Balanced ScoreCard

Para

DevSecOps

Version 0.1

Autor: Ronen Riesenfeld

Revisó y contribuyó:

Luciano Moreira da Cruz

Mateo Martinez

Introducción

Un Balanced Scorecard o Cuadro de Mando Integral, es una metodología de gestión estratégica que permite definir y realizar seguimiento a la estrategia de una organización. Esta metodología permite estructurar los objetivos estratégicos de forma dinámica e integral para poder medirlos y analizarlos mediante una serie de indicadores que evalúan el desempeño de las iniciativas y proyectos.

DevSecOps es una metodología ágil de desarrollo y despliegue de aplicación que a través de los años ha demostrado ser muy efectiva ayudando a las organizaciones a entregar aplicaciones más fiables, estables y seguras de una manera más rápida; DevSecOps es una práctica, no un rol, es la manera de hacer las cosas.

El presente documento plantea una serie de indicadores recomendados para DevSecOps alineados a la metodología de Balanced Scorecard.

El reporte del estado de DevOps (State of DevOps Report) presentado por Puppet muestra que las organizaciones que han adoptado herramientas de DevOps y siguen los principios de DevOps, han alcanzado a lograr:

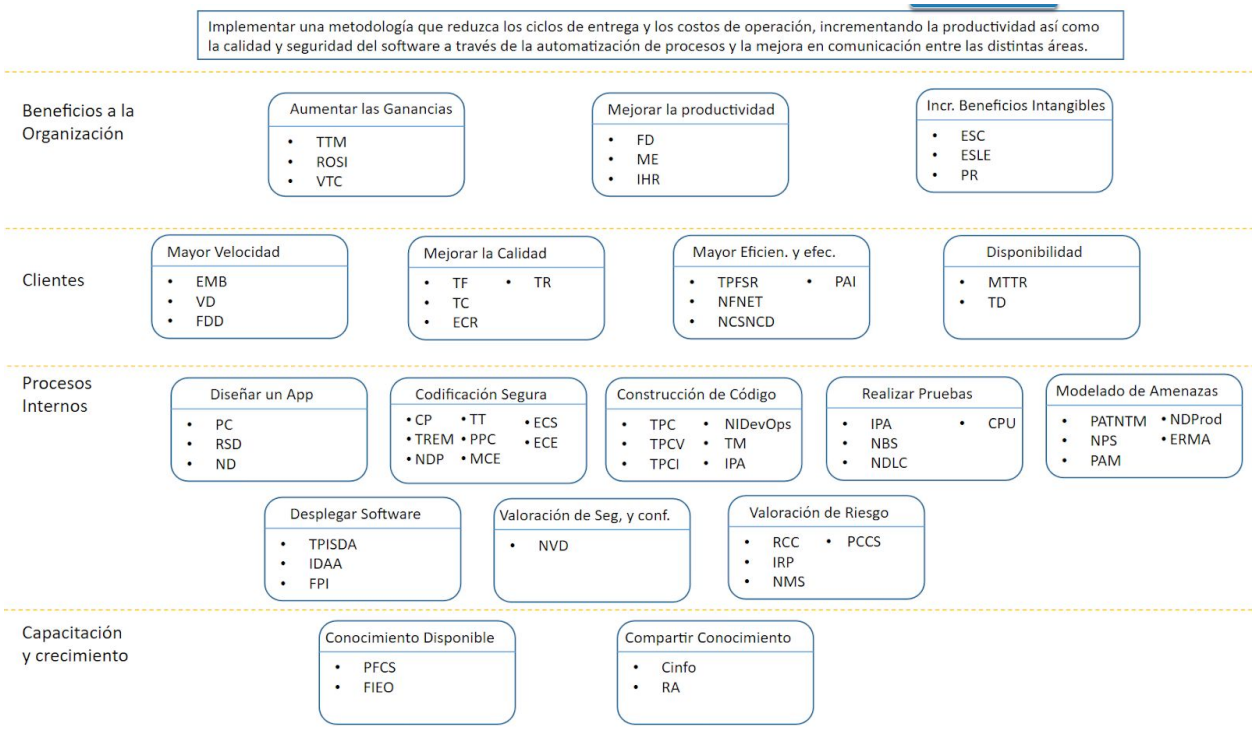
- Despliegues de software 46 veces más frecuentes que sus competidores.
- Recuperación de fallas 96 veces más rápida.
- 440 veces más rápido la entrega de cambios.
- 5 veces menor los fallos en cambios.
- Eficiencia operativa y mejores niveles de satisfacción del cliente.

