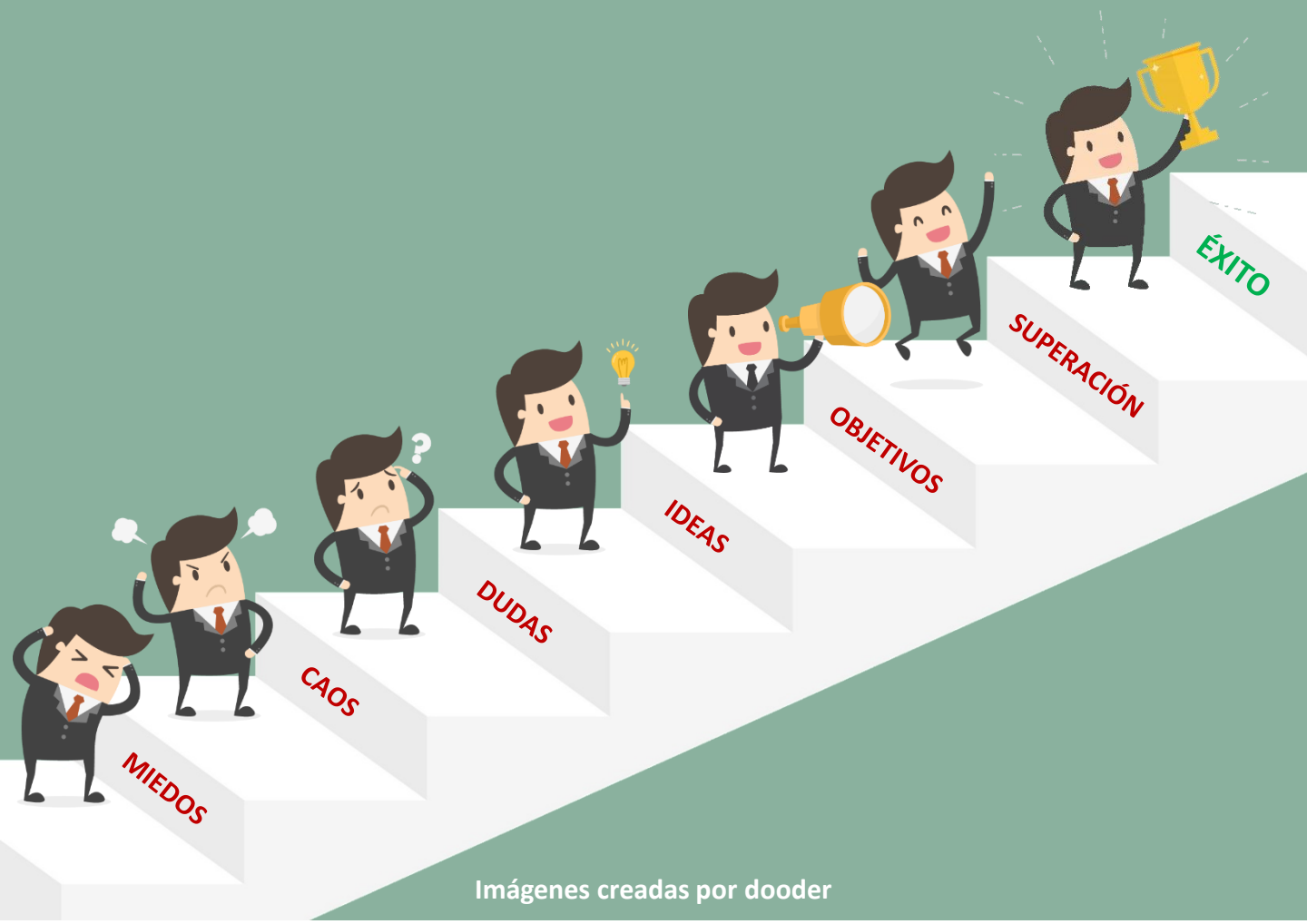
DEVSEC OOPS!...

Los casos de no éxito en DevSecOps 2.0



Imágenes creadas por dooder

Equivocarse hace parte del aprendizaje

CLOUD LEGION

DEVSECOPS Latinoamérica by Cloud Legion

¿Que veremos?

- Glosario
- Introducción a CALMS
- Fracaso vs Éxito
- Casos de no éxito de DevSecOps
- Errores Típicos
- ¿Como evitar o mitigar?
- 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, algunos casos de no éxito basándonos en los pilares principales que plantea CALMS, ¡Buena lectura!

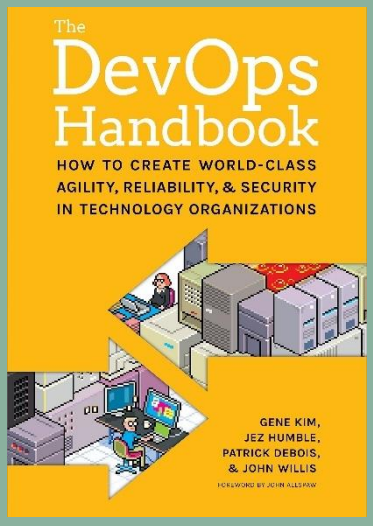


Introducción a CALMS

¿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.

