos argumentan falta de capacitación o no tener conocimientos suficientes en seguridad para ejecutar la práctica.
   - Prioridad por parte del negocio: Los distintos equipos de trabajo indican que los requerimientos de seguridad no son prioridad para el negocio.
   - Exceso de aprobaciones: Tal y como están definidas las prácticas, es necesario que haya varias aprobaciones para dar como aceptado un requerimiento de seguridad. Los equipos ágiles identificaron que estas aprobaciones son muchas y causan retrasos.
   - Tiempo insuficiente: Se identifica que llevar a cabo esta práctica tal y como está definida puede tomar mucho tiempo.

2. **Plantilla de requerimientos de seguridad**

   Impedimentos:

   - Falta de difusión: En general se encuentra que los equipos ágiles no conocen de la existencia de la plantilla ni cómo debe utilizarse este tipo de plantillas. Falta más difusión de la misma.
   - El Product Owner debe conocer de Tecnologías de Información y seguridad: Los equipos expresan que, si existe una plantilla de este tipo, el encargado de hacer los requerimientos formales en este caso el Product Owner, debería tener el conocimiento necesario en tecnologías de información y seguridad informática

para saber cuándo uno de los requerimientos expresados en la plantilla aplica en su proyecto.

3. **Desarrollo y ejecución de pruebas de casos de abuso.**

Impedimentos:

- Tiempo insuficiente: Los equipos ágiles identifican que para llevar a cabo un modelamiento de casos de abuso y su respectiva ejecución es necesario invertir gran cantidad de tiempo y dinero pues perciben que se debe hacer mucha documentación, lo que dificultaría las entregas constantes y oportunas.
- No hay responsable de seguridad: Los equipos perciben que esta tarea debería tener una persona asignada o un rol especifico con esta responsabilidad y que como no la hay no sería posible ejecutar esta práctica. Perciben que debería haber alguien encargado de especificar los casos de abuso y de ejecutarlos a manera de prueba. No contemplan las tareas de seguridad como un trabajo que debe hacer algún miembro actual del equipo.
- Desconocimiento en e inexperiencia en seguridad (equipo): a través de las encuestas se puede observar que algunas personas creen que los casos de abuso son realizados en otra fase. Los equipos no conocen como hacer casos de abuso, no saben cuándo hacerlos y dicen no tener personal capacitado para llevar a cabo este tipo de prácticas.

4. **Análisis de riesgos de las nuevas funcionalidades y diseño de seguridad.**

Impedimentos:

- No existe responsable desde el negocio: se identifica que para hacer un análisis de riesgo es necesario un conocimiento del negocio, sin embargo, no existe dentro de los equipos de trabajo una persona con este perfil.
- El usuario no se hace responsable del riesgo: dentro de las encuestan se puede los equipos expresan que los usuarios no se hacen responsables de las consecuencias de mitigar o no un riesgo.
- Implica más tiempo y costo: se asume que realizar esta práctica implicaría más tiempo por lo tanto más costos, el usuario está solo dispuesto a pagar por funcionalidades.
- Desconocimiento en seguridad: los equipos argumentan que generalmente quien solicita funcionalidades no tiene conocimientos en seguridad ni tampoco se cuenta

con recursos dentro del equipo para hacerlo. Afirman que necesitan personas capacitadas o capacitación.

5. **Codificación segura y revisión de código.**

Impedimentos:

- Falta de herramientas: los equipos indican que para llevar a cabo estas prácticas necesitarían acceso a herramientas que automatizan parte del proceso y que no se tiene acceso a ellas.
- Tiempo insuficiente: dentro de las encuestas se argumenta que no habría suficiente tiempo para llevar a cabo esta práctica debido a que tomaría mucho tiempo realizarla, la presión de las entregas lleva a que no se implementen buenas prácticas.
- Desconocimiento para resolver retos de codificación: se identifica como impedimento que no existe mucho conocimiento con respecto a estándares de codificación segura, que las revisiones requerirían expertos en seguridad y que no se cuenta con muchos.

6. **Pruebas de intrusión.**

Impedimentos:

- Tiempo y presupuesto insuficiente: se encuentra que estas pruebas requieren mucho tiempo y que no sería posible realizarlas en todos los Sprint, además las califican como extensas y costosas, se requeriría tiempo y presupuesto.
- No existe el conocimiento: no se tienen recursos que puedan planear y ejecutar estas pruebas de intrusión, no existe responsable de este tipo de pruebas.

7. LISTA CRUZADA IMPEDIMENTOS Y PRÁCTICAS DE SEGURIDAD

En este capítulo se realiza un análisis de la literatura para encontrar posibles soluciones a los diferentes impedimentos identificados. Con el fin de tener una visión más clara de cómo aportar a la solución se presentará una lista cruzada entre los impedimentos y las prácticas identificadas.

Finalmente se presentará una definición resultante de cada práctica identificada en la literatura, esta definición será importante para luego evaluar cada una como posible solución.

7.1. IMPEDIMENTOS Y PRÁCTICAS DE SEGURIDAD EN LA LITERATURA

A continuación, se presenta un estudio de la literatura en busca de posibles soluciones a los impedimentos identificados, y de otros impedimentos no encontrados en EPM, pero presentes en otros trabajos.

En [31] se identifican los siguientes impedimentos para ejecutar prácticas de seguridad en una metodología ágil:

- Tiempo insuficiente para abordar los requisitos de seguridad para cada release.
- Falta de conocimiento en seguridad por parte del equipo de desarrollo
- Poca conciencia sobre seguridad por parte del Product Owner.

La solución propuesta por los autores es crear un "security backlog" a partir del backlog, e incluir un rol denominado "security master" cuya función es marcar las historias de usuario que requieren atención con respecto a la seguridad. Adicionalmente el security master especifica cuáles son sus preocupaciones de seguridad en la funcionalidad marcada para que esta información sea usada en el desarrollo y pruebas, además se hace responsable de verificar las pruebas de estas funcionalidades. En [2] los autores también recomiendan la inclusión de un experto de seguridad, en este caso para guiar la creación de las historias de seguridad o casos de abuso, esta recomendación está orientada a remover impedimentos que tienen que ver con la falta de conocimiento en seguridad.

En [8] los autores hacen un recorrido por todo el ciclo de vida de una metodología ágil basada en Scrum dividiéndola en fases:

- Identificación
- Implementación
- Verificación
- Definición de terminado (definition of done).

Para la fase de identificación los autores proponen que se marquen las historias de usuario importantes. Estas marcas deben expresar los riesgos asociados mediante la documentación de historias de seguridad o casos de abuso, convirtiendo así la marca en un análisis de riesgos ligero. Esta práctica contribuye a remover impedimentos de tiempos insuficientes, reduciendo la documentación detallada.

En la fase de verificación se utiliza la marca de la fase anterior para hacer seguimiento de las historias de usuario, la marca indica que se debe tener especial cuidado en el desarrollo y

verificación. Esto aporta a la codificación segura de la historia de usuario y a las pruebas; facilitando que no se escape ninguna funcionalidad en la que se deba prestar atención en cuanto a seguridad

Finalmente, los autores sugieren la inclusión de recursos externos con conocimientos en seguridad en momentos específicos del proyecto para extender el conocimiento del equipo y resolver retos de seguridad en la implementación.

Por su parte en [12] los autores proponen la ejecución de un Sprint 0 cuyo foco es determinar las políticas y diseños de seguridad para el proyecto. Los autores también proponen que antes de cada sprint se utilicen las historias de usuario para identificar activos importantes de la compañía y a partir de estos crear historias de abuso, identificar vulnerabilidades y crear posibles escenarios de riesgo. Estas prácticas se orientan a reducir los problemas que representa para las prácticas de seguridad el constante cambio en los requerimientos.

En [14] los autores proponen dividir el Product Backlog en tres secciones, una de estas secciones dedicada a la seguridad. Para esta sección los autores sugieren asignar un equipo dedicado y especializado con su respectivo gestor responsable. Adicionalmente los autores recomiendan la creación de Sprints de seguridad los cuales deberían ejecutarse paralelamente y dentro de la caja de tiempo de los Sprints establecidos en el proyecto. Con estas propuestas se aporta a la reducción del desconocimiento en seguridad y a agilizar la ejecución de tareas de seguridad.

Para reducir problemas de desconocimiento de seguridad los autores de [11] proponen:

- La creación de un rol denominado security developer encargado del desarrollo, de las revisiones de seguridad y los casos de pruebas.
- La asignación de una persona como responsable de seguridad.
- Brindar entrenamiento en concienciación de seguridad al scrum master y al Product Owner, antes del inicio del proyecto.
- Realizar una auditoria o pruebas de seguridad mediante externos.

De esta manera se obtienen recursos especializados en el desarrollo de funcionalidades y pruebas de seguridad, también se brinda entrenamiento a personas con roles importantes dentro de la creación del producto.

En [29] los autores identifican que las metodologías ágiles utilizan eficientemente la posibilidad de automatizar pruebas en cada sprint, sin embargo, reconocen que en el campo de la seguridad

las pruebas de intrusión automatizadas no tendrían mucho sentido si las funcionalidades de seguridad no están terminadas. Reconociendo el gran valor que este tipo de pruebas pueden aportarle al desarrollo seguro de software los autores proponen que se priorice el desarrollo de las funcionalidades de seguridad dentro del Product Backlog para que las pruebas de intrusión automatizadas puedan ser ejecutadas desde las etapas iniciales de un proyecto.

Con respecto a las pruebas de intrusión manuales los autores identifican que estas generalmente se ejecutan sobre software listo para su despliegue en ambientes productivos, por lo tanto, proponen que este tipo de pruebas solo se hagan cuando el Product Owner decida que se va a hacer un despliegue en producción o cuando haya cambios importantes sobre la funcionalidad.

Otro trabajo que también se refiere a la automatización de pruebas de seguridad es [7], en este trabajo el autor brinda una guía para desarrolladores en donde explica cómo preparar pruebas de seguridad con el fin de alcanzar una mayor consciencia entre los desarrolladores y crear una manera fácil y efectiva para que verifiquen que su código es seguro. Tanto [7] como [29] pretenden reducir el tiempo que normalmente toma crear y ejecutar pruebas de seguridad.

En Handbook of the Secure Ágile Software Development Life Cycle [32] se ofrecen alternativas para el manejo de la seguridad en proyectos, cuando el conocimiento en seguridad de sus responsables es insuficiente. Una de estas alternativas es el uso de historias de seguridad genéricas las cuales deben estar disponibles para toda la organización. Estas historias funcionan como una guía de referencia que será útil entornos donde los responsables de los proyectos carecen de suficiente conocimiento en seguridad para generar por sus propios medios historias de este tipo. Otra ventaja de este tipo de historias es disminuir la necesidad de entrenamiento al personal o contratar externos enfocados en la seguridad.

Adicional al uso de historias genéricas, los autores proponen que alguien del equipo, con conocimiento en seguridad, tome la responsabilidad de este aspecto del proyecto en caso de que el Product Owner no pueda hacerlo. Paralelamente proponen que el análisis de riesgos y la mitigación del mismo se someta a aprobación por parte del Product Owner donde el riesgo que se decide no mitigar quede bajo su responsabilidad, con el fin de involucrarlo y hacer que se interese en él tema.

Finalmente, en el libro se proponen dos alternativas para hacer análisis de riesgos, hacerlo durante el sprint planning o marcar las funcionalidades que se consideran riesgosas para hacer el modelamiento de riesgos antes de iniciar el desarrollo de la funcionalidad.

Al igual que en uno de los capítulos del libro *Handbook of the Secure Ágile Software Development Life Cycle*, en [6] se identifica que en las metodologías ágiles hay ciclos/sprints que duran entre dos y cuatro semanas haciendo difícil, que los equipos de trabajo puedan cumplir con prácticas y tareas de seguridad en un proyecto. Para eliminar impedimentos en los tiempos de creación de historias de seguridad el autor recomienda la utilización de una lista preestablecida de historias y tareas de seguridad.

En [33] los autores presentan varios problemas al aplicar prácticas de aseguramiento de software:

- Cuando se utiliza una metodología ágil el tiempo para realizar pruebas de seguridad en un sprint es corto en comparación a cuando se realizaban en metodologías en cascada, además las herramientas actuales no siempre son las adecuadas, necesitan código completamente desarrollado.
- El conocimiento y el contexto de la seguridad en los proyectos no es generalizado en los desarrolladores y cuando los expertos de seguridad hacen pruebas y reportan vulnerabilidades, los equipos de trabajo no conocen del tema y no brindan soluciones efectivas.
- Las tareas de seguridad no están priorizadas de manera correcta, necesitan tener una mayor prioridad.

Para resolver parte de estos problemas los autores proponen a partir de historias de usuario hacer un modelamiento de amenazas, obtener requerimientos de seguridad y posibles pruebas, etiquetando historias de seguridad de manera que se les pueda hacer un seguimiento durante todo el proceso. Además, los autores recomiendan diseñar pruebas de regresión y de seguridad automatizadas de tal manera que se puedan ejecutar dentro de los tiempos brindados por un sprint.

En [13] los autores determinan que los principales impedimentos que se presentan para llevar a cabo actividades de seguridad en metodologías ágiles son; el poco tiempo otorgado para hacer análisis y diseños de seguridad con su respectiva documentación detallada y el constante cambio en los requerimientos que se presenta durante un proceso de desarrollo ágil. Con el fin de mitigar estos impedimentos los autores proponen la utilización de spikes de análisis y diseño de seguridad en cada uno de los sprints, utilizando documentación no muy detallada pero suficiente para brindar la información necesaria.

[34] Es un trabajo donde se identifica que es necesario realizar prácticas formales de especificación cuando se está desarrollando un sistema crítico, el autor hace visible que cuando se utilizan metodologías ágiles, no todos los miembros de un equipo de trabajo tienen el conocimiento para hacer especificaciones formales. Para mitigar la falta de conocimiento en especificaciones formales el autor propone incluir dentro del equipo de trabajo personas con el conocimiento en para hacerlo y que en el sprint se divida el trabajo entre especificaciones detalladas y desarrollo de funcionalidades, a su vez propone aumentar el nivel de los integrantes del equipo que no conocen del tema mediante la interacción de trabajar juntos.

En [26] los autores establecen que para realizar actividades como los requerimientos de seguridad, el análisis de riesgos y los casos de abuso se presenta como impedimento el desconocimiento de seguridad de los Product Owner y la poca responsabilidad que se les da sobre la aceptación de riesgo. Para resolver estos problemas, los autores proponen la participación del Product Owner en la creación de los requerimientos de seguridad, en los casos de abuso y los análisis de riesgo, para lo cual se les debe proporcionar entrenamiento en seguridad y brindar herramientas como plantillas con requerimientos de seguridad previamente construidos.

Para el análisis de riesgo los autores proponen que durante la planeación del sprint se determine si una historia de usuario debe tener un análisis de riesgo, de ser así este análisis se debe realizar durante un spike. Para la aceptación del riesgo el Product Owner debe ser el responsable de asumir o no un riesgo.

Finalmente, los autores hablan de la codificación e identifican que dejar las pruebas de seguridad para el final puede hacer que los programadores, olviden el código y sea más difícil la corrección. Para esto proponen realizar pruebas estáticas de código automatizadas y pruebas de regresión automatizadas.

En [35] los autores identifican que en las metodologías ágiles se usan las historias de usuario y que este método de especificación no facilita la creación de casos de abuso, por esto proponen el uso de historias de abuso las cuales se deben crear a partir de una historia de usuario durante la reunión de planeación del Sprint. De una historia de usuario puede que no se desprenda ninguna historia de abuso o puede que se desprendan varias. Los autores recomiendan modelar la historia de abuso como un árbol de ataque, en este articulo los autores también recomiendan automatizar las pruebas de seguridad.

En [36] el autor identifica que las metodologías ágiles pueden ser modificadas para incluir prácticas de seguridad, por ejemplo: requerimientos de seguridad, análisis de riesgos, pruebas de seguridad. Por esto los autores proponen la inclusión de un experto de seguridad en los proyectos ágiles para evaluar el riesgo junto con el cliente, proponer historias de usuario relacionadas con los riesgos identificados y realizar programación par junto con el equipo de trabajo. La idea principal es difundir conocimiento y conciencia en seguridad tanto a desarrolladores como al cliente.

En [35] y [37] se expresa que en la ejecución de proyectos ágiles, aparentemente hay insuficiencia de documentación en las actividades relacionadas con el diseño, especificaciones e interfaces cuando los proyectos tienen un componente critico de seguridad. Esto es presentado como un impedimento para llevar a cabo actividades de seguridad con eficiencia pues una documentación ligera no se complementa con las prácticas tradicionales de seguridad, los autores expresan la necesidad de ampliar los recursos de documentación más allá de los que sugieren las metodologías ágiles.

En [38], [39] nos presentan un contraste entre las metodologías ágiles y las que están dirigidas por un plan o las basadas en modelos de madurez. Los autores proponen un método para incluir un análisis que puede determinar si un proyecto es puramente ágil o no, comparando características como la aplicación del proyecto, la administración, características técnicas y de personal. Estos trabajos pretenden eliminar incompatibilidades al querer usar prácticas demasiado pesadas en proyectos que deben ser más ágiles, como puede llegar a ser un análisis de riesgos exhaustivo.

Los autores de [28] ven que uno de los principales problemas para integrar cualquier práctica de seguridad en el ciclo de vida de una metodología ágil tiene que ver con el nivel de agilidad que tiene la práctica de seguridad a incluir, por esto plantean un algoritmo y el uso de puntajes para determinar el grado de agilidad de una práctica de seguridad, de esta forma se puede encontrar un equilibrio entre los costos de reducir la agilidad y aumentar el nivel de seguridad.

Al igual, en [40] hacen una comparación entre las prácticas de seguridad y las prácticas de desarrollo ágil en donde las primeras son consideradas pesadas mientras que las segundas son consideradas livianas e informales. Para unir los dos mundos los autores sugieren que para integrar prácticas de seguridad en metodologías ágiles es necesario utilizar prácticas de seguridad más livianas. Sin embargo, el autor recomienda no perder de vista que si se incorpora una

práctica muy ágil pero que aporta poco a la disminución de riesgos de seguridad no se puede considerar benéfico. Para evitar que se pueda presentar esta problemática al adoptar prácticas de seguridad en metodologías ágiles el autor propone utilizar un método que mide la efectividad y agilidad de una práctica permitiendo seleccionar las más convenientes.

En [9], [41] los autores recomiendan identificar los objetos y sujetos relevantes para la seguridad, clasificarlos y hacer un a análisis de riesgo de los mismos. Facilitando que todos los participantes del equipo no solo el Product Owner conozca donde se debe prestar atención a la seguridad.

En [23] los autores afirman que las historias de usuario pueden ser inadecuadas para capturar requerimientos que tienen que ver con seguridad debido a que están centradas en las restricciones del sistema y no en los comportamientos del usuario. Además, que la captura de restricciones no está bien documentada en la literatura de las metodologías ágiles.

Debido a esto proponen extender las historias de usuario con historias de abuso en donde se establezcan comportamientos inadecuados de un usuario. Los autores proponen la automatización de pruebas a estos comportamientos usando herramientas como FitNesse, utilizada en las pruebas de aceptación.

Los autores también proponen la inclusión de expertos de seguridad para entrenar, hacer coaching y ser mentores de los miembros del equipo a su vez sugieren que debería haber un responsable por la seguridad y el riesgo que debe aconsejar y enseñar a los clientes a reconocer sus necesidades de seguridad.

Los autores de [42] consideran que los principales impedimentos para aplicar prácticas de seguridad en metodologías ágiles, son que los practicantes de las metodologías ágiles consideran estas prácticas costosas, pesadas y poco beneficiosas en un contexto ágil. Las prácticas de seguridad se apoyan en información detallada y formalizada contrario a lo que ocurre en una metodología ágil en donde hay constante cambio en los requerimientos. Para estos los autores en su trabajo no definen una solución específica para cada uno de los impedimentos identificados, pero proponen tomar fragmentos de distintitos métodos de aseguramiento para crear un nuevo método compatible con las metodologías ágiles.

