KEEP CALMS AND DEV SEC OPS



Toda historia tiene un inicio

DEVSECOPS Argentina

by Cloud Legion

¿Que veremos?

- GLOSARIO
- INTRODUCCIÓN A CALMS
- CALMS Y DEVSECOPS
- CALMS MATURITY MODEL
- ANTI PATRONES
- CONCLUSIONES
- AUTORES
- LINKS Y REFERENCIAS

Imágenes creadas por dooder

Glosario



Imágenes creadas por dooder

Glosario

Agile Organization: Una empresa flexible capaz de una respuesta rápida y adaptabilidad a las oportunidades y amenazas esperadas e inesperadas.

Agile Manifesto: La proclamación formal de valores y principios para guiar un enfoque iterativo y centrado en las personas para el desarrollo de software.

Agile Project Management: Un método iterativo e incremental de diseño y desarrollo de software en el que los desarrolladores trabajan en estrecha colaboración con los usuarios utilizando la información suficiente para comenzar a planificar y ejecutar.

Antifragile: Un término acuñado por el profesor Nassim Nicholas Taleb sobre una propiedad que permite a los sistemas aumentar en capacidad o rendimiento como resultado de estrés, errores, fallas o fallas.

Automation: La tecnología mediante la cual se realiza un proceso o procedimiento sin intervención manual. En DevOps, la automatización permite la creación de informes en tiempo real, integrando diversas herramientas utilizadas por diferentes partes interesadas y flujos de trabajo, integrando tecnología para reunir herramientas de diferentes dominios y descomponer los silos.

Agent: Un pequeño programa que se ejecuta en varias máquinas para controlar o informar del estado de cada uno. Es un proceso que se ejecuta en un servidor de destino como un usuario específico. La implementación de un agente requiere que mantenga las credenciales en ese sistema para las conexiones de script o cualquier otra conectividad. El agente ejecuta acciones de implementación como si estuviera en el equipo porque el agente lo está.

Agile Methodology: Una metodología de entrega de software que implica la iteración continua del desarrollo y las pruebas de software con un enfoque en la calidad del software y los comentarios de los usuarios. Cada iteración en caja de tiempo del ciclo de desarrollo continuo en Agile se conoce como Sprint. Cada Sprint en el desarrollo ágil debe dar lugar a un producto operativo para que cualquier cambio en los requisitos se pueda ajustar fácilmente, proporcionando flexibilidad y creatividad dentro de los equipos de desarrollo de software ágiles.

Glosario

Bottleneck (Lean): Un paso en un proceso que limita la capacidad total del proceso o sistema.

BDD: Behavior-driven development is an agile software development methodology where any application is documented and designed in accordance with the behavior a user expects to experience when interacting with the application. This encourages collaboration and teamwork among the quality analysts, developers, stakeholders, and any other business participants for a given project

Build: Una versión particular del código del programa de aplicación, a menudo conocido como la etapa de los nuevos desarrollos de características en el software.

Build Agent: A kind of agent used in the CI process, which can either be installed on a local system or remotely in relation to the CI server. A build agent sends and receives messages on handling various software builds.

CALMS Model: La esencia de DevOps: Cultura, Automatización, Lean, Medición, Compartir. Es un marco que se puede utilizar para evaluar la preparación de una organización para adoptar un proceso DevOps.

ChatOps: el uso de clientes de chat, chatbots y herramientas de comunicación en tiempo real para facilitar cómo se comunican y ejecutan las tareas de desarrollo y operación de software.

Constraints (Theory of): Una metodología para identificar el factor limitante más importante que se interpone en el camino para lograr un objetivo y luego mejorar sistemáticamente esa restricción hasta que ya no sea el factor limitante.

Containers: Un contenedor de desarrollo de software agrupa las aplicaciones y sus dependencias, lo que permite crear aplicaciones en las que los entornos de prueba y en vivo se pueden replicar fielmente.

Constraint: Una limitación o restricción en un sistema.

Glosario

Configuration Management: La administración de la configuración es un método automatizado para mantener la coherencia en todas las configuraciones del entorno en el que se hospeda la aplicación de software. En DevOps, las configuraciones se agrupan en forma de scripts o código que se controlan a través de la herramienta de control de versiones.

Commit: El proceso de insertar el código en un repositorio de código como Git y realizar un seguimiento de los cambios en el repositorio de código con un mensaje de registro que mejor describa los cambios realizados en el código.

Continuous Delivery: Una metodología que se centra en garantizar que el software esté siempre en un estado listo para la versión a lo largo de su ciclo de vida. El proceso de implementación se convierte en iterativo, lo que proporciona versiones más frecuentes al usuario final.

Continuous Integration: Una práctica de desarrollo de software que requiere que los desarrolladores combinen su código en un sistema de control de versiones compartido. El objetivo es localizar y solucionar los errores de software de una manera más oportuna, y reducir el tiempo que se tarda en publicar actualizaciones de software.

Culture: La totalidad de las ideas, valores, creencias, prácticas y comportamientos que son compartidos por los empleados en una organización. Es una actitud de responsabilidad compartida en un entorno de DevOps.

Definition of Done: En el desarrollo de software, una comprensión compartida de lo que significa para que el trabajo sea completo.

Deployment: El lanzamiento de actualizaciones de software para los usuarios. En entornos de DevOps, la implementación está totalmente automatizada para que los usuarios reciban actualizaciones tan pronto como se escriban y prueben.

DevOps: es una filosofía, una cultura que revoluciona el modo en que se gestiona el ciclo de desarrollo software

DevSecOps: acrónimo inglés de la unificación de development (desarrollo), Security (Seguridad) y operations (operaciones)

Glosario

Desarrollo Ágil: El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

Fail Fast: Una filosofía en la que se implementa una nueva versión del código, se produce un error, rápida y rápida. Se proporcionan comentarios y se adapta en consecuencia. Fallar rápido básicamente le anima a fallar rápido y temprano en lugar de posponer el error o trabajar con el fracaso. Fail fast philosophy tiene como objetivo reducir las pérdidas cuando las pruebas revelan que algo no está funcionando como se esperaba, y los desarrolladores pueden probar rápidamente otra cosa, a menudo conocido como un concepto de pivote.

Flow: Cómo las personas o los productos se mueven a través de un proceso. DevOps "First Way" se ocupa de optimizar el flujo a través de los sistemas

Gemba: Palabra japonesa para "el verdadero lugar". En los negocios a menudo equivale a donde se crea el valor.

IAC: A veces denominada "infraestructura programable", la infraestructura como código (IaC) trata la configuración de la infraestructura exactamente como el software de programación.

Iterations: Un único ciclo de desarrollo, normalmente medido como una semana o dos semanas.

Kaizen: Una filosofía empresarial japonesa de mejora continua de las prácticas de trabajo, eficiencia personal, etc. El objetivo es encontrar maneras de mejorar en un flujo de valor completo que conduce a mejores resultados de los clientes.

Kanban: Un método visual de control de la actividad que tira del flujo de trabajo a través de un proceso a un ritmo manejable.

Kanban Board: Una herramienta Kanban que ayuda a los equipos a organizar, visualizar y administrar el trabajo.

Glosario

Kata : En japonés, es un condicionamiento cultural o la idea de hacer las cosas de la manera "correcta". Es una forma sistemática de lograr metas y enfrentar desafíos que se pueden usar en toda la organización.

Lean: Una filosofía de producción que se centra en reducir los residuos y mejorar el flujo de procesos para mejorar el valor del cliente.

Lean IT: La aplicación de las ideas clave detrás de la inclinación al desarrollo y funcionamiento de los productos y servicios de TI.

Microservices: Una técnica de desarrollo de software alineada con la arquitectura orientada a servicios (SOA) que estructura una aplicación como una colección de servicios acoplados libremente. En una arquitectura de microservicios, los servicios son aún más detallados.

Muda: Una palabra japonesa que significa "futilidad; inutilidad; despilfarro. Muda es uno de los tres tipos de residuos en el pensamiento de proceso magro.

Mura: Una palabra japonesa que significa "desnivel; irregularidad; falta de uniformidad. Es uno de los tres tipos de residuos en el pensamiento de proceso magro.

Muri: Una palabra japonesa que significa "irrazonable; imposible; más allá del poder o demasiado difícil. Es uno de los tres tipos de residuos en el pensamiento de proceso magro.

Pipeline: es un concepto para evitar el desperdicio en el proceso de desarrollo de software , y se utiliza para proporcionar comentarios rápidos al equipo durante la implementación

Release: Uno o más cambios del sistema que se compilan, prueban e implementan juntos.

SDLC: son las siglas de: Systems Development Life Cycle, también conocido como "System Design Life Cycle" (ciclo vital del desarrollo/diseño de sistemas).

Scrum: Un marco ágil iterativo, con plazos e incrementalpara la realización de proyectos complejos.

Glosario

Serverless: Un modelo de ejecución de computación en la nube en el que el proveedor de nube administra dinámicamente la asignación de recursos de máquina. Los precios se basan en la cantidad real de recursos consumidos por una aplicación. Esto se conoce a menudo como "Función como un servicio."

Test Driven Development: Un proceso de desarrollo de software que se basa en la repetición de un ciclo de desarrollo corto donde los requisitos en forma de casos de prueba se utilizan para mejorar el software.

The Three Ways: Un conjunto de principios desarrollados por Gene Kim, galardonado CTO, autor e investigador, para definir de qué se trata Realmente DevOps.

- **First way:** Acelerar el flujo de trabajo, desde el negocio, pasando por el desarrollo, hasta las operaciones y el cliente.
- **Second way:** Aumente tanto el número de bucles de retroalimentación en su flujo como la rapidez con la que está recibiendo los comentarios.
- **Third way:** Desarrollar y fomentar una cultura donde se fomente la experimentación y el aprendizaje constantes.

Time to Value: Mida el tiempo que tarda el negocio en obtener valor de una característica.

Toolchain: Uso de un conjunto integrado de herramientas específicas de la tarea para automatizar un proceso de extremo a extremo. Por ejemplo, pruebas de código automatizadas, lanzamiento e implementación.

Value Stream Mapping: Herramienta lean que visualiza el flujo de datos, materiales y trabaja a través de un proceso con énfasis en la identificación y cuantificación de residuos.

Waste (Lean): Cualquier cosa que no agregue valor a un producto. (Véase Muda, Muri, Mura.)

Waterfall (Software Development): Enfoque lineal y secuencial para el diseño y desarrollo de software donde el progreso se considera que fluye constantemente hacia abajo.

Work in Progress (WIP): Cualquier trabajo que se haya iniciado pero aún no se ha completado.

Introducción a CALMS



Imágenes creadas por dooder

Introducción a CALMS

DevOps y DevSecOps son palabras de moda. Hay muchos artículos que describen qué son y qué no son. Creo que podemos estar de acuerdo en que son culturas, una forma de trabajo. También estoy seguro de que la mayoría de nosotros tenemos una impresión general de cómo debería ser: desarrollo, operaciones y seguridad trabajando juntos, rompiendo silos, entregando más rápido, automatizando, etc.

En la mayoría de las discusiones que hemos tenido con los profesionales de la industria, una pregunta que surge una y otra vez con respecto a DevSecOps, es "¿Hay un marco para ¿Adopción de DevSecOps?" Ahora hay buenas razones para esta pregunta y una es que muchas personas de operaciones empresariales conocen marcos como ITIL y Cobit. La respuesta a esa pregunta es: **”CALMS”**

Si nunca has oído hablar y no tienes idea de lo que podría ser " CALMS ", debes saber que esta es la línea que separa a un joven "Padawan" de un Maestro Jedi. Muchos ya conocen los beneficios proporcionados por el concepto DevOps y su importancia en el crecimiento sostenible de cualquier negocio, pero pocos logran forzar magistralmente la fuerza, aprovechando todo su potencial.



Imágenes creadas por dooder

Introducción a CALMS

DevOps es mucho más que un simple método para aumentar la velocidad de implementación, monitoreo y automatización de procesos. La adhesión a DevOps es una realidad, y la mayoría de las empresas de TI ya no pueden escapar de ella. Este marco , que a menudo se considera una verdadero cambio de cultura para implementar dentro de las organizaciones y esta alineado con las principales prácticas adoptadas después de la llegada de la transformación digital de la nueva era.

Las empresas que aún no han propuesto considerar unirse se quedarán atrás, aún lidiando con prácticas rígidas e inflexibles que pueden conducir a graves cuellos de botella en su organización.

Dentro de este contexto, aparece CALMS, un modelo que está alineado con la filosofía de DevOps y que puede guiar las mejores prácticas para los administradores de TI. Sin embargo, muchos aún desconocen qué es y cómo puede mejorar los resultados obtenidos con este marco.

En este E-book, mostraremos qué es CALMS, cuáles son sus pilares, sus beneficios y por qué puede ser un aliado importante para su equipo en este momento de transformación. ¡Buena lectura!

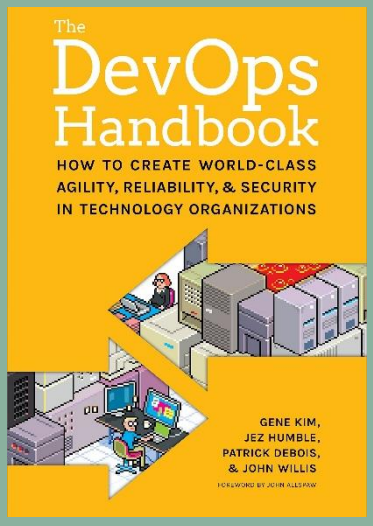


Introducción a CALMS

¿Que es CALMS?