Introducción a CALMS

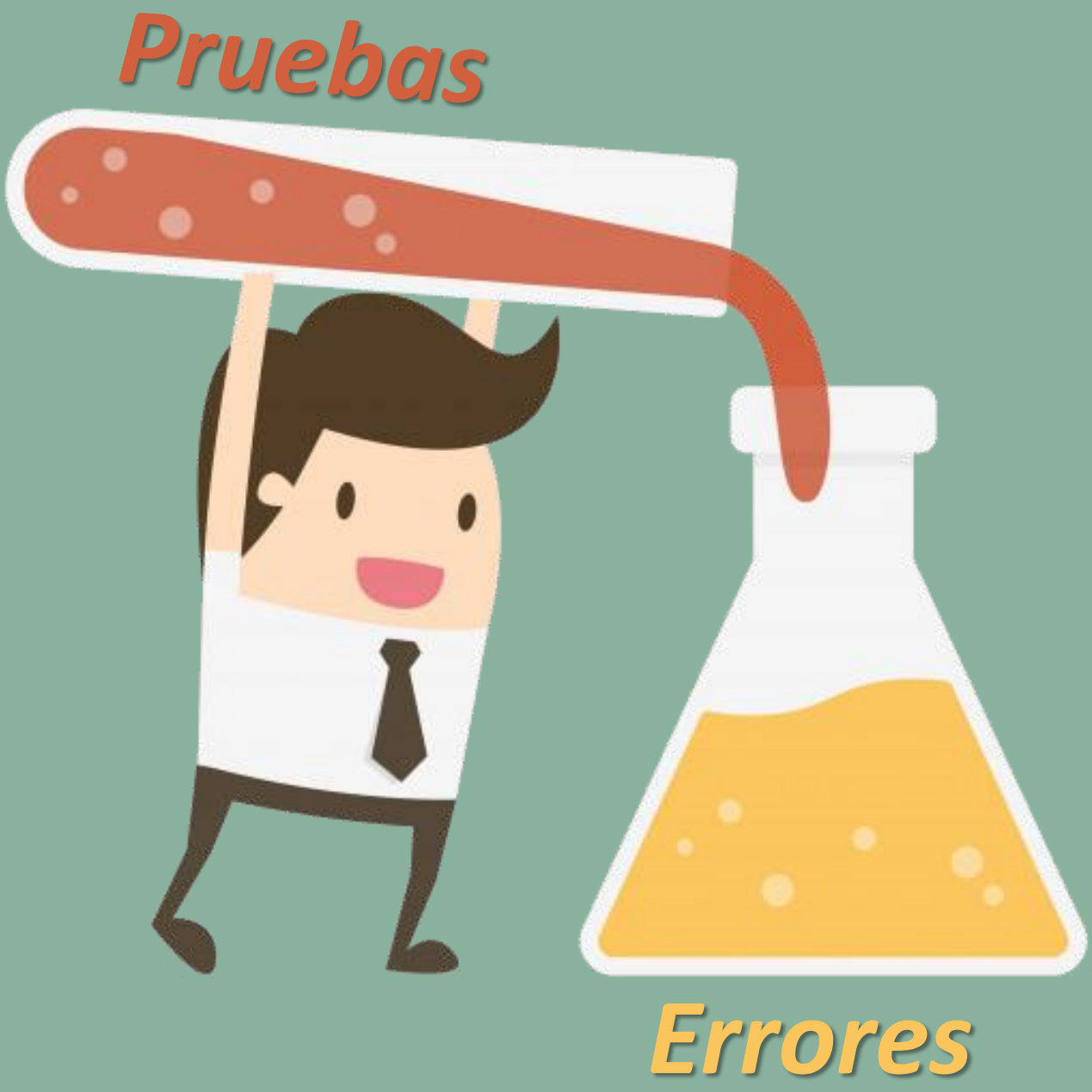
Culture	Todos los equipos tecnológicos son responsables de la seguridad; la seguridad es tarea de todos. Todos comprenden el sistema de extremo a extremo y colaboran regularmente para crear confianza.
Automation	La automatización ayuda a garantizar la seguridad mediante el uso estratégico de la orquestación y la automatización de tareas y procesos que tienen vulnerabilidades de seguridad cuando se hacen manualmente y donde la automatización puede mejorar las prácticas de seguridad.
Lean	La seguridad no es una limitación en el flujo de valor y los equipos no esperan a que se realicen las actividades de seguridad: el flujo se optimiza. El trabajo es visible a través de los backlogs compartidos.
Measurement	Se entiende el coste de la infracción, se comparten las métricas de negocio y de ataque y se sigue un enfoque centrado en el flujo de valor para optimizar el tiempo del ciclo y garantizar que no haya retrasos causados por la seguridad.
Sharing	Los ingenieros de seguridad y de software tienen competencias cruzadas y colaboran para automatizar los conocimientos. Las historias se comparten a través de wikis, standups y en el día a día...

Para profundizar los principios de CALMS y la temática en DevSecOps, súmate a **DevSecOps Latam**

https://devsecops-latam.org/



Fracaso vs Éxito



Imágenes creadas por dooder

A veces me pregunto si tanto nos gustan las historias de superación….

¿Porque ninguna empresa o consultor publica sus casos de no éxito y como lo superaron?



Imágenes creadas por dooder

La cultura del éxito (Un freno para la innovación)

Solemos reverenciar el éxito, el logro, la realización de lo que esperamos, pero qué pasaría con la humanidad si no existiera el intento, la falla, esa búsqueda permanente de ir más allá.

El éxito en realidad "frena el proceso de aprendizaje"

Aprendemos más de los fracasos que de los éxitos. Cuando algo sale como se esperaba, usamos ese proceso como una plantilla mental para futuros proyectos.

Somos usuarios de manuales y metodologías creemos que hemos establecido un patrón exitoso, incluso después de una sola instancia de éxito y esto, a su vez, tiende a transformarse en **"esta es la única forma de hacerlo“.**

El verdadero aprendizaje llega a través de la crisis. Si algo sale mal, tenemos que revolver, experimentar, hackear, gritas y seguir nuestro proceso.



La cultura del éxito (Un freno para la innovación)

“Nuestras mentes agitan las nuevas ideas, están más dispuestas a experimentar, están más abiertas a aportes externos cuando estamos en modo de crisis”

Incluso el fármaco relacionado con la famosa pastillita azul, la droga que más se vende en el mundo, no es usada con el propósito original. Comenzó originalmente para el tratamiento de angina de pecho, un problema cardíaco que afecta los vasos sanguíneos que llevan la sangre al corazón. **(¿Fallo épico o mente abierta para aprender y buscar nuevas aplicaciones?)**





En el libro de Demian Sterman nos muestra el camino del fracaso, el encanto y el aprendizaje que puede conllevar ir tras algo y no conseguirlo, la posibilidad incluso de encontrar algo más, algo distinto e inesperado. Henry Ford, Albert Einstein, Steve Jobs y hasta Walt Disney son algunos de los nombres y Historias de fracasos y fracasados que cambiaron el mundo

¿Se puede aprender del fracaso tan fácilmente?