En [43] los autores analizan las fases de Scrum e identifican que en la primera fase se realizan las historias de usuario las cuales son aprobadas por el cliente quien generalmente posee poco conocimiento de seguridad y riesgos. Para abordar este problema proponen la creación de un

backlog adicional para gestionar la seguridad sin afectar la agilidad. También proponen la creación de un "Security Master" quien será responsable de gestionar el backlog de seguridad, identificando funcionalidades con riesgos de seguridad, documentando el riesgo y haciendo pruebas de seguridad de las funcionalidades.

En [44] los autores identifican que integrar las prácticas de ingeniería de requisitos de seguridad tal y como fueron concebidas pueden agregar más pasos a las metodologías ágiles y hacer que estas pierdan los beneficios de usarlas. Por lo tanto, proponen un enfoque para encontrar un equilibrio entre las prácticas convencionales de seguridad y las prácticas ágiles de Extreme Programming (XP). Para lograrlo proponen la modificación de la metodología ágil XP, mediante la integración de varias actividades, entre ellas la identificación de activos sensitivos para la seguridad de la información, creación de historias de abuso con su respectivo análisis de riesgo, definición de historias de seguridad y la definición de estándares de codificación.

7.2.LISTA CRUZADA IMPEDIMENTOS Y PRÁCTICAS DE SEGURIDAD

A continuación, por cada una de las prácticas de seguridad identificadas en las políticas y lineamientos de seguridad se presenta una lista cruzada de los impedimentos para su ejecución y las prácticas que los podrían remover.

1. Requerimientos de seguridad de la información

Impedimento	Solución
• Desconocimiento en seguridad • El Product Owner no conoce de seguridad	• Creación de un responsable de seguridad[31] [14][32] [27] [36] [23] [43] • Incluir conocimiento en seguridad con un externo. [8] [11] • Equipo especializado en seguridad[14] [11] [34] • Entrenamiento en seguridad [11] [26] • Uso de historias de seguridad genéricas [32] • Utilizar plantilla requerimientos de seguridad[6] [26]
Tiempo insuficiente para especificar (proceso pesado)	• A partir de backlog obtener un security backlog[31] [33] [26] • A partir de análisis de riesgos ligero

	obtener requerimientos de seguridad y casos de abuso [8] [44] • Usar documentación resumida en vez de documentación extensa [8] [13] [44] • Evaluar el nivel de agilidad prácticas de seguridad [28] [40] • Realizar spikes para actividades de seguridad [13]
Desconocimiento de las funcionalidades que requieren atención en seguridad	• Sacar un security backlog a partir del backlog[31] • Etiquetas de marca de seguridad en el backlog [8] [33] • Incluir conocimiento en seguridad con un externo. [8] • Realizar esta actividad en las historias de usuario [33] • Mayor documentación [35][37] • Identificar sujetos y objetos relevantes para seguridad. [9], [41]

2. Plantilla de requerimientos de seguridad

Impedimento	Solución
Desconocimiento en seguridad	• Incluir conocimiento en seguridad con un externo. [8] • Creación de un responsable de seguridad[31] [14] [11] • Incluir un equipo especializado en seguridad [14] • Usar plantilla de historias de seguridad [32]
El Product Owner no conoce de seguridad	• Creación de un responsable de seguridad[31] [14][32] [27]

3. Desarrollo y ejecución de pruebas de casos de abuso (se refiere a las prácticas especificadas como Desarrollo de casos de abuso y Ejecución de casos de abuso).

Impedimento	Solución
• Desconocimiento en seguridad	• Incluir conocimiento en seguridad

• Desconocimiento como hacer casos de abuso en metodologías ágiles • El Product Owner no conoce de seguridad	con un externo. [8] • Creación de un responsable de seguridad[31] [14] [11] [23] [43] • Incluir un equipo especializado en seguridad [14] [34] • Realizar historias de abuso en el sprint planning, modelar como árbol de ataque. [35] • Entrenamiento en seguridad [11] [26]
Tiempo insuficiente para especificar (proceso pesado)	• A partir de análisis de riesgos ligero obtener requerimientos de seguridad y casos de abuso [8] [44] • Usar documentación resumida en vez de UMLsec [8] [13] • Realizar spikes para actividades de seguridad [13] • Evaluar nivel de agilidad [28] [40] • Automatizar pruebas de aceptación. [26] [35] [23]
Desconocimiento de las funcionalidades que requieren atención en pruebas de seguridad	• Sacar un security backlog a partir del backlog[31] • Etiquetas de marca de seguridad en el backlog [8] [33] • Incluir conocimiento en seguridad con un externo. [8] • Mayor documentación [35][37] • Identificar sujetos y objetos relevantes para seguridad. [9], [41]
Constantes cambios en los requerimientos	• Extracción de activos de las historias de usuario y creación de casos de abuso antes de cada sprint [12] • Utilización de Spikes en cada sprint para verificar la seguridad [13] [26]

4. Análisis de riesgos de las nuevas funcionalidades y diseño de seguridad (se refiere a las prácticas especificadas como análisis de riesgos de las nuevas funcionalidades y Diseño de la seguridad del sistema de información).

Impedimento	Solución
• Desconocimiento en seguridad • El Product Owner no conoce de seguridad	• Incluir conocimiento en seguridad con un externo. [8] • Incluir equipo especializado en seguridad [14] [34] • Capacitar al Product Owner y al scrum master [11] [26] [36] • Creación de un responsable de seguridad[31] [14][32] [27] [36] [23] [43] • Responsabilizar al Product Owner del riesgo no mitigado [32] [27] [26]
Tiempo insuficiente (proceso pesado)	• Realizar un análisis de riesgos ligero valorando en la afectación causada por un ataque. [8] [44] • Hacer la evaluación de riesgos en el sprint planning. [32] [27] • Marcar las funcionalidades con riesgos en el sprint planning y hacer el análisis antes de iniciar desarrollo[32] [27] • Realizar spikes para actividades de seguridad [13] [26] • Análisis previo para determinar prácticas de riesgos según tipo de proyecto [38], [39]
Un diseño de seguridad agrega complejidad al flujo de una metodología ágil basada en SCRUM.	• Realizar un sprint cero para realizar actividades de diseño. [12] [14]
Desconocimiento de las funcionalidades que requieren atención en pruebas de seguridad	• Identificar sujetos y objetos relevantes para seguridad. [9], [41]

5. Codificación segura y revisión de código (agrupa las siguientes prácticas: Codificación e Implementación de la Solución y Revisión de Código).

Impedimento	Solución
Desconocimiento de las funcionalidades que requieren atención en seguridad	• Etiquetas de marca de seguridad en el backlog [8] • Incluir un equipo especializado en seguridad [14] • Mayor documentación [35][37]
Desconocimiento en seguridad para resolver	• Incluir conocimiento en seguridad con

retos de codificación	un externo. [8] [23] • Incluir un equipo especializado en seguridad [14] • Crear un rol especializado de desarrollo de seguridad [11] • Programación par utilizando un experto en seguridad [36]

6. Pruebas de intrusión.

Impedimento	Solución
Las pruebas de intrusión se hacen sobre sistemas de información funcionando.	• No hacer pentesting todos los sprint solo hacerlo cuando el Product Owner decide hacer un reléase a producción [29] • Hacer pruebas de regresión de seguridad [33] [26]
Las pruebas de no se pueden hacer en los primeros sprints	• Automatizar pentesting sobre funcionalidades de seguridad priorizadas [29] [7] [33] [26]
Desconocimiento de las funcionalidades que requieren atención en seguridad	• Etiquetas de marca de seguridad [2] [33] • Creación de un responsable de seguridad especifica cuáles son sus preocupaciones de seguridad en la funcionalidad [31]
Desconocimiento en seguridad ejecutar casos de prueba	• Incluir conocimiento en seguridad con un externo. [8] [11] • Incluir un equipo especializado en seguridad [14] [11]

7.3. DEFINICION DE PRÁCTICAS

De la sección anterior se deprenden un conjunto de prácticas que pueden ser utilizadas para mitigar las diferentes causas por las cuales no se cumplen con los lineamientos establecidos para el desarrollo de software seguro en EPM. Es conveniente entonces tener la definición que se toma en este trabajo para cada una de las prácticas que más adelante serán evaluadas, esto se

hace necesario ya que uno o varios autores pueden proponer una misma práctica, pero puede haber diferencias entre lo que propone uno u otro autor.

Creación de un responsable de seguridad: Esta práctica consiste en la creación de un rol responsable de gestionar o ejecutar actividades de seguridad durante el proyecto, en el contexto de los requerimientos de seguridad este rol será el responsable de obtener los requerimientos de seguridad o historias de seguridad junto con el Product Owner. Esta persona deberá tener conocimiento de seguridad y podrá ser el ejecutor de actividades dentro de todo el proceso.

Incluir conocimiento de seguridad con externo: Incluir recursos externos con conocimientos en seguridad, distintos al equipo de trabajo ya conformado, para en momentos específicos resolver posibles retos en la implementación de la seguridad o ejecutar tareas específicas como desarrollar una historia con marca de seguridad o para planear y ejecutar casos de prueba.

Entrenamiento en seguridad: Por medio de esta práctica se pretende que los diferentes integrantes del equipo tengan conciencia de la seguridad y de sus implicaciones. Esta práctica se orienta principalmente al Scrum Master y al Product Owner.

Equipo especializado en seguridad - Recursos dedicados a actividades de seguridad: Dividir el equipo de trabajo y dedicar parte del equipo a realizar solo las actividades que tengan que ver con seguridad. Este equipo no realizaría otro tipo de actividades. Ejemplo generar roles como security developer y security master.

Utilización de una plantilla con requerimientos de seguridad transversales: esta práctica consiste en tener una lista prefabricada de requerimientos de seguridad, que en este caso serán historias de usuario de seguridad genéricas, la idea es que estas sean revisadas cuando se está creando el Product Backlog para definir si aplican en el proyecto. También se busca que esta plantilla sea alimentada por los diferentes proyectos.

Security Backlog a partir del Product Backlog: plantea recorrer cada una de las historias del Product Backlog para a partir de cada uno de los elementos del mismo generar historias de seguridad o casos de abuso y generar un backlog alterno llamado security backlog

Utilizar documentación resumida: las prácticas de seguridad se caracterizan por basarse en procesos donde existe una documentación bastante detallada, para el caso de los requerimientos de seguridad la recomendación es que se utilice una documentación muy sencilla que no

implique mucho tiempo pero que a la vez responda a las necesidades de las prácticas a introducir en el proceso. Para las actividades en seguridad se establecía que debía hacerse un modelado de los casos de abuso utilizando herramientas como UMLsec, en esta práctica lo que se busca es reducir al máximo esta documentación no utilizando este tipo de herramientas.

Realizar spikes para actividades de seguridad: Spikes de análisis y diseño de seguridad utilizando documentación muy ligera.

Mayor documentación: Lo expresado en la bibliografía es la necesidad de ampliar los recursos de documentación más allá de los que sugieren las metodologías ágiles en las actividades relacionadas con el diseño, especificaciones e interfaces.

Identificar sujetos y objetos relevantes para seguridad: para eliminar impedimentos en el desconocimiento de en donde aplicar actividades de seguridad en las diferentes fases del desarrollo, se identifican los objetos y sujetos relevantes para la seguridad, para después clasificarlos y hacer un a análisis de riesgo. Facilitando que todos los participantes del equipo conozcan donde se debe prestar atención.

Realizar historias de abuso en el sprint planning, modelar como árbol de ataque: plantea el uso de historias de abuso las cuales se deben crear a partir de una historia de usuario en la reunión de planeación de la iteración. De una historia de usuario puede que no se desprendan ninguna historia de abuso o puede que se desprendan varias.

A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso: utilización de técnicas de identificación de historias de usuario importantes, mediante la valoración de la afectación en caso de un ataque a esa funcionalidad. Este análisis se convierte en un análisis de riesgos ligero en donde se documenta la preocupación con la historia de usuario por medio de una historia de usuario de seguridad o como un caso de abuso.

Automatizar pruebas de aceptación: una vez se han definido las historias de seguridad se deben utilizar las herramientas adecuadas para automatizar las pruebas de aceptación de las mismas.

Etiquetas de marca de seguridad en el backlog: esta práctica consiste en dejar una marca en la historia de usuario que necesita especial atención, para que esta sea tenida en cuenta durante el proceso de desarrollo.

Responsabilizar al Product Owner del riesgo no mitigado: una vez se identifica un riesgo y sus consecuencias queda en manos del Product Owner decidir si implementar un control que mitigue el riesgo, la Práctica dice que se debe hacer responsable al Product Owner de aceptar el riesgo en caso de no querer implementar controles para mitigarlo.

Hacer evaluación de riesgos en el sprint planning: Práctica que plantea que durante la ejecución del sprint planning se tome cada una de las funcionalidades que se van desarrollar durante el sprint para en ese momento hacer un análisis del riesgo.

Marcar las funcionalidades con riesgos en el sprint planning y hacer el análisis antes de iniciar desarrollo: Práctica que plantea que durante la ejecución del sprint planning se tome cada una de las funcionalidades para marcar las tareas y funcionalidades riesgosas con el fin de luego hacer el modelamiento de riesgos antes de iniciar el desarrollo de la funcionalidad.

Realizar sprint de seguridad: Realizar un sprint cero para realizar actividades de análisis y diseño de seguridad

Programación par utilizando un experto en seguridad: en esta práctica se propone la inclusión de un experto de seguridad en los proyectos ágiles para desarrollar con el equipo de trabajo de modo que se asegure la implementación de código seguro y aumente la conciencia en seguridad.

No hacer pentesting todos los sprint solo hacerlo cuando el Product Owner decide hacer un release a producción: Está práctica propone solo hacer pentesting cuando el usuario decida que lo que se entrega es lo suficientemente funcional para ir a producción pues en este caso, las pruebas de pentesting se realizarán sobre software que en teoría debe estar listo para salir a producción.

Hacer pruebas de regresión de seguridad: las pruebas de regresión: consiste en planear y ejecutar pruebas de regresión pero que tengan que ver con seguridad, la idea de estas pruebas es que se automaticen y puedan ser ejecutadas cada sprint.

Automatizar pentesting: hay pruebas de seguridad que pueden ser lanzadas desde herramientas automáticas, algunas de estas pruebas revisan que el código sea seguro y otras revisan vulnerabilidades conocidas.

8. DEFINICIÓN PRÁCTICAS DE SEGURIDAD PARA METODOLOGÍA ÁGIL DE EPM

En este capítulo se evalúan las diferentes prácticas encontradas en la literatura que ayudan a remover los impedimentos identificados para el cumplimiento de las políticas y lineamientos de seguridad en el desarrollo de software. Una vez finalizada esta evaluación se realiza una guía de prácticas ágiles de seguridad para el desarrollo de software en EPM.

8.1.EVALUACIÓN PRÁCTICAS RELEVANTES

A continuación, se evalúan las prácticas encontradas para remover los impedimentos identificados tanto al interior de EPM como en la literatura consultada. El siguiente cuadro presenta 5 columnas:

- La primera columna llamada **Práctica** especifica la práctica evaluada.
- La segunda columna "**Compatibilidad Metodologías ágiles**" es en la que se da una puntuación con base a la compatibilidad que tiene la práctica con las metodologías ágiles y SCRUM, entre mayor sea el puntaje se considera que es más compatible,
- En la tercera columna **Relevancia Bibliográfica** se da un puntaje con base al número de autores que proponen está práctica y como esta se compara en número de autores con respecto a las otras prácticas.
    - Tres o más referencias = 3 puntos.
    - Dos referencias = 2 puntos.
    - Una referencia = 1 punto.
- En la cuarta columna **Compatibilidad cultural empresarial EPM** se da un puntaje basado en cómo podría ser recibida la práctica basado en el comportamiento cultural empresarial observado en EPM, esta valoración es subjetiva basada en mi conocimiento de 7 años en la compañía y de conversaciones con pares de otras áreas.
- Por último, está la columna **Total** donde se totalizan los puntajes obtenidos.

A cada uno de los ítems a evaluar se le asigna un puntaje de uno a tres siendo tres la mayor puntuación posible por cada uno de los ítems. Adicionalmente solo se seleccionarán prácticas

que obtengan un puntaje mayor o igual a 5.4 ósea un 60%. Si existen varias prácticas que cumplan este requisito y apunten a remover un mismo impedimento se podrán escoger las dos siempre y cuando sean complementarias, si no se escogerá la de mayor puntaje. En caso de que ninguna práctica cumpla con el puntaje exigido este lineamiento no se tomará en cuenta a la hora de realizar la guía producto de este trabajo.
Por último, cada una de las prácticas se evalúa teniendo en cuenta la definición que se brindó en la sección 7.3.

Impedimento encontrado en todos los lineamientos de seguridad

Práctica	**Compatibilidad Metodologías ágiles**	**Relevancia bibliográfica**	**Compatibilidad cultural empresarial EPM**	**TOTAL**
Creación de un responsable de seguridad.	1 Scrum no establece este rol [45]	3	3 En la cultura empresarial las personas reaccionan mejor si existe un responsable de un tema.	7
Incluir conocimiento de seguridad con externo	1 Sería un indicador que no se confía la ejecución del trabajo al equipo principio ágil 5 “Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.” confiaría en el equipo para	2	1 no existe disposición de adicionar más gastos	4

	hacer el trabajo No se cumplirían con las características de un equipo de desarrollo en scrum [45]			
Entrenamiento en seguridad	3	2	1	6
	Dentro de un proyecto de scrum se tienen en cuenta los entrenamientos que se requieren para el proyecto. [46]		las personas están abiertas a capacitaciones, pero los clientes no están abiertas al costo asociado	
Recursos dedicados a actividades de seguridad	1	1	2	4
	No se cumple con una de las características de un equipo de trabajo que establece que: "Scrum no reconoce ningún sub-equipo en el Equipo de Desarrollo, independientemente de los dominios particulares que necesitan ser abordados como pruebas o análisis de negocios; No hay excepciones para esta regla" [45]		Es aceptable dentro de la empresa siempre y cuando no se generen sobre costos y se hagan las entregas establecidas, sin embargo, se vela por que se siga la metodología ágil establecida.	

Equipo especializado en seguridad	1	2	2	5
	Proponen la creación de roles exclusivos de seguridad: security developer, estos roles no están establecido en SCRUM. [45]		Es aceptable dentro de la empresa siempre y cuando no se generen sobre costos y se hagan las entregas establecidas, sin embargo, se vela por que se siga la metodología ágil establecida.	

Prácticas seleccionadas:

- Creación de un responsable de seguridad
- Entrenamiento en seguridad

1. Requerimientos de seguridad de la información

Práctica	**Compatibilidad Metodologías ágiles**	**Relevancia bibliográfica**	**Compatibilidad cultural empresarial EPM**	**TOTAL**
Plantilla genérica de requerimientos de seguridad	3	2	1	6
	puede acortar el tiempo de creación del Product Backlog, no se establece como se hace el Product		Habría que adicionar historias de usuario a la plantilla genérica si aplica. En las áreas de TI no	

	Backlog[45]		hay una cultura de documentar.	
Security Backlog a partir del Product Backlog	1 Scrum no habla de un backlog diferente al sprint backlog o Product Backlog[45]	3	1 En EPM no hay muchas personas con conocimiento en seguridad, para hacer esta actividad los proveedores no cuentan con este recurso	4
A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso**.**	2 Aunque formalmente scrum no habla de riesgos, se puede enmarcar esta actividad en el evaluación y priorización de riesgo que es aplicable a cualquier entregable al usurio. [46]	2	2 En EPM no hay muchas personas con conocimiento en seguridad, al análisis de riesgos es específico para seguridad. Sin embargo, hay cultura de gestión de riesgos.	6
Usar documentación resumida	3 la filosofía de scrum evitar documentación innecesaria [46], mejor si para la seguridad se puede evitar el exceso de documentación.	3	3 Dentro de la cultura de la gerencia de TI se piensa que documentar toma mucho tiempo. Si esta actividad no va a ser muy	9

			prolongada mejor	
Realizar spikes para actividades de seguridad	2 No hay una definición concreta de spike en SCRUM. Sin embargo, son aplicados en metodologías ágiles como XP y hay evidencias de ser buen complemento de SCRUM [4]	2	2 Los usuarios podrían percibir menores funcionalidades en los entregables incrementales y considerarlo sobre costo	6
Mayor documentación	1 La filosofía de scrum evitar documentación innecesaria [46], la idea es hacer los procesos más ligeros.	2	1 Esta práctica no sería bien recibida por el personal de la gerencia de TI. Implicaría mayor documentación esta cultura no es bien aceptada	4
Identificar sujetos y objetos relevantes para seguridad.	2 En scrum ni en metodologías ágiles se habla al respecto, sin embargo, no se visualiza que no se pueda hacer	2	1 En EPM no hay muchas personas con conocimiento en seguridad, para hacer esta actividad los proveedores no cuentan con este recurso, sería	5

			una actividad resistida	

Las Prácticas seleccionadas fueron:

- Utilización de una plantilla con requerimientos de seguridad transversales.
- Utilizar documentación resumida.
- A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso.
- Realizar spikes para actividades de seguridad.