CALMS es un acrónimo en inglés de **Culture, Automation, LeanIT, Measurement and Sharing.** El término apareció entre 2008-2009 de la mano de **John Willis y Damon Edwards** que acuñaron el acrónimo CAMS y en 2010, fue ampliado a CALMS por **Jez Humble** coautor del "DevOps Handbook" , pero otras personas participaron en el proceso de creación del marco, como **Jene Kim**

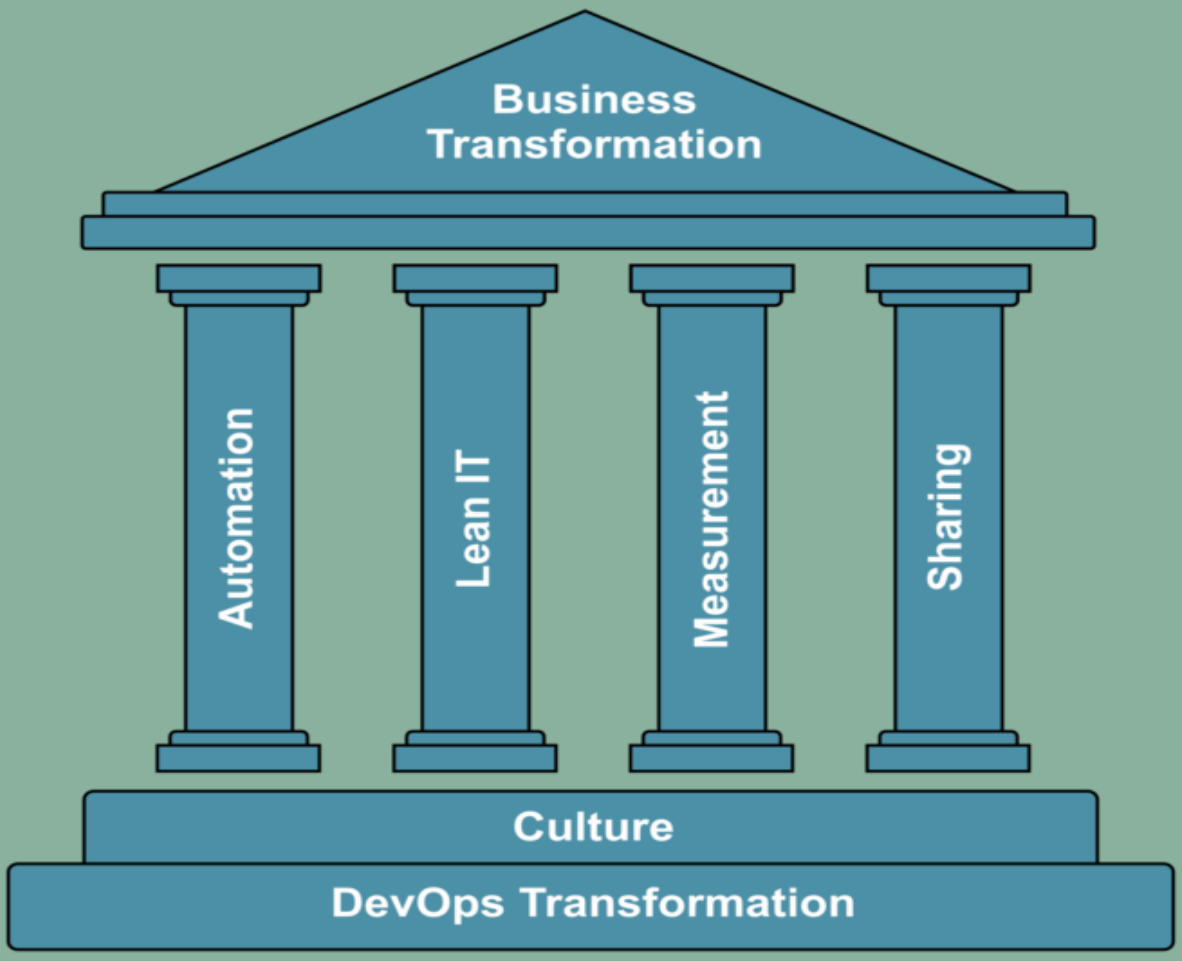
La idea en su concepción era romper la barrera existente entre los Desarrolladores y la Operación, que a menudo tenía gerentes, objetivos, plazos e indicadores totalmente diferentes. Debido a la constante fricción entre los dos sectores, se creó el término DevOps .



Tenga en cuenta que todos los elementos en el acrónimo se correlacionan y es por eso que crean una base tan firme para la transformación deseada por las empresas.

Cada letra de CALMS representa un tema que debe ser trabajado en su organización para implementar una verdadera cultura de DevOps en su empresa. A continuación, haremos una breve presentación sobre el significado de cada letra del acrónimo y cómo se relacionan entre sí.

Introducción a CALMS



El modelo CALMS establece los principios de DevOps

En este modelo, la cultura es la base de cualquier transformación empresarial que respalde los otros principios. Es de vital importancia que un equipo de DevOps adopte el cambio y trabaje para fomentar una cultura de colaboración, apertura y objetivos compartidos.

La **cultura** respalda los otros 4 principios clave para garantizar que DevOps sea exitoso.

Introducción a CALMS

La **automatización** es un 'pilar' significativo de DevOps, asegurando que la retroalimentación casi instantánea se pueda recopilar a través de procesos automatizados. **Lean IT** se trata de inculcar una mentalidad ágil en el equipo y sus procesos, asegurando que el enfoque siempre esté en producir valor para el usuario final mediante un enfoque lean para el pensamiento y los requisitos del sistema.

Este pensamiento Lean se extiende al siguiente principio; **Medición**. Es importante en cualquier entorno de entrega continua que los equipos no pasen tiempo midiendo KPI extraños. Manténgalo simple y mida las cosas correctas: esas estadísticas que lo ayudarán a guiarlo al valor del cliente y le permitirán encontrar y corregir fallas rápidamente. El principio final de este modelo es una aplicación práctica de la base cultural: asegúrese de **colaborar y compartir** experiencias y aprendizajes. Asumir la responsabilidad de compartir objetivos.

Si desea leer más sobre DevOps, dos libros que le recomendaría que agregue a su lista de lectura son **“The Goal: A Process of Ongoing Improvement"** de **Eliyahu M. Goldratt** (una mirada más antigua y menos centrada en mejora continua de TI), y " **The Phoenix Project** " de **Gene Kim, Kevin Behr y George Spafford** (más reciente y centrada en TI sobre DevOps).

Introducción a CALMS

CULTURE (CULTURA) 1/2

Quizás sea una de las letras más importantes y, en consecuencia, las letras más conocidas de CALMS. Esto se debe a que la importancia de que DevOps no sea una herramienta o una metodología siempre se propaga, sino una verdadera cultura , y el acrónimo que comienza con "C" no es para nada, lo que demuestra la importancia de este concepto.

Pero ciertamente en algunas ocasiones han tratado de vender la idea de implementación de una metodología o herramienta milagrosa, pero si analizamos no hay “H" o "T" de Herramientas/tools en el acrónimo "CALMS".

La cultura impregna cuestiones fundamentales, como las relaciones interpersonales de los miembros de los equipos de Desarrollo y Operaciones (Operaciones incluye SecOps), una premisa básica de DevOps.



Es por eso que es imperativo que la implementación de esta cultura perpetúe la forma en que las personas se relacionan entre sí en todos los sectores de TI , qué canales usan para comunicarse, el uso de metodologías ágiles en sus rutinas, entre otros.



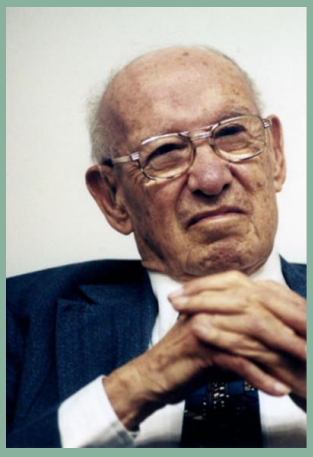


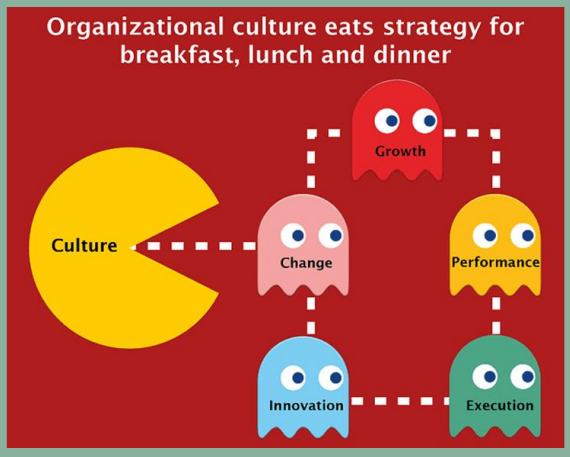
Imágenes creadas por dooder

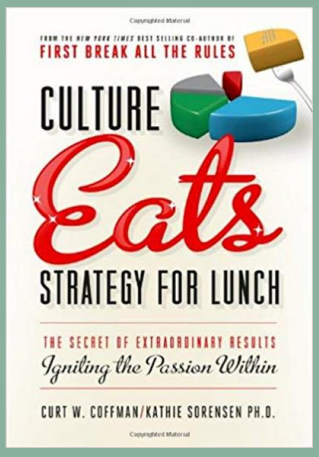
Introducción a CALMS

CULTURE (CULTURA) 2/2

Estas cuestiones están intrínsecamente presentes en la cultura organizacional centrada en DevOps y es uno de los puntos que necesita más atención de los gerentes de TI. Cuando esta cultura es frágil o no está alineada adecuadamente con los intereses de la organización, interfiere directamente con la calidad de los resultados de la institución en su conjunto.







"La cultura come la estrategia en la cena"

Una cita de Peter Drucker y hecha famosa por Mark Fields, presidente de Ford, es algo que suena muy cierto hoy en día. Es algo que todos "sentimos" que es verdad, no solo por las estadísticas que lo confirman, sino porque intuitivamente sabemos que sin una verdadera alineación de sus miembros, una organización no es nada.

Introducción a CALMS

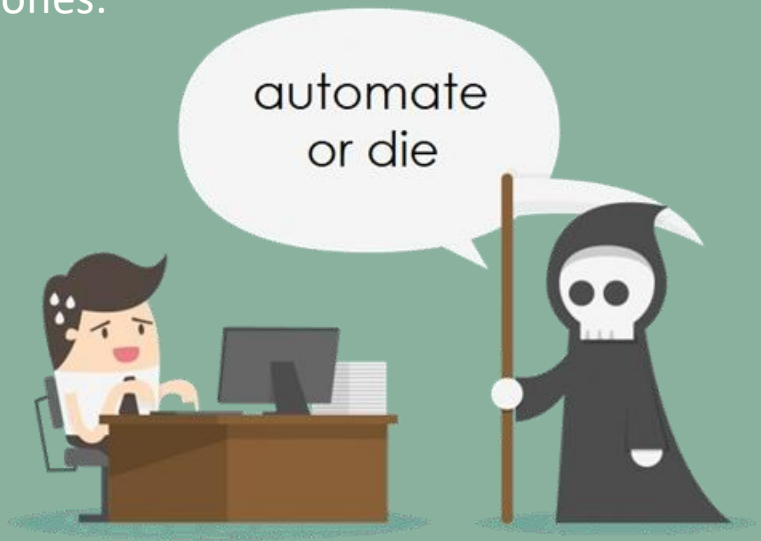
AUTOMATION (AUTOMATIZACIÓN) 1/2

La automatización es un punto fundamental no solo para DevOps, sino también para las empresas de TI que desean alinearse con el momento de la transformación digital. Por lo tanto, no sería diferente que el acrónimo CALMS incluya este término.

El objetivo es poder automatizar las tareas internas, especialmente aquellas etapas del flujo de creación, minimizando los errores humanos que pueden causar problemas y pérdidas tan perjudiciales para su negocio.

La automatización también realiza la documentación de cada proceso, también ayuda a estandarizar los pasos.

Es extremadamente ventajoso para los equipos de TI, no solo por la reducción de los errores humanos. Después de todo, la mano de obra cuesta tiempo y dinero. Por lo tanto, es mejor dejar que los sistemas automatizados realicen estas funciones.



Imágenes creadas por dooder

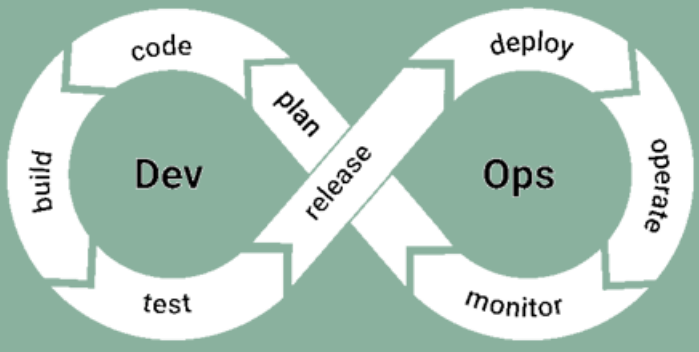
Introducción a CALMS

AUTOMATION (AUTOMATIZACIÓN) 2/2

Por ejemplo, los desarrolladores pueden pasar el 75% de su tiempo buscando errores y problemas de rendimiento manualmente, utilizando registros de clientes e informes que a menudo son demasiado vagos debido a la ignorancia técnica. Automáticamente, este proceso se realiza mucho más rápido, minimizando las pérdidas generadas.

En resumen buscamos automatizar completamente las etapas del flujo de creación, evitando errores humanos, documentando y estandarizando cada paso del proceso, aquí es donde entran las tuberías (pipelines).

Pipeline es la palabra del momento, sin embargo, cualquier otra automatización que ayude en este proceso también es bienvenida.



Este paso debe hacerse con cuidado y atención por parte de los gerentes. Esto se debe a que, si se hace con brechas y cuellos de botella, en lugar de ayudar, puede dañar el proceso de desarrollo por completo, brindando soluciones con defectos y problemas al cliente.

Introducción a CALMS

LEAN IT 1/2

Una cultura bien aplicada y sistemas de automatización eficientes no son suficientes para una implementación de DevOps madura. Después de todo, si todavía hay cuellos de botella en los procesos de Desarrollo y Operaciones, la tendencia es que se perpetúen con el tiempo.

Para minimizar esto, es esencial tener LeanIT y es por eso que forma parte de CALMS.

Los 5 principios clave de Lean están bien documentados y el movimiento Lean IT busca extender estos conceptos al mundo de la tecnología de la información y, en particular, al desarrollo de software.

Principios LeanIT				
Definir el valor	**Identificar el flujo de valor**	**Crear flujo**	**Establecer el Pull**	**Persigue la perfección**

Los textos clave son Lean Software Development de Tom & Mary Poppendieck y Lean Enterprise de Humble, Molesky y O'Reilly.

Introducción a CALMS

LEAN IT 2/2

Las metodologías Lean pueden diseñar de manera más eficiente las etapas de entrega para cada área de la organización. A partir de esto, es posible analizar, identificar y volver a dibujar los puntos que pueden estar causando grandes cuellos de botella y tasas de reprocesamiento.

Muchos recursos aplicados en las empresas son solo extras, que realmente hacen poca o ninguna diferencia en la experiencia del cliente con el tiempo. Por lo tanto, la aplicación de la metodología Lean es fundamental para que usted invierta solo en lo que, de hecho, puede ser el diferencial para su negocio.

Sobre la base de esta identificación y las modificaciones propuestas, es posible optimizar el flujo de entrega, con mayor velocidad y eficiencia, dos pilares fundamentales para el éxito del negocio de TI en la actualidad.

Con los cuellos de botella identificados, podemos optimizar el flujo, brindando más velocidad y mayor eficiencia.



Introducción a CALMS

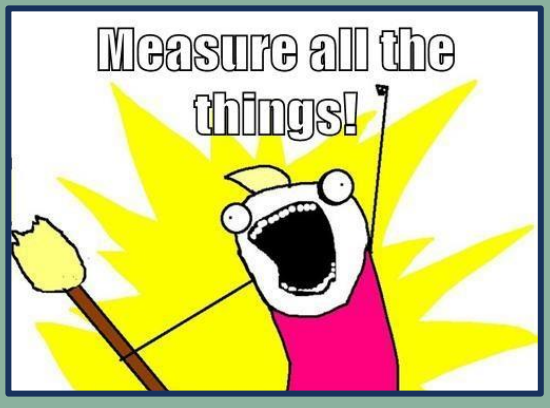
MEASUREMENT (MÉTRICA/MEDICIÓN) 1/3

¿Cómo saber si las medidas anteriores tienen, de hecho, los efectos necesarios o esperados? Sin el paso de medición, esto se vuelve prácticamente imposible. Una cultura organizacional bien aplicada, con estrategias de automatización y reducción de cuellos de botella, puede no ser suficiente.

Ciertas herramientas y técnicas pueden no ser lo que su empresa necesita, por lo que es necesario reformular los procesos. Pero, ¿cómo se puede reconocer esto? Ocurre a través de la medición.

Si bien es un truismo "lo que se mide se hace" (ver el gurú de la gestión **Tom Peters**) DevOps busca llevar esto al siguiente nivel. El modelo Build-Measure-Learn del libro de Eric Ries " **Lean Startup** " identifica el ciclo clave y, sin medición, nunca se puede aprender ni mejorar.