Si un proceso se puede automatizar, entonces hay que automatizarlo, esto nos permite asegurarnos que se cumplan los criterios y políticas de calidad y seguridad y de lo contrario podemos tomar acciones como por ejemplo, la suspensión de los procesos, por lo tanto, solo los componentes que cumplen las “mejores prácticas” pasan el pipeline de DevSecOps, garantizando que se hace entrega solo de aquellos que cumplen.

La implementación de DevSecOps no es de un dia para otro, sino es un proceso de cambio que las organizaciones deben de llevar a cabo, pero ¿por dónde empezar?. Bueno, primero entender la filosofía de DevSecOps, planear los cambios que se deban implementar tanto en la estructura organizaciones como en las políticas y los procedimientos, definir una estrategia de implementación y comenzar a llevarla a cabo.

Para entender cómo cada cambio que se implemente afecta a los procesos y objetivos de DevSecOps así como para entender que mejorar es importante poder medir cada uno de los

procesos involucrados. Por tal motivo, creamos este Balance Scorecard que su intención es dar visibilidad de los componentes y partes involucradas, ayudar a entender cómo cada una afecta y proporcionar una herramienta de medición que ayude a cumplir con los requerimientos de DevSecOps. A continuación se presenta un diagrama general



Métricas

Beneficios a la organización

Aumentar Ganancias

Incrementar la eficiencia de las inversiones en comparación a sus costos y tener un time to market adecuado.

- Reducir el Time to market (TTM)

  Es el tiempo requerido para poner un producto o servicio desde su concepción hasta estar disponible en producción. Un buen TTM se puede significar entregar a tiempo según un horario pre establecido o puede significar lanzar un producto antes de que aparezca otra para sustituirlo (puede ser de la competencia). Llegar después puede significar pérdidas importantes. "Los productos tecnológicos que llegan al mercado seis meses tarde pero dentro del presupuesto generan un 33% menos de ganancias en cinco años" (Becher, 2016)

  TTM = Fecha de puesta en producción - Fecha de concepción
  TTM = Fecha de aparición de sustituto - Fecha de puesta en producción
  TTM = Tiempo de vida del producto - (Fecha de puesta en producción - Fecha planeada)

  Donde:
  Fecha de puesta en producción: es la fecha en la que una organización liberó un producto o servicio al público esperado.
  Fecha de concepción: Es la fecha en donde la organización definió un producto o servicio y lo solicitó a producción.
  Fecha de aparición de sustituto: es la fecha en donde un producto o servicio sea de la competencia o no sea una alternativa al producto o servicio que ya está en producción.
  Tiempo de vida del producto: Es la duración esperada del producto en el mercado.
  Fecha planeada: Es la fecha en donde se espera poner el producto en producción para que tenga el tiempo de vida esperado.

- Return of Security Investment (ROSI)

  El ROSI está definido por OWASP como una medida que ayuda a determinar si la inversión en seguridad para prevenir o frustrar ataques se justifica como una inversión de seguridad a largo plazo: si el ROSI no es positivo, la inversión no se justifica, mientras que si es nulo no hay ahorro o rendimiento de la inversión. Hay varias fórmulas para calcular el ROSI (OWASP, 2013).

  $$ROSI = \frac{ALE * porcentaje\ de\ efectividad - costo\ de\ inversión}{Costo\ de\ Inversion}$$

  Donde:
  ALE (Annual Loss Expectancy): Es la pérdida anual esperada . Es la pérdida total anual esperada a causa de incidentes de seguridad.
  Porcentaje de efectividad: La probabilidad de mitigar el ataque
  Costo de inversión: El costo de implementar una solución de mitigacion y prevencion

  Para calcular la pérdida anual esperada se requiere la siguiente fórmula:

  $$ALE = ARO * SLE$$

  Donde:
  ARO (Annual Rate of Occurrence): Es la probabilidad de que un incidente ocurra en un año
  SLE Single Loss Expectancy: El costo total de un solo incidente. Debe incluir los costos directos de las pérdidas así como los costos indirectos asociados a las consecuencias del robo de datos.