Nuestra opinión es que no se aprende de fracasar, **se aprende de superar los fracasos**. Y eso es muy difícil en organizaciones que tiene el fracaso como algo negativo en vez de pasos esenciales para conseguir una innovación exitosa.

Si quieres mejorar tras un fracaso y tienes poco tiempo porque te persigue el **"Time to Market"** podes seguir esta técnica rápida de autoevaluación mediante seis preguntas:



¿Qué puedo aprender?

¿Qué hubiera hecho diferente?

¿Qué competencias debo entrenar?

¿De quién puedo aprender?

¿Cuál es el próximo paso?

¿te esforzaste lo suficiente?

Imágenes creadas por dooder

¿Se puede aprender del fracaso tan fácilmente?

1. Análisis. ¿Que puedo aprender de esto?

Responsabilízate de la parte que ha ido mal por ti. Quizá no todo ha sido culpa tuya pero quizá lo fue en parte. La gente con éxito no pone excusas ni culpa a los demás sino que se hacen cargo del asunto. Se crítico pero constructivo. Intenta analizar con objetividad. Haz una lista de las cuestiones clave de lo ocurrido y analiza la lista paso a paso para encontrar los puntos que puedan servirte de aprendizaje.

2. Alternativas. ¿Qué hubiese hecho de forma diferente?

¿Qué otras opciones tenías? ¿Qué decisiones tomaste? ¿Podrías haberlo hecho de modo diferente? Mirando con perspectiva, ¿qué pasos hubieses dado de modo diferente?

3. Competencias. ¿Necesito aprender o mejorar mis habilidades?

¿El problema te ha hecho ver que no tienes las habilidades necesarias? ¿Cómo podrías mejorarlas? Quizá puedas contar con libros, algún curso o alguna persona que pueda ayudar. Prepara un plan de desarrollo para mejorar o adquirir las habilidades y experiencia que necesitas.



¿Se puede aprender del fracaso tan fácilmente?

4. Modelos. ¿De quién puedo aprender?

¿A quién podrías pedir asesoramiento? Si te ha visto algún jefe, algún colega de trabajo o algún amigo les puedes pedir feedback y orientación. Mucha gente no pide ayuda porque lo consideran un signo de debilidad más que una fortaleza. No es así, sino que muestra que estás dispuesto a aprender y cambiar. Algún buen amigo estará dispuesto a ayudar.

5. Acción. ¿Cuál es el próximo paso? Elabora un plan.

¿Probarías algo nuevo o algo diferente? Revisa tus metas y objetivos. Esta inversión ha sido un revés en tu camino pero piensa que ha sido algo divertido en lugar de una interrupción. Ahora puedes centrar tu rumbo hacia el nuevo plan.

6. Aceptación. ¿Te has esforzado lo suficiente y has aprendido lo necesario?

Si ambas respuestas son síes, significa que posiblemente has hecho lo que estaba en tu mano para conseguir el objetivo. Tal vez tienes que plantearte un reto más asequible, mejorar tu planificación, tus competencias y tus técnicas o darte el tiempo necesario para alcanzar esa meta.



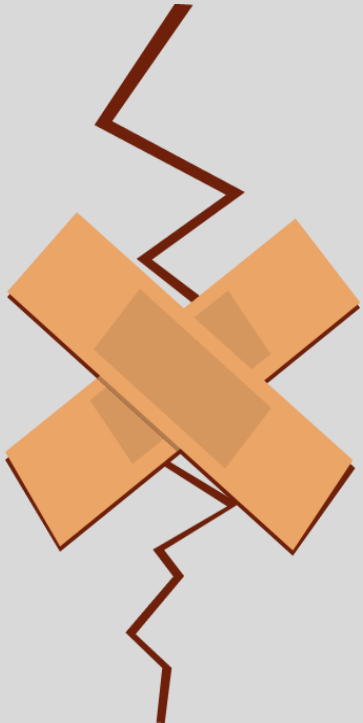
¡Fracasar es bueno! ¡Falla rápido, falla barato!

Estas afirmaciones parecen extrañas, sobre todo en nuestra cultura: **la del éxito.** Sin embargo, equivocarse es un factor clave en varias etapas del **Design Thinking.** Podríamos decir que este método espera el fracaso, o como otros lo llaman el pre- fracaso. De los errores se aprende y en eso basa su apuesta este proceso

Fallamos rápido, fallamos a menudo y aprendemos a diario de forma continua.