Entonces "medir todas las cosas" es el grito de guerra (y un gran meme).



Introducción a CALMS

MEASUREMENT (MÉTRICA/MEDICIÓN) 2/3

Este paso también es fundamental para la generación de feedback, ya sean positivos o negativos. Después de todo, ¿cómo van a identificar sus equipos que están en el camino correcto si no saben cómo señalarles cómo se está haciendo el trabajo con éxito?

La medición también permite generar conocimiento sobre la organización, así como generar previsibilidad sobre problemas, contingencias y accidentes que pueden ocurrir, identificando sus posibles causas y los primeros signos antes de que sucedan.



Introducción a CALMS

MEASUREMENT (MÉTRICA/MEDICIÓN) 3/3

De esta forma, los gerentes pueden salvaguardarse a sí mismos, evitando fallas o incluso creando planes de contingencia para reducir las pérdidas.

Con registros y paneles en la mano, los gerentes tienen suficiente información para poder analizar posibles fallas y generar mejoras constantes para sus negocios.

Estas mediciones deben realizarse en todos los niveles de la empresa, incluso las reglas de la empresa, de modo que se puedan realizar los cambios necesarios para tener una mayor eficiencia y mejores resultados en todas las áreas.



Imágenes creadas por dooder

Introducción a CALMS

SHARING (COMPARTIENDO) 1/2

La información organizacional ya no puede centrarse en los gerentes de cada departamento, especialmente cuando se implementa una cultura DevOps, que fomenta la integración entre los equipos.

El conocimiento, cuando se comparte, puede ayudar a los equipos a trabajar de una manera más organizada e integrada, evitando que los procesos dependan de otros departamentos y, principalmente, de ciertos empleados.

Imagine, por ejemplo, en un entorno de información centralizada, que solo un empleado es responsable de una determinada etapa del proyecto y solo él tiene la información sobre lo que estaba haciendo. En cierto punto, antes de que se complete el trabajo, decide abandonar la organización y lleva consigo información sobre la situación.

En vista de este escenario, el funcionamiento del flujo de desarrollo se ve muy afectado. Hasta que se resuelva el problema, podría resultar en una pérdida de tiempo que podría gastarse en optimizar el desarrollo de la solución.



Introducción a CALMS

SHARING (COMPARTIENDO) 2/2

En un modelo CALMS, los responsables de Desarrollo y Operaciones deben trabajar juntos para descubrir mejores soluciones, detectar y corregir errores antes de que causen problemas reales para el negocio.

Con el intercambio de información, si una persona abandona el proceso, no interfiere con el progreso del proyecto, ni interfiere con el funcionamiento del flujo, por lo que se vuelve autosostenible.

Del mismo modo, el intercambio de datos es fundamental para nivelar el conocimiento de los equipos, de modo que todos puedan colaborar, de la misma manera, en las etapas de desarrollo y operaciones.



Como puede ver, todos los pasos mencionados anteriormente se comunican, de modo que se vuelven fundamentales para crear un entorno DevOps sostenible, dentro de los parámetros de transformación digital y una cultura organizacional orientada hacia la adhesión de DevOps.

Introducción a CALMS

IMPORTANCIA DE CALMS 1/2

CALMS es responsable de guiar la adopción de DevOps en las organizaciones de TI, con el fin de identificar si es, de hecho, el mejor momento para esto. Si se identifica que todavía es necesario realizar algunos cambios, también ayuda a identificar qué se debe cambiar, para que el proceso se adopte de la manera más eficiente posible.

Además, CALMS también permite la supervisión de errores. Esto significa que los equipos tienen las herramientas y la autonomía para verificar las soluciones creadas constantemente y descubrir fallas y problemas de rendimiento incluso antes del informe del cliente, promoviendo un enfoque de mejora continua que puede deleitar a su audiencia con el tiempo.



CALMS, aplicado con plataformas de monitoreo de errores, es capaz de discernir qué puntos necesitan mejoras, proporciona un diagnóstico preciso de los errores existentes, de modo que los gerentes puedan identificar cuáles son los puntos de mejora y hacer los cambios necesarios. lo antes posible.

Imágenes creadas por dooder

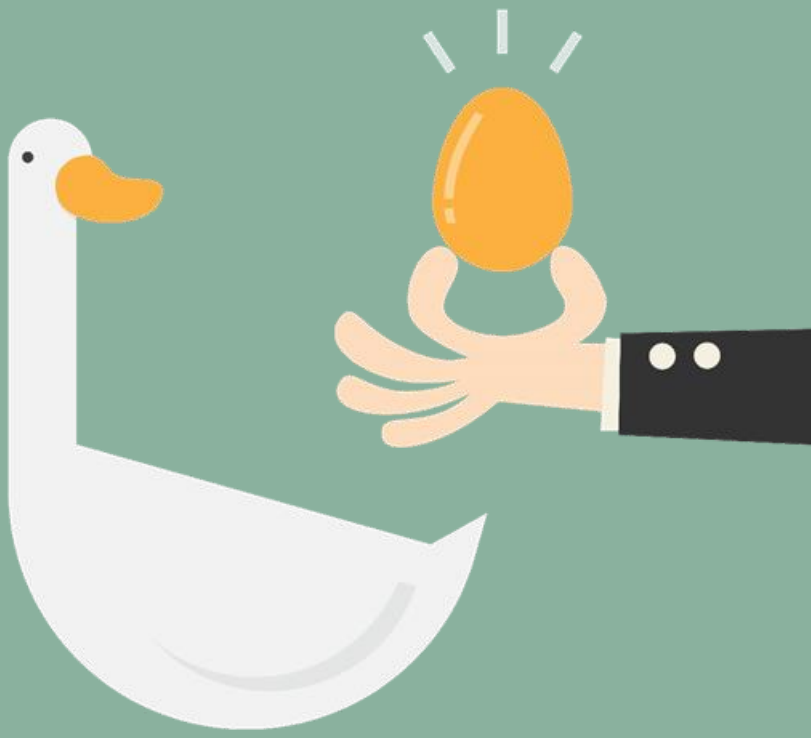
Introducción a CALMS

IMPORTANCIA DE CALMS 2/2

El modelo permite evaluar dónde el monitoreo de errores es más útil para los líderes y desarrolladores en un contexto DevOps.

También permite una adopción más responsable de DevOps, minimizando posibles fallas de implementación. Después de todo, si el marco ya está implementado con errores a priori, toda la ejecución de las herramientas posteriores también fallará.

Es por eso que el modelo CALMS es fundamental. Guía a los gerentes para identificar cuáles son las medidas que deben tomarse, exactamente qué puntos de mejora y cuánto necesita madurar la empresa para poder adoptar DevOps internamente.



Imágenes creadas por dooder

Introducción a CALMS

BENEFICIOS DE USAR CALMS 1/4

Pero, ¿por qué usar el modelo CALMS para implementar DevOps a expensas de otros modelos que también se pueden usar? ¿Cuáles son tus diferenciales? ¿Por qué es tan importante? Estos son algunos de los puntos que muestran la ventaja de CALMS en DevOps.

Es una forma de identificar la madurez de su empresa en el momento actual.

¿Cómo sabes dónde está tu empresa ahora? Después de todo, para que se aplique DevOps, es esencial que su negocio esté preparado para esto. Y el modelo CALMS es la mejor opción para esta evaluación.

Veremos un modelo de madurez mas adelante.



Imágenes creadas por dooder

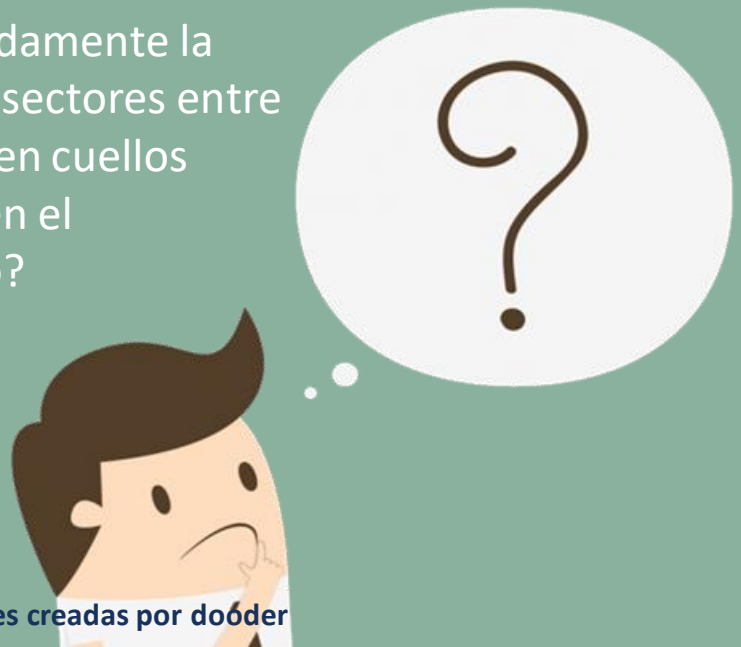
Introducción a CALMS

BENEFICIOS DE USAR CALMS 2/4

Le permite identificar y responder los siguientes puntos:

- ¿La cultura organizacional actual se centra en las prácticas de DevOps?
- ¿Es necesario implementar una mayor integración entre los equipos antes de comenzar a adoptar DevOps?
- ¿Mi empresa prioriza la automatización de procesos? Si no, ¿qué está evitando esto?
- ¿Mi empresa utiliza una política para reducir los procesos internos o todavía hay muchos procedimientos y estructuras superfluas que pueden hacer que sea más costoso y causar cuellos de botella en los procesos?
- ¿Estoy usando las métricas correctas para medir los resultados?

- ¿Estoy usando un exceso de métricas para evaluar los resultados?
- ¿Se comparte adecuadamente la información entre los sectores entre los empleados o existen cuellos de botella que impiden el flujo de conocimiento?



Imágenes creadas por dooder

Introducción a CALMS

BENEFICIOS DE USAR CALMS 3/4

Estas preguntas permiten a los gerentes responsables identificar lo que aún debe modificarse para poder comenzar a implementar DevOps en la empresa. De lo contrario, la adhesión puede incluso ser dañina.

Es por eso que el modelo CALMS es fundamental: guía a los gerentes para observar estos problemas y minimizar posibles fallas que podrían dañar los resultados de su negocio.

Proporciona la estructura necesaria para la aplicación de DevOps:

Aún relacionado con el elemento anterior, el modelo CALMS puede señalar las pautas que los gerentes deben tener en cuenta al elegir DevOps. Por lo tanto, podrá ver estos 5 pilares correctamente, observando sus detalles y sabiendo exactamente qué invertir en su negocio.



Imágenes creadas por dooder

Introducción a CALMS

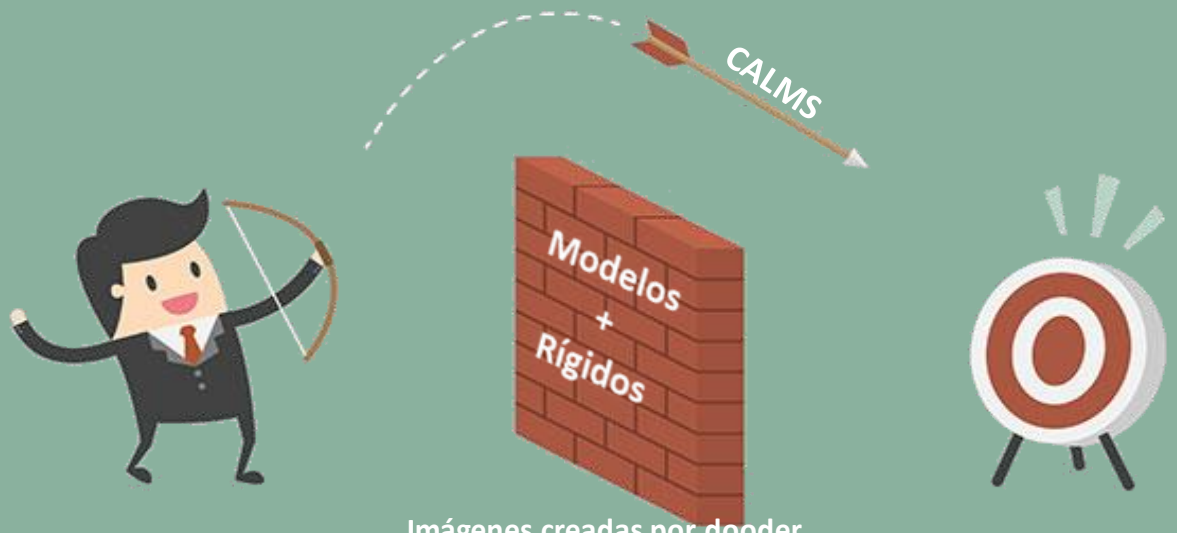
BENEFICIOS DE USAR CALMS 4/4

Muestra cómo los pilares son intercambiables.

Todos los pilares del modelo CALMS se comunican entre sí. Esto muestra la fuerza del modelo para ser aplicable en una estrategia DevOps. Por ejemplo, la adopción de estrategias de automatización debe estar dentro de una cultura organizacional fuerte y, además, implementarse dentro de una lógica Lean IT. De lo contrario, será solo otro procedimiento superfluo y costoso para su negocio.

Es una alternativa a los modelos más rígidos.

El modelo CALMS, por ejemplo, puede ser una alternativa interesante a otros modelos más rígidos y, en consecuencia, no están tan alineados con DevOps. Por ejemplo, es una alternativa a ITSM (Information Technology Service Management), un modelo de enfoque diseñado para diseñar, entregar, administrar y mejorar la forma en que se aplica la TI en una organización, siendo un modelo más cercano a ITIL .



Imágenes creadas por dooder

Introducción a CALMS

LA ADHESIÓN DE SOURCING A CALMS 1/2

Muchos especialistas en el área de DevOps creen que, hoy en día, es necesario adherirse a una nueva letra en el acrónimo CALMS, que se convertiría en CALMSS, con la adhesión de **Sourcing (outsourcing).**

El outsourcing se ha adoptado cada vez más en las empresas de TI como una forma, incluso, de reducir el personal y las rutinas, manteniendo ciertos procesos en manos de especialistas, que estarán en mejores condiciones para realizar algunas actividades cruciales para su negocio.

Por ejemplo, si las soluciones desarrolladas por su empresa necesitan expertos en Seguridad de la Información, una opción puede ser externalizar este servicio a una empresa que esencialmente trabaja con este punto.

Por lo tanto, permite que sus empleados trabajen con lo que realmente dominan, minimizando las fallas en este sector tan vital y estratégico para las soluciones de TI (seguridad de datos). Mientras tanto, la empresa subcontratada tiene la libertad de implementar las mejores herramientas destinadas a este propósito, minimizando los riesgos de cibercrimen y la fuga de información. Todos ganan con eso.



Introducción a CALMS

LA ADHESIÓN DE SOURCING A CALMS 2/2

Los consultores también entran en esta lista de la contratación externa, que puede ser de suma importancia para reducir las fallas y los cuellos de botella internos y que los administradores no están identificando.

