Seguridad informática en desarrollo de software con metodologías agiles

González Moreno Jorge Antonio, Universidad Piloto de Colombia

Resumen – El contenido del presente documento aborda cómo la seguridad informática debe ir de la mano con la gestión de proyectos en el desarrollo de software basado en la aplicabilidad de metodologías agiles, las cuales cada día son más implementadas por las organizaciones por su eficiencia, ligereza y flexibilidad para obtener mejores resultados en corto tiempo. En el tercer apartado del presente artículo se abordan las metodologías agiles y se toman como referencia algunas metodologías más acogidas por las organizaciones alrededor del mundo en los últimos años como la metodología SCRUM, KANBAN, y Programación Extrema XP, en el cuarto punto se realiza énfasis en el Seguridad informática, el sistema de información, los tipos de seguridad, y las propiedades con las cuales deben cumplir los sistemas seguros. Posteriormente se aborda las metodologías de desarrollo de software seguro, ya que estos permiten prevenir, identificar o corregir incidentes en el tema de seguridad informática, se aborda el ciclo de vida de desarrollo de un software y sus etapas, y las metodologías referenciadas son OWASP CLASP (Comprehensive Lightweight Application Security Process), OWASP Software Assurance Maturity Model (SAMM), OSSA (Oracle Software Security Assurance), por último se aborda las vulnerabilidades, fallos y los principales riesgos en los sistemas, así como son la Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT). El desarrollo de software ágil es importante pero la seguridad siempre debe estar inmersa en el desarrollo con el fin de mitigar vulnerabilidades, fallos y riesgos tanto en las aplicaciones como en la información que contienen**.**

Abstract – The content of this document addresses how computer security must go hand in hand with project management in software development based on the applicability of agile methodologies, which are increasingly implemented by organizations due to their efficiency, lightness and flexibility. to get better results in a short time. In the third section of this article, agile methodologies are addressed and some methodologies more accepted by organizations around the world in recent years are taken as a reference, such as the SCRUM, KANBAN, and Extreme XP Programming methodology, in the fourth point emphasis is made In Computer Security, the information system, the types of security, and the properties with which secure systems must comply. Subsequently, the secure software development methodologies are addressed, since these allow to prevent, identify or correct incidents in the field of computer security, the software development life cycle and its stages are addressed, and the referenced methodologies are OWASP CLASP (Comprehensive Lightweight Application Security Process), OWASP Software Assurance Maturity Model (SAMM), OSSA (Oracle Software Security Assurance), finally vulnerabilities, failures and the main risks in the systems are addressed, as well as the Analysis and Management Methodology of Information Systems Risks (MAGERIT). Agile software development is important, but security must always be embedded in development in order to mitigate vulnerabilities, failures, and risks in both applications and the information they contain.

Índice de Términos – Seguridad informática, Metodología Agiles, Vulnerabilidades, Metodologías de Software Seguro.

I. INTRODUCCIÓN

La evolución en la tecnología nos ha llevado que cada día más y más personas establezcan dentro de su rutina o cotidianidad el uso de herramientas o aplicaciones sistematizadas para satisfacer sus necesidades, motivo por el cual las empresas u organizaciones han tenido que evolucionar adaptándose al cambio y nuevas necesidades demandadas por los usuarios. Debido a la alta demanda de software fueron surgiendo metodologías agiles de desarrollo como lo es Scrum, XP, Kanban entre otras, con el fin de mejorar los tiempos de desarrollo del software para satisfacer las necesidades del cliente omitiendo requerimientos no funcionales como salvaguardar los principios de seguridad la información (disponibilidad, integridad y confidencialidad).

En el presente escrito se muestran algunas metodologías agiles utilizadas por las organizaciones en el desarrollo de software, así como también metodologías de desarrollo seguro cuyo objetivo es considerar la seguridad de los sistemas durante todo el ciclo de vida; como analizar los diferentes riesgos en los que pueden estar inmersos y de esta manera articular el desarrollo de software ágil pero seguro mitigando al máximo pérdida o robo de información y daños de imagen a las organizaciones.