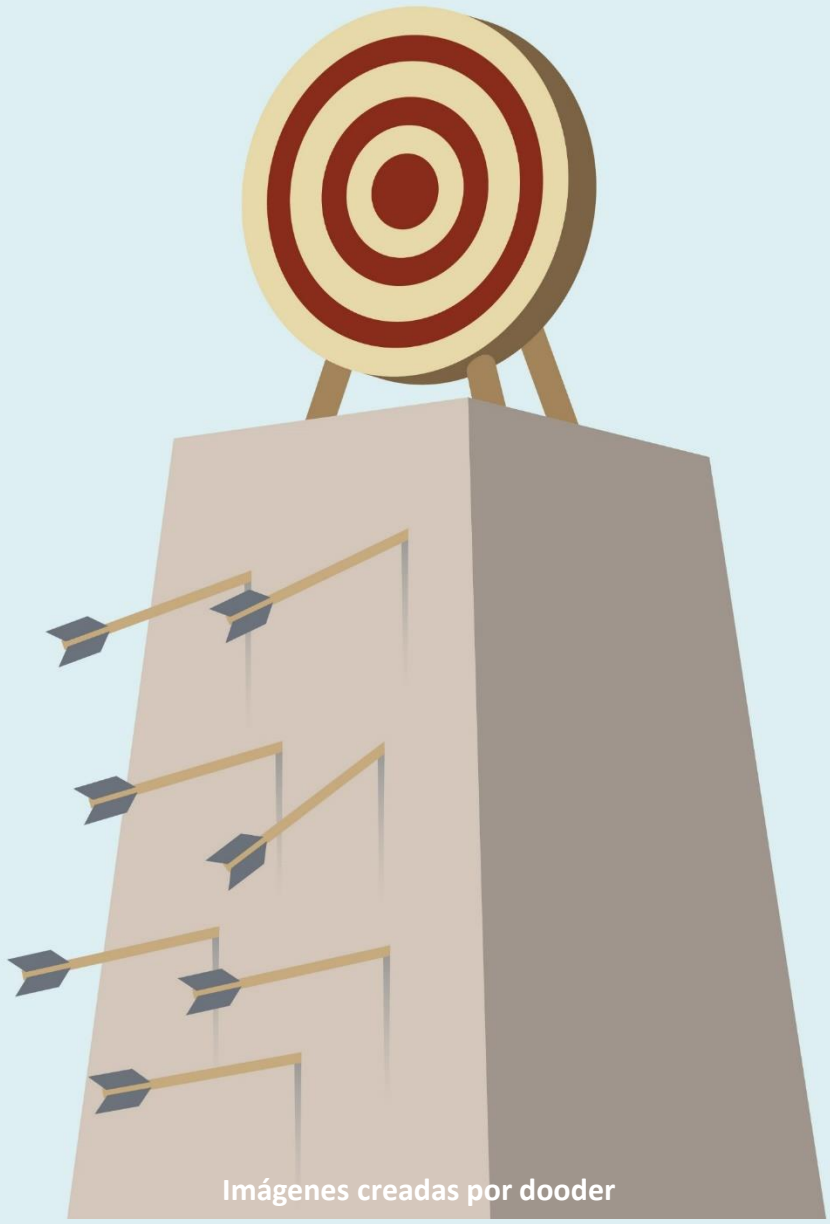
Si no fuéramos desafiados y no aprendiéramos, nuestros trabajos y nuestro trabajo serían extremadamente aburridos.

El fracaso no se debe ver como algo negativo, sino como pasos esenciales para conseguir una innovación exitosa. Simplemente es experimentación y repetición constante. Es simple: ensayo y error, y así sucesivamente hasta dar con la solución adecuada. El tema del fracaso es una cuestión de enfoque, como lo dijo Winston Churchill **“El éxito es la capacidad de ir de fracaso en fracaso sin perder el entusiasmo”.**



El fracaso es inevitable: con eso en mente empecemos con la diversión….

Los casos de no éxito de DevSecOps



Imágenes creadas por dooder

Los casos de no éxito de DevSecOps

Como lo mencionamos anteriormente usaremos CALMS, el acrónimo en inglés de **Culture, Automation, LeanIT, Measurement and Sharing** que es el pilar de DevOps / DevSecOps, para representar los casos en una modalidad de historias de café.





¿Qué salió mal?

¿Qué tan malo fue?

¿Cuáles fueron las lecciones aprendidas?

"Los casos que veremos en el E-book son representativos y están en alto nivel, también fueron modificados para no exponer información sobre clientes o conocidos"

Imágenes creadas por dooder

CASO #1

DevSec… que?



Imágenes creadas por dooder

DevSec… que? (Sharing & Culture)

Algunas empresas no logran comprender DevSecOps y como adoptarlo



En este primer caso de no éxito, luego de hablar con varios clientes y consultores amigos, vemos un número importante de empresas combinando sus equipos de desarrollo y operaciones de TI, pero la integración de DevOps con las operaciones de seguridad se está quedando atrás ya que no logran comprender como integrarse.

¿Pero porque sucede esto?

DevSec… que? (Sharing & Culture)

Luego de profundizar la cuestión con varios clientes sobre este fracaso de adopción **(Si, fracaso)** llegamos a la conclusión que uno de los grandes inhibidores de la confianza del mundo de Dev hacia DevSecOps son los anti-patrones generado por la necesidad de encajar o vender del mismo ecosistema, y estos generan múltiples discursos y dudas al momento de la adopción.

Veamos algunos ejemplos:

- **DevOps / DevSecOps es un proceso/metodología :** Es una filosofía, una forma de pensar, un nuevo paradigma.
- **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 o tarea.
- **Cambio de nombre:** Cambiar el nombre de un equipo o nombrar ingenieros con DevSecOps no significa que seas DevSecOps.
- **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.



Imágenes creadas por dooder

DevSec… que? (Sharing & Culture)

- **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? La seguridad forma parte de DevOps y antes de eso ya era una atributo de calidad
- **DevOps y la seguridad son enemigos:** DevOps y la seguridad no son enemigos. Claro, el enfoque principal de DevOps está en la entrega continua de valor. Pero eso no significa que DevOps deje completamente atrás el aspecto de la seguridad. (¿O la seguridad no es valor?)
- **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.



Imágenes creadas por dooder

DevSec… que? (Sharing & Culture)

- **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 "**



Imágenes creadas por dooder

DevSec… que? (Sharing & Culture)

Análisis de causas y consecuencias:



- Esta falta de confianza sobre la integración de la seguridad, conocida como DevSecOps podría conducir a problemas de seguridad y perdida de datos más adelante.

- Dados los frecuentes ciclos de desarrollo que son una característica inherente de DevOps, tener a seguridad como una entidad separada puede ralentizar los procesos y reducir la eficiencia.

- Comprometer la agilidad que es tan central en cualquier filosofía de DevOps o conducir a ventanas de tiempo donde se pueden liberar vulnerabilidades que no serán detectadas hasta el próximo ciclo de pruebas de seguridad.