2. Plantilla de requerimientos de seguridad

Para esta práctica no se realizará una evaluación adicional a la realizada en el punto anterior, pues gracias a la literatura, se comprueba que este lineamiento de seguridad es una de las prácticas de seguridad que ayudan a remover el impedimento de tiempo insuficiente para la creación de requerimientos de seguridad. Adicionalmente se observa que los principales impedimentos a solucionar con este lineamiento de EPM tienen que ver con el desconocimiento en seguridad y por parte del equipo y del Product Owner, estos mismos impedimentos fueron evaluados en el punto anterior dejando como resultado la inclusión de un responsable de seguridad en el proyecto.
Esto indica que una vez exista un responsable de seguridad este será el encargado de evaluar esta plantilla de requerimientos de seguridad junto con el Product Owner seleccionar, cuales aplican.

3. Desarrollo y ejecución de pruebas de casos de abuso (se refiere a las prácticas especificadas como Desarrollo de casos de abuso y Ejecución de casos de abuso).

Práctica	**Compatibilidad Metodologías ágiles**	**Relevancia bibliográfica**	**Compatibilidad cultural empresarial EPM**	**TOTAL**
Realizar	**1**	**1**	2	**4**

historias de abuso en el sprint planning, modelar como árbol de ataque.	En el sprint planning se dice que puede ser entregado en el incremento del sprint y como será alcanzado este objetivo, no se contemplan este tipo de actividades. Esta práctica puede aumentar el tiempo de la misma en más de 8 horas. (sprint planning [45])		Hacer este tipo ejercicios durante esta sesión aumentaría la conciencia de la seguridad, sin embargo, se necesitarían más personas con conocimiento en seguridad para llevar a cabo esta actividad	
A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	**6**
Usar documentación resumida	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	9
Realizar spikes para actividades de seguridad dentro de los sprint	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	6
Automatizar	**3**	**3**	**2**	**8**

pruebas de aceptación y regresión.	La automatización de pruebas de esta dentro de metodologías ágiles como XP [47] y algunos autores afirman que estas prácticas son compatibles con scrum [48]		EPM actualmente está trabajando fuertemente en la implementación de este tipo de pruebas automatizadas, pero no son muy conocidas	
Etiquetas de marca de seguridad en el backlog	**2** En las metodologías ágiles no se habla respecto al tema, pero no se observan problemas para realizarlo	**3**	1 requiere de personas con conocimiento en seguridad y no hay muchas	**6**
Identificar sujetos y objetos relevantes para seguridad.	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	5

Las Prácticas seleccionadas fueron:

- A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso
- Usar documentación resumida
- Automatizar pruebas de aceptación
- Etiquetas de marca de seguridad en el backlog

4. **Análisis de riesgos de las nuevas funcionalidades y diseño de seguridad (se refiere a las prácticas especificadas como análisis de riesgos de las nuevas funcionalidades y Diseño de la seguridad del sistema de información).**

Práctica	**Compatibilidad Metodologías ágiles**	**Relevancia bibliográfica**	**Compatibilidad cultural empresarial**	**TOTAL**

			EPM	
Responsabilizar al Product Owner del riesgo no mitigado	2 En metodologías ágiles ni en SCRUM se habla al respecto, no se observa incompatibilidad	**3**	2 los usuarios no acostumbran a asumir este tipo de responsabilidad puede haber resistencia. Por otro lado, en TI será recibida con gusto esta práctica	**7**
A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	**6**
Hacer evaluación de riesgos en el sprint planning.	1 En el sprint planning se planea que puede ser entregado y como será alcanzado este objetivo. No se contemplan este tipo de actividades. Implementar esta práctica puede aumentar el tiempo del sprint planning en más de lo establecido para esta actividad en SCRUM. (sprint planning [45])	2	2 Hacer esta evaluación en el sprint planning puede tomar más tiempo y normalmente allí se involucra a todo el equipo, como ventaja el equipo aprendería de este proceso	5
Marcar las	1	2	1	4

funcionalidades con riesgos en el sprint planning y hacer el análisis antes de iniciar desarrollo	En el sprint planning se planea que puede ser entregado y como será alcanzado este objetivo. No se contemplan este tipo de actividades. Implementar esta práctica puede aumentar el tiempo del sprint planning en más de 8 horas. (sprint planning [45])		Un análisis de riesgo debe involucrar a varias roles que no necesariamente serian fáciles de reunir en EPM una vez se comienza el desarrollo.	
Realizar spikes para actividades de seguridad	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	**6**
Análisis previo para determinar mediante riesgos si un tipo de proyecto es ágil				No se califica esta práctica pues está orientada a determinar si un proyecto debe ser ágil o no, para este trabajo nos basamos en proyectos ágiles
Realizar sprint de seguridad	1	2	2	**5**
	Un sprint de seguridad significa que no se tendrán software funcional entregable en un sprint		no se ve algún impedimento, sin embargo, necesitaría involucrar personas con conocimiento en seguridad, no hay suficientes para cubrir el proyecto.	

Identificar sujetos y objetos relevantes para seguridad.	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	5

Las Prácticas seleccionadas fueron:

- Responsabilizar al Product Owner del riesgo no mitigado
- A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso
- Realizar spikes para actividades de seguridad.

5. **Codificación segura y revisión de código (agrupa las siguientes prácticas: Codificación e Implementación de la Solución y Revisión de Código).**

Práctica	**Compatibilidad Metodologías ágiles**	**Relevancia bibliográfica**	**Compatibilidad cultural empresarial EPM**	**TOTAL**
Etiquetas de marca de seguridad en el backlog				**Ya fue seleccionada en otro lineamiento y aplica para el impedimento encontrado**
Mayor documentación	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	**4**
Incluir conocimiento en seguridad con un externo	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	**4**
Incluir un equipo especializado en seguridad	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	**5**

Crear un rol especializado de desarrollo de seguridad	**1**	**1**	2	**4**
	Proponen la creación de roles exclusivos de seguridad: security developer, este rol no está establecido en SCRUM. [45] No se cumple con una de las características de un equipo de trabajo que establece que: "Scrum no reconoce ningún sub-equipo en el Equipo de Desarrollo, independientemente de los dominios particulares que necesitan ser abordados como pruebas o análisis de negocios; No hay excepciones para esta regla" [45]		las personas pueden tener prevención con que se les asigne este rol por falta de conocimiento, como el proveedor es quien lo debe adicionar al grupo de trabajo y dada su escasez en el mercado puede aumentar costos, lo que no sería bien recibido. su presencia ayudaría a que otros sepan de seguridad	
Programación par utilizando un experto en seguridad	**2**	**1**	**2**	**5**
	La programación par es una práctica propia de XP Extreme programming y algunos autores		La bibliografía muestra que sería beneficioso a nivel de seguridad, sin embargo,	

	afirman que estas prácticas son compatibles con scrum [48]		tendría un rechazo debido a que generaría el costo de una persona que se paga al contratista y que no necesariamente se avanzaría al mismo ritmo si se está haciendo programación par.	

Prácticas seleccionadas:

Para este lineamiento solo se selecciona "Etiquetas de marca de seguridad en el backlog" mediante estas etiquetas los desarrolladores podrán identificar cuáles son las historias de usuario con las cuales deben tener más cuidado por el hecho de ser más críticas o sobre las cuales se tienen preocupación con respecto a la seguridad. No se selecciona ninguna otra práctica ya que no cumplen con el mínimo definido, En su mayoría estas prácticas obtuvieron poca calificación por su compatibilidad con las metodologías ágiles.

6. Pruebas de intrusión.

Práctica	**Compatibilidad Metodologías ágiles**	**Relevancia bibliográfica**	**Compatibilidad cultural empresarial EPM**	**TOTAL**
No hacer pentesting todos los sprint solo hacerlo cuando el Product Owner decide hacer un releáse a producción	**1**	**1**	**2**	**4**
	Hacer pentesting es un proceso largo, que puede agregar mucho tiempo y reducir la capacidad del equipo de hacer entregas		En EPM no existe la cultura de hacer pentesting, solo se hace algunas veces y al final, puede ayudar a fomentar la	

	constantes primer y tercer valor ágil [49]		Práctica	
Hacer pruebas de regresión de seguridad	**2**	**2**	2	**6**
	No se encuentra ninguna incompatibilidad, se harían cuando se ejecuten las pruebas dentro del sprint. Sin embargo, no se encuentra algo en SCRUM acerca de las pruebas.		Para EPM en algunos proyectos se acostumbra a hacer pruebas de regresión. Esta práctica no siempre se lleva a cabo.	
Automatizar pentesting.	**2**	**3**	**2**	**7**
	No se encuentra ninguna incompatibilidad, pero tampoco nada concreto del tema en metodologías ágiles ni específicamente en SCRUM		se deben licenciar herramientas que permitan automatizar pruebas de seguridad, pero se está realizando el proceso	
• Incluir conocimiento en seguridad con un externo. • Incluir un equipo especializado en seguridad	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya fue evaluada anteriormente	Ya habían sido evaluadas no cumplieron con el puntaje mínimo
Etiquetas de marca de seguridad				**Ya fue seleccionada en otro lineamiento y aplica para el**

				impedimento encontrado

Las Prácticas seleccionadas fueron:

- Hacer pruebas de regresión de seguridad.
- Automatizar pentesting.
- Etiquetas de marca de seguridad.

8.2. GUÍA DE PÁCTICAS ÁGILES DE SEGURIDAD PARA EL DESARROLLO DE SOFTWARE EN EPM

Como resultado del análisis presentado en las secciones y capítulos anteriores, se crea una guía preliminar de prácticas de seguridad para proyectos ágiles en EPM. A continuación, se presentan cada una de las prácticas que componen la guía.

1. Entrenamiento en seguridad

El entrenamiento en seguridad debe realizarse antes de la creación y refinamiento del Product Backlog. Debe ser recibido por el Product Owner y Scrum Master asignados al proyecto a quiénes por lo menos se les ofrece un entendimiento de la importancia de la seguridad y cómo debe ser abordada.

La capacitación es ofrecida por personal de TI con soporte de material de seguridad el cual estará al alcance de todos los equipos de desarrollo. En [50] - [52] se ofrecen recursos útiles para la formación en seguridad.

2. Establecer un responsable de seguridad

El responsable de seguridad debe ser una persona diferente al Product Owner. Sus responsabilidades son verificar que todas las prácticas descritas en la guía sean ejecutadas durante el proyecto y asesorar al resto del equipo en la toma de decisiones con respecto a seguridad. Dentro de sus funciones principales se encuentra:

- En la creación del Product Backlog, asesora al Product Owner en la creación de posibles historias de usuario que tengan que ver con seguridad y ayuda a la definición de criterios de aceptación
- Junto con el Product Owner y demás miembros del equipo debe asesorar en la utilización de historias de usuario de seguridad genéricas transversales.
- Una vez definidas las historias de usuario debe revisarlas y documentar de manera resumida los posibles riesgos de seguridad que la historia de usuario debe tener en cuenta.
- Realizar junto con el Product Owner una evaluación de riesgos ligera.
- Define junto con el equipo de desarrollo historias de abuso.
- Asesora al equipo de desarrollo en el análisis y diseño de la seguridad cuando este se realice.
- Establece etiqueta de seguridad en las historias de usuario que deben tener especial atención en materia de seguridad.
- Hace consciente al Product Owner que cuando no se mitiga un riesgo que es lo que se está asumiendo.
- Verifica la utilización de pruebas de automatizadas de seguridad.

3. Historias de usuario de seguridad Transversales

En esta práctica recomienda utilizar historias de usuario de seguridad genéricas que pueden aplicar a cualquier tipo de proyecto. Las historias de seguridad transversales deben ser revisadas durante la creación y actualizaciones del backlog para determinar si alguna de ellas aplica, según los requerimientos del negocio.

Si durante el desarrollo del proyecto se identifica una historia de seguridad transversal que no había sido identificada previamente, esta debe ser agregada para el beneficio de otros proyectos.
Utilizar un repositorio con historias de usuario de seguridad trasversales es importante porque asegura que todos los proyectos pueden incluir funcionalidades de seguridad genéricas mínimas sin importar la experiencia o el conocimiento en seguridad de los diferentes integrantes del equipo. De no existir este tipo historias de seguridad, se corre el riesgo que debido a la falta de experiencia o conocimiento el proyecto no cuente con funcionalidades mínimas de seguridad.

4. Análisis de riesgos ligero

La finalidad de esta práctica es realizar un análisis de riesgo sobre las funcionalidades propuestas para implementación durante el proyecto.

El primer paso es identificar en las funcionalidades si se puede presentar algún riesgo de seguridad para la compañía y el proyecto teniendo en cuenta si este riesgo se presenta por factores internos o externos.

Después de identificar los riesgos de seguridad, el Product Owner y un experto en seguridad deben hacer una ponderación del riesgo utilizando como herramienta la matriz de riesgo de la organización. La matriz de riesgo contiene dos ejes, uno con el impacto y otro con la probabilidad, el impacto es la medida en la que la materialización de un riesgo puede afectar al proyecto o la organización, la probabilidad es la variable en donde se evalúa que tan posible es que se materialice un riesgo.

Dependiendo del resultado de la evaluación se toma la decisión de mitigar o no el riesgo. Estos resultados deben quedar documentados brevemente especificando el riesgo identificado y su resultado (ver: **Responsabilizar al Product Owner del riesgo no mitigado**).

Mitigar un riesgo puede implicar realizar historias de seguridad detalladas, casos de abuso, tomar decisiones de arquitectura o de diseño. Una vez se ha identificado una historia épica o alguna historia de usuario esta debe ser etiquetada en el backlog mediante documentación del riesgo identificado y la ponderación otorgada (ver: **Etiquetas de marca de seguridad en el backlog**).

Se recomienda que esta práctica de seguridad se realice sobre las historias épicas una vez realizado el Product Backlog y antes de cada planeación de sprint sobre historias de usuarios menos generales derivadas del refinamiento de las historias épicas.

5. Responsabilizar al Product Owner del no mitigado

Una vez se identifica un riesgo y sus consecuencias queda en manos del Product Owner decidir si implementar un control que mitigue el riesgo identificado, si el Product Owner decide no mitigar el riesgo se debe le debe hacer responsable del riesgo asumido, dejando claro las consecuencias de no hacerlo.

6. Etiquetas de marca de seguridad en el backlog

Esta herramienta consiste en marcar las historias de usuario que fueron identificadas con riesgos de seguridad. A partir de una o varias historias marcadas se deriva una etiqueta, la cual consiste en una descripción detallada del problema o riesgo de seguridad identificado. Ilustración 6

Esta descripción del problema identificado se puede expresar como una historia de usuario de seguridad, historias de abuso o cualquier otro tipo de documentación que contenga orientación respecto a qué hacer para mitigar el riesgo Ilustración 7

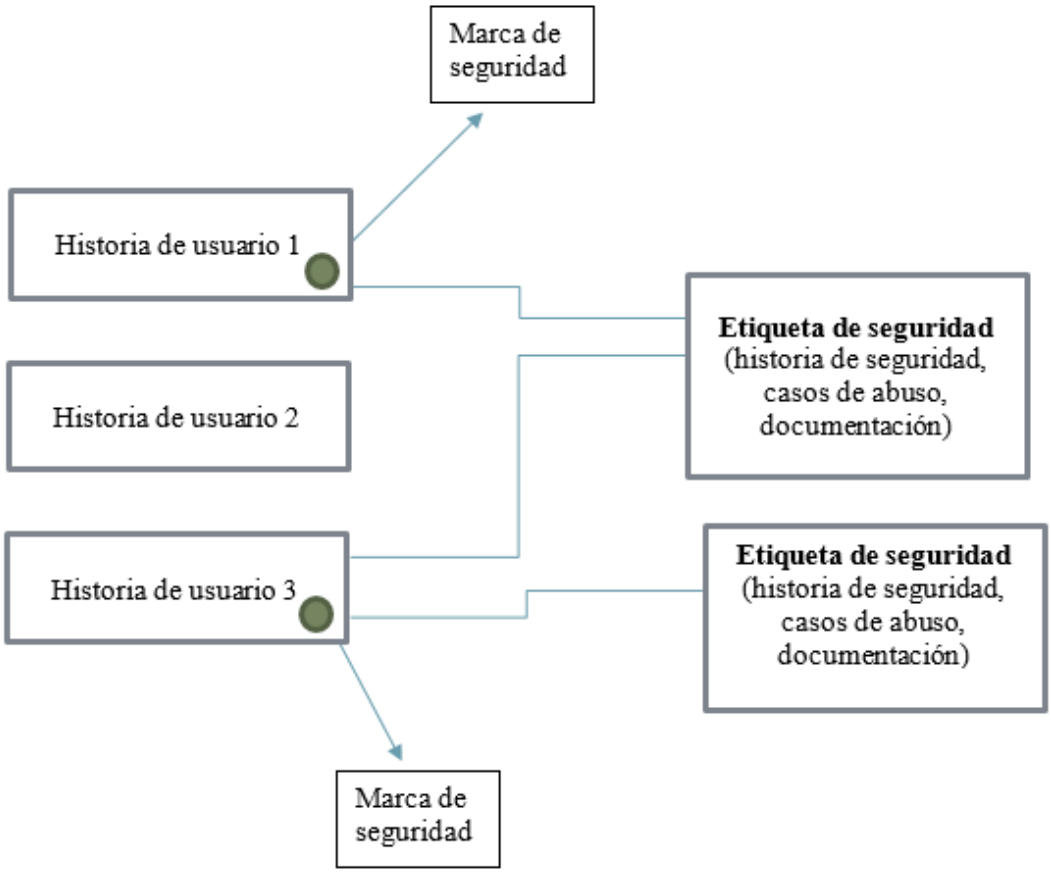


Ilustración 6 Marcas de seguridad

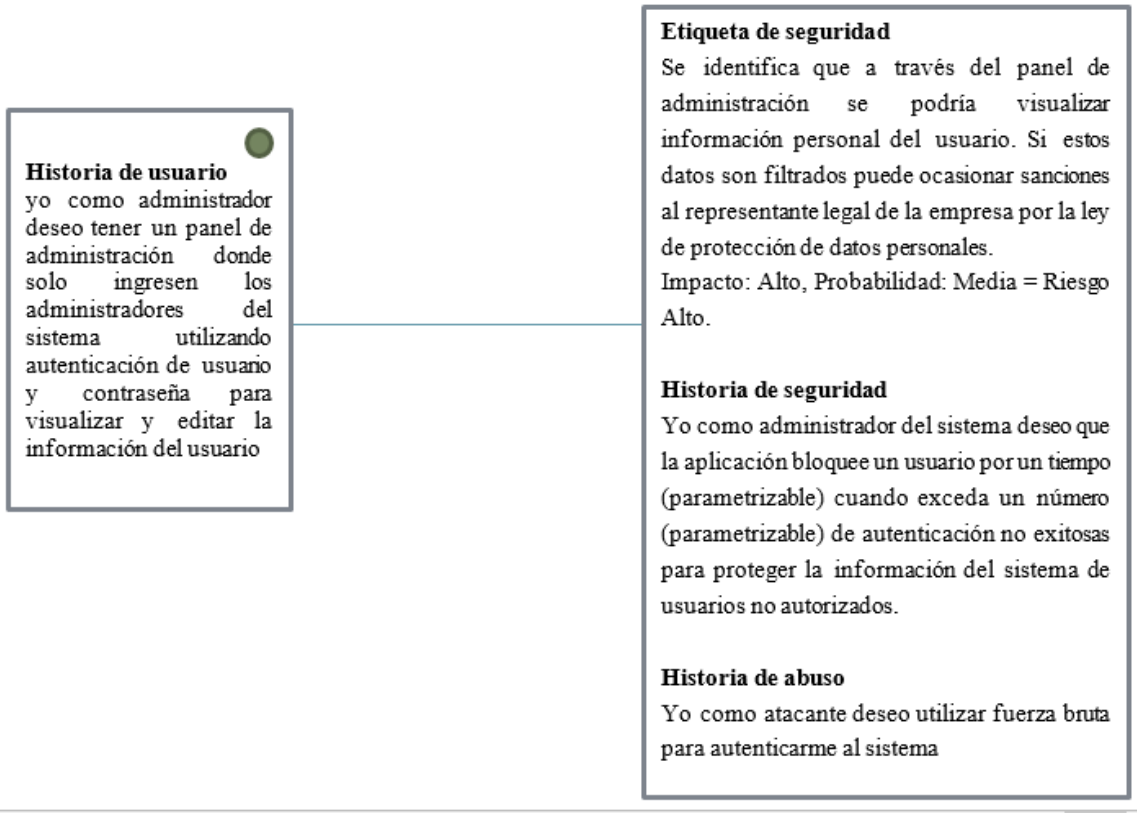


Ilustración 7 Detalle marca de seguridad

7. **A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso.**

El análisis de riesgos mencionado anteriormente es el insumo necesario para realizar esta actividad, una vez se identifican los riesgos en las historias épicas o en las historias de usuario se debe proceder a analizar la forma de mitigar este riesgo. Una posibilidad es especificar una historia de seguridad, complementarla con una o varias historias de abuso o mitigar el riesgo con una decisión de arquitectura. La decisión de qué herramientas utilizar depende del equipo de trabajo y de cómo consideran mitigar el riesgo de una mejor manera.