Por lo anterior es de suma importancia que se implementen proyectos aplicando tanto metodologías agiles como metodologías de desarrollo de software seguro para identificar, analizar, disminuir y corregir los riesgos con el fin de construir software de calidad y satisfacción del cliente.

II. GESTIÓN DE PROYECTOS

La gestión de proyectos se basa en agrupar los diferentes componentes de un proyecto para alcanzar un fin u objetivo

---

Documento creado para optar al grado de Especialista en Seguridad Informática de la Universidad Piloto de Colombia. Jorge Antonio González Moreno, Ingeniero de Sistemas, Estudiante de la especialización en Seguridad Informática en la Universidad Piloto de Colombia.

específico, el cual consta de grupos de proceso de inicio, planeación, ejecución, monitoreo y control y cierre, los cuales interactúan dentro de cada fase o ciclo de vida de un proyecto. Ahora bien, la gestión de proyectos de software es diferente a la tradicional ya que se dedica a la planificación, programación, ejecución, recursos, seguimiento y entrega de software los cuales tienen un ciclo de vida donde se deben realizar en repetidas ocasiones pruebas, actualizaciones y/o ajustes basados en las sugerencias de los usuarios, es por esto que un gerente de proyectos debe contar con cualidades y experticia idónea para gestionar con éxito un proyecto.

III. METODOLOGIAS AGILES

Las metodologías agiles son procedimientos encaminados a solucionar necesidades de forma evolutiva a través del tiempo, las cuales deben adaptarse para suplir las expectativas y requerimientos funcionales. De acuerdo a Pressman [1] "para lograr la adaptación incremental, un equipo ágil requiere retroalimentación con el cliente (de modo que sea posible hacer las adaptaciones apropiadas)", esto con el fin de permitir al cliente evaluar los avances en el proyecto, la retroalimentación con el equipo de desarrollo y realizar las adaptaciones a que tenga lugar. Beck establece 12 principios de agilidad los cuales son:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- 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.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia. [2]

Dentro de las metodologías agiles se destacan:

A. KANBAN

Es una metodología de entregas continuas la cual busca tener siempre visible las tareas asignadas, las que se están desarrollando, cuales están pendientes por asignar y cuales ya fueron culminadas por los diferentes grupos de trabajo, todo esto en tableros los cuales le permiten al personal visualizar las tareas y optimizar el flujo de actividades con el fin de no sobrecargar a los grupos o equipos de desarrollo, permitiéndoles flexibilizar y planificar mejor las actividades las cuales se ven reflejadas con resultados más rápidos.
Basados en esta metodología se logran tener ventajas como planificación de tareas, reducción de tiempo, rendimiento del grupo de trabajo y entregas de forma continua.

B. PROGRAMACIÓN EXTREMA XP:

La programación Extrema XP es una metodología escrita por Beck en la cual establece cinco valores como pilares para la implementación de proyectos; comunicación, simplicidad, retroalimentación, valentía y respeto. Abordando algunos aspectos importantes de XP:

- Comunicación: se enfatiza en una comunicación verbal continua con el fin de obtener conceptos y retroalimentación constante.
- Simplicidad: se enfoca puntualmente en las necesidades inmediatas, descuidando o no abordando consideraciones futuras.
- Retroalimentación: XP implementa una prueba a medida que se realiza un desarrollo de acuerdo a su funcionalidad e historias de usuario para su aceptación.
- Valentía: Las prácticas estrictas de la metodología XP hace que el grupo de trabajo requiera valentía o disciplina para asumir o afrontar los proyectos basados a la presión en la que se pueden ver inmersos.