Ante este escenario, las empresas que implementan DevOps han adoptado cada vez más el abastecimiento como una conducta estratégica para el negocio, con buenos resultados en este sentido. Otros

CALMS en DevOps puede representar una verdadera revolución para su negocio, por lo que puede reducir posibles cuellos de botella operativos, trabajar de manera eficiente dentro de una metodología Lean y ofrecer subsidios para una cultura organizacional sólida, automatizaciones que, de hecho, son útiles para el negocio, mediciones de métricas importantes e intercambio de información internamente.



Imágenes creadas por dooder

CALMS y DevSecOps



Imágenes creadas por dooder

CALMS y DevSecOps

DevSecOps es el principio de que todos los equipos de tecnología son responsables de la seguridad cibernética en una organización: la propiedad no está únicamente en la puerta de los profesionales y equipos de seguridad. La idea de que la ciberseguridad es el trabajo de todos surgió en parte porque las habilidades de ciberseguridad están limitadas, dentro del mercado en su conjunto y dentro de una organización específicamente. Un informe reciente de (ISC)2 afirma que hay una **escasez de personal de ciberseguridad global de tres millones** y que esto está aumentando. Vemos la misma tendencia con las organizaciones con las que trabajamos en Argentina y Latinoamérica.

Esta restricción se manifiesta a través de:

- La seguridad se considera (y a menudo puede ser) un bloqueador
- Prácticas laborales dolorosas que resultan en conflictos cotidianos.
- Traspasos, demoras y costos innecesarios para entregar valor
- Incumplimientos que cuestan dinero y daños a la reputación.



Imágenes creadas por dooder

CALMS y DevSecOps

La restricción se ve exacerbada por el diseño organizacional tradicional que tiene la seguridad como un equipo separado, a veces dos equipos separados entre ellos mismos. “Seguridad informática, de la información, física, etc…” Si bien sabemos que hay muchas consideraciones de seguridad física o perimetrales, las amenazas cibernéticas son variadas, crecientes y nadie lo negaría, absolutamente real. ¿Los equipos de seguridad tradicionales están preparados para la transformación digital?

Una de las primeras preguntas que solemos hacer en las organizaciones cuando empezamos a trabajar con ellas es sobre su diseño organizacional: tener equipos de seguridad separados crea tensión **(“la seguridad dice que no, no y no”),** traspasos y demoras en el proceso.



DevSecOps busca abordar estos desafíos, y una manera útil de desglosar cómo hacerlo, es usar el CALMS acrónimo de DevOps bien establecido como lo vimos anteriormente en este mismo e-book.

Imágenes creadas por dooder

CALMS y DevSecOps

CULTURE (CULTURA) Y DEVSECOPS 1/5

Una cultura de DevOps/DevSecOps se define principalmente, en mi opinión, por la alta confianza. Cuando tenemos altos niveles de confianza, hay bajos niveles de fricción y el trabajo puede fluir a alta velocidad, por lo que cuesta menos entregar valor. Es difícil hablar de cultura porque se siente como algo nebuloso, incluso más difícil para nosotros en tecnología hablar de eso, ya que estamos más entrenados para hablar en bits y bytes que en emociones y sentimientos. Creo que ayuda pensar en el comportamiento, como la clave de la cultura.



No solo porque eso nos da narrativa (y la narración de historias es una forma muy humana de aprendizaje), sino también porque podemos enfocarnos en cómo cambiar el comportamiento y eso nos da un enfoque práctico cuando queremos hacer cambios.

Para fomentar la confianza, necesitamos crear un lugar de seguridad psicológica, un lugar donde las personas puedan ser sinceras y experimentales sin temor a las consecuencias, especialmente si experimentan un fracaso.



Imágenes creadas por dooder

CALMS y DevSecOps

CULTURE (CULTURA) Y DEVSECOPS

Muchos de los líderes de hoy, particularmente de corporaciones más grandes, luchan con esta tolerancia al fracaso. Particularmente cuando su trabajo contiene infraestructura crítica nacional o internacional, como una red bancaria, es entendible que estos líderes provengan de un lugar donde ven una falla como un fracaso catastrófico. Pero hay muchos matices de falla: desde un defecto en una rama de desarrollo detectado durante la integración continua hasta un sistema inactivo durante horas para todos los usuarios globales.

Lo que también es entendible es lo que sucede en nuestro cerebro cuando escuchamos la palabra **'fracaso'.** La sociedad nos enseña que el fracaso es malo y que somos castigados por ello emocionalmente por sentimientos de vergüenza, bochorno y potencialmente también físicamente por ser rechazados o peor. Por eso a menudo tememos inherentemente al fracaso



El miedo causa actividad subcortical en la amígdala, que a su vez activa la red de memoria en el lóbulo frontal (donde ocurre la atención consciente), lo que dificulta el aprendizaje ya que la ansiedad es una distracción.

Imágenes creadas por dooder

CALMS y DevSecOps

CULTURE (CULTURA) Y DEVSECOPS

Cuando estamos en una situación, como estamos con seguridad, donde el conocimiento, las habilidades y la experiencia son una limitación y parte de la solución es mejorar el aprendizaje, no queremos una cultura de la culpa, donde las personas tienen miedo, ya que esto inhibirá el aprendizaje.

Además, vemos que muchas organizaciones empresariales tienen sistemas frágiles (lo que significa que pasan gran parte de su tiempo en la lucha contra incendios) y no tienen control sobre el flujo de trabajo (lo que significa que tienen demasiado trabajo en progreso y no tienen suficiente tiempo para hacer mejoras o realizar retrospectivas efectivas y actuadas). Toda esta presión, y un diseño organizativo aislado también a menudo da como resultado la definición de que alguien ha terminado siendo "hice mi trabajo".



Imágenes creadas por dooder

CALMS y DevSecOps

CULTURE (CULTURA) Y DEVSECOPS

Cuando hablamos de los comportamientos, queremos cambiarlos para evolucionar a una **'cultura DevSecOps',** de lo que estamos hablando es de personas que:

- Confiar entre nosotros y su proceso
- Participa activamente en el proceso y lo comprende de principio a fin
- Ten coraje y no temes a la culpa y al castigo
- Asumir la responsabilidad personal.
- Son reflexivos y actúan sobre la reflexión.
- Disfruta el aprendizaje



Imágenes creadas por dooder

CALMS y DevSecOps

CULTURE (CULTURA) Y DEVSECOPS

Entonces, sabemos lo que tenemos, y sabemos a qué queremos aspirar, aquí están algunos consejos sobre cosas prácticas que podemos hacer para influir en el cambio que desea ver:

"Mal" comportamiento	"Buen" comportamiento	Actividades para evolucionar el comportamiento
Desconfiar del otro	Confiar en el otro	Capacitar a los líderes para ser entrenadores, brindar plataformas colaborativas (físicas y virtuales), visibilizar el trabajo mediante atrasos y tableros Kanban (vea la próxima publicación de la parte 3 de esta serie en línea para obtener más información).
Desconfiar del proceso	Confiar en el proceso	Utilizar técnicas como el "Value Stream Mapping" (vea la próxima publicación de la parte 3 de esta serie en línea para obtener más información) para obtener una comprensión compartida del sistema de principio a fin.
Miedo/Cobardía	Coraje/Valentía	No juzgar los errores o los experimentos fallidos, atraer un ambiente de seguridad psicológica y sistémica donde la gente pueda ser valiente y el valor sea premiado.
"Hice mi trabajo"	Responsable/Sensato	Tener en cuenta los fracasos como un aprendizaje e informarse con los demás sobre el descubrimiento y la solución - no ajustar más controles. Usar el "Value Stream Mapping" como un camino hacia la responsabilidad compartida.
Reactivo/Imprudente	Reflexivo/Pensativo	No terminar el incidente hasta que el experimento haya sido completado con una hipotesis exitosa para su solución. Varias organizaciones me cuentan que tienen tiempo exclusivamente para resolver el problema, pero no para reflexionar sobre él.
Estático/Inactivo/Invariable	Aprendiz/Emprendedor/Dinámico/Activo	Ayudar a las personas a abandonar los comportamientos existentes y utilizar la capacitación y el aprendizaje mediante los experimentos para fomentar el dominio del saber. Colaborar con las personas para que vean la incomprensión, la neuroplasticidad, la práctica y la automaticidad.

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

En el **The DevOps Handbook**, **Christopher Little** diciendo:

"La automatización es para DevOps como los telescopios son para la astronomía".

Nos encanta esta frase y la usaremos para la siguiente analogía.

- **Las estrellas:** son los clientes que estamos mirando o los resultados de valor que queremos ofrecerles.
- **Los astrónomos:** somos nosotros, los practicantes de DevOps.
- **El planeta:** es nuestra organización
- **Los otros planetas:** son nuestros socios y competidores.
- **Los meteoritos:** son amenazas e incidentes con los que tenemos que lidiar.
- **Los telescopios:** son esenciales para permitirnos ver, recopilar datos y comentarios y acelerar la entrega del cambio.



Imágenes creadas por dooder

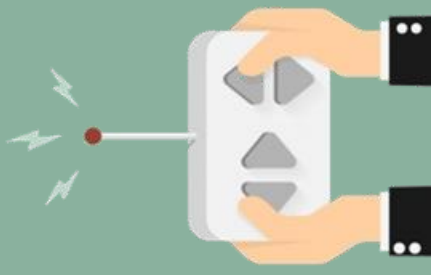
CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Pero no podemos ni debemos intentar automatizar todo en la vida. Creo que pasará bastante tiempo hasta que podamos ir a tumbarnos a la playa mientras las máquinas hacen todo el trabajo.

Pero definitivamente podemos automatizar para dejar de hacer trabajos repetitivos y aburridos y romper las restricciones que vemos en nuestro trabajo: **la seguridad, que es una de las restricciones clave**. Tengo algunas herramientas de seguridad favoritas que he aprendido con mis clientes y recomiendo regularmente que nos ayuden a ser más DevSecOps, ya sea aliviando la presión sobre el lado de la estabilidad (IT Ops) o permitiéndonos garantizar que las consideraciones de seguridad se tengan en cuenta antes. para no ralentizar el rendimiento (desde el desarrollo).





Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Todos ellos deben considerarse en el contexto de la entrega continua (CD), es decir, la práctica de tener siempre el software en un estado liberable. En realidad, me gusta llevar cosas más allá de CD y cadenas de herramientas de arquitectura con mis clientes que optimizan el flujo de trabajo desde la idea hasta la realización; es decir, las cadenas de herramientas de DevOps que registran la idea lo antes posible, la rastrean a través de cada paso del ciclo de vida de entrega continua, incluida la liberación y la operación, y la retroalimentación sobre su valor en el registro de ideas (o la cartera de pedidos si lo prefiere). Yo llamo a esto una cadena de herramientas DevOps o DevOps Loop .

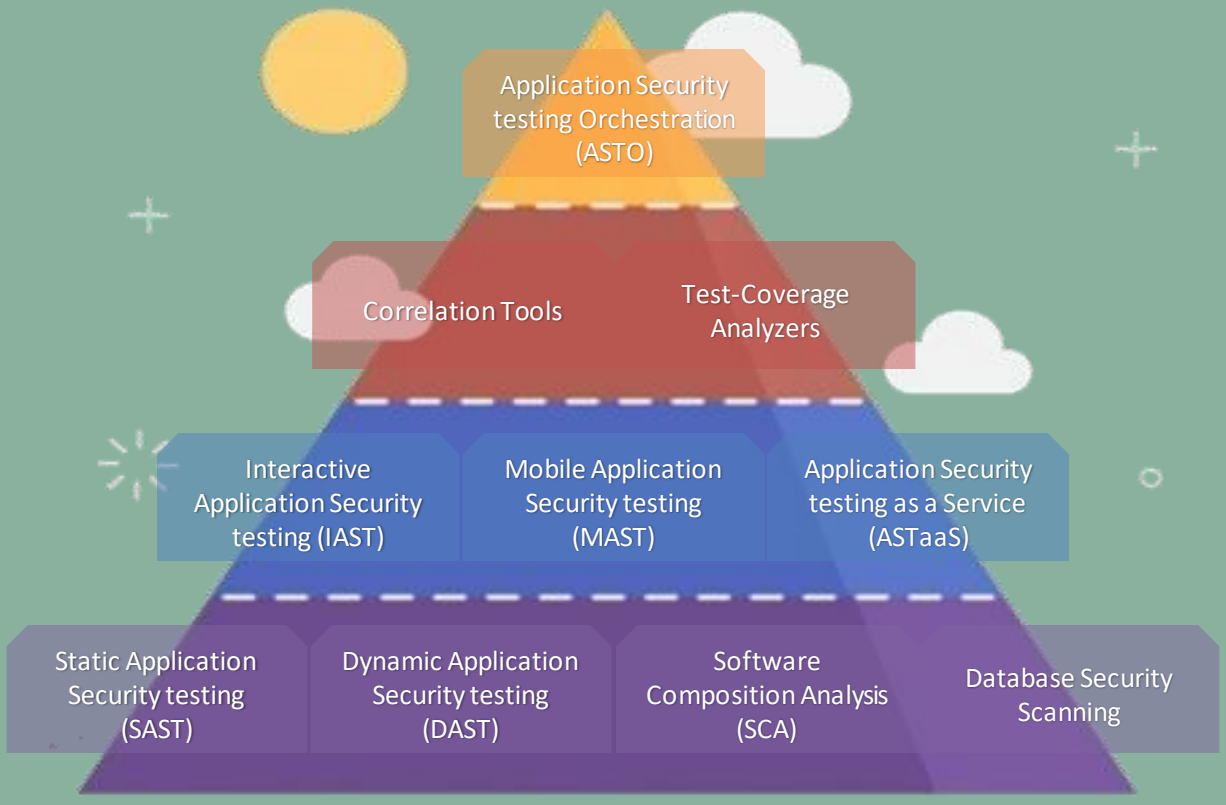


Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Application Security testing tools



Este gráfico representa clases o categorías de Application Security testing tools . Los límites se difuminan a veces, ya que determinados productos podrían realizar elementos de varias categorías, pero son aproximadamente las clases de herramientas dentro de este dominio. Hay una jerarquía áspera en que las herramientas en la parte inferior de la pirámide son fundacional y como la competencia se gana con ellos, las organizaciones podrían buscar utilizar algunos de los métodos más progresivos más altos en la pirámide

CALMS y DevSecOps

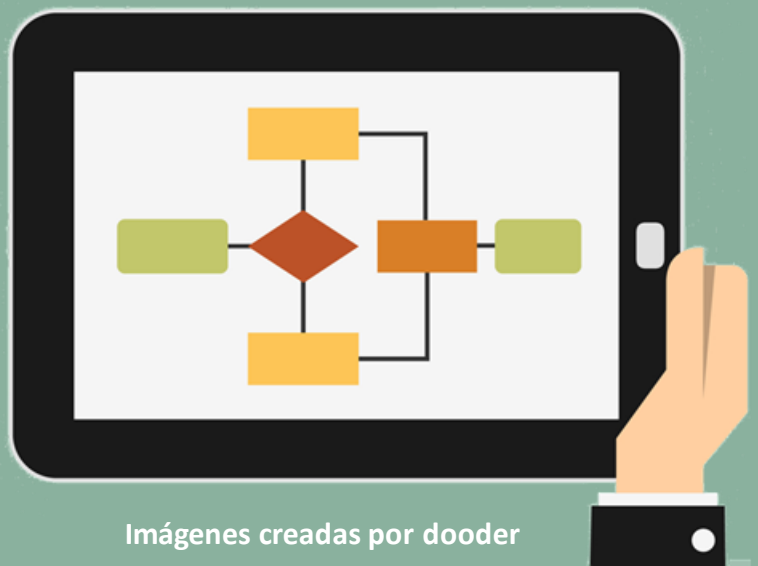
AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Static Application Security Testing (SAST)

Las herramientas de la categoría SAST se conocen como herramientas de testeos de sombrero blanco o de la caja blanca, donde el evaluador conoce información sobre el sistema o software que se está probando, incluyendo un diagrama de arquitectura, acceso al código fuente, etc.