- Entregar un producto de mier…. a nuestros clientes. Si, así es, si tu producto no tiene seguridad no tiene **CALIDAD**, ya que sus atributos son: simplicidad, consistencia /completitud, robustez, flexibilidad, performance, usabilidad, constructibilidad, escalabilidad y a ver si adivinan? **SEGURIDAD**

Imágenes creadas por dooder

DevSec… que? (Sharing & Culture)

Lecciones aprendidas

Un patrón que recomendamos en los caso de nos éxito de adopción de DevSecOps 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.

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 a más equipos de productos: el conocimiento que habrán adquirido durante este proceso acelerará el intercambio de conocimientos en los nuevos equipos que visiten.



Imágenes creadas por dooder

DevSec… que? (Sharing & Culture)

Lecciones aprendidas

As veces la colaboración de un tercero para la realización de este cambio cultural es recomendable **(la visión externa siempre es la mejor**), puede ser privado o con la contribución de las comunidades.

Gamificación: Compartir es una de las mejores forma de generar nuevas costumbre y si se hace jugando aun mejor. (Modelado de amenazas con cartas) una de las mejores formas de integrar Dev y Sec.

Nuestra recomendación final para compartir es el dojo. Popularizado por Target , los dojos ponen uno o dos equipos de productos nuevos en un nuevo entorno, en el mismo lugar durante unas semanas y establecen y persiguen un objetivo agresivo mientras practican nuevas formas de trabajo en conjunto. https://dojo.target.com/



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'** https://itrevolution.com/devops-dojo-captial-one/

Imágenes creadas por dooder

DevSec… que? (Sharing & Culture)

Lecciones aprendidas

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 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.

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 si no se mantiene en el tiempo.

DevSecOps Latinoamérica fue fundado en el año 2017, Originalmente como DevSecOps Argentina por el equipo da Cloud Legion como respuesta a las tendencias y necesidades de la comunidad fundamos la iniciativa y organización DevSecOps Argentina para profesionales de seguridad dedicados a la ciencia de cómo incorporar la seguridad dentro de las prácticas ágiles y DevOps.



CASO #2



Herramientitis aguda

Imágenes creadas por dooder

Herramientitis aguda (Automation & Culture)

Centrarse en herramientas puede ser una receta peligrosa

El caso de no éxito por excelencia en el mundo de TI y en nuestra experiencia el caso mas complicado de remediarse en las grandes empresas o las mas tradicionales (**Billetera**).

“Este proyecto fue para mejorar lo que veníamos haciendo mal, no para automatizar la caga….piiiiiiiiii”

Frase en plena reunión de seguimiento del proyecto de uno CISO de uno de los grupos de empresas mas grande de Latinoamérica, creo que esta frase es la representación fiel para este tipo de casos de no éxito, ustedes se preguntaran porque?



La automatización no es el problema, y tampoco lo es tener más opciones de herramientas. El problema radica en que los equipos de seguridad dominan la narrativa sobre cómo se debe construir el software y cómo los procesos de desarrollo deben funcionar, y obligan a la adopción de herramientas que dan forma al proceso y al flujo de trabajo en lugar de que lo inverso sea cierto.

Imágenes creadas por dooder

Herramientitis aguda (Automation & Culture)

Muchos clientes y conocidos empezaron sus proyecto de DevOps/DevSecOps por las herramientas.

Pronto se dieron cuenta lo difícil que fue encontrar las herramientas adecuadas para los procesos de cada equipo de la organización porque cada equipo (desarrolladores, operaciones de TI, seguridad, etc.) querrá usar una herramienta específica para sus prácticas en DevOps, incluso si esto dificulta la tarea.

¿Quien no tiene un vendor amigo en el placard?

Además, surgen nuevas herramientas todo el tiempo, incluso hay herramientas que ayudan a integrar otras herramientas.



Imágenes creadas por dooder

Herramientitis aguda (Automation & Culture)

Análisis de causas y consecuencias:



- Empezar DevSecOps por las herramientas nos puede traer consecuencias a medida que avanza el proyecto y nos damos cuenta que no sirve pero tenemos que seguir porque ya se compro o a veces seguir invirtiendo (El collar sale mas caro que el perro)

- No tener las herramientas adecuadas puede evitar que los equipos vean el máximo beneficio de sus esfuerzos reflejados en el famoso **“Time to market”**

- Muchos equipos de seguridad fallan en la integración con DevOps porque no entienden las herramientas, intervienen demasiado tarde y rápido influenciados por vendors e interrumpen el flujo de trabajo de ingeniería.

- Si seguridad no se comprende en el proceso de ingeniería, y las herramientas que permiten a los equipos de DevOps moverse rápidamente antes de sumarse en el pipeline Preparen la casilla de correo para recibir falsos positivos.

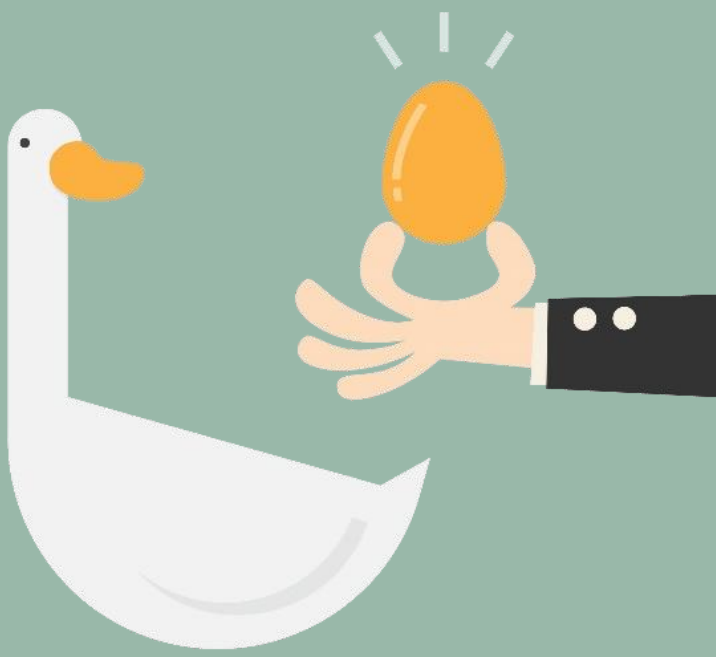
Imágenes creadas por dooder

Herramientitis aguda (Automation & Culture)

Lecciones aprendidas

No vamos a negar que la Automatización basada en herramientas no sea buena, pero no comiences tu estrategia DevSecOps discutiendo y seleccionando un montón de herramientas sin conocer la madurez de tu organización.