El proceso XP se basa en el desarrollo de programación orientada a objetos el cual se centra en cuatro actividades estructurales las cuales son la planeación, diseño, codificación y pruebas como se puede apreciar en la siguiente gráfica:

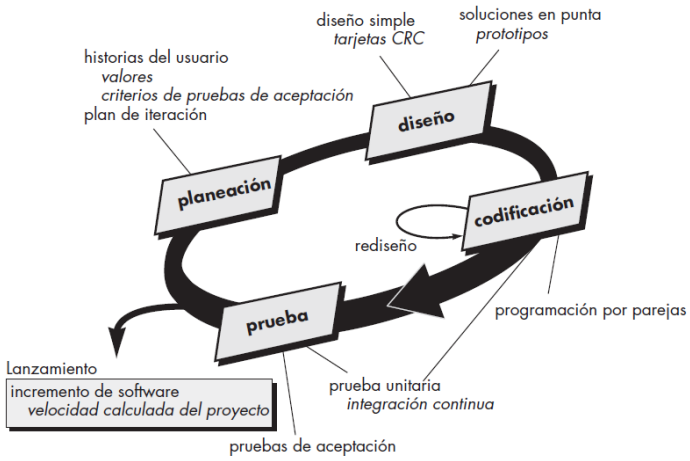

Fig. 1. Ingeniería del software Un enfoque práctico.

La programación extrema XP puede ser implementada en proyectos grandes o pequeños por su adaptabilidad y mejora en los tiempos de desarrollo, sin embargo, una falencia o desventaja puede ser al momento de cada entrega ya que el cliente puede realizar requerimientos lo que harán que el sistema siga creciendo.

C. SCRUM

La metodología Scrum se basa en el trabajo en equipo, por el cual mediante reuniones, herramientas y funciones de forma organizada permiten estructurar y gestionar mejor el trabajo.
Esta metodología está orientada principalmente sobre cinco valores los cuales son:

- Foco: Los Equipos Scrum se enfocan en un conjunto acotado de características por vez. Esto permite que al final de cada Sprint se entregue un producto de alta calidad y, adicionalmente, se reduce el time-tomarket.
- Coraje: Debido a que los Equipos Scrum trabajan como verdaderos equipos, pueden apoyarse entre compañeros, y así tener el coraje de asumir compromisos desafiantes que les permitan crecer como profesionales y como equipo.
- Apertura: Los Equipos Scrum privilegian la transparencia y la discusión abierta de los problemas. No hay agendas ocultas ni triangulación de conflictos. La sinceridad se agradece y la información está disponible para todos, todo el tiempo.
- Compromiso: Los Equipos Scrum tienen mayor control sobre sus actividades, por eso se espera de su parte el compromiso profesional para el logro del éxito.
- Respeto: Debido a que los miembros de un Equipo Scrum trabajan de forma conjunta, compartiendo éxitos y fracasos, se fomenta el respeto mutuo, y la ayuda entre pares es una cuestión a respetar. [3]

Scrum utiliza iteraciones o ciclos los cuales son llamados Sprints y contienen una serie de actividades donde se establecen los objetivos, conformación de los equipos y actividades grupales, establecimiento de reuniones cortas de seguimiento o avance de los proyectos y se evalúa el progreso general del mismo y por último se evalúa todo el trabajo realizado y se realiza un plan de mejora para el siguiente.

De acuerdo a [1] “Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega”, en cada una de estas actividades ocurre un sprint.

A continuación, se puede apreciar el flujo para un Sprint.