Ejemplo:

Se identifica que existen un alto riesgo que personas no autorizadas quieran autenticarse al módulo administrativo de un sistema. Para mitigar este riesgo se decide que se va a especificar una historia de usuario de seguridad y que se probará con un caso de abuso:

Historia de usuario de seguridad:
Yo como administrador del sistema deseo que la aplicación bloquee un usuario por un tiempo (parametrizable) cuando exceda un número (parametrizable) de autenticaciones no exitosas para proteger la información del sistema de usuarios no autorizados.

Historia de abuso:
Yo como atacante deseo utilizar fuerza bruta para autenticarme al sistema

Otra posible solución para mitigar este riesgo puede venir dada por una decisión de arquitectura, por ejemplo: *la autenticación de la solución debe utilizar el Directorio Activo de Microsoft, pues este componente ya incluye medidas similares a las expresadas en la historia de usuarios como el bloqueo de usuarios por intentos fallidos.*

Las herramientas anteriormente mencionadas pueden utilizarse en conjunto o individualmente, esta decisión depende del proyecto, y lo que consideren necesario para mitigar el riesgo.

8. **Realizar spikes para actividades de seguridad.**

El objetivo de los spikes propuestos es la de profundizar en la solución de problemas, la mitigación de riesgos y mejorar o realizar diseños de seguridad. Cuando en una historia de usuario se determina que se debe tomar una acción para eliminar el problema de seguridad identificado, esta acción puede implicar la revisión de un diseño, análisis de amenazas, preparar pruebas de aceptación y realizar diseños que requieran investigación y pruebas de concepto.

No es aconsejable en la mayoría de los casos que este tipo de actividades se realicen durante la reunión de planeación del sprint pues estas reuniones suelen ser cortas y si no existe el detalle de cómo implementar alguna acción es importante negociar la realización de un spike de seguridad.

9. **Automatizar pruebas de aceptación**

Como se definió en la práctica (**A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso**) el equipo puede utilizar historias de abuso e historias de seguridad, los criterios de aceptación de estas historias son el parámetro para determinar si el sistema se comporta según lo esperado por el equipo.

Para hacer un correcto seguimiento de estos comportamientos durante todo el proyecto se recomienda modelar pruebas de aceptación en herramientas que automaticen el proceso de

pruebas. Con el fin de ejecutarla múltiples veces durante todos los sprint y verificar que la aplicación se comporta según lo esperado a pesar de los constantemente cambios en el proyecto. Herramientas para este tipo de pruebas pueden encontrarse libremente en internet algunas de ellas son [53], [54]

10. Automatizar pentesting

A diferencia de los fallos y riesgos identificados por medio de historias de seguridad e historias de abuso cuyas pruebas se mencionan en (**Automatizar pruebas de aceptación),** pueden existir fallos de seguridad que no son identificados o mitigados, esto puede deberse a que el equipo las pasó por alto y no las definió o porque durante el desarrollo pudieron presentarse problemas de codificación insegura.

Para reducir el riesgo de que este tipo de fallos se presenten y para corregirlos de manera temprana se recomienda que durante los sprint se realice revisión del código de la aplicación en desarrollo, también se recomienda que todo entregable sea revisado en busca de vulnerabilidades conocidas.

Tradicionalmente estas revisiones y pruebas se pueden ejecutar por parte de una persona conocedora de seguridad. Sin embargo, en entornos donde no hay suficiente personal con este conocimiento es recomendable acudir a herramientas que se pueden configurar para que hagan este tipo de revisiones.

Actualmente existen herramientas con capacidades para automatizar su ejecución y que además brinda la posibilidad de ser integradas a herramientas de integración continua de software. En el mercado las herramientas que se encuentran para estos fines están maduras y brindan la posibilidad de evaluar codificación segura e identificar vulnerabilidades conocidas en el software.

9. VALIDACIÓN

Para validar el modelo presentado en el punto anterior, se procede a realizar en una encuesta con grupos de trabajo entre ellos, varios metodólogos ágiles de Empresas Públicas de Medellín, muchos de ellos encargados de implementar y difundir las prácticas de desarrollo ágil al interior de la empresa.

La metodología para hacer esta validación es utilizar una encuesta con preguntas abiertas, para esto se realiza una presentación explicando en que consiste cada una de las prácticas seleccionadas, con el fin que cada una de las personas exprese su opinión con respecto, a cada una de las prácticas.

La encuesta de validación realizada fue la siguiente (ver anexo 2):

> Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas se seguridad propuestas para su implementación.
>
> A continuación, se lista cada práctica independiente para que haga sus observaciones de considerarlo necesario.
>
> 1. Entrenamiento en seguridad.
> 2. Designar responsable de seguridad.
> 3. Utilizar plantilla de historias de seguridad.
> 4. Análisis de riesgo ligero.
> 5. Responsabilizar al usuario del riesgo no mitigado.
> 6. Utilizar marcas de seguridad.
> 7. Obtener historias de seguridad e historias de abuso.
> 8. Realizar spikes de seguridad.
> 9. Automatizar pruebas de aceptación.
> 10. Automatizar pentesting.

De las encuestas realizadas se sacan las siguientes conclusiones.

1. **Entrenamiento en seguridad.**

   Para está práctica las principales recomendaciones son las siguientes:

   - En EPM se dictan capacitaciones en marcos ágiles, previas al inicio de los proyectos recomiendan utilizar estos espacios para también capacitar sobre seguridad.
     - **Respuesta:** La práctica no riñe con que se aprovechen espacios ya establecidos para la capacitación que se den antes del inicio del proyecto.

La idea es aprovechar la mejor oportunidad para hacer este tipo de actividad.

- Se debe incluir un mayor número de personas como los lideres técnicos y personas del equipo de desarrollo de los proyectos en esta actividad debido a que se había contemplado inicialmente para los roles de Product Owner y Scrum Master.
- Utilizar dos enfoques uno básico y otro más técnico.
    - **Respuesta:** La capacitación en seguridad no se limita solo a los dos roles indicados scrum master y Product Owner, pero el sentido de la práctica es que se asegure como mínimo la capacitación de estos dos roles. Adicionalmente si se contempla ampliar el número de roles involucrados en el entrenamiento de seguridad es pertinente pensar en entrenamientos más técnicos dependiendo del rol.
- Debido a que el equipo de desarrollo de los proyectos está integrado por fábricas de software, recomiendan que se exija este conocimiento en algunos integrantes de los proveedores.
    - **Respuesta:** La exigencia de personal con conocimiento de seguridad recomendada para los proveedores es un elemento que obedece a las realidades contractuales y que es deseable, más a nivel contractual que de esta práctica.

Ajustes y aclaraciones sobre la práctica planteada:

Se adiciona a la redacción de la práctica planteada que el mínimo de roles a los que se les debe exigir este entrenamiento, siendo estos los roles inicialmente planteados, adicionalmente sería deseable que todo el personal del proyecto reciba entrenamiento en seguridad.

Práctica actualizada

El entrenamiento en seguridad debe realizarse antes de la creación y refinamiento del Product Backlog. ***Debe ser recibido como mínimo*** por el Product Owner y Scrum Master asignados al proyecto a quiénes se les ofrece un entendimiento de la importancia de la seguridad y cómo debe ser abordada. ***Si se decide brindar entrenamiento a roles distintos a los nombrados inicialmente, es recomendable analizar la posibilidad de brindar entrenamiento más técnico dependiendo de los roles que tendrán las personas en el proyecto.***

La capacitación es ofrecida por personal de TI con soporte de material de seguridad el cual estará al alcance de todos los equipos de desarrollo. En [50] - [52] se ofrecen recursos útiles para la formación en seguridad.

2. **Designar responsable de seguridad.**

   Las recomendaciones relevantes son las siguientes:

   - Designar un responsable de la seguridad a nivel no solo del proyecto sino también a nivel de la gerencia de TI que podría también ser un equipo de personas.
     - **Respuesta:** El alcance de este proyecto es el de generar una guía con prácticas de seguridad para la metodología ágil utilizada por EPM y comprende prácticas a utilizar dentro del ciclo de vida de Scrum, la recomendación de hacer responsables de seguridad a nivel de la gerencia de TI es un tema que se contempla más a nivel organizacional que de la metodología utilizada dentro del contexto de un proyecto de desarrollo.
   - La persona designada como responsable no necesariamente debe tener conocimiento en seguridad, pero lo puede acompañar un experto en el tema.
     - **Respuesta:** Es aceptable que la persona designada como responsable de seguridad no tenga el conocimiento en el tema desde que exista el acompañamiento de alguien que lo tenga. Esta sugerencia es entendible dado que inicialmente este conocimiento es escaso a nivel organizacional. Pero es deseable que este conocimiento exista al seleccionar a la persona debido a sus funciones.
   - Se sugiere que no se cree un rol aparte, y que sea el Product Owner o el arquitecto de la aplicación (varias personas apuntan a que sea designado el arquitecto de la aplicación).
     - **Respuesta:** La práctica solo limita a Product Owner para ser el responsable de seguridad, por lo tanto, cualquier otro rol incluyendo al arquitecto de la aplicación puede ser el responsable de seguridad.

       Cuando se entiende el Product Owner como: el representante de las personas interesadas en los resultados del proyecto y con la capacidad de tomar decisiones, se descarta que esta persona sea el encargado de seguridad, debido a que en este trabajo se identificó tanto en la literatura como en la investigación realizada, que las personas que actualmente ejercen este rol, no tienen motivación para implementar funcionalidades

de seguridad, no tienen un conocimiento lo suficiente técnico para entender acerca de la ciberseguridad, y generalmente su conocimiento es mas de negocio.

La idea de un responsable de seguridad es que acompañe al Product Owner y lo asesore en temas de seguridad para mitigar la posible falta de conocimiento en el tema.

- Tercerizar este rol debido a la capacidad operativa.
    - **Respuesta:** La recomendación de que sea un tercero cabe dentro de la definición dada en la práctica ya que no exige que se alguien de la empresa, no obstante, si esto sucede se debe garantizar que sea un miembro del equipo durante la duración del proyecto de lo contrario se podría ver esta práctica como la de traer conocimiento de externos la cual fue anteriormente descartada.

Ajustes y aclaraciones sobre la práctica planteada:
Se aclara que no es necesario que la personas que hacen el rol de responsable de seguridad tengan conocimientos en el tema siempre y cuando haya acompañamiento.

Práctica actualizada

El responsable de seguridad debe ser una persona diferente al Product Owner e***s deseable que tenga conocimiento técnico general acerca de seguridad, pero no es necesario siempre y cuando exista acompañamiento de una persona experta en el área de seguridad.***
Sus responsabilidades son verificar que todas las prácticas descritas en la guía sean ejecutadas durante el proyecto y asesorar al resto del equipo en la toma de decisiones con respecto a seguridad. Dentro de sus funciones principales se encuentra:

- En la creación del Product Backlog, asesora al Product Owner en la creación de posibles historias de usuario que tengan que ver con seguridad y ayuda a la definición de criterios de aceptación
- Junto con el Product Owner y demás miembros del equipo debe asesorar en la utilización de la plantilla genérica de historias de usuario de seguridad.
- Una vez definidas las historias de usuario debe revisarlas y documentar de manera resumida los posibles riesgos de seguridad que la historia de usuario debe tener en cuenta.

- Realizar junto con el Product Owner una evaluación de riesgos ligera.
- Define junto con el equipo de desarrollo historias de abuso.
- Asesora al equipo de desarrollo en el análisis y diseño de la seguridad cuando este se realice.
- Establece etiqueta de seguridad en las historias de usuario que deben tener especial atención en materia de seguridad.
- Hace consciente al Product Owner que cuando no se mitiga un riesgo es él Product Owner quien está asumiendo el riesgo.
- Verifica la utilización de pruebas de automatizadas de seguridad.

3. **Historias de usuario de seguridad transversales**

   Las recomendaciones relevantes son las siguientes:

   - Vigilar que estas historias si sean revisadas a conciencia.
   - No limitarse a lo que está consignado en la plantilla de historias de seguridad.
     - **Respuesta:** Vigilar que las plantillas sean utilizadas de manera correcta es parte de las tareas de la persona responsable de seguridad, igualmente el no limitarse a lo que está escrito en estas plantillas también es labor del responsable de seguridad, mediante la ejecución del análisis de riesgo de seguridad ligero.
   - Si el Product Owner no conoce de Tecnologías de información se debe apoyar en el personal de TI para aplicarlas.
   - Se podrían socializar estas historias de seguridad genéricas a los arquitectos de aplicación incluso si no están trabajando de forma ágil.
     - **Respuesta:** La práctica no limita su utilización dentro de un contexto de proyecto ágil, adicionalmente las historias de seguridad genéricas transversales tiene su equivalencia dentro de los lineamientos establecidos de los cuales partió este trabajo.
   - Segmentar plantillas por tipos de proyecto.
     - **Respuesta:** No se ve inconveniente que a medida que se madure la se puedan segmentar las historias de seguridad por tipo de proyecto.

Ajustes y aclaraciones sobre la práctica planteada:

La práctica se mantiene igual a como se definió.

4. **Análisis de riesgo ligero**

   Las recomendaciones relevantes son las siguientes:

   - Deben identificarse y definirse su mitigación continuamente no solo al principio o al final.
   - No está claro el momento en el cual se debe llevar a cabo esta actividad.
     - **Respuesta:** La idea es que este análisis se haga después de haber definido el Product Backlog y durante el tiempo que hay antes de empezar el Sprint planning, "el refinamiento" las etapas de refinamiento del backlog (grooming).
   - Se debe utilizar una matriz de riesgos comunes en los proyectos.
     - **Respuesta**: El utilizar una matriz de riesgos comunes entre proyectos es una idea interesante pero no fue encontrada dentro de la bibliografía de prácticas ágiles de seguridad para mitigar los impedimentos identificados.
   - Ampliar el concepto a Análisis de riesgo ligero, comprende riesgos vulnerabilidades, brechas etc.
     - **Respuesta:** El concepto de análisis de riesgo confundió a personas que diligenciaron la encuesta pues llegaron a pensar que se realizaría un análisis de los riesgos de todo el proyecto, y no solo los riesgos de seguridad por lo que se renombrará la práctica a análisis de riesgos de seguridad ligero.
   - Que se haga por objetivo o por reléase para que la mirada sea amplia y no tan detallada.
     - En la terminología utilizada en seguridad se puede ver el riesgo como el agrupador más amplio pues finalmente términos como vulnerabilidades, brechas, intrusiones, son riesgos del sistema, no es mandatorio que el análisis de riesgos de seguridad se realice exclusivamente sobre las historias de usuario o historias épicas, si hay agrupadores de como los objetivos o el reléase se puede hacer el análisis a este nivel, pero la idea es ser granulares en la revisión, no toda historia de usuario implica un riesgo en las funcionalidades del sistema.

Ajustes y aclaraciones sobre la práctica planteada:

En la actualización cambia el título de la práctica y se aclara a que nivel se aplica esta práctica: historias épicas, historias de usuario, reléase o funcionalidad.

Práctica actualizada

Análisis de riesgos de seguridad ligero

La finalidad de esta práctica es realizar un análisis de riesgo sobre las funcionalidades propuestas para implementación durante el proyecto.

El primer paso es identificar en las funcionalidades si se puede presentar algún riesgo de seguridad para la compañía y el proyecto teniendo en cuenta si este riesgo se presenta por factores internos o externos. ***Esto implica revisar las funcionalidades planteadas a nivel de historias épicas, historias de usuario, reléase o funcionalidad. El objetivo de este análisis es identificar rápidamente cualquier riesgo del producto.***

Después de identificar los riesgos de seguridad, el Product Owner y un experto en seguridad deben hacer una ponderación del riesgo utilizando como herramienta la matriz de riesgo de la organización. La matriz de riesgo contiene dos ejes, uno con el impacto y otro con la probabilidad, el impacto es la medida en la que la materialización de un riesgo puede afectar al proyecto o la organización, la probabilidad es la variable en donde se evalúa que tan posible es que se materialice un riesgo.

Dependiendo del resultado de la evaluación se toma la decisión de mitigar o no el riesgo. Estos resultados deben quedar documentados brevemente especificando el riesgo identificado y su resultado (ver: **Responsabilizar al Product Owner del riesgo no mitigado**).

Mitigar un riesgo puede implicar realizar historias de seguridad detalladas, casos de abuso, tomar decisiones de arquitectura o de diseño. Una vez se ha identificado una historia épica o alguna historia de usuario esta debe ser etiquetada en el backlog mediante documentación del riesgo identificado y la ponderación otorgada (ver: **Etiquetas de marca de seguridad en el backlog**).
Se recomienda que esta práctica de seguridad se realice sobre las historias épicas una vez realizado el Product Backlog y antes de cada planeación de sprint sobre historias de usuarios menos generales derivadas del refinamiento de las historias épicas.