- Incrementar el Valor Total del Cliente (VTC)

  Indica los ingresos totales que una empresa puede esperar razonablemente de una sola cuenta/cliente. Nos dice cuántos ingresos se puede esperar que genere un cliente en el transcurso de la relación comercial. Mientras más tiempo un cliente continúe comprando en la compañía, mayor será su valor de por vida (TLV). Para calcularlo se requieren de diversos cálculos intermedios.

  $$CLTV = \frac{Valor\ del\ cliente}{Vida\ media\ del\ cliente}$$

  Donde:
  CLTV: customer lifetime value

  $$Valor\ del\ cliente = \frac{Valor\ promedio\ de\ compra}{Tasa\ de\ frecuencia\ de\ compra\ promedio}$$

  $$Valor\ promedio\ de\ compra = \frac{Ingresos\ totales}{Numero\ de\ ordenes}$$

$$Tasa\ de\ frecuencia\ de\ compra\ promedio = \frac{Numero\ de\ compras}{Numero\ de\ clientes}$$

$$Vida\ media\ del\ cliente\ = \frac{suma\ de\ vida\ útil\ de\ los\ clientes}{Numero\ de\ clientes}$$

Mejorar la Productividad

Tener equipos de trabajo interdisciplinarios, motivados y eficientes.

- Frecuencia de Despliegue (FD)

  Nos indica la velocidad real de una empresa, nos permite entregar valor a los clientes, encontrar sus necesidades y resolver problemas. Esta medida no se puede tratar de forma segura como la medida definitiva de velocidad de una organización, debido a que la entrega frecuente se puede deber a mayor número de bugs, por lo que se debe de buscar un balance adecuado (Smith, 2018).

  $$Frecuencia\ = \frac{Numero\ de\ liberaciones}{Por\ semana}$$

- Motivación de Empleados (ME)

  La idea es llevar a cabo una encuesta interna de motivación de los empleados. Con esta se puede obtener información sobre qué mejorar. Estas encuestas deben de poder identificar y medir distintos aspectos de la motivación de los empleados

- Información de HR sobre los empleados (IHR)

  El departamento de recursos humanos debe de proporcionar información sobre el estado de los empleados, esta información puede indicar su motivación y perspectiva sobre la compañía

Incrementar los Beneficios Intangibles

Los stakeholders están satisfechos con los productos generados y perciben disponibilidad constante de los equipos.

- Encuesta de Satisfacción de los Clientes (ESC)

  Los clientes en este contexto son todos los stakeholders, y con esta encuesta se busca identificar el nivel de satisfacción de la organización con el servicio y los productos generados. La información que se busca obtener es sobre la experiencia a nivel micro y llevar estas a abordar situaciones a nivel macro. Debemos de poder responder ¿A dónde va la organización?, ¿Cómo está nuestro servicio en comparación a las necesidades del mercado? , ¿Dónde necesitamos mejorar?

- Encuesta Satisfacción Laboral de los Empleados (ESLE)

  La intención es entender qué tan satisfechos están los empleados y cuales son las medidas probables que se tendrían que tomar para llevar a cabo mejoras. Una organización es un lugar que está hecho del estado de ánimo, moral, ambiente, cultura y motivación de todos sus empleados. La mejor manera de medir, analizar y obtener información sobre la organización es a través de una encuesta de satisfacción de los empleados.

- Peer Review (PR)

  Recibir retroalimentación de compañeros del mismo equipo. El objetivo es garantizar que el trabajo que se está realizando es de la calidad esperada, recibir sugerencias de mejoras de nuestros pares y recibir retroalimentación en general sobre cómo se están cumpliendo con la expectativas del equipo.

Clientes

Mayor Velocidad

Incrementar la velocidad en que una versión puede ser desplegada desde el punto de vista del cliente (desde la solicitud). Los índices de estabilidad del Main branch, velocidad de despliegue, y frecuencia, fueron tomados del reporte "*Three Critical Development Metrics for Engineering*." de CircleCI (CircleCI, n.d.).

- Estabilidad del Main Branch (EMB)

  La estabilidad del main branch es una medida de que listos estamos para un despliegue. Si se necesita hacer un despliegue ahorita, ¿se podría? Esta métrica afecta directamente la frecuencia de despliegue. Esta métrica, se mide como el porcentaje de tiempo que el main branch está en un estado fallido. (CircleCI, n.d.)