Es probable que escuches que algunos proveedores afirman tener la herramienta perfecta **(El ganso de oro)** para las prácticas de DevSecOps, pero eso es un cuento infantil.



Adopta un enfoque agnóstico y tene en cuenta que no existe una herramienta única que pueda cubrir todo lo que necesitas.

Imágenes creadas por dooder

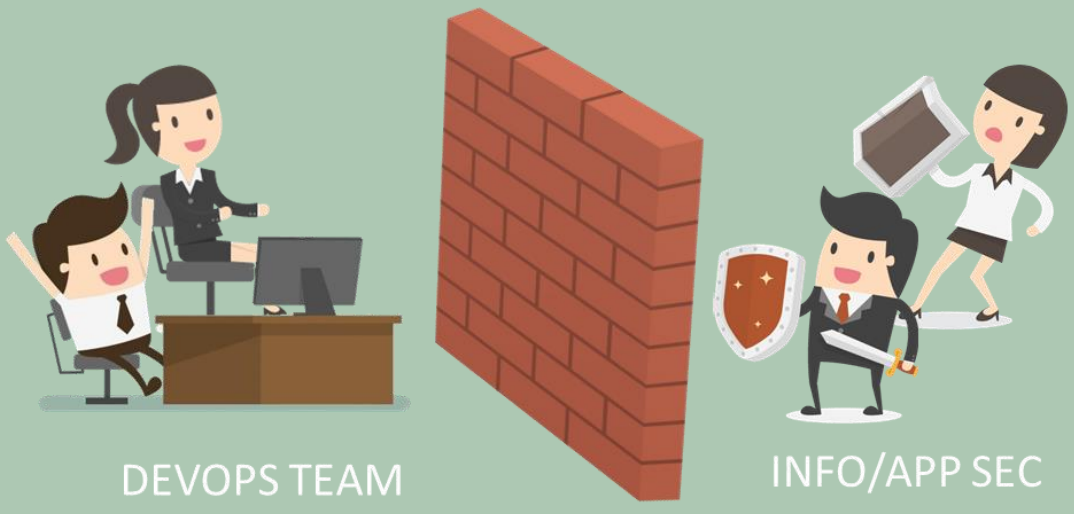
Herramientitis aguda (Automation & Culture)

Lecciones aprendidas

Muchas empresas han empezado sus proyectos implementando típicamente SONAR, OWASP ZAP, ETC… porque querían tener mayor visibilidad. **“Cuidado con lo que deseas, porque podrías conseguirlo”.**

Luego se dieron cuenta que era la caja de pandora, no solo por los posibles falsos positivos en algunos casos, sino que nunca habían estimado los FTEs necesarios del equipo de seguridad para analizarlos, o un flujo de como derivar y gestionar los hallazgos y gestionar las vulnerabiliades.

No dejes al equipo de DevOps fuera de las decisiones importantes de herramientas. ¡Hazlos tu compañero! No seas el próximo **Muro del cumplimiento**



Imágenes creadas por dooder

Herramientitis aguda (Automation & Culture)

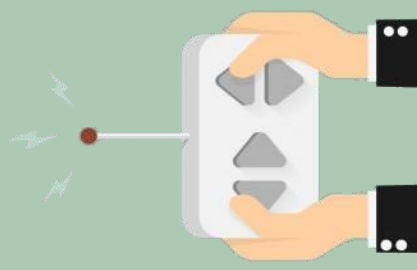
Lecciones aprendidas

Pero no podemos, y no debemos, tratar de automatizar todo en la vida. Pasará bastante tiempo antes de que todos podamos ir a tumbarnos en la playa, mientras las máquinas hacen todo el trabajo….

Pero 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 cuando pensamos en proyectos Agiles.



Todo esto se debe considerar en el contexto de la entrega continua (CD), es decir, la práctica de tener siempre el software en un estado liberable y seguro.