5. **Responsabilizar al usuario del riesgo no mitigado.**

Las recomendaciones relevantes son las siguientes:

- Es una política que debe ser obligatoria no es un asunto para que el Product Owner decida.
    - **Respuesta:** Dentro de la literatura no se encuentra que se haya estudiado o probado el hacer obligatorio el mitigar un riesgo identificado, ocurren situaciones en las que cliente valora que mitigar un riesgo es más caro que asumirlo, por lo tanto, esto no debe ser obligatorio pues el cliente puede decidir no mitigarlo.
- Se debería generar un artefacto donde quede explicito que riesgos se van a mitigar y cuáles no, para que quede clara la responsabilidad que debe asumir el Product Owner
    - **Respuesta:** No había sido especificado el generar un artefacto con los riesgos que se mitigan y con los que no, pero puede ser un complemento para la práctica y para motivar al Product Owner a dar prioridad a la seguridad.
- Concientizar es importante para que el Product Owner entienda la importancia de no implementar los controles a los riesgos identificados.
    - **Respuesta:** La idea de todo el proceso aquí planteado y de la identificación de los posibles riesgos de seguridad es hacer consciente al Product Owner de las consecuencias de la materialización de un riesgo.
- Culturalmente puede no ser aceptado, se propone que el riesgo debe ser compartido, el marco propone que el Product Owner entienda, pero no lo asuma.
    - **Respuesta:** Con respecto a que el riesgo sea compartido, porque puede no ser culturalmente aceptado, se toma la recomendación y se corregirá el texto de la práctica planteada. La motivación de la práctica es que el Product Owner tenga responsabilidad de las decisiones que se están tomando para incentivarlo a mitigar posibles riesgos.
- ¿Cómo se podría responsabilizar algo que se materializó, pero no se identificó al inicio?
    - **Respuesta:** En la literatura analizada no existe nada relacionado a hacer responsable al cliente de un riesgo que no se identifica hasta una vez se materializa.
- Es válido tener riesgo no mitigado visible, pero no debe ser solamente asumido por el Product Owner sino también por el negocio.
    - **Respuesta:** Bajo la definición un de la metodología ágil un Product Owner es el representante del cliente con el poder para tomar decisiones,

en este caso un representante del área de negocio, responsable por los requerimientos que se hacen a las áreas de TI. Sin embargo, la práctica no es especifica en como responsabilizar al Product Owner o al negocio, en este caso se podrían utilizar correos electrónicos o un documento firmado por el jefe responsable (se aclara al lector que EPM es una empresa pública que tiene que ser rigurosa en justificar las acciones que se toman).

Ajustes y aclaraciones sobre la práctica planteada:

Se toman las siguientes recomendaciones: el riesgo es compartido y no solo responsabilidad del Product Owner, se debe generar un artefacto con los riesgos que se mitigan y con los que no, se debe responsabilizar también al negocio.
Se hacen las correcciones a la práctica

Práctica actualizada

Co-Responsabilizar al Product Owner del riesgo no mitigado:

Una vez se identifica un riesgo y sus consecuencias, queda en manos del Product Owner priorizar y decidir si implementar un control que mitigue el riesgo identificado, si el usuario decide no ***mitigar el riesgo se le debe co-responsabilizar del riesgo aceptado, dejando claro las consecuencias de no mitigarlo. Para esto se debe preparar un documento que especifique el riesgo identificado, su valoración en la matriz de riesgo y las razones por las cuales se decide no mitigar el riesgo. Este documento debe firmarse por el jefe de la dependencia cliente y por el responsable de la gerencia de TI.***

6. **Utilizar marcas de seguridad**

Las recomendaciones relevantes son las siguientes:

- No hay recomendaciones a la práctica.

7. Obtener historias de seguridad e historias de abuso

Las recomendaciones relevantes son las siguientes:

- No se necesitan historias de seguridad, deben ser historias de usuario con criterios de aceptación de seguridad, criterios de aceptación no negociables.

- Tener un banco de historias de seguridad, y de abuso que puedan ser consultados por aquellos que apenas empiezan la metodología.
- Corresponde a un criterio de aceptación de la historia de usuario.

Ajustes y aclaraciones sobre la práctica planteada:

Respuesta: Los conceptos son válidos, pero no riñen con la práctica, finalmente las historias de seguridad son historias de usuario orientadas a funcionalidades de seguridad, y según lo planteado se deben hacer cuando sea requerido para mitigar un riesgo. El nombre sirve para identificarlas como de seguridad, pero realmente es una historia de usuario.

Práctica actualizada

No se realizan cambios en la práctica planteada.

8. **Realizar spikes de seguridad**

Las recomendaciones relevantes son las siguientes:

- Promoverlos, usar expertos, referenciarse, dojos
- Utilizar dojos que se están realizando para reforzar conceptos.
- ¿A qué nivel fábrica de desarrollo o EPM?
- Validar temas contractuales de costos.

Ajustes y aclaraciones sobre la práctica planteada:

Respuesta: La idea es hacer estos spikes cuando sea necesario y se desprenda de alguna actividad que se deba realizar durante un sprint. Lo expresado son recomendaciones para mejorar el conocimiento, de seguridad. Esta actividad debe ser realizada por la fábrica de software o EPM, quien quiera que deba intervenir en el spike dependiendo de la actividad a realizar.

Práctica actualizada

No se realizan actualizaciones sobre la práctica.

9. **Automatizar pruebas de aceptación**

Las recomendaciones relevantes son las siguientes:

- Es un tema es muy incipiente en la compañía, se debe pensar en una evolución de la metodología.

Ajustes y aclaraciones sobre la práctica planteada:

Respuesta: La idea es que la práctica propuesta ayude a apalancar y promover la implementación de este tipo de pruebas en la metodología ágil de EPM.

Práctica actualizada

No se realizan actualizaciones sobre la práctica.

10. Automatizar pentesting

Las recomendaciones relevantes son las siguientes:

- Práctica en proceso de aprendizaje
- Hay diferentes herramientas veracode, swopp, zap acunetix

Ajustes y aclaraciones sobre la práctica planteada:

Ninguna.

Práctica actualizada

No se realizan actualizaciones sobre la práctica.

9.1.GUÍA VALIDADA

Guía de prácticas de seguridad para el marco ágil de desarrollo en Empresas Públicas de Medellín.

Introducción

Las prácticas de seguridad aquí expuestas fueron extraídas de la literatura relevante de la

seguridad en metodologías ágiles, con el fin de que al interior de EPM sean aplicadas para reducir la posibilidad de producir software con fallos de seguridad.

La aplicación de estas prácticas no asegura que el software resultante no vaya a tener ningún fallo de seguridad, pero es una gran contribución para reducir esta posibilidad. Las prácticas presentadas a continuación se encuentran distribuidas durante todo el ciclo de vida de la metodología ágil de EPM.

Alcance

Estas prácticas pueden aplicarse en proyectos de desarrollo de software nuevo o en proyectos de mantenimiento, no se descarta que algunas prácticas de las aquí establecidas puedan funcionar en proyectos de diferentes tipos.

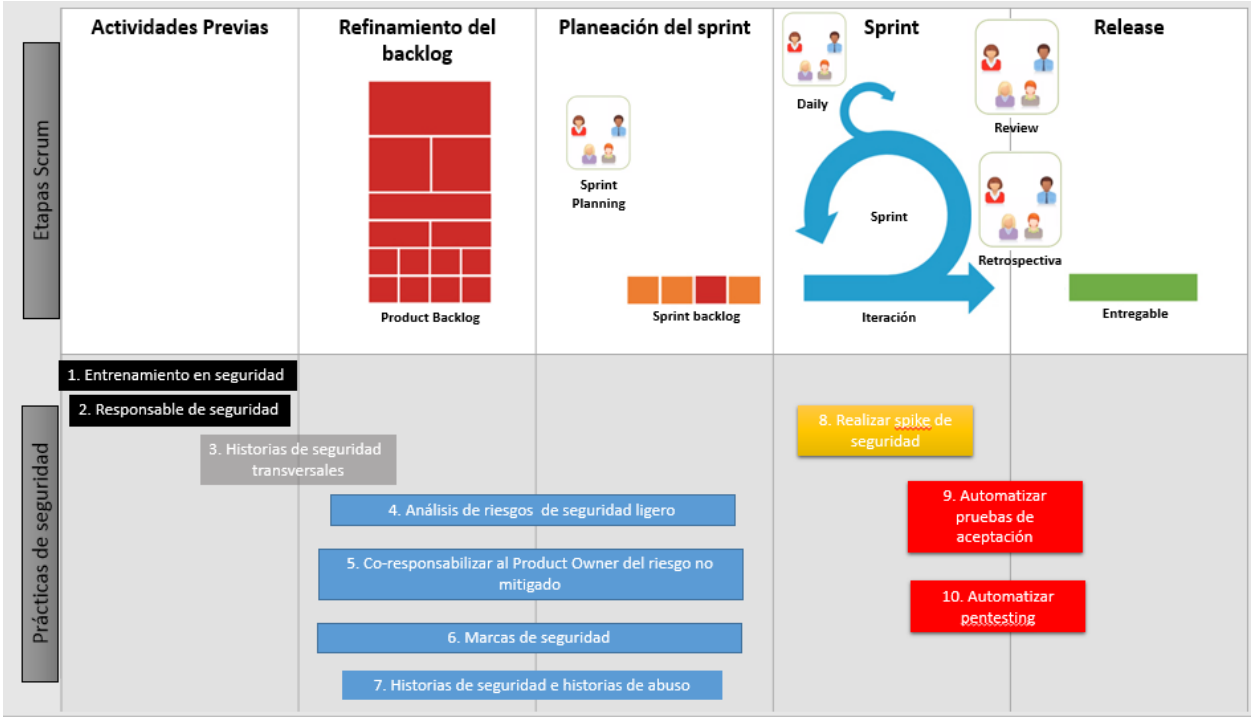


Ilustración 8 Prácticas de seguridad

1. **Entrenamiento en seguridad.**

El entrenamiento en seguridad debe realizarse antes de la creación y refinamiento del Product Backlog. Debe ser recibido como mínimo por el Product Owner y Scrum Master asignados al proyecto a quiénes se les ofrece un entendimiento de la importancia de la seguridad y cómo debe ser abordada. Si se decide brindar entrenamiento a roles distintos a los nombrados inicialmente, es recomendable analizar la posibilidad de brindar entrenamiento más técnico dependiendo de los roles que tendrán las personas en el proyecto.

La capacitación es ofrecida por personal de TI con soporte de material de seguridad el cual estará al alcance de todos los equipos de desarrollo. En [50] - [52] se ofrecen recursos útiles para la formación en seguridad.

2. **Designar responsable de seguridad.**

El responsable de seguridad debe ser una persona diferente al Producto Owner es deseable que tenga conocimiento técnico general acerca de seguridad, pero no necesario siempre y cuando exista acompañamiento de una persona experta en el área de seguridad.
Sus responsabilidades son verificar que todas las prácticas descritas en la guía sean ejecutadas durante el proyecto y asesorar al resto del equipo en la toma de decisiones con respecto a seguridad. Dentro de sus funciones principales se encuentra:

- En la creación del Product Backlog, asesora al Product Owner en la creación de posibles historias de usuario que tengan que ver con seguridad y ayuda a la definición de criterios de aceptación
- Junto con el Product Owner y demás miembros del equipo debe asesorar en la utilización de la plantilla genérica de historias de usuario de seguridad.
- Una vez definidas las historias de usuario debe revisarlas y documentar de manera resumida los posibles riesgos de seguridad que la historia de usuario debe tener en cuenta.
- Realizar junto con el Product Owner una evaluación de riesgos ligera.
- Define junto con el equipo de desarrollo historias de abuso.
- Asesora al equipo de desarrollo en el análisis y diseño de la seguridad cuando este se realice.
- Establece etiqueta de seguridad en las historias de usuario que deben tener especial atención en materia de seguridad.
- Hace consciente al Product Owner que cuando no se mitiga un riesgo es él Product Owner quien está asumiendo el riesgo.

- Verifica la utilización de pruebas de automatizadas de seguridad.

3. Historias de usuario de seguridad transversales

En esta práctica recomienda utilizar historias de usuario de seguridad genéricas que pueden aplicar a cualquier tipo de proyecto. Las historias de seguridad transversales deben ser revisadas durante la creación y actualizaciones del backlog para determinar si alguna de ellas aplica, según los requerimientos del negocio.

Si durante el desarrollo del proyecto se identifica una historia de seguridad transversal que no había sido identificada previamente, esta debe ser agregada para el beneficio de otros proyectos. Utilizar un repositorio con historias de usuario de seguridad trasversales es importante porque asegura que todos los proyectos pueden incluir funcionalidades de seguridad genéricas mínimas sin importar la experiencia o el conocimiento en seguridad de los diferentes integrantes del equipo. De no existir este tipo historias de seguridad, se corre el riesgo que debido a la falta de experiencia o conocimiento el proyecto no cuente con funcionalidades mínimas de seguridad.

4. Análisis de riesgo de seguridad ligero

La finalidad de esta práctica es realizar un análisis de riesgo sobre las funcionalidades propuestas para implementación durante el proyecto.

El primer paso es identificar en las funcionalidades si se puede presentar algún riesgo de seguridad para la compañía y el proyecto teniendo en cuenta si este riesgo se presenta por factores internos o externos. Esto implica revisar las funcionalidades planteadas a nivel de historias épicas, historias de usuario, reléase o funcionalidad. El objetivo de este análisis es identificar rápidamente cualquier riesgo del producto.

Después de identificar los riesgos de seguridad, el Product Owner y un experto en seguridad deben hacer una ponderación del riesgo utilizando como herramienta la matriz de riesgo de la organización. La matriz de riesgo contiene dos ejes, uno con el impacto y otro con la probabilidad, el impacto es la medida en la que la materialización de un riesgo puede afectar al proyecto o la organización, la probabilidad es la variable en donde se evalúa que tan posible es que se materialice un riesgo.

Dependiendo del resultado de la evaluación se toma la decisión de mitigar o no el riesgo. Estos resultados deben quedar documentados brevemente especificando el riesgo identificado y su resultado (ver: **Responsabilizar al Product Owner del riesgo no mitigado**).

Mitigar un riesgo puede implicar realizar historias de seguridad detalladas, casos de abuso, tomar decisiones de arquitectura o de diseño. Una vez se ha identificado una historia épica o alguna historia de usuario esta debe ser etiquetada en el backlog mediante documentación del riesgo identificado y la ponderación otorgada (ver: **Etiquetas de marca de seguridad en el backlog**).

Se recomienda que esta práctica de seguridad se realice sobre las historias épicas una vez realizado el Product Backlog y antes de cada planeación de sprint sobre historias de usuarios menos generales derivadas del refinamiento de las historias épicas.

5. Co-Responsabilizar al Product Owner del riesgo no mitigado

Una vez se identifica un riesgo y sus consecuencias, queda en manos del Product Owner priorizar y decidir si implementar un control que mitigue el riesgo identificado, si el usuario decide no mitigar el riesgo se le debe co-responsabilizar del riesgo aceptado, dejando claro las consecuencias de no mitigarlo. Para esto se debe preparar un documento que especifique el riesgo identificado, su valoración en la matriz de riesgo y las razones por las cuales se decide no mitigar el riesgo. Este documento debe firmarse por el jefe de la dependencia cliente y por el responsable de la gerencia de TI.

6. Etiquetas de marca de seguridad en el backlog

Esta herramienta consiste en marcar las historias de usuario que fueron identificadas con riesgos de seguridad. A partir de una o varias historias marcadas se deriva una etiqueta, la cual consiste en una descripción detallada del problema o riesgo de seguridad identificado. Ilustración 9

Esta descripción del problema identificado se puede expresar como una historia de usuario de seguridad, historias de abuso o cualquier otro tipo de documentación que contenga orientación respecto a qué hacer para mitigar el riesgo. Ilustración 10

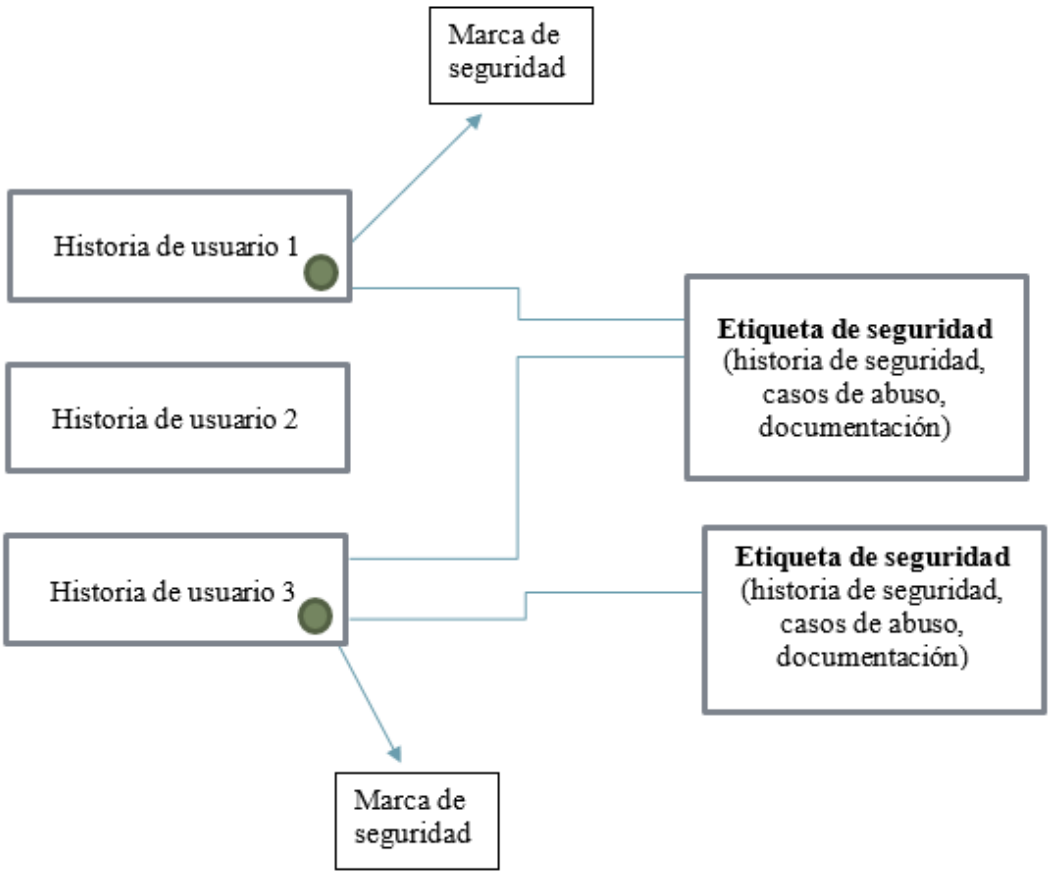


Ilustración 9 Marcas de seguridad

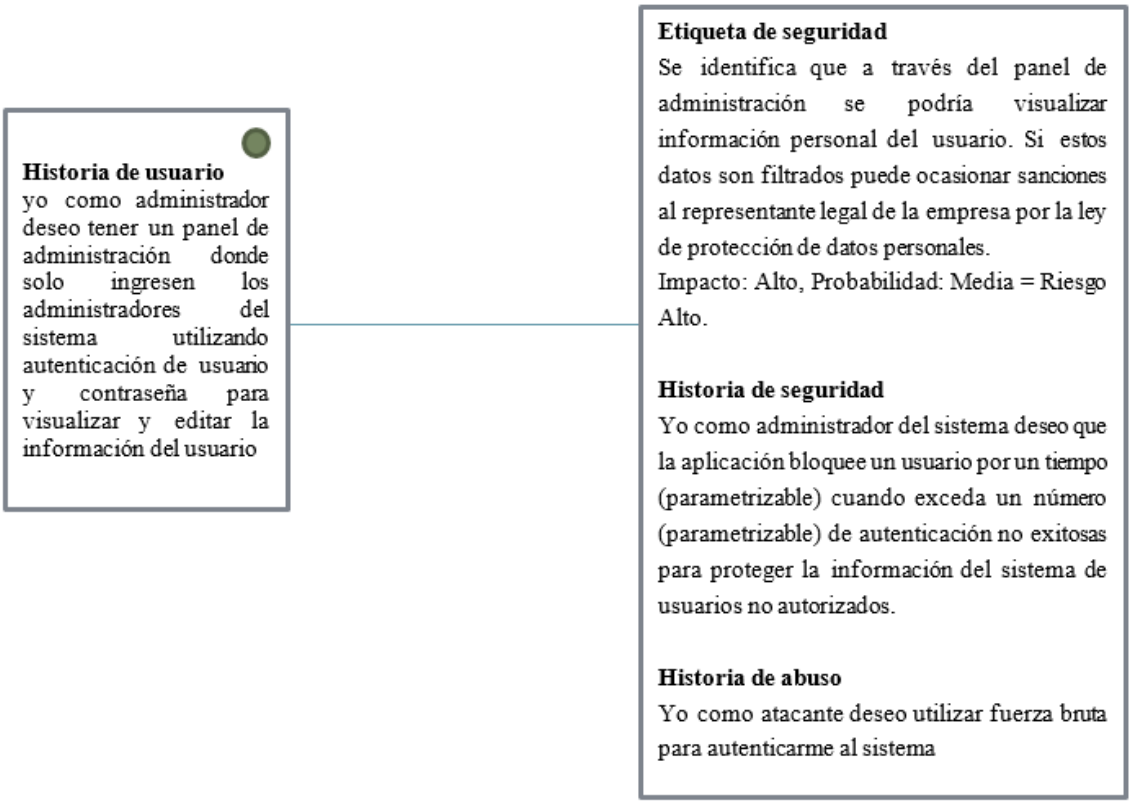


Ilustración 10 Detalle marca de seguridad

7. **A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso.**

El análisis de riesgos mencionado anteriormente es el insumo necesario para realizar esta actividad, una vez se identifican los riesgos en las historias épicas o en las historias de usuario se debe proceder a analizar la forma de mitigar este riesgo. Una posibilidad es especificar una historia de seguridad, complementarla con una o varias historias de abuso ó mitigar el riesgo con una decisión de arquitectura. La decisión de qué herramientas utilizar depende del equipo de trabajo y de cómo consideran mitigar el riesgo de una mejor manera.