  Valores de Referencia de acuerdo con Alexa Internet Ranked orgs:

5 percentile	46% del tiempo está en un estado fallido
La Media	98.5% del tiempo está en un estado estable
80% de las org	90% del tiempo está en un estado estable

- Velocidad de Despliegue (VD)

  Es el tiempo que tarda el código en pasar del main branch a producción, esto puede variar desde unos minutos hasta muchas horas. Este costo es incurrido cada vez que se cambia el código ya sea por una nueva función o corrección. Cuanto menor es el tiempo de implementación, menos costoso es cambiar tu producto. El tiempo de despliegue es una medida del costo de despliegue. Cuanto menor sea el tiempo de implementación, menos costoso será cambiar su producto.

  Esta métrica es el tiempo de despliegue medido como la cantidad de minutos entre que un trabajo entra a la cola y cuando termina el despliegue.

  Valores de referencia (AIR):
  - 80% de las organizaciones despliegan en 17 minutos o menos
  - Las organizaciones más rápidas en 2.6 minutos
  - El promedio es 7.9 minutos
  - El 5 percentil está en 36.1 minutos

- Frecuencia de despliegue (FDD)

  Esta métrica indica realmente que tan rápido se mueve una organización. Esta métrica está directamente influenciada por la velocidad de despliegue y la estabilidad del main branch. Las organizaciones con alto desempeño pueden desplegar bajo demanda, mientras que las organizaciones con bajo desempeño sólo pueden desplegar cada 4 semanas o más.

  $$Frecuencia = \frac{Numero\ de\ liberaciones}{Por\ semana}$$

  Valores de referencia (AIR):
  - 95 percentile despliega 42 veces por semana
  - El 5 percentil despliega 1 vez por semana
  - La media son 8 despliegues por semana
  - 75% de las organizaciones despliega menos de 16 veces por semana

Mejorar la Calidad

Desplegar software que cumple con sus requisitos sin fallas ni vulnerabilidades

- Total de Fallas (TF)

  El número de fallas tanto funcionales como de seguridad detectadas en un periodo de tiempo establecido. Entendiéndose como fallas funcionales cualquier error o falla en el

programa o sistema que hace que produzca un resultado incorrecto o inesperado. Las fallas de seguridad se refiere a que el programa o sistema está expuesto a la posibilidad de ser atacado, danado o explotado.

- Tasa de Completación (TC)

  Necesitamos entender cómo los usuarios utilizan nuestros productos y si tienen una buena experiencia. Las pruebas de usabilidad nos ayudan a entender que tan probable los usuarios pueden completar tareas en el software en un tiempo preestablecido. Esto es especialmente útil para medir el impacto potencial en la productividad del usuario.

$$Tasa\ de\ Completación\ = \frac{número\ de\ usuarios\ que\ completaron\ una\ tarea\ en\ un\ tiempo\ preestablecido}{Total\ de\ usuarios}$$

- Esfuerzo en Cambios por Regresiones (ECR)

  Indica el trabajo necesario que se hace para corregir problemas encontrados una vez que una versión ha sido liberada.

$$Esfuerzo\ de\ cambios\ = \frac{Número\ de\ correcciones\ (parches)}{Número\ de\ características}$$

- Tiempo de Respuesta (TR)

  Es el tiempo que toma cada interacción entre el usuario y el sistema. Según el estudio de Hoxmeier, J.A., & Dicesare, C. (2000), "Por cada unidad (tres segundos) de aumento en el tiempo de respuesta, hay un promedio de caída de .22 en promedio satisfacción." Una buena percepción del sistema se mantuvo cuando los tiempos de respuesta eran menores a 6 segundos.

$$Tiempo\ de\ Respuesta\ =\ tiempo\ entre\ la\ solicitud\ y\ la\ respuesta\ esperada$$

Mayor Eficiencia y Efectividad

Resolver cualquier falla, vulnerabilidad o discrepancia de manera rápida

- Tiempo Que Permanece una Falla sin Resolver (TPFSR)

  Este indicador mide el tiempo desde que una falla se detecta hasta que se resuelve en producción.

$$TPFSR\ =\ Fecha\ de\ puesta\ en\ producción\ -\ fecha\ de\ apertura\ de\ ticket$$

- Número de Funciones Nuevas Entregadas a Tiempo (NFNET)

  Es el número de características nuevas que fueron solicitadas y planeadas y son entregadas en el sprint definido.

  $$NFNET = \frac{Número\ de\ características\ puestas\ en\ producción}{Número\ de\ características\ solicitadas}$$

- Número de Cambios Solicitados por no Cumplir con lo Definido (NCSNCD)

  Es el número de cambios realizados a la aplicación por haberse desviado de la solicitud original.