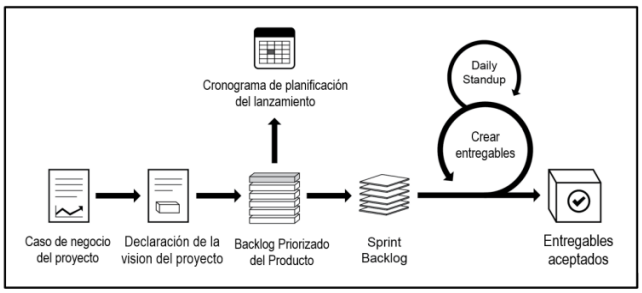

Fig. 2. Flujo de Scrum para un sprint

El equipo Scrum está conformado por tres actores principales, el primero es el dueño del producto el cual tiene como principales responsabilidades determinar la visión del producto, gestionas las expectativas de las partes interesadas, determinar y conocer las características funcionales, generar y mantener el plan de entregas, maximizar la rentabilidad del producto, establecer y cambiar las características del producto, participar en revisión del sprint para obtener retroalimentación de las partes interesadas, además debe tener el respaldo de la organización para la toma de decisiones; el segundo actor es el grupo de desarrollo, el cual está conformado por personal idóneo encargado de realización y calidad del producto, hacer las entregas y puestas en producción; y el tercer actor es el facilitador el cual es un líder que ayuda a los miembros del equipo a entender el proyecto para maximizar los esfuerzos, dentro de sus responsabilidades están velar por el correcto empleo y evolución del Scrum, asegurar que el equipo de desarrollo sea funcional y eficiente, detectar, monitorear y superar los impedimentos del proyecto y garantizar la cooperación y comunicación dentro del equipo.

IV. SEGURIDAD INFORMATICA

Según Baca la seguridad informática es:

> La disciplina que, con base en políticas y normas internas o externas de la empresa, se encarga de proteger la integridad y privacidad de la información que se encuentra almacenada en un sistema informático, contra cualquier

tipo de amenazas minimizando los riesgos tanto físicos como lógicos a los que está expuesta. [4]

Un aspecto importante dentro de la seguridad informática es el análisis de riesgos, el riesgo se define como la posibilidad de que no se obtengan los resultados deseados. Dentro del análisis de riesgo es importante la planeación la cual implica identificar las vulnerabilidades y medir el daño que pueden causar de tal manera que al disminuirlas se disminuirá el riesgo de sufrir daños, además también implica considerar los costos.

Otro aspecto relevante dentro de la seguridad informática es el sistema de información, para Aguilera [5] un sistema de información es "un conjunto de elementos organizados, relacionados y coordinados entre sí, encargados de facilitar el funcionamiento global de una empresa o de cualquier otra actividad humana para conseguir sus objetivos" además se establece que el sistema de información puede tener unos elementos como el recurso, el equipo humano, la información y las actividades.

La seguridad define dos tipos, la seguridad activa la cual comprende las medidas para reducir o evitar los riesgos en los que puede verse expuesto el sistema, y la seguridad pasiva la cual establece acciones para recuperar el sistema en el momento en el cual se produzca un incidente.

La falta de seguridad en los sistemas de información puede causar pérdidas económicas o de posicionamiento a la organización, las cuales pueden ser generados por incidentes fortuitos es decir errores cometidos accidentalmente por funcionarios, factores ambientales o fallos del sistema; o por los incidentes fraudulentos los cuales son originados por software malicioso, acciones mal intencionadas de funcionarios, robos o accidentes provocados en la organización.
Los sistemas seguros deben cumplir con las propiedades de integridad, confidencialidad y disponibilidad de la información para minimizar o evitar los riegos en las organizaciones.

- Integridad: propiedad que establece la consistencia y precisión de la información, ósea que no fue modificada sin autorización o que contengan errores con el fin de garantizar su veracidad.
- Confidencialidad: propiedad relacionada con la privacidad de la información, la cual establece el acceso de la información a un grupo de personas determinado.
- Disponibilidad: propiedad de la información con el fin que este accesible en modo tiempo y lugar cuando se requiera.

V. METODOLOGIA DESARROLLO DE SOFTWARE SEGURO

El desarrollo de software seguro se basa en la realización de controles periódicos para corregir o identificar cualquier error al inicio o durante la ejecución de un proyecto con el objetivo de optimizar tiempo, reducir costos y evitar accesos no autorizados. Por tal razón es necesario hablar de SDLC (Software Development Life Cycle) o ciclo de vida de desarrollo de un software el cual está conformado por las siguientes etapas:

- Análisis: corresponde al proceso en el cual se establece la necesidad y los requerimientos del sistema.
- Diseño: Se realizan los diagramas o modelados que expresan el flujo o componentes del sistema de acuerdo a la etapa de análisis.
- Desarrollo: es el proceso en el cual se realiza la codificación de los modelos de acuerdo a la etapa del diseño con el lenguaje de programación y herramientas establecidas.
- Pruebas: Etapa donde se evalúan los entregables de codificación de acuerdo a los requerimientos establecidos en la etapa del diseño con el fin de constatar que no existan errores.
- Implementación: Consiste en colocar el sistema a disposición del cliente para que lo utilice en ambiente productivo y resuelva la necesidad desarrollada.
- Mantenimiento: Esta etapa consiste en corregir los problemas del sistema que se puedan presentar generando una nueva versión más eficiente.

Basado en el desarrollo de software seguro y el ciclo de vida del software han surgido varias metodologías entre las cuales se encuentran:

Microsoft Security Development Lifecycle MS SDL:

Microsoft SDL es un proceso de desarrollo de software el cual utiliza un modelo en espiral con el fin de ayudar a los desarrolladores a crear software reduciendo problemas y vulnerabilidades de seguridad, así como también costos de desarrollo y mantenimiento.

EL proceso de MS SDL está conformado o dividido por siete fases las cuales son:

Fig. 3. El ciclo de vida de desarrollo de seguridad (SDL)

- Entrenamiento: Establece una formación sobre seguridad a todo el personal incluyendo desarrolladores e ingenieros, así como una específica de acuerdo a su rol o función, la cual debe ser actualizada analmente.
- Requerimientos: Los equipos de desarrollo los definen de acuerdo a factores como el tipo de datos, amenazas conocidas, regulaciones establecidas, mejores prácticas, lecciones aprendidas, funcionalidad y privacidad son documentados para su posterior trazabilidad.
- Diseño: Se crean modelos los cuales deben mantenerse y actualizarse a lo largo del ciclo de vida para poder identificar, calificar y categorizar las amenazas acordes al riesgo. De igual forma los desarrolladores deben utilizar la herramienta de modelado de amenazas lo que permite al equipo comunicar y analizar el diseño de seguridad del sistema en busca de posibles problemas y mitigación de los mismos.
- Implementación: En esta etapa los desarrolladores escriben el código acorde con el diseño y desarrollo, caben mencionar que cuentan con herramientas de desarrollo, compiladores y comprobaciones de seguridad para la implementación y seguridad del software.
- Verificación: Antes de publicar cualquier sistema o funcionalidad requiere ser verificada de acuerdo a los requisitos de diseño y que no contenga errores de codificación para este fin se emplea un revisor ajeno al personal al grupo de desarrollo con el fin de ejercer un control de separación de funciones. En esta etapa se realizan controles de análisis de código estático, análisis binario, escáner de credenciales y secretos, escaneo cifrado, fuzz testing, validación y component governance, si el revisor encuentra algún inconveniente con el código se informa al responsable para que realice los ajustes y envíen nuevamente para su revisión.
- Lanzamiento: Luego de aprobar todas las pruebas y revisiones el sistema no se lanza inmediatamente a todos los clientes si no que se hacen progresivamente a grupos cada vez más grandes.
- Respuesta: Posterior al lanzamiento se realiza una supervisión del sistema identificando posibles incidentes de seguridad en tiempo real.

OWASP CLASP (Comprehensive Lightweight Application Security Process)

OWASP [6] define el modelo como “un modelo prescriptivo, basado en roles y buenas prácticas, que permite a los equipos de desarrollo implementar seguridad en cada una de las fases del ciclo de vida de desarrollo en forma estructurada, medible y repetible.”