Ejemplo:

Se identifica que existen un alto riesgo que personas no autorizadas quieran autenticarse al módulo administrativo de un sistema. Para mitigar este riesgo se decide que se va a especificar una historia de usuario de seguridad y que se probará con un caso de abuso:

Historia de usuario de seguridad:
Yo como administrador del sistema deseo que la aplicación bloquee un usuario por un tiempo (parametrizable) cuando exceda un número (parametrizable) de autenticaciones no exitosas para proteger la información del sistema de usuarios no autorizados.

Historia de abuso:
Yo como atacante deseo utilizar fuerza bruta para autenticarme al sistema

Otra posible solución para mitigar este riesgo puede venir dada por una decisión de arquitectura, por ejemplo: *la autenticación de la solución como la de utilizar el Directorio Activo de Microsoft para manejar la autenticación de la solución, pues este componente ya incluye medidas similares a las expresadas en la historia de usuarios como el bloqueo de usuarios por intentos fallidos.*

Las herramientas anteriormente mencionadas pueden utilizarse en conjunto o individualmente, esta decisión depende del proyecto, y lo que consideren necesario para mitigar el riesgo.

8. **Realizar spikes para actividades de seguridad.**

El objetivo de los spikes propuestos es la de profundizar en la solución de problemas, la mitigación de riesgos y mejorar o realizar diseños de seguridad. Cuando en una historia de usuario se determina que se debe tomar una acción para eliminar el problema de seguridad identificado, esta acción puede implicar la revisión de un diseño, análisis de amenazas, preparar pruebas de aceptación y realizar diseños que requieran investigación y pruebas de concepto.

No es aconsejable en la mayoría de los casos que este tipo de actividades se realicen durante la reunión de planeación del sprint pues estas reuniones suelen ser cortas y si no existe el detalle de cómo implementar alguna acción es importante negociar la realización de un spike de seguridad.

9. **Automatizar pruebas de aceptación**

Como se definió en la práctica (**A partir de análisis de riesgos ligero obtener requerimientos de seguridad, casos de abuso**) el equipo puede utilizar historias de abuso e historias de seguridad, los criterios de aceptación de estas historias son el parámetro para determinar si el sistema se comporta según lo esperado por el equipo.

Para hacer un correcto seguimiento de estos comportamientos durante todo el proyecto se recomienda modelar pruebas de aceptación en herramientas que automaticen el proceso de pruebas. Con el fin de ejecutarla múltiples veces durante todos los sprint y verificar que la aplicación se comporta según lo esperado a pesar de los constantemente cambios en el proyecto.
Herramientas para este tipo de pruebas pueden encontrarse libremente en internet algunas de ellas son [53], [54]

10. Automatizar pentesting

A diferencia de los fallos y riesgos identificados por medio de historias de seguridad e historias de abuso cuyas pruebas se mencionan en (**Automatizar pruebas de aceptación),** pueden existir fallos de seguridad que no son identificados o mitigados, esto puede deberse a que el equipo las paso por alto y no las definió o porque durante el desarrollo pudieron presentarse problemas de codificación insegura.

Para reducir el riesgo de que este tipo de fallos se presenten y para corregirlos de manera temprana se recomienda que durante los sprint se realice revisión del código de la aplicación en desarrollo, también se recomienda que todo entregable sea revisado en busca de vulnerabilidades conocidas.

Tradicionalmente estas revisiones y pruebas se pueden ejecutar por parte de una persona conocedora de seguridad. Sin embargo, en entornos donde no hay suficiente personal con este conocimiento es recomendable acudir a herramientas que se pueden configurar para que hagan este tipo de revisiones.

Actualmente existen herramientas con capacidades para automatizar su ejecución y que además brinda la posibilidad de ser integradas a herramientas de integración continua de software. En este mercado las herramientas que se encuentran están maduras y brindan la posibilidad de evaluar codificación segura e identificar vulnerabilidades conocidas en el software.

10. CONCLUSIONES

- Se identifica que EPM tiene definidas lineamientos de desarrollo de software seguro, sin embargo, estos están orientados a las diferentes fases del desarrollo de software en cascada, definiendo prácticas de seguridad a implementar en las fases de requisitos, diseño, implementación, verificación y mantenimiento. No se encontró ningún anexo o guía para poder aplicar esta serie de recomendaciones cuando se utilizan metodologías de desarrollo que no necesariamente incluyan las fases anteriormente mencionadas. Basado en los resultados de las encuestas se puede decir que esta situación dificulta la aplicación de las prácticas definidas pues las personas de los equipos de desarrollo no necesariamente tienen el conocimiento para adaptar y aplicar las prácticas cuando no se trata de una metodología en cascada.

- Aunque los lineamientos de seguridad para el desarrollo están publicados en la intranet corporativa no es fácil encontrarlos pues los lineamientos no están en un solo sitio, ni en un solo documento. Adicionalmente en algunos documentos es necesario realizar la extracción de la práctica ó lineamiento del contexto del documento. Según lo evidenciado en las encuestas esto contribuye a que las prácticas no se conozcan en su totalidad.

- Se evidencia que tanto en EPM como en sus proveedores de fábrica de software no existe suficiente conocimiento y experiencia en el aseguramiento de software, pues como se evidenció en los impedimentos identificados, la falta de conocimiento o acompañamiento por parte de expertos fue una de las principales causas para no seguir los lineamientos de seguridad. Aunque EPM y sus proveedores son solo una pequeña muestra se puede vislumbrar una falencia en temas de seguridad en la industria local del desarrollo de software.

- Se evidencia la falta de conciencia de seguridad que tienen los product owner de EPM, los cuales están orientados a la entrega de funcionalidad dándole baja prioridad a los temas que tienen que ver con la seguridad informática que en ocasiones deben ser abordados con más prioridad que algunas funcionalidades. Como posibilidad de mejora se debe plantear la posibilidad de adelantar un programa de conciencia en seguridad al interior de EPM para los empleados que tienen roles activos en proyectos con componentes de Tecnologías de información.

- Las prácticas de seguridad establecidas en EPM son percibidas como rígidas y llenas de aprobaciones innecesarias que hace que los procesos de desarrollo no se ejecuten con la agilidad requerida en la actualidad. Esto es un claro indicador que EPM debe iniciar un proceso de adaptación de los procesos de seguridad informática a los nuevos paradigmas de agilidad y flexibilidad en el desarrollo de software.

- Existe suficiente literatura de prácticas de seguridad que han sido utilizadas y en muchos casos probadas dentro de marcos ágiles de desarrollo. Adicionalmente en la misma literatura hay evidencia que en muchos casos se puede presentar incompatibilidad entre las prácticas tradicionales de seguridad y las metodologías ágiles. Por lo tanto, es recomendable utilizar las prácticas de seguridad que ya han sido adaptadas o fueron pensadas para encajar en los marcos ágiles de desarrollo. En el caso de EPM hay evidencias que al utilizar prácticas que no están orientadas a metodologías ágiles ha sido un impedimento para el uso activo.

- Dentro de la literatura del tema no hay consensos acerca de un conjunto de mejores prácticas de seguridad orientadas a metodologías ágiles como si lo había para metodologías de desarrollo en cascada.

- La mayor parte de la literatura de prácticas de seguridad está orientada a proponer alternativas para obtener desde las historias de usuario insumos para asegurar el software que va ser creado, utilizando técnicas como las historias de seguridad, historias de abuso, análisis de riesgos. Por el contrario, no hay mucha literatura en prácticas específicas para el momento de la codificación.

- Debido a que este trabajo se enfoca en la metodología ágil utilizada por EPM, no todas las prácticas expuestas en la literatura consultada aplican directamente para la compañía seleccionada. Adicionalmente muchas de las prácticas encontradas se orientan a objetivos que otras prácticas también intentan cubrir. Se concluyó entonces que cada práctica debe ser evaluada para encontrar un ajuste al contexto de organizacional.

- Se debe evaluar muy bien la conveniencia de escoger una práctica sobre otra pues cada práctica es compatible en mayor o menor medida con los marcos ágiles y con las realidades organizacionales en donde se va aplicar. Uno de los puntos más relevantes para utilizar una determinada práctica tiene que ver con la cultura organizacional.

11. TRABAJOS FUTUROS

Como trabajo futuro se podría realizar la evaluación de la efectividad de las prácticas seleccionadas mediante mediciones después de su implementación, comparando los datos que arrojen los productos desarrollados utilizando estas prácticas y los datos encontrados en la actualidad.
También se podría utilizar esta misma guía de desarrollo en empresas con contextos organizacionales similares y observar sus resultados.

Otra de los trabajos que se podrían realizar a partir de este es uno en donde se utilice la metodología aquí planteada poder plantear soluciones similares a esta para otras compañías que estén viviendo problemas como los que tiene EPM actualmente con respecto a la seguridad en sus desarrollos de software internos.

12. REFERENCIAS

[1] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Ágile software development methods review and analysis," *VTT Publ.*, no. 478, pp. 3–107, 2002.

[2] MinTIC, "Decreto Número 2573 de 2014," pp. 1–9, 2014.

[3] D. Parra, C. P. Hernández, J. P. Zucchet, G. Romero, M. F. Revelo, and C. T. Martínez, "No . 108 DE 2010 ' Visión y Prospectiva del Gobierno en línea en Colombia 2019 ,'" no. 108, 2010.

[4] K. Beznosov and P. Kruchten, "Towards Ágile Security Assurance," pp. 47–54, 2005.

[5] H. Oueslati, M. Masudur Rahman, and L. ben Othmane, "Literature Review of the Challenges of Developing Secure Software Using the Ágile Approach." IEEE, 2015.

[6] V. Asthana, I. Tarandach, N. O'Donoghue, B. Sullivan, and M. Saario, *Práctical Security Stories and Security Tasks for Ágile Development Environments*. SAFECode, 2012.

[7] A. BROSTRÖM, "Integrating Automated Security Testing in the Ágile Development Process," 2015.

[8] C. Pohl and H.-J. Hof, "Secure Scrum : Development of Secure Software with Scrum," 2015.

[9] M. Siponen, R. Baskerville, and T. Kuivalainen, "Integrating Security into Ágile Development Methods," vol. 0, no. C, pp. 1–7, 2005.

[10] Microsoft, "Security Development Lifecycle for Ágile Development," *Development*, pp. 0–20, 2009.

[11] K. Rindell, S. Hyrynsalmi, and V. Leppänen, "Securing Scrum for VAHTI," vol. 1525, pp. 236–250, 2015.

[12] R. E. Maria, L. A. Rodrigues, and N. A. Pinto, "ScrumS - A Model for Safe Ágile Development," *Proc. 7th Int. Conf. Manag. Comput. Collect. Intell. Digit. Ecosyst. - MEDES '15*, pp. 43–47, 2015.

[13] D. Mougouei, N. Fazlida, M. Sani, and M. M. Almasi, "S-Scrum : a Secure Methodology for Ágile Development of Web Services," *World Comput. Sci. Inf. Technol. J.*, vol. 3, no. 1, pp. 15–19, 2013.

[14] U. Rafi, T. Mustafa, N. Iqbal, and W. Zafar, "US-Scrum: A Methodology for Developing Software with Enhanced Correctness, Usability and Security," vol. 6, no. 9, 2015.

[15] NIST, "NVD Common Vulnerability Scoring System Support v2." [Online]. Available: https://nvd.nist.gov/cvss.cfm.

[16] G. Tassey, "The Economic impacts of inadequate infraesrtucture for software testing," 2002.

[17] K. A. Briski, P. Chitale, V. Hamilton, A. Pratt, B. Starr, J. Veroulis, and B. Villard, "Minimizing code defects to improve software quality and lower development costs .," no. October, 2008.

[18] G. Lee and W. Xia, "Toward ágile: An integrated analysis of quantitative and qualitative field data on software development ágility," *MIS Q.*, vol. 34, no. 1, pp. 87–114, 2010.

[19] L. Williams, "A Survey of Ágile Development Methodologies," *Univercsty West Engl.*, pp. 209–227, 2007.

[20] K. Schwaber, *Ágile Project Management with Scrum*. Microsoft Press, 2004.

[21] A. Martín, *Proyectos ágiles con #scrum*, 1st ed. kleer, 2013.

[22] P. Letelier, C. Penadés, J. Canós, and E. Sánchez, "Metodologías Ágiles en el Desarrollo de Software," *Val. Val.*, p. 59, 2009.

[23] K. M. Goertzel, T. Winograd, H. L. McKinley, and P. (Booz A. H. Holley, "Security in the Software Lifecycle: Making Software Development Processes —and the Software Produced by Them—More Secure, Draft Version 1.2 (August 2006), U.S. Department of Homeland Security," no. August, 2006.

[24] R. Baskerville, "Ágile Security for Information Warfare: A Call for Research," *ECIS 2004 Proc.*, p. 10, 2004.

[25] F. Roeser, "Can Software Security be successfully implemented in Ágile software development? A systematic literature review," 2011.

[26] A. Vähä-sipilä, “Software security in ágile product management,” 2011.
[27] V. Ylimannela, “Security activities in scrum control points,” 2011.
[28] H. Keramati, Seyed-Hassan Mirian-Hosseinabadi, and S. H. Mirian-Hosseinabadi, “Integrating software development security activities with ágile methodologies,” 2008.
[29] M. Tomanek and T. Klima, “Penetration testing in ágile software development projects,” vol. 5, no. 1, pp. 1–7, 2015.
[30] R. D. Nagy and I. Mohamed Abdelwahab, “A Security Testing Framework for Scrum based Projects A Security Testing Framework for Scrum based Projects,” no. April, 2016.
[31] S. R. Ghani, Imran; Azham, Zulkarnain; Jeong, I. Ghani, Z. Azham, and S. R. Jeong, “Integrating Software Security into Ágile-Scrum Method,” *KSII Trans. Internet Inf. Syst.*, vol. 8, no. February, pp. 646–663, 2016.
[32] J. Ahola, C. Frühwirth, M. Helenius, L. Kutvonen, J. Myllylahti, T. Nyberg, A. Pietikäinen, P. Pietikäinen, J. Röning, S. Ruohomaa, and C. Särs, *Handbook of the Secure Ágile Software Development Life Cycle*. University of Oulu, 2014.
[33] Veracode Securosis, “Secure Ágile Development,” pp. 1–25, 2014.
[34] S. Wolff, “Scrum goes formal: Ágile methods for safety-critical systems,” *2012 1st Int. Work. Form. Methods Softw. Eng. Rigorous Ágil. Approaches, FormSERA 2012 - Proc.*, pp. 23–29, 2012.
[35] V. Kongsli, “Towards ágile security in web applications,” *21st ACM SIGPLAN Symp. Object-Oriented Program. Syst. Lang. Appl.*, pp. 805–808, 2006.
[36] J. Wäyrynen, M. Bod\′en, and G. Boström, “Security Engineering and eXtreme Programming: An Impossible Marriage?,” *Extrem. Program. Ágil. Methods - XP/Ágile Universe 2004*, pp. 117–128, 2004.
[37] J. Viega and G. McGraw, *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley.
[38] B. Boehm and R. Turner, *Balancing Ágility and Discipline: A Guide for the Perplexed*, vol. 22. 2003.
[39] B. Boehm and R. Turner, “Using risk to balance ágile and plan-driven methods,” *Computer (Long. Beach. Calif).*, vol. 36, no. 6, pp. 57–66, 2003.
[40] A. Singhal, “Selection of Security Activities for Integration with Ágile Methods after Combining their Ágility and Effectiveness,” vol. 6, no. 2, pp. 57–67.
[41] M. T. Siponen, “An Analysis of the Recent IS Security Development Approaches,” *Inf. Secur. Manag.*, pp. 101–124, 1AD.
[42] S. Renatus, C. Teichmann, and J. Eichler, “Method selection and tailoring for ágile threat assessment and mitigation,” in *Proceedings - 10th International Conference on Availability, Reliability and Security, ARES 2015*, 2015, pp. 548–555.
[43] Z. Azham, I. Ghani, and N. Ithnin, “Security backlog in scrum security practices,” *2011 5th Malaysian Conf. Softw. Eng. MySEC 2011*, pp. 414–417, 2011.
[44] G. Boström, J. Wäyrynen, M. Bodén, K. Beznosov, and P. Kruchten, “Extending xp practices to support security requirements engineering,” *Proc. - Int. Conf. Softw. Eng.*, vol. 2006–May, pp. 11–17, 2006.
[45] J. Sutherland and K. Schwaber, “The Scrum Guide,” vol. 1. p. 17, 2016.
[46] T. Satpathy, *A Guide to the SCRUM BODY OF KNOWLEDGE (SBOK™ Guide)*, 2016th ed. SCRUMstudy™, a brand of VMEdu, Inc, 2016.
[47] D. Janzen and H. Saiedian, “Test-driven development concepts, taxonomy, and future direction,” *Computer (Long. Beach. Calif).*, vol. 38, no. 9, pp. 43–50, 2005.
[48] H. Kniberg, *Scrum and XP desde las trincheras*. 2007.
[49] D. Thomas, J. Sutherland, K. Schwaber, R. C. Martin, B. Marick, J. Kern, R. Jeffries, A. Hunt, J. Highsmith, M. Fowler, W. Cunningham, A. Cockburn, A. van Bennekum, and M. Beedle, “Manifiesto ágil.” [Online]. Available: http://ágilemanifesto.org/iso/es/principles.html. [Accessed: 22-May-2017].

[50] Incibe, “Ciberseguridad para micropymes y autónomos.” [Online]. Available: https://www.incibe.es/formacion/ciberseguridad-para-micropymes-y-autonomos. [Accessed: 01-Jun-2017].
[51] ESET, “Academia ESET.” [Online]. Available: https://www.academiaeset.com. [Accessed: 01-Jun-2017].
[52] RTVE, “Mundo hacker.” [Online]. Available: http://www.rtve.es/alacarta/videos/mundo-hacker/. [Accessed: 01-May-2017].
[53] Fitnesse, “Fitnesse.” [Online]. Available: http://fitnesse.org/. [Accessed: 01-Jun-2017].
[54] Seleniumhq, “Selenium.” [Online]. Available: http://www.seleniumhq.org/. [Accessed: 01-Jun-2017].

ANEXOS

Anexo 1. Encuestas de diagnostico

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

60%.

- No hay una persona de seguridad en los equipos ágiles.
- No hay un Security Owner, similar al product Owner quien es el dueño del producto. Para que el sea el dueño de la seguridad.
- No se han escrito historias de usuario de seguridad para implementar en los equipos.
- El porcentaje que se cumple es 100% reactivo

2. Plantilla de requerimientos de seguridad

0%.

- No conocemos la plantilla.
- Nunca ha habido quien nos la presente.

3. Desarrollo y ejecución de casos de abuso

20%.