  $$NCSNCD = Número\ de\ cambios$$

- Plazo de aprobación de implementación (PAI)

  Tiempo transcurrido entre la solicitud de implementación de un cambio aprobado y la implementación real en producción

Disponibilidad

Mantener la disponibilidad definida en la aplicación

- Tiempo medio de recuperación (MTTR)

  Tiempo en que la aplicación no está disponible por un despliegue fallido. Tiempo entre un despliegue de producción fallido hasta la restauración completa de las operaciones de producción

  $$MTTR = fecha - hora\ de\ despliegue\ de\ la\ aplicación - fecha - hora\ en\ que\ la\ aplicacion\ no\ esta\ disponible$$

- Tiempo Disponible (TD)

  Indica el tiempo total que la aplicación está disponible en comparación con el tiempo total disponible requerido. El tiempo requerido está definido en el SLA y se deben de considerar los tiempos de mantenimiento programados así como los tiempos de despliegues y otras actividades que pueden causar una interrupción pero que sean necesarias para la operación..

  $$TD = \frac{Tiempo\ total\ disponible}{Tiempo\ total\ requerido}$$

Procesos Internos

Diseñar una Aplicación

Define los objetivos específicos que cumplen con los requerimientos y donde se especifica la arquitectura, un diseño detallado y la seguridad suficiente para satisfacer las necesidades operativas, bajo condiciones y eventos inesperados.D

- Pruebas de Carga (PC)

  Esta prueba se realiza para verificar el rendimiento del sistema bajo una carga esperada. La aplicación no debe de reportar errores ni fugas de memoria (memory leaks). Se debe cargar la aplicación al máximo esperado por un periodo de tiempo definido. El tiempo promedio de respuesta debe de cumplir con lo requerido.

- Riesgo de Seguridad por Diseño (RSD)

  Nos indica en qué medida los requisitos han sido verificados y probados contra los riesgos de seguridad.

  $$RSD = \frac{Requisitos\ verificados}{Total\ de\ Requisitos}$$

- Número de Defectos (ND)

  Número de defectos (bug y vulnerabilidades) ya sea funcionales, no funcionales y de seguridad que fueron detectados en otra etapa del proceso de desarrollo, construcción y despliegue. El valor esperado es cero.

Codificación Segura

Es el proceso de escribir el código de manera estandarizada, repetible y utilizando componentes seguros, basado en el diseño y los requerimientos siguiendo las reglas de estilo y las prácticas del código.

- Cumplimiento de Políticas (CP)

  Indica el porcentaje de controles relacionados con la retención necesaria por alguna norma que están automatizados.

  $$CP = \frac{Número\ de\ controles\ automatizados}{Total\ de\ controles}$$

- Tasa de Remediación (TREM)

  Bugs de Calidad

Es el tiempo promedio que toma remediar cada uno de los defectos funcionales como no funcionales reportados.

Bugs de Seguridad
Es el tiempo promedio que toma remediar cada una de las vulnerabilidades de seguridad reportadas.

- Número de Defectos en Producción (NDP)

  Es el número de defectos (bugs/vulnerabilidades) funcionales, no funcionales y de seguridad descubiertos en producción y no detectados durante su construcción.

- Administración de Defectos (TT)

  Es el tiempo de triage.

  Calidad
  Es el tiempo que toma en determinar qué tan crítico es un defecto, qué impacto tendrá en la funcionalidad de la aplicación o de todo el sistema, esfuerzo de remediación y la asignación del mismo a un responsable para su mitigación.

  Seguridad
  Es el tiempo que toma en determinar qué tan crítico es un defecto, qué impacto tendrá en la funcionalidad de la aplicación o de todo el sistema, que riesgo tiene, dificultad de mitigación y la asignación del mismo a un responsable para su mitigación.

- Porcentaje de Pruebas al Código (PPC)

  Se refiere a escribir código que sea fácil de probar para un sistema automatizado como a escribir las pruebas. Las pruebas actúan como documentación para que los cambios al código de un desarrollador muestran rápidamente un error. Esto también ayuda a incorporar a nuevos desarrolladores (Osbourn, 2020).

  Indica que tantas pruebas se tienen definidas en comparación al código. Por ejemplo, las pruebas cubren el 97% del código base.