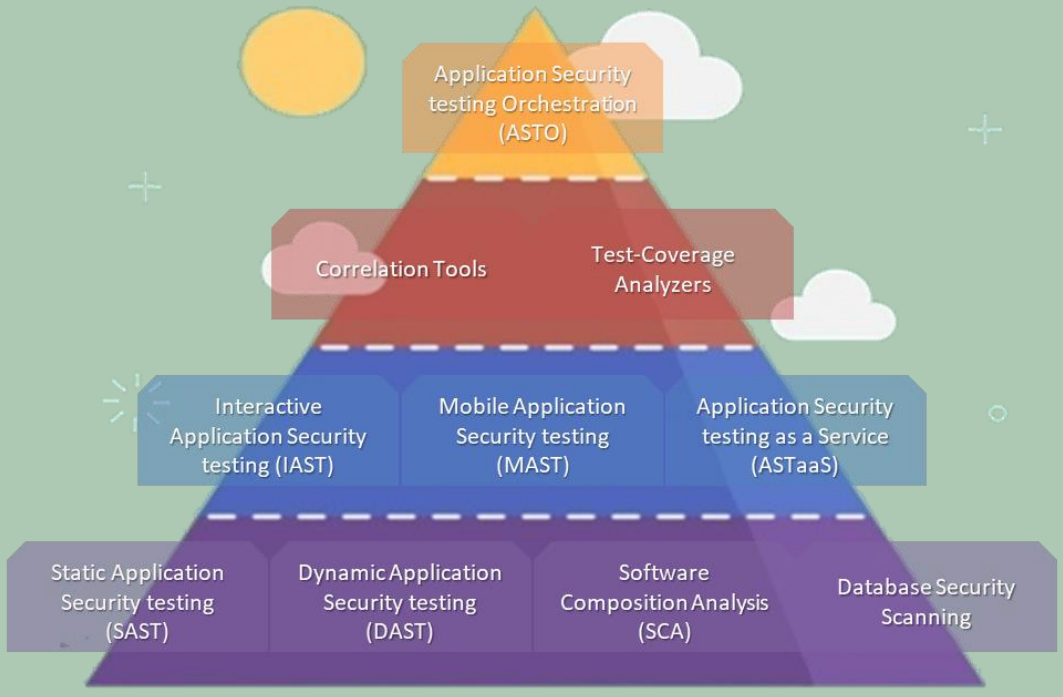


Imágenes creadas por dooder

Herramientitis aguda (Automation & Culture)

Lecciones aprendidas

Esta pirámide representa clases o categorías de Application Security testing tools .



Queres saber mas sobre automatización de los testeos de seguridad y las AST, no dejes de leer la guía práctica sobre la nueva era del testing de seguridad en https://devsecops-latam.org/



Imágenes creadas por dooder

CASO #3

TWD





Imágenes creadas por dooder

TWD (Measurement & Culture)

TWD

¿Seguro pensaron en una seria de Zombies verdad? Pero también se puede leer **The World of Dinosaurs.**

En este caso de no éxito nos basaremos en una auditoria a una empresa del rubro financiero. Donde el equipo de DevSecOps explico al auditor su proceso de backup automático en la nube utilizando almacenamiento de objetos **(Podes pensar en un Amazon Glacier…)** y algunas cositas mágicas más, pero…..

La reacción del equipo fue sorpresa cuando el auditor seguía preguntando por sus cintas de backups, más aún cuando recibieron el informe con un desvió por no poder evidenciar las misma en su estrategias de backup.



Imágenes creadas por dooder

TWD (Measurement & Culture)

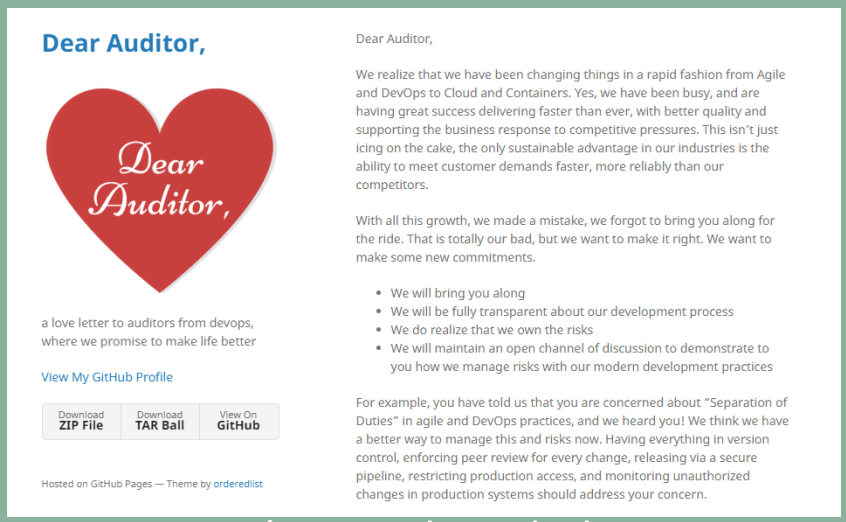
A menudo vemos a empresas implementar nuevas técnicas o practicas sin tener en cuenta todo el ecosistema del negocio y las expectativas o requerimientos de sus partes interesadas. Este caso de no éxito nos muestra un fracaso compartido:



El auditor que cual persona que **"sale a regar cuando llueve tipo checklist"**, mantuvo su postura aunque equivocada **Business as usual is good enough.**

El equipo de auditoria interna de la entidad por no lograr ser coach o nexo entre las partes **(Recuerden que en una auditoria ambas partes deben ser competentes)**

Podemos observar esta anécdota donde el equipo DevOps le escribe una carta al auditor. http://dearauditor.org/



Imágenes creadas por dooder

TWD (Measurement & Culture)

Análisis de causas y consecuencias:



- No prepararse para el cambio cultural, a nivel empresa puede resultar en problemas con las partes interesadas del negocio y hasta perdidas de puntos en una auditoria de cumplimiento.

- Falta de cumplimiento o multas por no lograr mostrar métricas y evidencias objetivas para presentar a entidades regulatorias.

- Requisitos tecnológicos cambiantes, en base a las nuevas demandas del negocio, cuando mal implementado puede generar perdida de visibilidad y trazabilidad

- Identificar los requisitos de seguridad y cumplimiento de antemano y automatizar la mayor cantidad posible. Hemos visto algunas organizaciones donde los registros de auditoría van directamente a una consola donde solo en auditor interno tiene acceso. (si no cambias la forma de pensar de tu auditor interno menos lograras con el externo)

Imágenes creadas por dooder

TWD (Measurement & Culture)