Las Herramientas SAST examinan el código fuente (en reposo o estático cono se suele nombrar) para detectar y reportar debilidades que podrían convertirse en vulnerabilidades seguridad.

Los analizadores de código fuente podrían ejecutarse en código no compilado para comprobar defectos como errores numéricos, validación de entrada, condiciones, path traversals, punteros y referencias, etc. Los analizadores de código binario y de byte hacen lo mismo en el código compilado. Algunas herramientas se ejecutan únicamente en código fuente, algunas solo en código compilado y otras en ambas.



Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Dynamic Application Security Testing (DAST)

A diferencia de las herramientas SAST, las herramientas DAST se podrían denominar como herramientas de testeos Black-Hat o Black-Box, donde el probador no tiene conocimiento previo del sistema. Detectan condiciones que indiquen una vulnerabilidad de seguridad en una aplicación en su estado de ejecución.

Las herramientas DAST se ejecutan en código operativo para detectar problemas con interfaces, solicitudes, respuestas, scripts (es decir, Javascript), inyección de datos, sesiones, autenticación y más.

Las herramientas DAST emplean técnicas de Fuzzing: lanzar casos de testeos conocidos, no válidos e inesperados en una aplicación, as veces en gran volumen.



Imágenes creadas por dooder

CALMS y DevSecOps



AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Software Composition Analysis (SCA)

Los procesos de gobernanza de software que dependen de la inspección manual son propensos al fracaso. Las herramientas SCA examinan el software para determinar los orígenes de todos los componentes y librerias dentro del software. Estas herramientas son altamente efectivas para identificar y encontrar vulnerabilidades en componentes comunes y populares, particularmente en componentes de código abierto. Sin embargo, no suelen detectar vulnerabilidades para los componentes desarrollados a medida por la propia empresa. Funcionan comparando los módulos conocidos que se encuentran en el código con una lista de vulnerabilidades conocidas. Las herramientas SCA encuentran componentes que tienen vulnerabilidades conocidas y documentadas, aconsejan si los componentes están desactualizados o tienen parches disponibles. Para hacer esta comparación, casi todas las herramientas SCA utilizan la Base de datos de vulnerabilidad nacional del NIST scannea y exposiciones comunes (CVEs) como fuente de vulnerabilidades conocidas. Muchos productos SCA comerciales también utilizan el VulnDB base de datos de vulnerabilidad comercial como fuente, así como algunas otras fuentes públicas y propietarias.



Las herramientas SCA podrían ejecutarse en código fuente, código de byte, código binario o alguna combinación.

Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Database Security Scanning

El gusano SQL Slammer del 2003 explotó una vulnerabilidad conocida en un sistema de administración de bases de datos que tuvo un parche lanzado más de un año antes del ataque. Aunque las bases de datos no siempre se consideran parte de una aplicación, los desarrolladores de aplicaciones a menudo dependen en gran medida de la base de datos, y las aplicaciones pueden afectar mucho a las bases de datos. Las herramientas de escaneo de seguridad de la base de datos verifican parches y versiones actualizadas, contraseñas débiles, errores de configuración, lista de control de acceso (ACL) y más.

Algunas herramientas pueden extraer registros que buscan patrones o acciones irregulares, como acciones administrativas excesivas.



Los escáneres de bases de datos generalmente se ejecutan en los datos estáticos que están en reposo mientras el sistema de administración de bases de datos está en funcionamiento. Algunos escáneres pueden monitorear los datos que están en tránsito

Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Interactive Application Security Testing (IAST)

Los enfoques híbridos han estado disponibles durante mucho tiempo, pero más recientemente se han categorizado y discutido usando el término IAST. Las herramientas IAST utilizan una combinación de técnicas de análisis estático y dinámico. Pueden probar si las vulnerabilidades conocidas en el código son realmente explotables en la aplicación en ejecución.

Las herramientas IAST utilizan el conocimiento del flujo de la aplicación y el flujo de datos para crear escenarios avanzados de ataque y utilizan los resultados del análisis dinámico de forma recursiva: a medida que se realiza un análisis dinámico, la herramienta aprenderá cosas sobre la aplicación en función de cómo responde a los casos de prueba. Algunas herramientas utilizarán este conocimiento para crear casos de prueba adicionales, que luego podrían generar más conocimiento para más casos de prueba y así sucesivamente. Las herramientas IAST son expertas en reducir el número de falsos positivos, y funcionan bien en entornos Agile y DevOps, donde las herramientas tradicionales independientes DAST y SAST pueden requerir demasiado tiempo para el ciclo de desarrollo.



Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Mobile Application Security Testing (MAST)

El Open Web Application Security Project (OWASP) enumeró los 10 principales riesgos móviles en 2016 y 2017 tales como:

- M1: Improper Platform Usage
- M2: Insecure Data Storage
- M3: Insecure Communication
- M4: Insecure Authentication
- M5: Insufficient Cryptography
- M6: Insecure Authorization
- M7: Client Code Quality
- M8: Code Tampering
- M9: Reverse Engineering
- M10: Extraneous Functionality



Las herramientas MAST son una combinación de análisis estático, dinámico y forense. Realizan algunas de las mismas funciones que los analizadores estáticos y dinámicos tradicionales, pero permiten que el código móvil se ejecute también en muchos de esos analizadores. Las herramientas MAST tienen características especializadas que se centran en problemas específicos de las aplicaciones móviles, como el jailbreak o el rooting del dispositivo, las conexiones WI-FI falsas, el manejo y la validación de certificados, la prevención de fugas de datos y más.

Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Application Security Testing as a Service (ASTaaS)

Como su nombre lo indica, con ASTaaS, le paga a alguien para que realice pruebas de seguridad en su aplicación. El servicio generalmente será una combinación de análisis estático y dinámico, pruebas de penetración, pruebas de interfaces de programación de aplicaciones (API), evaluaciones de riesgo y más. ASTaaS puede usarse en aplicaciones tradicionales, especialmente aplicaciones móviles y web.

El impulso para el uso de ASTaaS proviene del uso de aplicaciones en la nube ,donde los recursos para las pruebas son más fáciles de reunir. Se proyecta que el gasto mundial en computación en nube pública de $67B en 2015 a $162B en 2020.



Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Correlation Tools

Tratar con falsos positivos es un gran problema en las pruebas de seguridad de aplicaciones. Las herramientas de correlación pueden ayudar a reducir parte del ruido al proporcionar un depósito central para los hallazgos de otras herramientas AST.

Las diferentes herramientas de AST tendrán diferentes hallazgos, por lo que las herramientas de correlación analizan los resultados de diferentes herramientas de AST y ayudan con la validación y la priorización de los hallazgos, incluidos los flujos de trabajo de remediación. Mientras que algunas herramientas de correlación incluyen escáneres de códigos, son útiles principalmente para importar resultados de otras herramientas.



Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Test-Coverage Analyzers

Los Test-coverage analyzers miden cuánto del código total del programa ha sido analizado. Los resultados se pueden presentar en términos statement coverage (porcentaje de líneas de código probado) o branch coverage (porcentaje de rutas disponibles probadas).

Para aplicaciones grandes, los niveles de cobertura aceptables pueden determinarse por adelantado y luego compararse con los resultados producidos por los analizadores de cobertura de prueba para acelerar el proceso de prueba y liberación. Estas herramientas también pueden detectar si determinadas líneas de código o ramas de lógica no pueden alcanzarse durante la ejecución del programa, lo que es ineficiente y una posible preocupación de seguridad.

Algunas herramientas SAST incorporan esta funcionalidad en sus productos, pero también existen productos independientes



Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Application Security Testing Orchestration (ASTO)

ASTO integra herramientas de seguridad a lo largo de un ciclo de vida de desarrollo de software (SDLC).

Si bien el término ASTO fue recientemente acuñado por Gartner ya que este es un campo emergente, hay herramientas que ya han estado haciendo ASTO, principalmente aquellas creadas por proveedores de herramientas de correlación. La idea de ASTO es tener una administración centralizada y coordinada y la generación de informes de todas las diferentes herramientas de AST que se ejecutan en un ecosistema.



Todavía es demasiado pronto para saber si el término y las líneas de productos durarán, pero a medida que las pruebas automáticas se vuelven más omnipresentes, ASTO satisface una necesidad.

Imágenes creadas por dooder

CALMS y DevSecOps

AUTOMATION (AUTOMATIZACIÓN) Y DEVSECOPS

Hay muchos factores a considerar cuando se selecciona entre estos diferentes tipos de herramientas AST. Si se está preguntando cómo comenzar, la decisión más importante que tomará es comenzar a usar las herramientas. (Si queres conocer mas sobre las AST, como seleccionar y el mejor uso para cada una de ellas no dejes de leer nuestro el E-book anterior:



Imágenes creadas por dooder

CALMS y DevSecOps

LEAN IT Y DEVSECOPS

En la introducción a LeanIT vimos que el foco esta en optimizar su flujo valor desde la idea hasta la realización, un marco de conectividad que elimina el dolor de escribir y administrar integraciones y visualizar sus tiempos de plomo y ciclo.

Existen varias herramientas de DevSecOps que pueden ayudar con la implementación de “LEAN” una de ellas es TaskTop. Estas tools ayudan en la optimización del flujo de la idea a la realización, un marco de conectividad que le quita el dolor de escribir y gestionar las integraciones y visualizar los tiempos de espera y los ciclos. nos muestra dónde están los cuellos de botella

Los que formamos DevSecOps Argentina realizamos una gran cantidad de mapeo de flujo de valor con nuestros clientes como parte de su viaje; eso nos brinda métricas increíblemente poderosas, pero difiere de las típicas herramientas como TaskTop, en que no está impulsado por datos de los sistemas sino de las cabezas de las personas. No es que esto sea necesariamente bueno o malo: estoy firmemente en el campo de una combinación de hombre y máquina que funciona mejor en el clima comercial que tenemos hoy.

“Los seres humanos todavía necesitan interpretar las tendencias y, en su mayoría, decidir cómo actuar sobre la base de lo que los datos les dicen. Y a menudo los datos aún no están disponibles.”

CALMS y DevSecOps

LEAN IT Y DEVSECOPS

Recientemente, en un ejercicio de mapeo de flujo de valor, facilitamos que el equipo asignara un tiempo de espera de ocho semanas para la actividad de seguridad en su flujo de valor debido a las limitaciones de recursos y la separación entre los equipos. Fue el mayor retraso con respecto al ciclo general, por lo que una de las áreas clave que abordamos. Hay dos formas principales de abordar esto cuando lo vemos en una secuencia de valor:

Hombre: Un patrón que funcionó realmente bien en muchos de nuestros clientes es colocar a los profesionales de seguridad en los equipos autónomos o en los equipos de productos / características por un período de tiempo (generalmente alrededor de 3 meses).

Máquina: cuando vimos la A en CALMS, Automatización y cómo desplazarme a la izquierda integrando herramientas desde el principio en la tubería para controles de seguridad y también cómo proporcionar operaciones como servicio



Imágenes creadas por dooder

CALMS y DevSecOps

LEAN IT Y DEVSECOPS

Otra herramienta Lean para pensar en el contexto de DevOps es **Kanban** y la capacidad, como con Value Stream Mapping, de colaborar visualmente en el flujo de trabajo. Estas herramientas son tan poderosas en parte porque tienen que ver con la visibilidad, y la visibilidad genera confianza que es, como aprendimos cuando vimos Cultura, fundamental para fomentar un entorno DevOps.

Kanban puede ayudar a mostrar dónde están los bloqueadores y dónde tenemos demasiado trabajo en progreso también. Esto es fundamental para impulsar conversaciones sobre mejoras.

Hablando de mejora, alentamos a todas las organizaciones a ser experimentales para impulsar la innovación y reducir el riesgo.

El uso del "**kata**" de mejora de Lean es clave para que esto se vuelva habitual: separarse de una cultura de reuniones y planificación, hacia pequeñas mejoras incrementales frecuentes.



Imágenes creadas por dooder

CALMS y DevSecOps

LEAN IT Y DEVSECOPS

Las personas que se han entrenado en artes marciales están familiarizadas con el **kata**: se trata de practicar el mismo patrón repetidamente hasta que alcanza la automaticidad en el cerebro y realmente es un hábito o "la forma en que hacemos las cosas por aquí". Comenzamos considerando la visión o dirección a largo plazo, consideramos el estado actual y decidimos nuestro próximo estado objetivo. Luego hacemos **PDCA** entre los dos estados: planificamos, luego hacemos, luego verificamos y actuamos en un ciclo continuo.

Esta idea de mejora incremental y continua también se refleja en DevOps **Kaizen** , un modelo que nos impulsa a alentar a nuestros clientes a pensar en un viaje de DevOps en términos evolutivos en lugar de transformacionales, alcanzando un estado general de mejora más rápido.

Dado que el impacto de los pequeños cambios en la productividad es menor y el sistema organizacional tiene tiempo para recuperarse más rápido y hacer otro pequeño cambio más rápidamente.



CALMS y DevSecOps

LEAN IT Y DEVSECOPS

Sin embargo, mientras mejoramos el sistema, todavía lo estamos ejecutando. John Allspaw dijo una vez:

"Un incidente es una inversión no planificada, y si no lo ve así como un líder, no obtendrá un retorno de la inversión que ya se realizó en su nombre".

Sin embargo, a menudo no nos sentimos así, particularmente con las brechas de seguridad. Pero esto nos lleva a otro concepto Lean: el Cordón de Andón y su relación con una cultura organizacional de alta confianza. Agradecer a las personas por una oportunidad de aprendizaje y trabajar juntos para lograr una mejora ayuda a construir nuestra postura de seguridad y eliminar el desperdicio de nuestro flujo, y elimina un enfoque de culpa y castigo, generando confianza.

En esencia, lean se preocupa por identificar y eliminar los desechos de la línea de producción, por lo que la métrica que más comúnmente asociamos es el tiempo de ciclo. Es súper efectivo al mostrarnos cuándo nuestro modelo de seguridad está actuando como un cuello de botella, pero ¿qué otras medidas son útiles en el mundo DevSecOps? Contestaré esto en la próxima publicación de blog de esta serie: CALMS for DevSecOps: Part Four: Measurement.

CALMS y DevSecOps

LEAN IT Y DEVSECOPS

¿Cómo Lean mejora el rendimiento?

Se trata fundamentalmente de crear una cultura de aprendizaje. Esto está en contraste con la naturaleza prescriptiva y muy planificada del pasado en seguridad.

En lugar de tratar de reunir todos los requisitos y comprender todos los casos de uso, y luego decirle a la gente qué hacer, aprendemos a medida que avanzamos.

DevSecOps ya nace tomando la cultura de aprendizaje, invitando explícitamente a seguridad al juego de ofrecer valor al cliente más rápido.

En la era del "Time to Market" la seguridad ya no puede mantenerse en secreto y ser vista como una barrera para la entrega, sino que debe adoptar la automatización, el aprendizaje, la medición, el intercambio y aprender a convertirse en un acelerador.