- Modificación del Código Existente (MCE)

  Se refiere a la frecuencia con que el código cambia con el tiempo. Si bien es una parte natural del ciclo de desarrollo, las modificaciones al código existente debe minimizarse siempre que sea posible. Mientras menos código se necesite cambiar para acomodar una nueva característica, mejor. (Osbourn, 2020)

  Por ejemplo, en cada sprint se deberá de modificar menos del 10% de código existente

- Escribir Código Simple (ECS) (cyclomatic complexity)

  Cuanto más simple es el código, más fácil es probarlo y cambiarlo. Cada vez que se incorpora un nuevo desarrollador, les lleva tiempo aprender el nuevo código, se puede reducir este tiempo haciendo que el código se escriban de manera simple (Osbourn, 2020). Mantener un nivel menor a 10 es lo ideal.

- Escribir Código Estable (ECE)

  Se refiere a que los cambios en el producto no perjudican al resto del negocio. Si el lanzamiento de una nueva característica rompe una característica existente, esto es perjudicial, por lo que se busca estable para que el equipo no esté enfocado en la mitigación de fallas sino pueda concentrarse en desarrollar nuevas funciones. Una medición podría ser "Menos del 2% del código implementado durante el año debería resultar en un tiempo de inactividad de la aplicación". (Osbourn, 2020)

Construcción del Código

Conjunto de procesos mediante el cual el código fuente se convierte en una aplicación independiente que se puede ejecutar en un dispositivo. El proceso incluye diversos pasos como la compilación y el empaquetado de librerías.

- Tiempo promedio de construcción (TPC)

  Es el tiempo promedio que se toma en desplegar una aplicación. Es el promedio de los tiempos que el proceso de construcción toma desde que se dispara un pipeline (o se da clic en construir) hasta su completación. (Vriesman, 2017)

- Tiempo dedicado a problemas con el control de versiones (TPCV)

  Es el tiempo en que los desarrolladores están bloqueados con problemas en el control de versiones. Este dato se puede obtener de los equipos de soporte de DevOps / Tools. (Vriesman, 2017).

- Tiempo dedicado a problemas con integración continua (TPCI)

  Es el tiempo en que los procesos están bloqueados debido a problemas con el sistema de integración. Este dato se puede obtener de los equipos de soporte de DevOps / Tools. (Vriesman, 2017)

- Número de incumplimiento en las prácticas de DevOps (NIDevOps)

  Para este indicador se deben de tener definidas políticas y procedimientos de DevSecOps, como por ejemplo, los procesos de integración continua, los procesos de liberación y despliegue, etc. Este indicador se obtiene de revisar que tanto esas políticas y procedimientos se cumplen. (Vriesman, 2017)

- Tiempo de Merge (TM)

  En la integración continua se espera llevar a cabo las integraciones de una manera continua y temprana, por lo que esta medida nos indica cuanto tiempo tarda una característica creada en un feature branch en estar integrada en el main branch. (Vriesman, 2017)

Realizar Pruebas

Conjunto de procesos destinados a evaluar y determinar la integridad y calidad del software así como el cumplimiento con los requisitos regulatorios, funcionales, no funcionales,de negocio y de uso.

- Indice de pruebas automatizadas (IPA)

  La intención es medir si realmente se están ejecutando pruebas automatizadas. Por ejemplo, si las pruebas automatizadas se deben de ejecutar todas las noches durante el proceso de construcción/integración continua debemos esperar 30 ejecuciones por mes. (Vriesman, 2017)

  $$IPA = \frac{Número\ de\ ejecuciones}{Periodo}$$

- Número de Bugs por Severidad (NBS)

  Bug de Calidad

  Indica el número de defectos tanto funcionales como no funcionales clasificados por su criticidad, por ejemplo, alta, media, baja e información. El número de defectos deseado es 0 en todas sus severidades.

  Bugs de Seguridad

  Indica el número de defectos de seguridad clasificados por su criticidad, por ejemplo, alta, media, baja e información. El número de defectos deseado es 0 en todas sus severidades.

- Número de Defectos por Líneas de Código (NDLC)

  Muestra la calidad de defectos tanto de calidad como de seguridad por líneas de código.

  $$Densidad = \frac{Numero\ de\ Defectos}{lineas\ \ de\ Código}$$

- Cobertura de pruebas unitarias (CPU)

  Es la cantidad de código que las pruebas unitarias cubren. Se debe de incluir el número total de casos, tomando en cuenta los casos de abuso y de uso. El número de casos de

abuso debe de ser por lo menos igual a los casos de uso. Esto como resultado del modelado de amenazas.

$$CPU = \frac{Métodos\ cubiertos\ en\ los\ casos\ de\ uso}{Total\ de\ Metodos}$$

Modelado de amenazas

Uso de un modelo para identificar y comprender los riesgos y amenazas que puede tener la aplicación basados en su funcionalidad y en las características de su entorno de ejecución.