OWAS CLASP está conformada por:

- CLASP View: Perspectivas de alto nivel, estas se dividen en cinco las cuales son conceptos, roles, evaluación de actividades, implementación de CLASP Best Practices: Buenas practicas, agrupación de actividades como programas institucionales de sensibilización, evaluar el rendimiento de las aplicaciones, obtener los requerimientos de seguridad, implementar prácticas de desarrollo seguro, construir procedimientos de solución de vulnerabilidades y publicar guías operacionales de seguridad.
- actividades y vulnerabilidades que contienen procesos que interactúan entre sí.
- CLASP Activities: Actividades, implementadas para facilitar una interacción entre seguridad y el SDLC.
- CLASP Resources: Recursos, contribuyen a la planificación, ejecución y cumplimiento de las actividades relacionadas con la seguridad del software.
- CLASP Taxonomy: Taxonomía, clasificación de tipos de problemas o vulnerabilidades, divididos en 5 categorías que son errores de tipo y rangos, problemas del entorno, errores de tiempo y sincronización, errores de protocolo y errores de lógica.

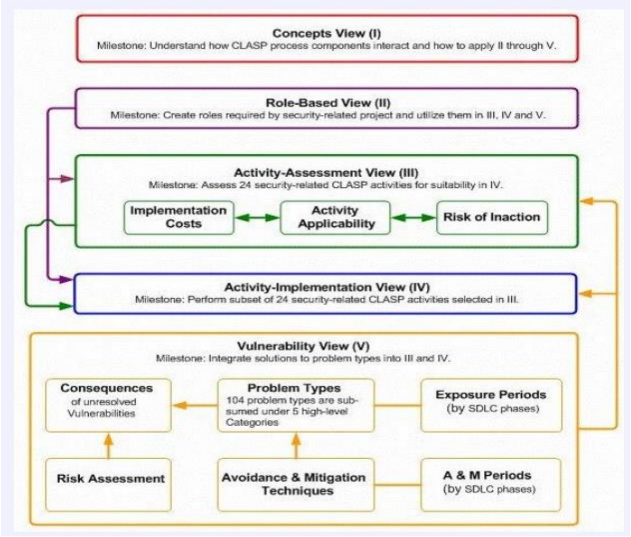

Fig. 4. OWASP CLASP

OWASP Software Assurance Maturity Model (SAMM)

Es un modelo de madurez efectivo y medible para analizar y mejorar la seguridad del software el cual pretende enseñar a las organizaciones como diseñar, desarrollar e implementar software seguro acorde a sus necesidades. Este modelo ayudará a evaluar las prácticas de seguridad existentes, construir un programa balanceado con iteraciones definidas, demostrar mejorar en el software y definir las actividades relacionadas con la seguridad.

Algo importante de resaltar es que SAMM puede ser utilizado por cualquier organización así sea pequeña, mediana o grande independiente del estilo de desarrollo que utilicen ya que fue construido con los principios de cambios de comportamiento de la organización a través del tiempo, así como también es flexible lo cual permite personalizar las opciones basado en su tolerancia al riesgo y como utiliza el software.
Las bases de este modelo están definidas en tres niveles de madurez para las doce prácticas de seguridad como lo ilustra la siguiente imagen:

Fig. 5. Desarrollo de Software

OSSA (Oracle Software Security Assurance)

Es una metodología creada para brindar seguridad en el diseño, desarrollo, pruebas y mantenimiento abarcando cada fase del ciclo de vida del desarrollo del producto.
Dentro del desarrollo seguro OSSA implementa:

- Estándares de codificación: en el cual establece la hoja de ruta, guía y capacitación a los desarrolladores para la producción de código seguro.
- Análisis y pruebas de seguridad: establece los tipos de pruebas realizadas incluyendo el análisis estático, dinámico y funcional.
- Hackeo ético: realización de pruebas de penetración y análisis de seguridad operativa.
- Configuración segura: política en la cual establece que los productos deben ser seguros por defecto.
- Evaluaciones externas de seguridad: Evaluación con organizaciones acreditadas de productos de acuerdo a criterios comunes. [7]

Por otra parte, dentro de la gestión de vulnerabilidades realiza actualización de parches y alertas de seguridad para la corrección de errores de seguridad, así como instrucciones de como informar una supuesta vulnerabilidad.