CALMS y DevSecOps

LEAN IT Y DEVSECOPS

Necesitas una cultura que estimule y recompense a los empleados a pensar como **intra-emprendedores.**

Imagina que Sony no le hubiera dejado a Ken Kutaragi trabajar en su idea...

O que Google no dejara que Paul Buchheit trabajara en su idea...

En 1975, un empleado de Kodak inventó la cámara digital. Sus jefes le hicieron esconderla.



CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Ante pudimos observar DevSecOps desde la óptica de Lean y pudimos concluir que las métricas son clave para Lean, en el contexto del flujo, es el tiempo del ciclo. Esencialmente, queremos resultados de valor en las manos de nuestros clientes lo más rápido posible y este es el elemento clave de un caso de negocios de DevOps y, al mismo tiempo, no compromete la integridad del sistema: rendimiento Y estabilidad.

En el patrón DevSecOps queremos seguridad incrustada en el flujo de la cadena de valor. Eso significa:

- Los requisitos de seguridad (NFR) están integrados en la cartera de pedidos de desarrollo
- No se pierde tiempo esperando las aprobaciones de seguridad o las pruebas durante el desarrollo
- Las pruebas de seguridad están integradas en las canalizaciones de CI / CD
- Las pruebas de seguridad no requieren supervisión ni espera
- Las acciones de seguridad de operaciones de TI están disponibles como autoservicio gobernado para desarrolladores



Imágenes creadas por dooder

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Seguir este patrón significa que la seguridad no representa desperdicio en el flujo de valor, por lo que el tiempo de ciclo se optimiza desde esta perspectiva. ¿Pero qué otras métricas son de interés para DevSecOps? **Peter Drucker** dijo: **"Si no puedes medirlo, no puedes mejorarlo".** y si DevOps no es otra cosa, es un conjunto de patrones de formas de trabajo que mejoran el desempeño organizacional. Entonces, ¿cómo medimos la eficacia de DevSecOps y la entrega de valor? ¿Cómo determinamos cuándo DevSecOps ya no es una 'cosa' ?

Existen innumerables métricas posibles en una plataforma DevSecOps. La decisión de qué métricas seguir se basa en gran medida en las necesidades del negocio y en los requisitos de cumplimiento.

Para facilitar esta visión podemos verlas en dos grupos:

Métricas de alto valor

Métricas de apoyo



Imágenes creadas por dooder

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Métricas de alto valor

Estas métricas deben implementarse primero en una nueva plataforma. No todas las plataformas tendrán estas métricas disponibles de inmediato, pero un entorno completamente maduro generalmente tendrá todas estas métricas disponibles.

Métrica	Descripción
Frecuencia de implementación	Número de implementaciones en producción en un período de tiempo determinado
Cambiar el tiempo de entrega (para aplicaciones)	Tiempo entre una confirmación de código y la implementación de producción de ese código
Cambiar volumen (para aplicaciones)	Número de historias de usuario implementadas en un período de tiempo determinado
Cambiar tasa de falla	Porcentaje de despliegues de producción que fallaron
Tiempo medio de recuperación (MTTR) (para aplicaciones)	Tiempo entre un despliegue de producción fallido hasta la restauración completa de las operaciones de producción.
Disponibilidad	Cantidad de tiempo de actividad / tiempo de inactividad en un período de tiempo determinado, de acuerdo con el SLA
Volumen de problemas del cliente	Número de problemas reportados por los clientes en un período de tiempo determinado
Tiempo de resolución de problemas del cliente	Tiempo medio para resolver un problema informado por el cliente
Tiempo para valorar	Tiempo entre una solicitud de función (creación de la historia del usuario) y la realización del valor comercial de esa función
Hora de ATO	Tiempo entre el comienzo de Sprint 0 para lograr un ATO
Hora de parchear vulnerabilidades	Tiempo entre la identificación de una vulnerabilidad en la plataforma o aplicación y la implementación exitosa de la producción de un parche

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Métricas de apoyo

Estas métricas proporcionan información útil y se recomienda que las métricas en esta categoría se implementen después de que se hayan implementado las métricas de alto valor.

Métrica	Descripción
Prueba de cobertura	Porcentaje de código cubierto por pruebas automatizadas
Cambiar tipos	Porcentaje de características frente a correcciones frente a parches de seguridad
Tiempo de disponibilidad de información del evento.	Tiempo desde un evento hasta información sobre el evento que está disponible para el equipo DevSecOps o los usuarios finales
Incorporación de desarrolladores	Tiempo desde que un desarrollador se une al equipo hasta la capacidad de confirmar el código para la implementación de producción
Cambiar tiempo de resolución	Tiempo entre una propuesta de cambio y el cierre (implementación o rechazo)
Cambiar el tiempo de entrega (para la plataforma DevSecOps)	Tiempo entre un cambio (por ejemplo, confirmación de código) y la implementación de la plataforma de ese cambio
Cambiar volumen (para la plataforma DevSecOps)	Número de historias de usuario implementadas en un período de tiempo determinado
Cambiar la tasa de falla (para la plataforma DevSecOps)	Porcentaje de despliegues de plataforma que fallaron
Tiempo medio de recuperación (MTTR) (para la plataforma DevSecOps)	Tiempo desde un despliegue fallido de la plataforma hasta la restauración completa de las operaciones de la plataforma
Cambiar el tiempo de entrega de las imágenes	Tiempo desde la identificación de la necesidad de una imagen nueva / actualizada hasta su disponibilidad para uso en producción
Frecuencia de publicación de imágenes	Número de imágenes nuevas / actualizadas publicadas en un período de tiempo determinado

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Métricas de apoyo

Métrica	Descripción
Disponibilidad de registro	Cantidad de tiempo de actividad / tiempo de inactividad del sistema de registro en un período de tiempo determinado
Número de alertas de monitoreo	Cantidad de alertas de monitoreo activadas en un período de tiempo determinado
Número de pruebas de unidad / integración	Número de unidades automatizadas o pruebas de integración para una aplicación.
Número de pruebas funcionales / de aceptación.	Número de pruebas funcionales o de aceptación automatizadas para una aplicación
Punto de recuperación medio	Rango de tiempo medio de datos que se pierden debido a un incidente
Cumplimiento de control de retención	Porcentaje de controles relacionados con la retención (por ejemplo, AU-11, SI-12) que están automatizados
Instanciación de imagen	Tiempo transcurrido entre el momento en que un desarrollador solicita la creación de instancias de imagen y su creación de instancias real
Desviación de referencia de seguridad	Desviación entre los puntos de referencia de seguridad aplicados a una imagen y los puntos de referencia de seguridad en una imagen instanciada
Controles técnicos	Número de controles técnicos de seguridad implementados parcial o totalmente
Frecuencia de parches de vulnerabilidad	Con qué frecuencia los parches de vulnerabilidad se implementan regularmente en producción
Tiempo de entrega de parches de vulnerabilidad	Tiempo entre el descubrimiento de una nueva vulnerabilidad (es decir, su publicación) y el parcheo en la producción
A & A tiempo de entrega	Tiempo entre el inicio de una evaluación de cumplimiento de seguridad y la finalización de los procesos de A&A
Los hallazgos de SAR cuentan	Número de hallazgos en el Informe de evaluación de seguridad (SAR)
Recuento de POA y M	Número de POA y Ms
Plazo de entrega de revisión de seguridad de nueva arquitectura	Tiempo transcurrido entre el inicio de una solicitud de revisión de seguridad de una nueva arquitectura y la finalización

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Métricas de apoyo

Métrica	Descripción
Experimentos y alternativas	Número de experimentos tecnológicos y componentes alternativos probados.
Cuenta de registro del sistema	Número de sistemas (aplicaciones, SO, servicios) en una plataforma DevSecOps que están registrando
Cumplimiento de control de seguridad de AU	Lista y porcentaje de controles de seguridad de AU que se satisfacen mediante las prácticas de registro de la plataforma DevSecOps
Plazo de aprobación de CM	Tiempo entre el envío de una solicitud de cambio y su aprobación
Plazo de aprobación de implementación	Tiempo transcurrido entre la solicitud de implementación de un cambio aprobado y la implementación real en producción
Tiempo de entrega de notificaciones	Tiempo entre cualquier paso dado del proceso de MC y la notificación de todas las partes interesadas
Tiempo de entrega de aprovisionamiento de usuarios	Tiempo entre la solicitud de un nuevo usuario en la plataforma y el usuario que puede iniciar sesión
Cumplimiento de control de seguridad de CA	Lista y porcentaje de controles de seguridad de CA que se satisfacen mediante las prácticas de administración de cuentas de la plataforma DevSecOps
Frecuencia de auditoría de privilegios	Número de veces en un período de tiempo determinado que los usuarios y sus privilegios son auditados
Recuento de administrador	Lista y número de usuarios que tienen privilegios de administrador
Frecuencia de rotación secreta	Número de veces en un período de tiempo dado que los secretos se cambian y actualizan cuando se ven afectados
Plazo de incorporación	Tiempo entre una solicitud de una nueva aplicación para usar la plataforma DevSecOps y la aplicación que se está implementando en la plataforma
Tiempo de entrega de gastos	Tiempo entre un gasto y el informe del gasto.

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Otra forma de encarar las métricas en DevSecOps es dividirlas en dos categorías.

- **Costo de infracciones**
- **Costo de evitar infracciones**

El costo de las infracciones es interesante y difícil de obtener de un sistema. Es un poco como el costo de la interrupción, ya que no todas las aplicaciones web son transaccionales, por lo que no es tan simple como decir que la aplicación estuvo inactiva durante una hora y realiza transacciones de $ 1 millón por hora, por lo que perdimos $ 1 millón.

Imaginemos un Instituto que si bien uno puede solicitar y pagar su membresía al instituto en el sitio web, ese no es su propósito principal.





El instituto de DevOps está allí para apoyar su propósito organizacional de reunir, inspirar, guiar, representar y celebrar a todos los que comparten una pasión en común y garantizar que DevOps brinde su potencial excepcional para beneficiar a la comunidad.

Imágenes creadas por dooder

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Del mismo modo, pensemos en un banco con el que puede solicitar una línea de crédito a través del sitio web, pero su gobierno, desde una perspectiva de vulnerabilidad, exige que el crédito solo se apruebe después de una conversación telefónica. El núcleo de las transacciones que hacen en línea es alrededor de la recolección de pagos; si eso cae, simplemente retoma donde lo dejó más tarde.

El costo de una brecha no es solo el tiempo de inactividad del sistema, muchos tipos de brechas no requieren ningún tiempo de inactividad.

Cuando un agente de amenazas logra ingresar a su sistema, robando los datos de sus clientes y uno se entera nueve meses después, ese no es su problema. Su problema ahora es la operación de limpieza, su daño a la reputación y cualquier multa (ver en particular GDPR) que se le imponen. Oh, y si alguien va a ir a la cárcel

Ese comportamiento **CYA (Cover your asset)** 'firma esto para decir que conoces el riesgo' no es tan inteligente ahora, ¿verdad?)



Imágenes creadas por dooder

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

Entonces, ¿cuál es el costo de evitar la violación? Es una inversión significativa en personas y sistemas, y aquí es donde medir las mejoras de DevSecOps resulta útil. Estos son agunos ejemplos:

- Ahorro de tiempo al evitar las aprobaciones de seguridad: la seguridad está involucrada en el proceso y el equipo autónomo ha adquirido el conocimiento requerido y las pruebas automatizadas
- Tiempo ahorrado desde la administración de secretos y certificados usando una solución de protección de identidad
- Tiempo ahorrado de IT Ops haciendo cambios en el firewall porque los equipos de productos se están sirviendo a sí mismos, de forma segura



Recuerde: el tiempo ahorrado en última instancia significa el resultado del valor recibido antes.

Imágenes creadas por dooder

CALMS y DevSecOps

MEASUREMENT (MÉTRICA/MEDICIÓN) Y DEVSECOPS

En resumen, las mediciones DevSecOps buscan hacer tres cosas:

1. Mostrar tiempo / costo adicional en el flujo de cadena de valor para actividades de seguridad
2. Evite todas las infracciones o tenga remedio casi instantáneo
3. Optimice y automatice todas las actividades de seguridad a través de la canalización de extremo a extremo

Cuando se logran estas tres cosas, ya no necesitamos hablar sobre DevSecOps . Es solo DevOps, o incluso mejor, solo la forma de trabajar



Imágenes creadas por dooder

CALMS y DevSecOps

SHARING (COMPARTIENDO) Y DEVSECOPS

Podemos resumir que, DevSecOps tiene dos impulsores principales: la seguridad es una limitación en la mayoría de las organizaciones y ha sido una ocurrencia tardía en el mundo de DevOps. DevOps busca optimizar el flujo de resultados de valor desde la idea hasta la realización mediante la eliminación de cuellos de botella y la amplificación de la retroalimentación.

Antes en automatización, describimos cómo podemos abordar esta limitación con las prácticas de automatización, pero también hemos dado a entender que existe un enfoque cultural para abordar este problema, y todo se centra en el **intercambio efectivo de conocimientos**.

Un patrón que aconsejamos y que hemos visto funcionar con eficacia en varias ocasiones, es que uno o dos miembros de la **'torre de marfil'** de seguridad se integren y se ubiquen en un equipo de productos por un período temporal, alrededor de tres meses funciona bien.



Imágenes creadas por dooder

CALMS y DevSecOps

SHARING (COMPARTIENDO) Y DEVSECOPS

La regla 80:20 parece aplicarse aquí; Se necesita el 20% del conocimiento en el 80% de las situaciones, por lo que la cantidad requerida de conocimiento se contagia rápidamente cuando un pequeño grupo de humanos trabaja en estrecha colaboración. Los expertos en seguridad pueden pasar por varios equipos de productos: el conocimiento que habrán adquirido durante este proceso acelerará el intercambio de conocimientos en los nuevos equipos que visiten.

Esto requiere que los equipos de seguridad se suban a bordo con la visión de DevOps por la que trabajan los equipos de producto y necesita algunas personas de mente abierta dispuestas a cambiar sus prácticas de trabajo diarias. El liderazgo tiene que ser un sello de goma y tenemos que hacer el intercambio entre 'no podemos encontrar tiempo para ahorrar tiempo' y 'si se hace romper la cadencia ahora, cosechará la recompensa de una mayor capacidad de inyección más adelante'



Imágenes creadas por dooder

CALMS y DevSecOps

SHARING (COMPARTIENDO) Y DEVSECOPS

Estos ágiles, DevOps, defensores de DevSecOps, llámalos como quieras, también pueden ser alentados a hacer otras cosas para ayudar a compartir el conocimiento. Por ejemplo, podrían comenzar una comunidad de **DevSecOps de interés** / práctica o gremio donde inviten a las personas (cualquier persona interesada) a participar en talleres, reuniones, espacios en línea o incluso hackatones y compartir sus ideas de esa manera.



Si bien todo este intercambio de conocimientos se realiza de manera conversacional, también pueden ocurrir otras cosas para que el conocimiento esté ampliamente disponible a través del autoservicio y para acelerar el aprendizaje. La mayoría de las organizaciones con las que trabajamos usan algún tipo de wiki, ya sea Confluence o Teams u otra cosa. Sin embargo, un error común en este tipo de plataformas es que se vuelven extensas y difíciles de navegar. Ellos también necesitan un poco de limpieza.