- Hacemos baterías de pruebas, pero no propuestas por EPM, sino generadas por el equipo de desarrollo de IG.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

10%

- No tenemos una persona de parte del cliente que participe e identifique cuales son esos riesgos de seguridad. para EPM
- No son visibles para IG desde EPM.

5. Codificación segura y revisión de código

90%

- Manejamos buenas practicas de desarrollo
- Manejamos analizadores de código estatico Sonar
- Hacemos TDD
- Hacemos revisiones Pares.

6. Pruebas de intrusión

90%

- Hemos recibido el feedBack de las aplicaciones ~~de seguri~~ MiBatica y Epp.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica. < 0 %
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. No hay un acompañamiento de seguridad.
Se hace seguimiento a modo reactivo.
Todas las personas involucradas no Tienen contexto relacionado con la seguridad.

2. Plantilla de requerimientos de seguridad Se desconoce la plantilla. de seguridad. es importante socializar.

3. Desarrollo y ejecución de casos de abuso Hacer casos de abuso.
Esto implica que en cada sprint no se deben incluir tantas historias funcionales ya que se deben incluir Historias de abuso.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. No se identifican claramente los riesgos de parte de EPM. estos se deben hacer visible. e incluir en backLog.

5. Codificación segura y revisión de código En este sentido se usa Jenkisn, Sonar, revisiones pares. sin embargo no se tienen duras las reglas por parte de EPM

6. Pruebas de intrusión solo lo he visto en APPs Moviles de EPM. pero no lo he visto para otras aplicaciones.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

los requerimientos de seguridad se visualizan en mayor medida al inicio del proyecto en los inception pero luego en el transcurso, estos requerimientos se convierten en solicitudes directas del equipo ágil al equipo se seguridad.

2. Plantilla de requerimientos de seguridad

Estas plantillas no son muy conocidas y al parecer poco socializadas en los equipos ágiles. Se realiza por parte del equipo de seguridad un acompañamiento muy reactivo y en algunas casas coercitivo.

3. Desarrollo y ejecución de casos de abuso

Hasta el momento los casos de abuso no son una práctica estandarizada y se realizan casi que por demanda. Las pruebas que se realizan son muy funcionales y en pocos casos se diseña un caso de abuso.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

5. Codificación segura y revisión de código

6. Pruebas de intrusión

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. 40%

Aunque ~~en~~ los inception se deben invitar los stakeholder, brillaron por su ausencia en cada inception de las apps & modulos.

2. Plantilla de requerimientos de seguridad 60%

Esta plantilla ha ido saliendo ~~En los con~~ en los sprints debido a que la revisión de las personas en EPM a dado con algunos problemas.

3. Desarrollo y ejecución de casos de abuso 0%

SE EJECUTA UN ANALISIS DE ETICAL HACKING EN LOS SERVIDORES DE IG.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. 40%

AUNQUE AL MOMENTO DE ELEGIR LOS DRIVERS DE DESARROLLO DE LA ARQUITECTURA DE MOVILIDAD LA SEGURIDAD SE ENCUENTRA A NIVEL GENERAL

ES NECESARIO CRUZAR INFO CON EPM Y HACER LA PÚBLICA EN EL EQUIPO.

5. Codificación segura y revisión de código 90%.

ESTO ES LO QUE HAY:
- TDD
- BUENAS PRACTICAS DE CODIFICACIÓN
- SONAR Y JENKINS

6. Pruebas de intrusión 20%

- ETICAL HACKING EN I6
- ATAQUES PLANEADOS POR LUIS

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. 25%

- Responsables de seguridad en el equipo ágil
- ~~Cero~~ Retroalimentación de seguridad hacia el equipo

2. Plantilla de requerimientos de seguridad 10%

- No se conoce la plantilla de seguridad en los Proyectos.

3. Desarrollo y ejecución de casos de abuso 0%

- No se realizan casos de abuso.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. 10%

- Se realizan análisis de riesgos básicos
- No se aplican análisis generales de las funcionalidades.

5. Codificación segura y revisión de código 95%

- Revisiones Revises continuamente.

6. Pruebas de intrusión 90%

- Pruebas continuamente. y retroalimentación de la misma

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. 20%

- No hay rol responsable, presente en los proyectos
- El rol existe reactivamente.
- No se hace continuidad a los planes de acción emprendidos
- Las auditorías se realizan improvisadas y de manera los hallazgos llegan como imposiciones que deben cumplirse, pero no habían sido comunicados

2. Plantilla de requerimientos de seguridad 0 %

- No se conocen por los proyectos.
- Deben ser socializadas a todos los proyectos

3. Desarrollo y ejecución de casos de abuso %

- No conocemos los servidores para ejecutar Pruebas de estrés

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. 0%.

- El tema de seguridad no es visible al equipo resulta siempre cuando se hacen auditorías
- Es importante socializar, divulgar la información con el fin de levantar riesgos en esta área

5. Codificación segura y revisión de código 97%.

- Se realizan através de revisiones pares

6. Pruebas de intrusión

- En EPM no conozco que se haga.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. (60%)

- Exige mucha doc como pre-requisito

2. Plantilla de requerimientos de seguridad (20%)

- No se conoce acerca de esta práctica

3. Desarrollo y ejecución de casos de abuso (60%)

- Debe ser un trabajo conjunto

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. (30%)

- No se trabajan de cerca con las personas del negocio.

5. Codificación segura y revisión de código (80%)

- Revision par

6. Pruebas de intrusión (70%)

-Falta de documentación

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica. 90%.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

• Tiempo con usuarios
• Tomar como base un rol de acceso ya existente

1. Requerimientos de seguridad de la información

• Aunque ya se tiene un esquema de seguridad y acceso por roles, muchas veces no se valida exhaustivamente, por tiempo del cliente.
• Clientes predispuestos que ya está la seguridad implantada, sin ponerlo en contexto

2. Plantilla de requerimientos de seguridad

• Desconocimiento de la plantilla y su finalidad
• El cliente cuenta su necesidad funcional, más no su parte técnica, la da por sentado.
• Cliente deja parte técnica a los desarrolladores.

3. Desarrollo y ejecución de casos de abuso

• No hay presupuesto para hacer pruebas exhaustivas
• Se asume que ya se hacen las pruebas integrales van a este nivel de detalle
• Desconocimiento de casos de abuso

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- No tener tiempo o presupuesto para diseñar seguridad.
- falta de conocimiento a fondo por parte del cliente.

5. Codificación segura y revisión de código

- falta de conocimiento de estándares de seguridad mundial
- Revisión continua de código por parte de expertos en seguridad.

6. Pruebas de intrusión

- Tiempo y presupuesto para preparar y ejecutar pruebas
- Indicar al cliente el valor agregado de esta actividad

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

Por lo general dintro del marco ágil si se realizan requerimientos de seguridad. No veo impedimentos.
- 100%

2. Plantilla de requerimientos de seguridad

- 0%
No se conoce la plantilla, solo se diligencian Historias de Usuario, si se requieren muchas aprobaciones no es ágil.
No hay responsables de seguridad en el equipo

3. Desarrollo y ejecución de casos de abuso

- 30%
Algunos criterios de aceptación corresponden a casos de abuso, se podrian considerar este tipo de criterios. Si existiera un responsable de seguridad en el equipo se podrian tener mas visibles.
No es una práctica común en el equipo.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- 20%

Algunas veces se hace el diseño de riesgos, pero no diseño de seguridad porque no se conoce esta práctica, y no existen recursos con el conocimiento para hacerlo.
El responsable del negocio no sabe de seguridad.

5. Codificación segura y revisión de código

- 0%

No se maneja este concepto, ni se conocen herramientas para hacerlo, falta capacitación en el equipo.

6. Pruebas de intrusión

- 0%
- Son muy extensas y costosas.
- No se conocen herramientas
- No hay una persona de seguridad en el equipo.
- Los requerimientos de seguridad no son prioridad para el negocio.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. 70%
   - Exige Capacitación de practicas
   - No existe rol responsable
   - No se da la importancia a la practica

2. Plantilla de requerimientos de seguridad 60%
   - No Existe rol Responsable
   - No se da la importancia
   - No es prioridad para el negocio

3. Desarrollo y ejecución de casos de abuso 30%
   - No Existe Rol Responsable
   - No se le da la importancia
   - Exige capacitación de la practica

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. 60 %

- No existe rol Responsable
- No se da la importaua
- Falta capacitación de la practica

5. Codificación segura y revisión de código 95%

- No se le da la importacia

6. Pruebas de intrusión 95 %

- No se le di la importacia.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

- No hay rol responsable de esta actividad. Sin embargo, no se presentan muchos impedimentos debido a que se definen marcos de seguridad en el equipo.
- Falta capacitación en cuanto a temas de seguridad.

2. Plantilla de requerimientos de seguridad

- Aceptación malas prácticas.
- No se conoce mucho de esta práctica.

3. Desarrollo y ejecución de casos de abuso

- No se ha llegado a realizar un caso de abuso en el equipo, en ágiles pruebas enfocan de aceptación como debe funcionar el aplicativo, pero no de seguridad.
- Por cuestiones de tiempo, Bancolombia tenía visto en la entrega.
- Falta capacitación. Falta los temas de seguridad.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- no hay responsables de seguridad en el equipo y falta concienciación sobre seguridad informática.
- el responsable del negocio no sabe de seguridad informática.

5. Codificación segura y revisión de código

- se necesita tener más información sobre herramientas que nos ayuden con la revisión pre de código implementado

6. Pruebas de intrusión

- tomaría mucho tiempo, puede generar retrasos en entregas ajustadas.
- Requiere muchas dedicaciones para ejecutarlas.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica. 90%.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

- Tiempo con usuarios
- Tomar como base un rol de acceso ya existente

1. Requerimientos de seguridad de la información.

- Aunque ya se tiene un esquema de seguridad y acceso por roles, muchas veces no se valida exhaustivamente, por tiempo del cliente.
- Clientes predisponen que ya está la seguridad implantada, sin ponerlo en contexto

2. Plantilla de requerimientos de seguridad

- Desconocimiento de la plantilla y su finalidad
- El cliente cuenta su necesidad funcional, más no su parte técnica, la da por sentado.
- Cliente deja parte técnica a los desarrolladores.

3. Desarrollo y ejecución de casos de abuso

- No hay presupuesto para hacer pruebas exhaustivas
- Se asume que ya se hacen las pruebas integrales van a este nivel de detalle
- Desconocimiento de casos de abuso

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- No tener tiempo o presupuesto para diseñar seguridad.
- falta de conocimiento, a fondo por parte del cliente.

5. Codificación segura y revisión de código

- falta de conocimiento de estándares de seguridad
- Revisión continua de código por parte de expertos en seguridad.

6. Pruebas de intrusión

- Tiempo y presupuesto para preparar y ejecutar pruebas
- Indicar al cliente el valor agregado de esta actividad

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

- No hay rol responsable de este aspecto. Sin embargo se presentan muchos impedimentos debido a que definan marco de seguridad en el equipo.
- Falta capacitación en cuanto a temas de seguridad.

2. Plantilla de requerimientos de seguridad

- Necesitaría incluir pruebas.
- No se conoce ningún caso práctico.

3. Desarrollo y ejecución de casos de abuso

- No se ha llegado a realizar un caso de abuso en el equipo, en algún momento debemos capacitar como debe ser el aplicativo, pero no de seguridad.
- Por cuestiones de tiempo, podríamos tener incluso en la pruebas.
- Falta capacitación sobre los temas de seguridad.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- No hay responsable de seguridad en el equipo y falta capacitación sobre seguridad informática.
- El responsable del negocio no sabe de seguridad informática.

5. Codificación segura y revisión de código

- Se necesita tener más información sobre herramientas que nos ayuden con la revisión de código implementado

6. Pruebas de intrusión

- Tomaría mucho tiempo, puede generar reversos para entregas oportunas
- Requiere muchas autorizaciones para ejecutarlas.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. 30 %
   - No hay lineamientos claros en seguridad para desarrollos en B.I
   - Los roles para acceder a la información no se manejan por aplicación sino por fuentes de información.
   - Requerimientos de seguridad no son prioridad para el negocio.

2. Plantilla de requerimientos de seguridad 0 %
   - No existe una plantilla definido.
   - Si se crea, el cliente debe conocer temas de T.I y seguridad.

3. Desarrollo y ejecución de casos de abuso 0 %.
   - No ~~ha~~ se tiene conocimientos en el tema.
   - Se requiere mucha documentación.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. 0%.
   - Se requiere capacitación en el tema.
   - No es prioritario para el negocio.

5. Codificación segura y revisión de código 30%.
   - Se requiere capacitación en el tema.
   - Se requieren herramientas que apoyen las pruebas y la automatización.

6. Pruebas de intrusión 50%.
   - Requiere mucho tiempo y dificultaría la ejecución de sprints cortos en metodologías ágiles.
   - No hay un responsable específico para temas de seguridad en el equipo de trabajo.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

80%
los permisos solo los dan con aprovación del jefe los permisos no se dan tan facilmente.

2. Plantilla de requerimientos de seguridad

90%
La plantilla es por catalogo. Eso es bastante seguro.

3. Desarrollo y ejecución de casos de abuso

30%

- Faltan pruebas de seguridad en los aplicativos.
- Falta un Rol que se encargue de estas Pruebas

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- No se hace para mucho caso de inyección de base de datos

5. Codificación segura y revisión de código

- No se revisa el codigo basado en seguridad
- No se que tanto aplique en BI, ya que la visualizacion es por herramientas microsoft que de por si tienen la seguridad enebida.
- En base de datos falta aplicar mas codificacion segura.

6. Pruebas de intrusión

- falta estas pruebas, No se realizan en BI

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.
   - 10%
   - No hay rol responsable de esa actividad.
   - Exige mucha documentación.
   - No hay responsables de seguridad en el equipo.

2. Plantilla de requerimientos de seguridad
   - 10%
   - Darla a conocer
   - El responsable del negocio no sabe de seguridad informática

3. Desarrollo y ejecución de casos de abuso
   - 10%
   - No hay responsables de seguridad en el equipo.
   - Tomaría mucho tiempo para hacer esta actividad

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.
   - 10%
   - Se exige mucha documentación
   - Tomaría mucho tiempo hacer esta actividad

5. Codificación segura y revisión de código
   - 80%
   - Tomaría mucho tiempo para hacer esta actividad.

6. Pruebas de intrusión
   - 10%
   - Los requerimientos de seguridad no son prioridad para el negocio.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

~~40%~~ → ~~5%~~ 10%.

→ No hay conocimiento suficientes en seguridad

→ Muchos pasos intermedios para aprobación de Requerimientos

2. Plantilla de requerimientos de seguridad

0%.

→ No se conoce la plantilla

3. Desarrollo y ejecución de casos de abuso

0%.

→ No hay personal con conocimiento en pruebas de Seguridad.

→ No hay he

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

0%.

→ No hay personas con Conocimientos en Seguridad.

→ No se realizan estas actividades

5. Codificación segura y revisión de código

70%.

→ No se realizan buenas revisiones pares al código debido por que el código sea seguro

6. Pruebas de intrusión

~~60%~~. 60%.

→ Poca documentación de la seguridad.

→ Pocos conocimientos en pruebas de seguridad

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. 10%
   - No hay rol responsable
   - Dentro de las prioridades del negocio esta la funcionalidad y es poco visible la seguridad.

2. Plantilla de requerimientos de seguridad 10%.
   - Dar a conocer la plantilla de seguridad a todos los equipos.
   -

3. Desarrollo y ejecución de casos de abuso 0%.
   - Se requiere de capacitación
   - las personas con este perfil para ejecutar dichas tareas son escasas.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. 10%

No siempre hay responsables de seguridad en el equipo

5. Codificación segura y revisión de código 80%.

- Que por la presión de entrega no se implementen buenas practicas.
- Que EPM solicite que se entregue sin pruebas y se hagan responsables de esta etapa.
- Que las pruebas se hagan en producción y no en ambiente dllo qe.

6. Pruebas de intrusión 20%.

Que se incluyan en el plan de pruebas y sean aceptadas por el EPM

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

- No hay una correcta definición por parte de los marcos ágiles.
- No hay rol responsable de esa actividad

2. Plantilla de requerimientos de seguridad

- No hay rol responsable de esa actividad.

3. Desarrollo y ejecución de casos de abuso

- No se conoce acerca de esta práctica.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- Exige mucha documentación como prerrequisito.
- No hay rol responsable de esa actividad.

5. Codificación segura y revisión de código

- No hay rol responsable de esa actividad.
- Tomaría mucho tiempo hacer esta actividad y no se haría una entrega oportuna.

6. Pruebas de intrusión

- No hay rol responsable de esa actividad.
- Tomaría mucho tiempo hacer esta actividad y no se haría una entrega oportuna

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

Por lo general dentro del marco ágil si se realizan requerimientos de seguridad. No veo impedimentos.
- 100%

2. Plantilla de requerimientos de seguridad

- 0%
No se conoce la plantilla, solo se diligencian Historias de Usuario, si se requieren muchas aprobaciones no es ágil.
No hay responsable de seguridad en el equipo

3. Desarrollo y ejecución de casos de abuso

- 30%
Algunos criterios de aceptación corresponden a casos de abuso, se podrían considerar este tipo de criterios. Si existiera un responsable de seguridad en el equipo se podrían tener mas visibles.
No es una práctica común en el equipo.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- 20%

Algunas veces se hace el diseño de riesgos, pero no diseño de seguridad porque no se conoce esta práctica, y no existen recursos con el conocimiento para hacerlo.
El responsable del negocio no sabe de seguridad.

5. Codificación segura y revisión de código

- 0%

No se maneja este concepto, ni se conocen herramientas para hacerlo, falta capacitación en el equipo.

6. Pruebas de intrusión

- 0%
- Son muy extensas y costosas.
- No se conocen herramientas
- No hay una persona de seguridad en el equipo.
- Los requerimientos de seguridad no son prioridad para el negocio.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

Actualmente los usuarios líderes indican que roles deben acceder a las funcionalidades

El impedimento puede ser que muchas veces los usuarios desconocen el proceso integral y muchas veces quedan grupos de usuarios que no se tomaron en cuenta por fuera de la funcionalidad

2. Plantilla de requerimientos de seguridad

Pensaría que es a nivel del aplicativo CRM que se maneje, pero considero que Si puede ser totalmente posible.
Se debe hacer una socialización a los equipos ya que es totalmente "O" poco conocidos para muchos.

3. Desarrollo y ejecución de casos de abuso

*. No se conoce mucho acerca de esta práctica

*. No se ha incluido dentro de los criterios de aceptación en las Historia de usuarios.

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- No hay rol responsable de esa actividad
- No hay responsables de seguridad en el equipo
- El responsable del negocio no sabe de seguridad informática

5. Codificación segura y revisión de código

- Estamos tratando de empezar con estas prácticas →
- Mas capacitación y seguimiento a los desarrolladores.

→ Implica mas tiempo y costo el negocio/usuario es quien paga los requerimientos, no lo pagan las funcionalidades dicen que somos muy caros

6. Pruebas de intrusión

*. Costo quien lo pagaría?
*. Quien las llevaría a cabo
* Los requerimientos de seguridad no son prioridad para el negocio.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información.

20%

- El responsable del negocio no sabe de seguridad informática
- No se conoce acerca de esta practica.
- Tomaría mucho tiempo aplicarlo

2. Plantilla de requerimientos de seguridad

30%

- No hay Rol responsable de esa actividad
- No hay responsable de Seguridad en el equipo

3. Desarrollo y ejecución de casos de abuso

0%

- No hay responsables de Seguridad en el equipo
- No se conoce acerca de esta practica

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

0%

- No hay responsable de seguridad
- Las personas que lo podrían hacer no tienen el conocimiento

5. Codificación segura y revisión de código

80%

- Tomaría mucho tiempo hacer esta actividad y no se harían entregas oportunas.
- Se requiere herramientas q. la empresa no posee y que disminuir el tiempo.