VI. VULNERABILIDADES, FALLOS Y RIESGOS

Al hablar de seguridad informática hay que tener en cuenta todos los elementos que la componen, los cuales son fundamentales para valorar, determinar y minimizar el impacto de los incidentes, estos elementos son amenazas, vulnerabilidad, riesgo y ataques.

En sistemas de información se entiende por amenazas la presencia de uno o más factores que pueden atacar al sistema produciendo daño, teniendo en cuenta el estado de vulnerabilidad. Existen diferentes tipos de amenazas entre los cuales se encuentran las físicas, electrónicas, incidentes intencionales, los incidentes maliciosos entre otros. Las amenazas se pueden clasificar en cuatro grupos; de interrupción, de interceptación, de modificación y de fabricación.

- De interrupción: El objetivo es deshabilitar el acceso a la información.
- De interceptación: El acceso no autorizado a un determinado recurso o información del sistema
- De modificación: Consiste en la modificación no autorizada de datos o programas.
- De fabricación: Agregar información errónea o falsa en la información del sistema.

En cuanto a los riesgos, se refieren a la posibilidad de que se materialicen las amenazas aprovechando la vulnerabilidad, ante un riesgo a organización cuenta con tres alternativas que son asumirlo, disminuirlo o anularlo y transferirlo; por otra parte, las vulnerabilidades son probabilidades que existen de que una amenaza se materialice, de las cuales existen dos tipos las físicas y las lógicas.
Las vulnerabilidades físicas son las que afectan a la infraestructura como por ejemplo los desastres naturales terremotos, inundaciones entre otros. De igual forma también incluyen los controles de acceso físico como a los centros de cómputo o lugares dentro de la organización.

Las vulnerabilidades lógicas por el contrario de las físicas afectan el desarrollo de la operación por ejemplo la configuración del sistema operativo, desactualizaciones y de desarrollo como inyecciones de código SQL, Cross Site Scripting por mencionar algunas.

Los 10 principales riesgos de seguridad de las aplicaciones web para el 2021 según OWASP son:

- A01:2021-Control de acceso roto
- A02:2021-Fallas criptográficas
- A03:2021-Inyección
- A04:2021-Diseño inseguro
- A05:2021-Configuración incorrecta de seguridad
- A06:2021-Componentes vulnerables y obsoletos
- A07:2021-Fallas de identificación y autenticación
- A08:2021-Fallas de integridad de datos y software
- A09:2021-Fallas de registro y monitoreo de seguridad
- A10:2021- Falsificación de solicitudes del lado del servidor

De acuerdo con los riesgos presentes en la actualidad, es recomendable que las organizaciones cuenten con un área de

administración de riesgos con el objetivo de identificar problemas antes que estos ocurran y así generar actividades dirigidas a mitigar los impactos que pueden causar en la organización. La administración del riesgo se contempla en tres etapas las cuales son identificar y analizar el riesgo, definir una estrategia para administrar el riesgo e implementar planes de mitigación cuando sean necesarios. Dentro del control del riesgo existen mecanismos de seguridad preventivos, detectores y correctores.

MAGERIT: Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información

El MAGERIT [8] "es la metodología de análisis y gestión de riesgos de la información desarrollada por el Consejo Superior de Administración Electrónica". Los elementos principales para el análisis de riesgos son activos, amenazas, vulnerabilidades, impacto, riesgo, funciones servicios y mecanismos.

Esta metodología establece el proceso de gestión del riesgo con el fin que se tomen decisiones inferidas en riesgos por la utilización de tecnologías de información.