CALMS y DevSecOps

SHARING (COMPARTIENDO) Y DEVSECOPS

Vemos que mucho del mundo ágil y DevOps ahora opera en Jira (hay otras herramientas de gestión de pedidos pendientes disponibles, pero esta es, con mucho, la más popular y ubicua en nuestra experiencia). Y todos saben cómo usar una herramienta de chat grupal (¿tiene WhatsApp, no?).

Slack es, por lejos, el canal de chat más popular en equipos de tecnología, pero Office365 está disponible en casi todas partes..
Ambos tienen la capacidad de integrarse con herramientas en la cadena de herramientas DevOps/DevSecOps y chat grupal + herramientas = ChatOps.



Esta es uno de los quick wins favoritos para las evoluciones de DevOps, y siempre me piden quick wins. Rápido y fácil de configurar y usar, es una plataforma de colaboración que genera confianza a través de la visibilidad y puede mejorar los tiempos de recuperación y la estabilidad del sistema

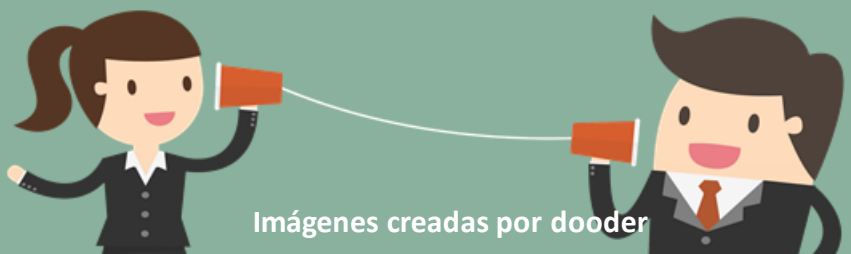
Imágenes creadas por dooder

CALMS y DevSecOps

SHARING (COMPARTIENDO) Y DEVSECOPS

Entonces, por ejemplo, en lugar de entrar en una sala de guerra y poner a la gente en una línea de conferencia cuando ocurre un incidente, si está usando **ChatOps**, abrirá un canal y todos se juntaran allí. Su equipo puede consultar a herramientas como “Dynatrace” para obtener un diagnóstico de fallas, o a Jenkins para obtener información sobre los últimos cambios o puede presionar un nuevo cambio para solucionarlo a través de Puppet, todos compartiendo el mismo panel de vidrio. Cuando haya terminado, la retro este completa y se ejecute un experimento para asegurarse de que el incidente no vuelva a ocurrir, puede guardar el registro del chat en un ticket de Jira o otra tool.

O otro ejemplo, un equipo ha llegado al final de su sprint y está listo para lanzar. Algunos de ellos están en una oficina en Kent, el jardín de Inglaterra como nos gusta llamarlo, y algunos de sus compañeros de equipo están en Bangalore. Al configurar un canal de chat, pueden monitorear todas las confirmaciones de código en tiempo real, verificar que el servidor de integración continua esté pasando las pruebas (¡incluidas las de seguridad!) Y verlo entrar en producción, e incluso quedarse y monitorearlo allí si lo desean. Nuevamente, el registro de chat se puede guardar nuevamente en la cartera de pedidos del producto.



Imágenes creadas por dooder

CALMS y DevSecOps

SHARING (COMPARTIENDO) Y DEVSECOPS

Nuestra recomendación final para compartir en DevSecOps es el dojo. Popularizado por Target , los dojos ponen uno o dos equipos de productos nuevos (ish) en un nuevo entorno con PYME (como la seguridad) en el mismo lugar durante unas semanas y establecen y persiguen un objetivo agresivo mientras practican nuevas formas de trabajo.

Disfruté especialmente la charla de Capital One de DevOps Enterprise Summit London en 2018, donde John, el propietario del producto, establece el tono para la seguridad psicológica, con la declaración 'las cosas se van a poner desordenadas' que nos lleva al círculo completo en la primera publicación en este serie, sobre los elementos culturales de DevSecOps.



Imágenes creadas por dooder

CALMS y DevSecOps

SHARING (COMPARTIENDO) Y DEVSECOPS

Sobre el dojo

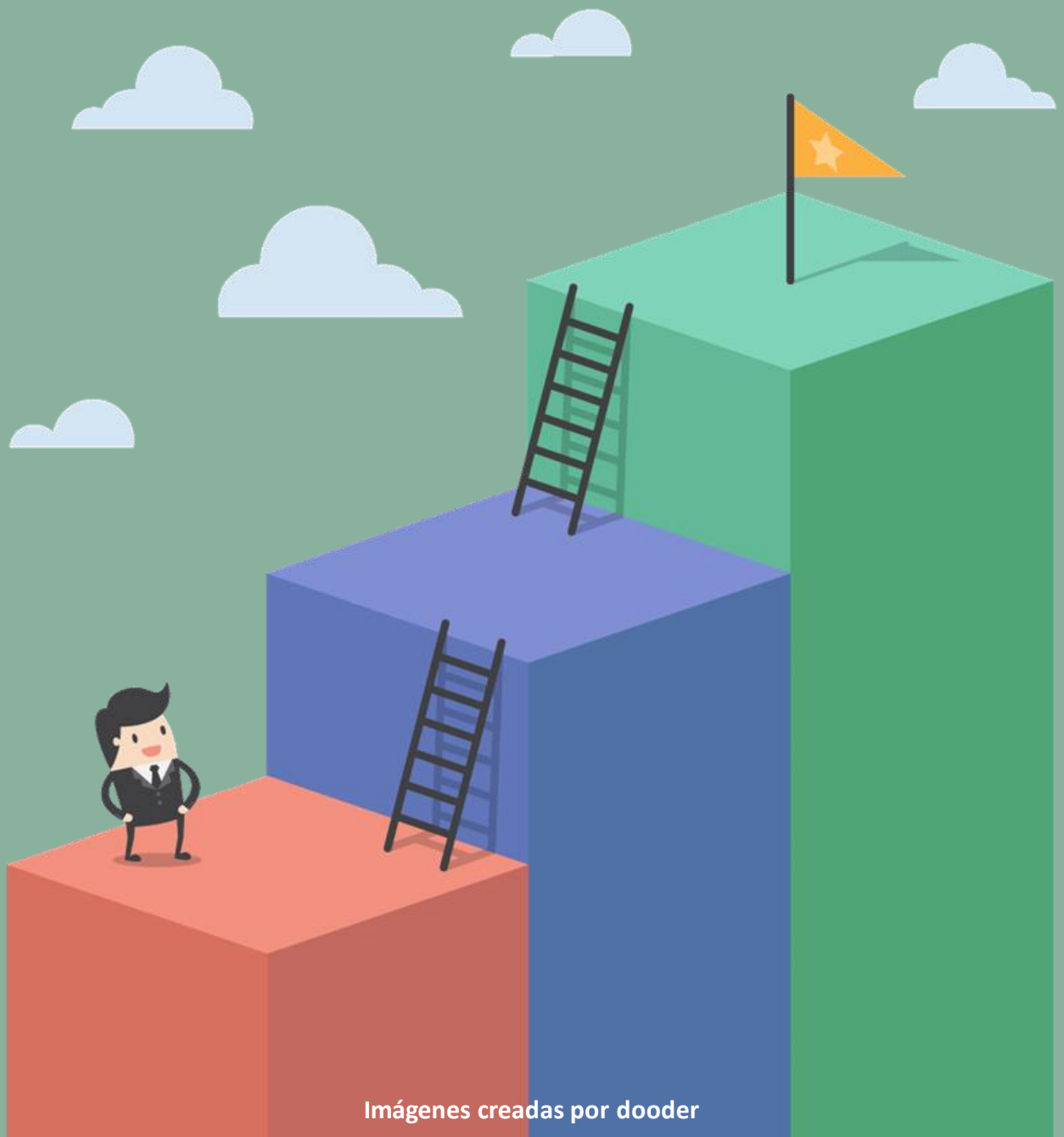
Como el entorno de aprendizaje inmersivo único de Target, el Dojo desarrolla o mejora los músculos de Producto, Lean, Agile y DevOps. Target fue el primero en crear este entorno único para ayudar a la organización a cambiar su cultura y mentalidad rápidamente. Los equipos abandonan su espacio de trabajo normal y bajan al Dojo por un período de tiempo para lograr dos objetivos:

- Primero, cumplir con su trabajo actual . Los equipos no crean aplicaciones falsas; se enfocan en su trabajo de productos del mundo real, entregan contra su cartera de pedidos y agregan valor a sus clientes.

- Segundo, aprenda a adquirir y desarrollar habilidades de larga data . El aprendizaje es esencial para la cultura Target y para mantenerse al día con el mundo de la tecnología en constante cambio. Los equipos deben estar dispuestos a comprometerse con el aprendizaje y aceptar que pueden necesitar reducir la velocidad para acelerar ese aprendizaje.

Estos dos objetivos son esenciales para entrar en el Dojo; continuar entregando valor mientras aprende algo nuevo o mejora una habilidad actual.

CALMS Y DMM



Imágenes creadas por dooder

CALMS Y DMM

Uso de CALMS para evaluar los DevOps de una organización

De los marcos probados que permiten a las empresas evaluar DevOps en su organización y cómo se puede mejorar, el modelo CALMS sigue siendo particularmente útil.

CALMS, que como vimos anteriormente significa Colaboración, Automatización, Lean, Measurement and Sharing, es particularmente útil para analizar la estructura DevOps de una organización y, en última instancia, su utilidad en cualquier organización. El marco CALMS cubre a todos los interesados dentro de DevOps, incluidos los negocios, las operaciones de TI, QA, InfoSec y los equipos de desarrollo, y cómo entregan, implementan e integran colectivamente procesos automatizados que tienen sentido comercial.

"El modelo CALMS proporciona un buen marco de referencia para comparar la madurez de un equipo DevOps y, como tal, es invaluable para evaluar el estado de los equipos para el cambio transformacional que conlleva".



Imágenes creadas por dooder

CALMS Y DMM

Práctica	Cultura	Automatización	Soporte/Apoyo	Medidas	Comunicación
NIVEL 5: OPTIMIZACIÓN	El "POR QUÉ" en las organizaciones, está enfocado y sostenido en el cliente, día a día junto con la tecnología involucrada y apropiada.	Los procesos empresariales de soporte tecnológico tienen un esfuerzo mínimo de TI. La innovación y soporte de flujos de valor son automatizados.	El enfoque está en el cliente, la resolución de problemas es una norma, el "management coaching" y el aprendizaje son continuos.	Medición del valor de las ideas para su futura realización.	Intercambio asertivo de conocimientos y fortalecimiento individual.
NIVEL 4: NATURALIZACIÓN	La cultura es vista como un activo a gestionar. Se establece la capacidad para adaptar cambios empresariales según las necesidades.	La automatización respalda los objetivos y procesos comerciales, no solo los procesos tecnológicos.	La atención al cliente como pauta, el personal está comprometido para apoyar la innovación y la mejora constante.	Monitoreo desde el enfoque del cliente hasta el flujo de valor, con una retroalimentación adecuada y ciclos de mejoras.	La comunicación se dirige hacia la cooperación y la mejora basada en indicadores y objetivos acordados.
NIVEL 3: EVOLUCIÓN	La actitud y el comportamiento deseados se identifican, la gerencia o los entrenadores comienzan a apoyar los cambios.	Los procesos están automatizados y centrales en todo el ciclo de vida del flujo de valor. La tecnología es asignada a las áreas o procesos de negocio de los silos.	Comienza el interés hacia la atención del cliente. El personal se capacita para aprender en un entorno "sin culpa/faltas".	Monitoreo de recursos (personas, procesos, herramientas, proveedores) vinculados a indicadores del desempeño esencial. Existen algunas alertas e incrementos.	La colaboración, la toma de decisiones compartidas y la rendición de cuentas comienzan y se chequean para encontrar formas de avanzar.
NIVEL 2: MEJORAS	Estar al tanto de los aspectos culturales que pueden ayudar u obstaculizar el valor centrado en el cliente. La ayuda solicitada o la gestión lidera los esfuerzos del cambio.	Los procesos de automatización del silo permiten una integración mínima o una comunicación de flujo de valor cruzado.	El organigrama tradicional empieza a ser cuestionado, se considera cambiar la estructura organizativa del enfoque del valor o del producto.	Las medidas se centran en el proyecto, las decisiones son reactivas o no son lo suficientemente "alertas".	Comienzo de la gestión visual, la cual no está integrada para permitir el conocimiento del flujo de valor o el intercambio de decisiones.
NIVEL 1: COMIENZO	La estrategia no está alineada, (roles de personas, KPIs, herramientas, valores) impactando así en el negocio diario.	No hay automatización por lo que varias herramientas similares no están integradas.	El enfoque es reactivo/irreflexivo en la resolución de problemas, hay poca o nula participación directa en la gestión, el aprendizaje es ad-hoc.	Toma de medidas vanidosas, no reactivas o simplemente informes básicos, estas medidas no guían el desarrollo, el cambio o mejora alguna.	Mala o escasa comunicación y coordinación. Muchas reuniones e informes no son leídos o tenidos en cuenta.

Anti patrones de CALMS



Imágenes creadas por dooder

Anti patrones de DevSecOps

Recientemente leímos en alguna parte que los buenos líderes tienen una lista de "empezar a hacer" y una lista de "dejar de hacer". Este concepto está en Good to Great (Collins) pero también lo he leído en otros lugares.

La lista 'Dejar de hacer' normalmente comprende las cosas que no agregan valor o probablemente más importantemente 'interfieren' con el trabajo que agrega valor. Los he escuchado como antipatrones.

De manera similar, DevOps / DevSecOps tienen antipatrones, cosas que destruyen las bases de DevOps / DevSecOps en las organizaciones.



Imágenes creadas por dooder

Anti patrones de DevSecOps

DevOps / DevSecOps es un proceso/metodología : No lo es, es una filosofía, una forma de pensar.

Agile es igual a DevOps / DevSecOps: No lo es, DevOps y Agile se complementan, pero no son lo mismo. DevOps lo ayuda a administrar todo su proceso de ingeniería. Agile, por otro lado, es útil cuando se trata de proyectos complejos. Agile lo guía a través de cómo puede desarrollar su software.

Cambio de nombre: Cambiar el nombre de un equipo o nombrar ingenieros con DevOps no significa que seas DevOps.

Iniciar un grupo DevOps separado: Creó otro silo, ¿verdad? El núcleo de DevOps se centra en eliminar los silos entre las unidades de una organización.

DevOps / DevSecOps es una palabra de moda: En realidad, algunas personas usan esto como una palabrota ... "¿Puedes hacer esto de una manera DevOps?", Es decir, ¿puedes hacerlo rápidamente y romper las reglas mientras estás en ello? Pienso que es un estado mental y está respaldado por la excelencia en ingeniería.

Anti patrones de DevSecOps

DevOps significa que los desarrolladores administran la producción: No, DevOps no asume la responsabilidad de administrar la producción de personas cuya responsabilidad principal es la estabilidad del sistema de producción.

Venden DevOps / DevSecOps como una bala de plata: Algunas personas piensan que poner estas palabras en la misma oración es como poner 'Queso' y ‘Dulce' en la misma palabra, no lo es, es una de las cosas más difíciles que intentarán y es algo que una organización debe hacer por sí misma con la tutoría y educación adecuadas.

La adquisición hostil: Desarrollo no se hace cargo de las operaciones. Hay una parte igual de responsabilidad entre los equipos de desarrollo y Operaciones.