- ### Porcentaje de amenazas técnicas y no técnicas modeladas (PATNTM)

  Se debe de contar con la lista total de amenazas y el número de estas amenazas que fueron parte del proceso de modelado.

  $$PATNTM = \frac{Amenazas\ modeladas}{Total\ de\ amenazas}$$

- ### Nivel de participación de los stakeholdes (NPS)

  Para tener un modelado de amenazas completo es requerido que todos los stakeholders participen, incluyendo los desarrolladores, arquitectos, dueños del producto, analistas (business analyst), gerentes, ejecutivos, etc. Este índice mide el nivel de participación. Por lo menos deben de participar 3 equipos, los desarrolladores, arquitectos, y dueños del producto.

  $$NPS = \frac{Número\ de\ equipos\ que\ participaron}{Número\ total\ de\ equipos}$$

- ### Porcentaje de artefactos modelados (PAM)

  Indica la cantidad de elementos identificados que han sido clasificados. Esto permite saber el nivel de visibilidad que se tiene de los activos más preciados.

  $$PAM = \frac{Numero\ de\ elementos\ clasificados}{Total\ de\ elementos\ indentificados}$$

- ### Numero de desvios (NDProd)

  Es la cantidad de hallazgos en producción que no fueron modelados y que no fueron descubiertos en las fases tempranas de desarrollo y construcción.

- ### Ejecución regular de modelado de amenazas (ERMA)

  No existe una medición estándar de qué tan frecuente se debe de llevar a cabo, un modelado de amenazas se puede considerar como una sesión de diseño con el equipo. Muchos equipos llevan a cabo esta tarea cada sprint y otros menos maduros la llevan a

cabo después de varios sprints.También depende de lo que se está desarrollado, pudiera haber equipos con un perfil de riesgo que justifiquen el modelado de amenazas con cada sprint. Se deben incluir hallazgos, amenazas, agentes de amenazas, activos y mitigaciones.

Una vez definida la frecuencia, este indicador deberá medir qué tanto se cumple, por ejemplo, si la frecuencia es mensual y en un año solo se llevarán a cabo 6 sesiones estaríamos en el 50%.

Desplegar el software

Es el proceso de entregar el software en un formato listo para consumo.

- ### Tiempo promedio de indisponibilidad del sistema durante la actualización (TPISDA)

  Este indicador está relacionado a la operación y contribuye a valorar su eficiencia. (Vriesman, 2017). Cuando un sistema no está disponible tiene una afectación a su producción, si el valor es alto y adicionalmente se tiene un problema, este tiempo se puede llegar a incrementar exponencialmente y aunado a eso se deben de considerar los reintentos, por lo que se busca que este valor sea lo más pequeño posible. (Vriesman, 2017).

  Por ejemplo, Si el tiempo es de 1 hora y llegara a suceder un problema, este tiempo podría incrementarse exponencialmente aunado a los tiempos de cada reintento.

- ### Índice de despliegue automático de aplicaciones (IDAA)

  Nos indica la magnitud en la que se tienen procesos de despliegue automatizados en comparación al número total de aplicaciones a desplegar. El valor ideal es 1 indicando que todas las aplicaciones se despliegan automáticamente.

  $$IDAA = \frac{Número\ de\ aplicaciones\ que\ se\ despliegan\ automatizadamente}{Total\ de\ aplicaciones\ a\ desplegar}$$

- ### Frecuencia de publicación de imágenes (FPI)

  Se refiere al número de imágenes nuevas / actualizadas publicadas en un período de tiempo determinado (a veces esto no se controla).

  Esta medida debe de compararse con el valor deseado.

- ### Tiempo promedio de despliegue (TPD)

  Es el tiempo promedio que toma desplegar una aplicación. Se mide desde que se envía la solicitud (dando clic en desplegar) o cuando el calendario en un sistema de despliegue como Jenkins lo dispara hasta que la aplicación queda desplegada.

Esta medida debe de compararse con el valor deseado.

Valoración de la seguridad y configuración

Pruebas minuciosas de seguridad basadas en el buen conocimiento de la aplicación y su lógica de negocios.