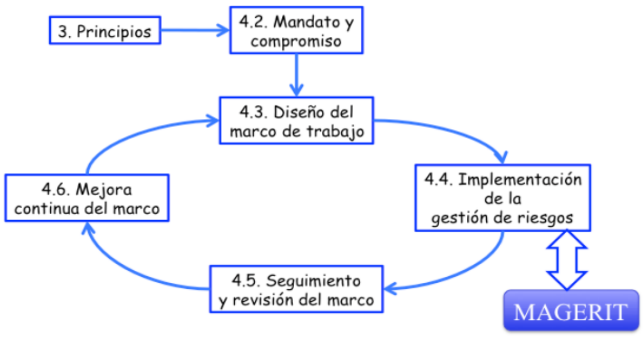

Fig. 6. ISO 31000 -Marco de trabajo para la gestión de riesgos

Esta metodología establece los siguientes objetivos:

- Concienciar a los responsables de las organizaciones de información de la existencia de riesgos y de la necesidad de gestionarlos
- Ofrecer un método sistemático para analizar los riesgos derivados del uso de tecnologías de la información y comunicaciones (TIC)
- Ayudar a descubrir y planificar el tratamiento oportuno para mantener los riesgos bajo control Indirectos
- Preparar a la Organización para procesos de evaluación, auditoría, certificación o acreditación. [8]

VII. CONCLUSIÓN

- En la realización de un proyecto todo el equipo de trabajo debe estar debidamente capacitado acerca de software seguro y se debe profundizar en el conocimiento llegado el caso de acuerdo a su rol o función en la organización, esto con el fin que todo el personal involucrado en el proyecto tenga conocimiento adecuado para generar software de alta calidad.
- En todo proyecto de desarrollo de software la seguridad debe estar inmersa desde el inicio contemplando cada una de las fases que conforman el ciclo de vida hasta la finalización y de esta manera poder realizar una mejor planificación y seguimiento en búsqueda de fallos o vulnerabilidades que acarrearían en pérdidas de información, recursos y tiempo, con el fin de evitar reprocesos y optimizar costos independientes de la metodología ágil que se esté o vaya a utilizar.
- Realizar pruebas de seguridad con herramientas de automatización en cada iteración del proceso de desarrollo del software facilita la detección de amenazas y vulnerabilidades, así como su corrección.
- Establecer una metodología de análisis de riesgo en los proyectos es importante porque permite al equipo de trabajo tomar medidas de prevención, aminorar posibles pérdidas, evitar potenciales peligros o minimizar su impacto.
- Crear software utilizando metodologías agiles nos brinda grandes beneficios en cuando autonomía, flexibilidad y eficacia de los proyectos, con una mayor calidad en los productos por sus revisiones constantes generando adaptabilidad a los posibles cambios, así como satisfacción del cliente por la agilidad en la entrega de los productos acorde a sus necesidades.

REFERENCIAS

[1] R. S. Pressman, "*Ingenería del software un enfoque práctico"* 2ª ed. México: Mc Graw Hill, 2010, pp. 58-69.
[2] K. Beck, "*Manifiesto por el desarrollo ágil del software"* 2001. [En línea] Disponible en https://agilemanifesto.org/iso/es/principles.html [Consultado: 05- nov-2022]
[3] D. M. Alaimo, "*Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos*", Buenos Aires : Kleer, 2013, pag 24.
[4] G. Baca Urbina, "Introducción a la seguridad Informática" México: Grupo Editorial Patria, 2016, pp.12
[5] P. Aguilera, "*Seguridad Informática"* Madrid: Editex, 2010, pp. 8
[6] OWASP, *"Comprehensive, lightweight application security process*", 2006. [En línea] Disponible en http://www.owasp.org, [Consultado: 05- nov-2022]

[7] ORACLE, Oracle Foftwarw Security Assueance, [En Línea] Disponible en https://www.oracle.com/mx/corporate/security-practices/ , [Consultado: 05- nov-2022]

[8] Ministerio de Hacienda y Administraciones Públicas, "*Magerit v.3: Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información",* Madrid, 2012 [En línea]. Disponible en: http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html#.VRMl5_yG8ms [Consultado: 05- nov-2022]

Autor. González Moreno Jorge Antonio, Nació en el Socorro-Santander el 16 de julio 1986, Ingeniero de Sistemas de la Universidad Industrial de Santander en 2013, estudiante de Especialización en Seguridad Informática en la Universidad Piloto de Colombia.