6. Pruebas de intrusión

0%

- No hay Responsable.
- Exige capacitación
- No se conoce acerca de esta práctica

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.
- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

1. Requerimientos de seguridad de la información. 70%
   - Exige Capacitación de practicas
   - No existe rol responsable
   - No se da la importancia a la practica

2. Plantilla de requerimientos de seguridad 60%
   - No Existe rol Responsable
   - No se da la importancia
   - No es prioridad para el negocio

3. Desarrollo y ejecución de casos de abuso 30%
   - No Existe Rol Responsable
   - No se le da la importancia
   - Exige capacitación de la practica

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad. 60 %

- No existe rol Responsible
- No se da la importancia
- Falta capacitación de la practica

5. Codificación segura y revisión de código 95%

- No se le da la importancia

6. Pruebas de intrusión 95 %

- No se le da la importancia.

- En qué porcentaje cree usted que en EPM bajo su marco ágil de desarrollo, se sigue esta práctica.

- Si su respuesta no es un 100% cuales cree usted que son los impedimentos para que no se cumpla con la práctica bajo el marco ágil de desarrollo de EPM.

~~No hay rol responsable de esa actividad~~
~~Necesitaria muchas aprobaciones~~

1. Requerimientos de seguridad de la información.

- 10%
- No hay un rol responsable de esa actividad.
- Necesitaria muchas aprobaciones

2. Plantilla de requerimientos de seguridad

- 0%
- No se conoce de la existencia de la plantilla
- No se conoce acerca de esta practica

3. Desarrollo y ejecución de casos de abuso

- 0%
- Exige mucha documentacion como pie requisito
- Tomaria mucho tiempo hacer esta actividad y no se haria una entrega opoituna

4. Análisis de riesgos de las nuevas funcionalidades y diseño seguridad.

- 0%
- No se tiene en cuenta en el proceso de analisis del ciclo de desarrollo

5. Codificación segura y revisión de código

- 10%
- Las revisiones de seguridad actualmente no se tienen presentes en la validacion del codigo

6. Pruebas de intrusión

- 0%
- Requiere tiempo extra en pruebas y conocimiento de como realizarlas

Anexo 2. Encuestas de validación

Nombre _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

- ME PARECE IMPORTANTE INCLUIR EN ESTE ENTRENAMIENTO A LOS INTEGRANTES DEL EQUIPO DEVOPS.
- EL ARQUITECTO DE APLICACIÓN (ACTUAL LÍDER TÉCNICO) DEBERÍA PARTICIPAR EN ESTOS ENTRENAMIENTOS.

2. Designar responsable de seguridad

PIENSO QUE EL RESPONSABLE PODRÍA SER, EN EL MEDIANO PLAZO, EL MISMO ARQUITECTO DE APLICACIÓN. ESTO PUEDE LOGRARSE CON EL ENTRENAMIENTO CONSTANTE. ESTO PUEDE LOGRAR LA RECUPERACIÓN DE LAS CAPACIDADES TÉCNICAS QUE SE HAN PERDIDO EN LAS UNIDADES DE SOLUCIONES.

3. Utilizar plantilla historias de seguridad

- ESTAS PLANTILLAS SE PUEDEN SOCIALIZAR A TODOS LOS ARQUITECTOS DE APLICACIÓN, INCLUSO PARA SÍ NO ESTÁN TRABAJANDO DE FORMA ÁGIL.
- ES IMPORTANTE RECORDAR LA IMPORTANCIA DE VALIDAR LA SEGURIDAD DE FORMA TEMPRANA Y NO AL LLEGAR AL COMITÉ. (VALIDAR EL PROCESO ACTUAL).

4. Análisis de riesgo ligero

5. Responsabilizar al usuario del riesgo residual

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.

- EL ARQUITECTO DE APLICACIÓN DEBERÍA SER EL MISMO RESPONSABLE DE SEGURIDAD (Repito!)

8. Realizar spikes de seguridad

9. Automatizar pruebas de aceptación

10. Automatizar pentesting

Nombre: _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad
   -

2. Designar responsable de seguridad
   - Podrían automatizar pruebas de seguridad - equipo DevOps
   - Intentar el equipo multi disciplinario para tener un equipo integral de Respaldo (Microsoft)

3. Utilizar plantilla historias de seguridad

4. Análisis de riesgo ligero
   desde el inception → para mitigar desde el principio

5. Responsabilizar al usuario del riesgo residual
   - No olvidar que el PO actúa es de TI

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.

8. Realizar spikes de seguridad
validar temas contractuales de costo

9. Automatizar pruebas de aceptación
importante – trabajar

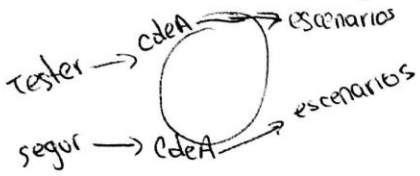
10. Automatizar pentesting
– Vera code
Suite

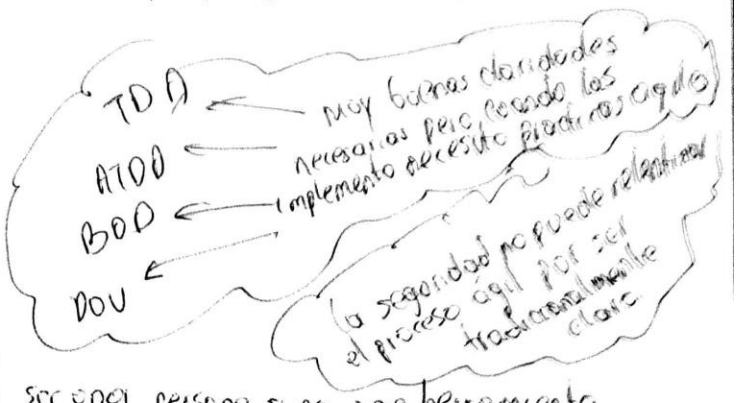
Nombre: _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

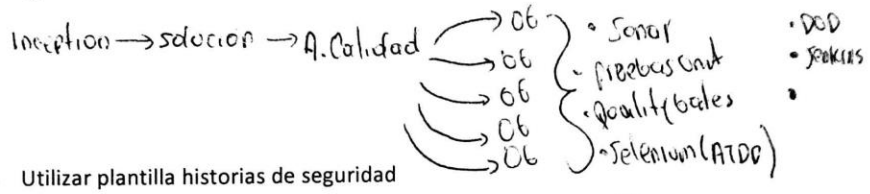




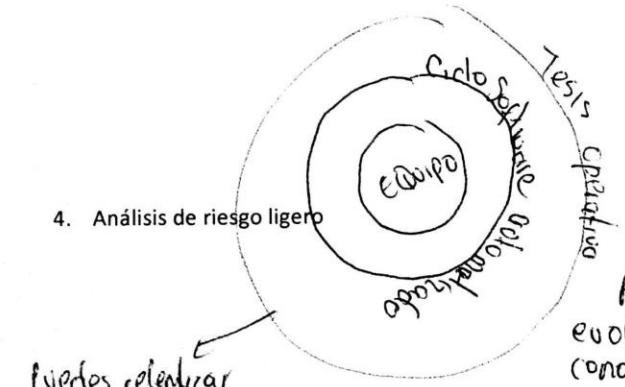
2. Designar responsable de seguridad

¿El responsable no debería ser una persona si no una herramienta

3. Utilizar plantilla historias de seguridad



4. Análisis de riesgo ligero



5. Responsabilizar al usuario del riesgo residual

¿Historias de Abuso?

Una historia debe ser con calidad y la seguridad esta incluida

DOAceptacion

ARQ

Seguridad agilismo

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.

8. Realizar spikes de seguridad

9. Automatizar pruebas de aceptación

10. Automatizar pentesting

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

- El P.O es de negocio, se trata de que no sea técnico
- El SM es del proveedor en el nuevo esquema
- Sugiero que el que debe tener el entrenamiento debe ser el Arquitecto de Aplicaciones o Servicio.

2. Designar responsable de seguridad

Si debe existir este responsable no solamente por equipo sino alguien responsable a nivel de Gerencia de TI. (Puede ser un equipo).

3. Utilizar plantilla historias de seguridad

Totalmente de acuerdo deberían existir plantillas de seguridad

4. Análisis de riesgo ligero

Válido el análisis de Riesgo en las H.U. pero no solo que sean lógicas.

5. Responsabilizar al usuario del riesgo residual

Es muy válido tener el riesgo residual visible porque hoy no se hace, sin embargo no debe ser asignando ~~ad~~ solamente al P.O. sino al negocio también.

6. Utilizar Marcas de seguridad

OK.

7. Obtener historias de seguridad e historias de abuso.

no sé si con el cumplimiento de políticas se requieren historias adicionales.

8. Realizar spikes de seguridad

OK.

9. Automatizar pruebas de aceptación

OK.

10. Automatizar pentesting

OK.

Comentario Final:

- Hay términos que desconozco por lo que para mí todo lo que mejore la seguridad lo veo con buenos ojos, sin contar los costos, como $, Acquidecidia, escalabilidad, etc.
- Sugiero una imágen con los beneficios y los problemas que puede presentarse con la implementación de uno de los ítems o los problemas de no implementarse.

Nombre: [redacted] IDDUFT

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

¿Es posible solicitar alguna constancia de parte del proveedor que el personal que prestará el servicio haya tenido un entrenamiento al respecto? ¿hay algún curso recomendado?

2. Designar responsable de seguridad

3. Utilizar plantilla historias de seguridad

4. Análisis de riesgo ligero

¿Qué momentos pueden ser los más indicados para hacer este análisis? ¿hace parte de una actividad de inception o es parte de una ceremonia de planning?

¿Debe llegar solo a identificar escenarios de riesgo o también el nivel de controles actuales y la documentación de controles propuestos?

5. Responsabilizar al usuario del riesgo residual

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.
   Teniendo en cuenta que las historias de usuario son por definición muy independientes entre sí, ¿a esta corresponde realmente a una historia de usuario o a un criterio de aceptación de la historia con la que está relacionado?

8. Realizar spikes de seguridad

9. Automatizar pruebas de aceptación

10. Automatizar pentesting

Nombre: [redacted]

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

- Product owner, Scrum Master
- No aplica para Scrum Master. Se debe enfocar así:
  - PO: generar conciencia
  - Arquitectos/analistas EPM: entrenamiento técnico en seguridad.
- Se debería "exigir" conocimiento en seguridad en el eq. desarrollo de la fábrica

2. Designar responsable de seguridad

- Inicialmente debe estar acompañado por experto.

3. Utilizar plantilla historias de seguridad

- Muy buena.

4. Análisis de riesgo ligero

5. Responsabilizar al usuario del riesgo residual

- Sobre riesgos identificados antes de; ¿Cómo responsabilizar lo que se materializó, pero no se identificó al inicio?

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.

8. Realizar spikes de seguridad

¿A qué nivel? ¿Fabrica de desarrollo? ¿EPM?

9. Automatizar pruebas de aceptación

✓

10. Automatizar pentesting

- Zap
- Acunetix : licencias

Nombre: _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

Es necesaria. No se debe hacer con el proveedor.
Creo que no debe ser "básico", sugiero que sea Intermedio.
Debe servirnos para generar mayores conversaciones y Criterios de Aceptación en las H.U.

2. Designar responsable de seguridad

Debe hacer preguntas poderosas que hagan visible los aspectos de Seguridad

NO sugiero tener un rol aparte. Debe ser del ADN del PO o Arquitecto de Aplicación. Deberían tener lineamiento (ejemplos autentizados) para entrega a las Fábricas de sw.
Duda: Se podría tercerizar? No lo descartaría.

Muy Capacidad operativa?

3. Utilizar plantilla historias de seguridad

Me parecen valiosas y útiles. Creo que Si el P.O. es de TI, aplican como lineamiento, no como H.U independiente.
Si el P.O. es de negocio, les puede conocer, pero se apoya con el personal TI para entenderlos y aplicarlos.

4. Análisis de riesgo ligero

Si, útil. Sugiero ampliar el concepto a: Diagnóstico ligero.
Min: Riesgos, Vulnerabilidades, Brechas, etc. Para que sea un concepto amplio y de valor a las aplicaciones.
Se requiere por H.U? propongo que sea por Objetivo ó Release para que la minuta sea amplia y no tan detallada.

5. Responsabilizar al usuario del riesgo residual

Culturalmente NO
Propongo que el riesgo debe ser compartido, así como el marco propone responsabilidad de equipo.
Que el P.O. lo entienda, no que lo asuma

Nota: Sale el SM de EPM.

6. Utilizar Marcas de seguridad

Buena idea. Hacer visibles las posibles vulnerabilidades.
Hacer énfasis en estas HU. Cards rojas.
Tener muy buenas pruebas para estas.

7. Obtener historias de seguridad e historias de abuso.

Como en el (3) no creo que se requieran HU. Stks de seg. deberían estar incluidos en las HU en sus Criterios de Aceptación
Podrían ser escenarios de pruebas.

8. Realizar spikes de seguridad

Útil y necesario.
Tengo dudas con la cultura, porque no somos muy dados a investigar. Hay que promoverlo.
Usar expertos, referenciamientos, Dojos, entre otros.

9. Automatizar pruebas de aceptación

Clave y creo que el punto más impactante (positivo) de lo que hoy estamos proponiendo.
Va muy de la mano con el Marco y aquella con costos y tiempos.

10. Automatizar pentesting

→ +1. Hay prácticas de seguridad en proceso de aprendizaje.

En general muy bien. Creo que cabe completamente en nuestro contexto.
Siguiendo la metodología sugiero iniciar de lo poco y simple a lo más complejo. Por ejemplo: Arrancaría con (8) y (6).
Me preocupa la capacidad operativa para hacer los apoyos: Hay suficiente gente lista en seguridad? Cuántos podemos tener? Se empezaría con lo más crítico?

Nombre _____ _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

Estoy de acuerdo que se haga el entrenamiento a todo el equipo del proyecto (Scrum, product owner y equipo de desarrollo) se haga una inducción a todos para que conozcan la importancia de la seguridad y se puede hacer en dos niveles básico y más técnico

2. Designar responsable de seguridad

Es importante designar una persona para que vele por que se este cumpliendo y dicha persona estar continuamente estudiando porque hay nuevas vulnerabilidades

3. Utilizar plantilla historias de seguridad

Un modelo de referencia es importante pero no debe limitarse a esto, sino siempre analizar que más puede aplicarse y que no este allí

4. Análisis de riesgo ligero

Deben identificarse y definir cómo mitigarlos. Esto debe hacerse continuamente, no al principio no al fin, durante todo el proyecto

5. Responsabilizar al usuario del riesgo residual

Concientizar es importante y que sienta la importancia de no hacerse

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.

8. Realizar spikes de seguridad

9. Automatizar pruebas de aceptación

10. Automatizar pentesting

General:
~~Es~~ Sería bueno crear un grupo de seguridad donde se definan casos específicos para hacer pruebas. tipo dojo allí se aprenden y practican cosas nuevas

Nombre: _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

Buena propuesta, se debe incluir en el entrenam/m a los analistas responsables de las aplicaciones, es necesaria la capacitación y actualización constante en temas de seguridad informatica.

2. Designar responsable de seguridad

3. Utilizar plantilla historias de seguridad

4. Análisis de riesgo ligero

Se debe manejar también una matriz de riesgos comunes o habituales, los cuales siempre deban ser cubiertos en los proyectos.

5. Responsabilizar al usuario del riesgo residual

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.

8. Realizar spikes de seguridad

9. Automatizar pruebas de aceptación

10. Automatizar pentesting

Nombre: _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

Actualmente se tienen capacitaciones en marcos ágiles, previas al inicio del uso en los proyectos y se puede incluir un módulo de seguridad genérico para aprovechar las sesiones de capacitación.

2. Designar responsable de seguridad

3. Utilizar plantilla historias de seguridad

Se debe cuidar que estas plantillas no se vuelvan paisaje. En otros momentos se han definido checklist y ~~criterios~~ requisitos no funcionales que se asumen pero no se revisan a conciencia.

4. Análisis de riesgo ligero

5. Responsabilizar al usuario del riesgo residual

Sería importante que se genere un artefacto (documento) donde quede explícito el análisis del riesgo ligero y las acciones que se van a implementar y los que no, para que quede mas clara la responsabilidad que se asume

6. Utilizar Marcas de seguridad

7. Obtener historias de seguridad e historias de abuso.

Sería bueno tener un "banco" de marcas e historias de seguridad y abuso que pueda ser consultado por aquellos que apenas empiezan a usar la metodología.

8. Realizar spikes de seguridad.

Cada mes se están realizando Dojos para reforzar conceptos de las metodologías ágiles. Sería bueno no solo pensar en Spikes sino en dojos que refuercen los conocimientos y se apliquen las experiencias obtenidas en otros proyectos.

9. Automatizar pruebas de aceptación

Este tema aún loveo muy crudo; no sé si es posible que se incluya como una evolución a la metodología, ya que, por ejemplo, de todas las aplicaciones que conozco ni siquiera se ha llegado a automatizar pruebas funcionales.

10. Automatizar pentesting

Se debería hacer un Enfasis en los momentos donde se debe realizar el analisis de seguridad de las historias de usuario. Mi recomendación sería que se realice como una actividad dentro de la planeacion de c/ release para que cada salida a producción considere la seguridad

Nombre: _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad

2. Designar responsable de seguridad

Veo mas adecuado designar la responsabilidad al arquitecto que tenga un enlace a alguien especializado de seguridad que participe en el "Sprint 0" e inception donde se den las guias y lineamientos

3. Utilizar plantilla historias de seguridad

Podria considerar que las plantillas tengan adaptaciones para tipos de proyectos y se utilicen como herramienta en su ejecución.

4. Análisis de riesgo ligero

5. Responsabilizar al usuario del riesgo residual

Es una politica de TI, debe ser obligatorio no es un asunto para que el PO resuelva o decida.

6. Utilizar Marcas de seguridad

Es posible crear casos de prueba de seguridad y automatizarlo?

7. Obtener historias de seguridad e historias de abuso.

Esto parte de tener unos lineamientos claros aplicables al proyecto, ~~si~~ podríamos optar como criterios de aceptación de historias de usuario para no ser negociables.

8. Realizar spikes de seguridad

Es un concepto genérico aplicable.

9. Automatizar pruebas de aceptación

Sí, verificar como enlazar al integrador. Igual punto 9.

10. Automatizar pentesting

Nombre: _____

Basado en su conocimiento cultural de EPM, conocimiento del marco ágil de desarrollo adoptado por EPM y/o conocimiento en seguridad, encuentra usted algún aspecto que se deba mejorar o cambiar dentro de las siguientes prácticas de seguridad propuestas para su implementación.

A continuación, se lista cada práctica independientemente para que haga sus observaciones de considerarlo necesario.

1. Entrenamiento en seguridad
   - El entrenamiento para el P.O. debe ser obligatorio
   - El equipo de dllo debe estar tambien entrenado
   - El arquitecto de la app debe tener conocimiento técnico

2. Designar responsable de seguridad
   - Podría ser el P.O.
   - Debería ser alguien de EPM de TI
   - Puede ser el arquitecto de la app
   - Revisar si aplica para los equipos de BI.

3. Utilizar plantilla historias de seguridad

4. Análisis de riesgo ligero

   Se debe hacer entrenamiento en análisis de riesgo

5. Responsabilizar al usuario del riesgo residual
   - Mas que responsabilizar, concientizar.
     Tal vez mostrar casos en los que no se tuvo en cuenta la seguridad y se materializó el riesgo

6. Utilizar Marcas de seguridad
   - Usar las políticas como marcas predecesoras (decisiones arquitectónicas genéricas)
   - Guías para que sea más fácil para los usuarios

7. Obtener historias de seguridad e historias de abuso.

Hacer acompañamiento con pedrinos como se hace hoy con los colibrí
Hacer un DOJO dentro del proyecto marcos ágiles (actualmente se hacen)

8. Realizar spikes de seguridad

Actual/ la carga de ceremonias y tareas distintas a desarrollo es muy alta y ocupa mucho tiempo los usuarios funcionales cuestionan mucho los tiempos altos (que se incrementan con ceremonias scrum) y TI no tiene ppto para estas actividades

Automatizar pruebas de aceptación

Revisar para BI que herramienta se podría utilizar o si es necesario

Automatizar pentesting