DevOps es gestión de lanzamiento impulsada por el desarrollo: Cualquier cosa que sugiera que DevOps reemplaza las operaciones de TI no tiene sentido. Claro, acelerar y automatizar, pero DevOps no es un proceso o una capacidad de automatización y no es el reemplazo de las operaciones de TI.

No podemos hacer DevOps - Somos únicos: Sí, lo sos, esta es la razón principal por la que DevOps funciona, ya que no es una receta única para todos los cocineros en 5 minutos. Aplica las filosofías y principios y tendrás tu propio sabor DevOps. Una de nuestra frases favoritas es " Usa la cultura para cambiar la cultura "

Anti patrones de DevSecOps

DevOps y la seguridad son enemigos: DevOps y la seguridad no son enemigos. Claro, el enfoque principal de DevOps está en la entrega continua. Pero eso no significa que DevOps deje completamente atrás el aspecto de la seguridad. Si eres un ingeniero de seguridad, no tienes que detestar a DevOps.

Por el contrario, DevOps en realidad proporciona una plataforma para incorporar el elemento de seguridad en las primeras etapas del proceso de desarrollo. Al hacerlo, se aumenta la seguridad del código antes de su implementación. Por supuesto, tendría que tener las herramientas operativas y de automatización adecuadas para hacerlo. De hecho, el término "DevSecOps" ha sido acuñado y se usa ampliamente. Abarca los principios y pautas para integrar la seguridad en su enfoque DevOps.

DevOps se deshace de las operaciones: La suposición de que DevOps elimina por completo las operaciones de TI no es cierta. Si en su implementación de DevOps, el equipo de producción se queja de que una aplicación tiene algunos problemas relacionados con las operaciones, solo significa que lo está haciendo mal. No significa que DevOps fomente un entorno de NoOps. Los equipos de operaciones continúan jugando un papel crucial.

Conclusiones



Imágenes creadas por dooder

Conclusiones

A medida que las tecnologías avanzan y aumenta el valor del dato, nos encontramos en un terreno en el que la seguridad se convierte no solo en un valor añadido, sino en una parte esencial del diseño, tanto como puede serlo el rendimiento.

Esta filosofía no es nada nuevo y se basa en la adaptabilidad. Como dijo el célebre Bruce Lee:

No te establezcas en una forma, adáptala y construye la tuya propia, y déjala crecer, sé como el agua. Vacía tu mente, sé amorfo, moldeable, como el agua. Si pones agua en una taza se convierte en la taza, si pones agua en una botella se convierte en la botella, si la pones en una tetera se convierte en la tetera. El agua puede fluir o puede aplastar. Sé como el agua, amigo, el agua que corre nunca se estanca; así es que hay que seguir fluyendo.

Bruce Lee



Conclusiones

No hay duda de que DevSecOps impulsado por CALMS es un ejemplo de los que decía Bruce Lee y esta revolucionando la forma en que las organizaciones manejan la seguridad. Sin embargo, debido a una variedad de razones, como la falta de conciencia de lo que es DevSecOps, un cambio cultural no solicitado para los empleados, restricciones presupuestarias y, a veces, solo la ambigüedad del término, muchas organizaciones siguen siendo escépticas. sobre cambiar a DevSecOps.

Los beneficios técnicos, así como los beneficios comerciales que las organizaciones pueden obtener al implementar DevSecOps, son muy prometedores. Aunque seguramente se encontrará con algunos inconvenientes cuando comience, la implementación de DevSecOps puede ser muy beneficiosa para su organización a largo plazo. Es por eso que sumarse a comunidades como DevSecOps Argentina para estar al dia o contratar buenos proveedores puede marcar la diferencia.



AUTORES



Imágenes creadas por dooder

Luciano Moreira



https://www.linkedin.com/in/lucianomoreiradacruz/

https://twitter.com/Luciano_m_cruz

Soy una persona apasionada por la tecnología y sobre todo por la seguridad de la misma en todos sus estados. creo que el conocimiento debe ser libre y que las personas deberían buscarlo todo el tiempo.

Tengo un Master en Ciberseguridad por la Universidad Camilo José Cela. DevOps Institute Ambassador, uno de los seleccionados por la gente de peerlyst como, "50 Influential DevSecOps Professionals" Primer Auditor CSA STAR certificado en la región SOLA, Fui Elegido Cybersecurity Consultant of the Year en los premios Cybersecurity Excellence Awards 2016, 2017 y 2018, MVP - Most Valuable Professional Azure (Security, Infrastructure & Storage), Auditor Líder ISO 27001:2013, 27018, 27017 y 9001:2015.
Cofundador y presidente de DevSecOps Argentina, Presidente del capítulo argentino de la CSA Cloud Security Alliance. Miembro de ISSA, ISACA, OWASP, del comité académico de los eventos E-gisart y Cyber de ISACA y del comité científico en Ciberseguridad del evento IEEE ARGENCON, Orador e Instructor de seguridad en EducacionIT, FSA, I-SEC, entre otros.

Cuento con más de 17 años de experiencia en IT, de los cuales los últimos 13 años trabaje en una docena de proyectos en todas las capas de seguridad y infraestructura hibridas.

Actualmente me desempeño como CDSO - Chief DevSecOps strategy Officer en Cloud Legion

Christian Ibiri



www.linkedin.com/in/christian-ibiri

https://twitter.com/Christianibiri

Soy una persona apasionada por la tecnología y sobre todo las disruptivas o las que transforman la forma en que estamos acostumbrados de hacer las cosas, como computación en la nube, metodologías Agile, o Infraestructuras Agiles.

La verdad me encanta la época que estamos viviendo hoy por hoy donde uno puede levantar varias instancias para correr lo que se necesite sin tener que contar con un parque computacional enorme, mucho mas si me acuerdo de mis principios donde teníamos que desplegar maquinas físicas, armar clusters, “virtualización si teníamos suerte”.

Uno de los seleccionados por la gente de peerlyst como, "50 Influential DevSecOps Professionals"

Cofundador de DevSecOps Argentina, tiene 12 años de experiencia en IT, de los cuales los últimos 7 años en las áreas de Infraestructuras hibridas, cloud, DevOps y tooling automation.

Participe en varios proyectos de infraestructura, networking, migración y implementación de clouds privadas y publicas, automatización, comunicaciones unificadas y colaboración, acompañando a las demás partes involucradas en los mismos, desde la etapa de requerimientos hasta la implementación y pos-implementación. Creo en la interoperabilidad y que el mundo opensouce puede tranquilamente coexistir con tecnologías propietarias sin ningún tipo de limitaciones.

Entusiasta de DevOps, y infraestructuras agiles o infra como código.

Helen Beal



https://www.linkedin.com/in/helenjbeal/

https://twitter.com/HelenRanger4

Helen Beal ayuda a las personas a practicar los principios de DevOps en organizaciones del mundo real para Ranger4. Se describe a sí misma como DevOpsologist, ya que su papel principal en su vida laboral es estudiar las entradas y salidas de los sistemas de pensamiento que conforman DevOps y qué resultados de valor ofrecen y podemos medir.
Oradora, escritora, consultora, entrenadora y facilitadora de aprendizaje.

- Miembro de la Junta de Regentes del Instituto DevOps
- En el jurado de los Premios DevOps Dozen
- Miembro de la Junta Consultiva Mundial de DevOps
- Miembro de la Junta Consultiva de DevOps de Jax
- Editor de DevOps en InfoQ
- Colaborador de devops.com

Listada en DevOps 100 de Techbeacon: Top Leaders, Practitioners , Expertos a seguir
Enumerada en las 20 mujeres influyentes de TechBeacon en DevOps
Enumerada en las 51 influenciadoras de DevOps de PowerAdmin para comenzar a seguir hoy
Enlistada en Mujeres en DevOps.com DevOps Trabajos de intervención

Fuera de DevOps, es ecologista y novelista.

Santiago Agustín Fernandez



https://www.linkedin.com/in/safernandez666/

https://twitter.com/safernandez666

Empecé mi vida laboral en el Grupo Prisa. Multimedia, comunicación, televisión, radio, prensa escrita y editorial española, presente en 22 países de Europa y América, principalmente de habla hispana y portuguesa. A través de diferentes cargos en la dirección del área de Microinformática.
Luego trabajé como Jefe de Seguridad de la información La tercera compañía de seguros más grande del mundo. Hemos establecido y mantenido una estrategia de Seguridad de la Información alineada con las metas y objetivos de la organización para guiar el establecimiento y la gestión del programa continuo de Seguridad de la Información para LATAM, obteniendo la certificación ISO 27001.
Actualmente me estoy desarrollando como Gerente de Seguridad de la Información | CISO en Naranja. Principal emisor de tarjetas de crédito en Argentina, con más de 10 millones de plásticos, con una red de adquisición de 280.000 comerciantes, y más de 220 sucursales. Naranja está actualmente expandiendo su negocio a través de soluciones tecnológicas innovadoras, que incluyen Digital Financial Services, Naranja Travels, plataforma de eCommerce, y servicios a comerciantes como Naranja POS a Payment Facilitator, entre otros.
Al mismo tiempo, proporciono consultoría y servicios a diferentes organizaciones.Soy Licenciado en Informática por la Universidad de Palermo y estoy certificado como CISSP, CISM, CSX CyberSecurity, CCSK, MCSA & SMAC™ (Scrum Master). En este momento soy un entusiasta de DevSecOps. Revisa algunos proyectos en Github.com/safernandez666

Links y referencias



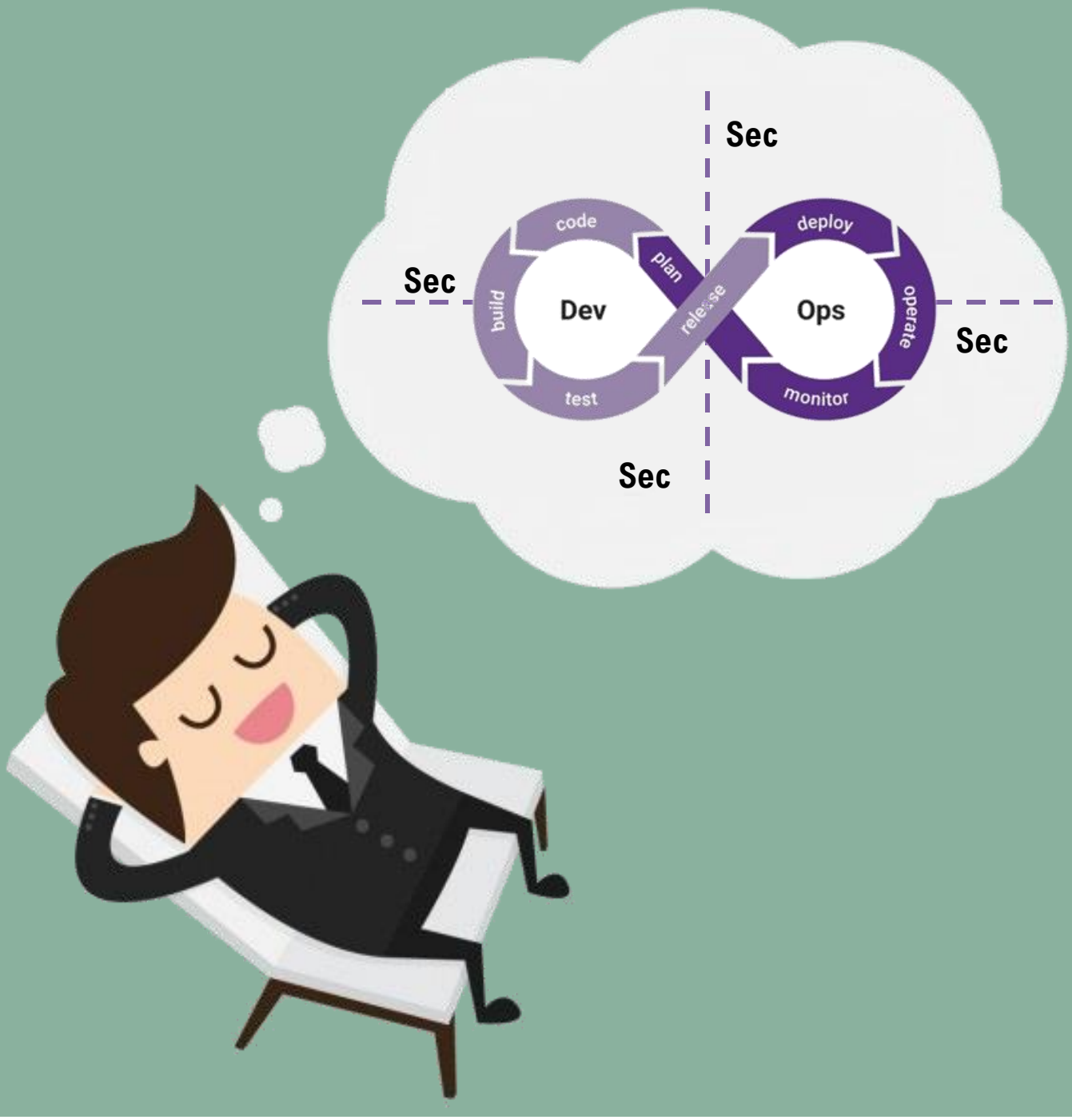
Imágenes creadas por dooder

Links y referencias

- https://cloudlegion.com.ar/
- https://www.ruggedsoftware.org/
- https://devsecops.org/
- https://devsecops-argentina.org/
- https://devopsinstitute.com/
- https://news.netcraft.com/archives/2017/03/24/march-2017-web-server-survey.html
- https://itrevolution.com/the-unicorn-project/
- https://twitter.com/realgenekim
- https://itrevolution.com/book/the-phoenix-project/
- http://www.amazon.com/DevOps-Dummies-Computer-Tech/dp/1119552222
- http://theleanstartup.com/
- https://dzone.com/articles/the-devops-loop
- http://teamtopologies.com/
- http://www.manning.com/books/cloud-native-patterns
- http://itrevolution.com/war-and-peace-and-it/
- https://itrevolution.com/book/the-art-of-business-value/
- https://www.amazon.com/Engineering-DevOps-Continuous-Improvement-Beyond-ebook/dp/B07ZZLH7KM

Links y referencias

- https://itrevolution.com/devops-dojo-captial-one/
- https://dojo.target.com/
- https://www.tandfonline.com/doi/abs/10.1080/10919392.2019.1568713?journalCode=hoce20
- https://www.developer.com/mgmt/a-kaizen-approach-for-devops-how-to-help-teams-find-and-fix-their-own-problems.html
- https://www.gartner.com/doc/3772095/hype-cycle-application-security
- https://en.wikipedia.org/wiki/False_positives_and_false_negatives
- https://www.forbes.com/sites/louiscolumbus/2017/04/29/roundup-of-cloud-computing-forecasts-2017/#449ad54e31e8
- https://en.wikipedia.org/wiki/Data_loss_prevention_software
- https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_10_Mobile_Risks
- https://vulndb.cyberriskanalytics.com/
- https://nvd.nist.gov/
- https://en.wikipedia.org/wiki/Black-box_testing
- https://itrevolution.com/book/the-devops-handbook/
- https://www.esecurityplanet.com/network-security/2019-it-security-employment-outlook.html
- https://www.forrester.com/report/Use+DevOps+And+Supply+Chain+Principles+To+Automate+Application+Delivery+Governance/-/E-RES118681



DEVSECOPS Argentina
by Cloud Legion

CLOUD LEGION

https://devsecops-argentina.org/

https://www.linkedin.com/groups/12033278

https://twitter.com/devsecops_ar

CC BY SA