- Number de vulnerabilidades dinámicas (NVD)

  Número de vulnerabilidades que se encontraron como resultado de llevar a cabo pruebas dinámicas de seguridad y de penetración.

  $$NVD = N\acute{u}mero\ de\ vulnerabilidades\ cr\acute{i}ticas,\ altas,\ medias\ y\ bajas$$

Valoración de riesgo

Es un proceso de identificación, análisis y priorización de riesgos que pueden causar el fracaso o la pérdida del proyecto si ocurren.

- Riesgo comunicado al cliente (RCC)

  Se indica el número de riesgos identificados que han sido mitigados o retenidos (aceptados) y han sido documentados y comunicados al cliente.

- Identificación de reglamentos y políticas (IRP)

  Se deben de identificar todas las políticas y reglamentos que la aplicación deben de cumplir, muchas veces en base a la ubicación de sus usuarios o de la misma aplicación así como con referencia a los datos que esta maneja.

  Algunas políticas que se deben de considerar:

  - Políticas de privacidad
  - Manejo de Cookies
  - Definir términos y condiciones de uso
  - Cumplir con las regulaciones y estándares de la industria donde aplique
  - Guardar registros de consentimiento por el uso de datos (de aplicar)
  - Registros de cómo se procesan los datos
  - Seguir cambios en la regulación y estándares

- Nivel de madurez en seguridad (NMS)

  Una mediación genérica que debe de considerar los hallazgos por herramientas automatizadas junto con la valoración de un experto en la que se evalúan los siguientes puntos:

  - Confidencialidad de datos
  - Integridad de datos
  - Disponibilidad de los datos
  - Logica de negocio
  - Arquitectura

  Los valores pueden ir del 1 al 3 o al 5.

- Percepción de los clientes en cuanto a seguridad (PCCS)

  Que valoración del 1 al 3 o al 5 le daría un usuario a la aplicación en base a su experiencia como usuario. ¿Se sentiría seguro?

Capacitación y crecimiento

Conocimiento Disponible

Proveer de toda la información que los empleados necesitan cuando la necesitan.

- Participación en foros, comunidades o similares (PFCS)

  Número de preguntas realizadas en foros, comunidades o similares y que son respondidas.

- Fuentes de información externa a la organización (FIEO)

  Porcentaje de información que se busca en fuentes externas a la organización. Se busca minimizar este número lo más posible.

Compartir conocimiento

Crear un ambiente donde la experiencia y el conocimiento es compartido entre los empleados

- Compartir información (CInfo)

  Es la manera más eficiente y fácil de distribuir información entre los miembros de un equipo. Para medir este dato se pueden incluir métricas dentro de las valoraciones o revisiones de desempeño de cada miembro así como también la cantidad de artículos publicados en las plataformas de información.

- Ratio de adopción (RA)

  Es un índice que nos indica que tanta información se está reutilizando o compartiendo.

$$RA = \frac{numero\ de\ veces\ que\ se\ compartio\ información}{Total\ de\ oportunidades\ que\ el\ equipo\ tuvo\ para\ compartir\ información}$$

Referencias

Alexa Internet Ranking. https://www.alexa.com/plans

Becher, J. (2016, August 11). *Time is Now Money: Why Time-to-Market Beats ROI as the New Indicator of Profitability*. Forbes. https://www.forbes.com/sites/sap/2016/08/11/time-is-now-money-why-time-to-market-beats-roi-as-the-new-indicator-of-profitability/

CircleCI. (n.d.). *Three Critical Development Metrics for Engineering*. Retrieved June 29, 2020, from https://circleci.com/resources/velocity-report/

Hoxmeier, J.A., & Dicesare, C. (2000). System Response Time and User Satisfaction: An Experimental Study of Browser-based Applications.

Osbourn, T. (2020, January 22). *6 Essential KPIs for Software Development Teams*. TextExpander. https://textexpander.com/blog/kpis-software-development-teams/

OWASP. (2013, November). *Application Security Guide For CISOs*. https://owasp.org/www-pdf-archive/Owasp-ciso-guide.pdf

Smith, T. (2018, February 14). *Three Critical Metrics for Engineering Velocity*. Dzone.Com. https://dzone.com/articles/three-critical-metrics-for-engineering-velocity

Vriesman, D. (2017, 09 20). Constructing a DevOps Tactical View using Balanced Scorecard. Linkedin. Retrieved 10 18, 2020, from https://www.linkedin.com/pulse/constructing-devops-tactical-view-using-balanced-denny-vriesman/