Lecciones aprendidas

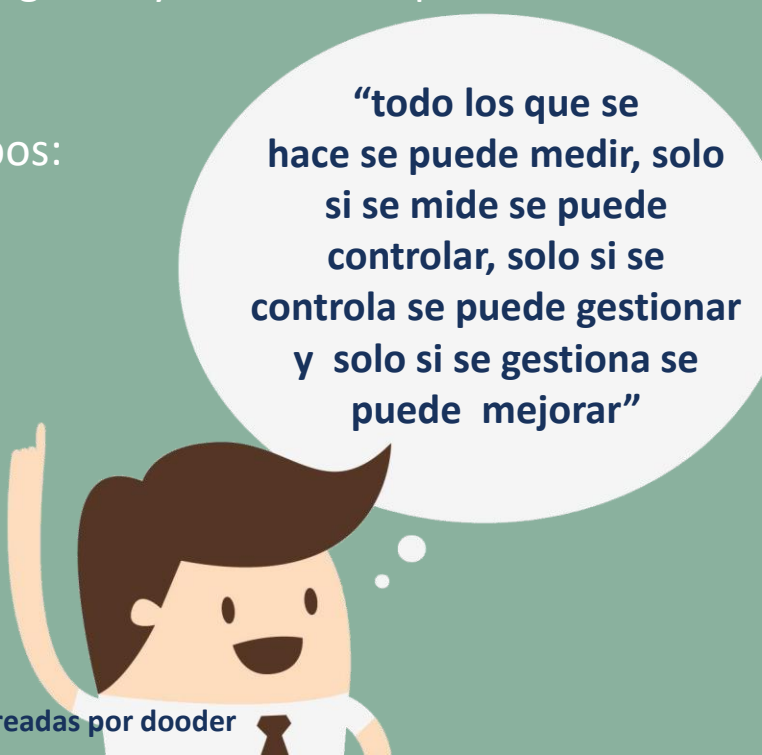
DevSecOps no es otra cosa que, un conjunto de patrones de formas de trabajo conocidas que juntas y ordenadas pueden mejoran el desempeño organizacional.

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

TWD (Measurement & Culture)

Lecciones aprendidas



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
Tiempo de cambio (para aplicaciones)	Tiempo que transcurre entre la confirmación de un código y el despliegue en producción de dicho código
Volumen de cambios (para aplicaciones)	Número de historias de usuario desplegadas en un plazo determinado
Tasa de fracaso de los cambios	Porcentaje de implantaciones en producción que han fracasado
Tiempo medio de recuperación (MTTR)	Tiempo entre un despliegue de producción fallido y la restauración completa de las operaciones de producción
Disponibilidad	Cantidad de tiempo de actividad/tiempo de inactividad en un periodo de tiempo determinado, de acuerdo con el SLA
Volumen de solicitudes de los clientes	Número de problemas comunicados por los clientes en un periodo de tiempo determinado
Tiempo de resolución de problemas del cliente	Tiempo medio para resolver un problema informado por el cliente
Time to value	Tiempo que transcurre entre la solicitud de una función (creación de una historia de usuario) y la realización del valor comercial de esa función
Time to ATO	Tiempo entre el inicio del Sprint 0 y la consecución de 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

Imágenes creadas por dooder

TWD (Measurement & Culture)

Lecciones aprendidas



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
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

Imágenes creadas por dooder

TWD (Measurement & Culture)

Lecciones aprendidas



Métricas de apoyo:

Métrico	Descripción
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
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.

Imágenes creadas por dooder

TWD (Measurement & Culture)

Lecciones aprendidas

Un Balanced Scorecard o Cuadro de Mando Integral, es una metodología de gestión estratégica que permite definir y realizar seguimiento a la estrategia de una organización.



Esta metodología permite estructurar los objetivos estratégicos de forma dinámica e integral para poder medirlos y analizarlos mediante una serie de indicadores que evalúan el desempeño de las iniciativas y proyectos.



https://riesenfeld.com.mx/wp-content/uploads/2021/09/BSC_para_DevSecOps.pdf

Imágenes creadas por dooder

CASO #4

3M



Imágenes creadas por dooder

3M (Lean, Sharing & Culture)

3M Este caso usaremos los términos japoneses **Muda, Mura y Muri** , para analizarlo.

“lo hemos hecho siempre así”
Creo que si nos pagaran por cada vez que escuchamos esta oración seriamos millonarios, La organización estatal en cuestión estaba en pleno proyecto de implementación de un pipeline DevOps de punta a punta, y alguien se acordó que deberían sumar a seguridad **(tarde pero seguro…)** se imaginan que pudo pasar no?



Antes de profundizar en el caso repasemos **Muda, Mura y Muri** son tres conceptos claves de la mejora continua Kaizen, que la producción Lean incorpora como variables protagonistas para incrementar la eficiencia.

- **Muda:** Desperdicio
- **Mura:** Discrepancia. Variabilidad del flujo de trabajo. Interrupciones en el flujo de trabajo. Tiempo muerto.
- **Muri:** Tensión. Sobrecarga de trabajo que produce cuellos de botella

Imágenes creadas por dooder

3M (Lean, Sharing & Culture)

Si pensamos en **Muri**, podemos decir que la postura del oficial de seguridad y muchas de sus políticas eran ineficientes **(Criptonita para Lean)** planillas, políticas y tareas que se podían automatizar sin ningún esfuerzo pero su irracionalidad se daba por no ser parte del proyecto desde el inicio nunca se sintió parte de mismo, prefiere la sobrecarga **(falla en la comunicación).**

Esta situación solo logro sumar irregularidad **(Mura)** inconsistencia en procesos y actividades. La postura del CISO Estuvo muy relacionado con los cuellos de botella del MVP. Es decir, existía **Mura** cada vez que se interrumpía el flujo normal del trabajo en la tarea del proyecto relacionada con procesos no optimizados de seguridad.

La relación era de pura tensión y solo generaba requerimientos o solicitudes poco razonables **(Muri)** Condiciones estresantes tanto para los trabajadores como para los procesos de trabajo.



Imágenes creadas por dooder

3M (Lean, Sharing & Culture)

Pero otra postura típica que vemos en las empresas con seguridad tradicional se acerca mas a **MUDA.** Actividades que no añaden valor desde la perspectiva del cliente y que tampoco son necesarias para operar el negocio y desde seguridad nos brinda la sensación de nunca llegar a la meta.



Burocracia: Procedimientos, documentación y papeleo innecesario que no aportan valor al resultado.

Sobreproducción: Desarrollar más características de las necesarias.

Multiproyecto: Alternar el tiempo de trabajo entre varios proyectos e interrupciones del flujo de trabajo.

Esperas: Tiempos de espera por falta de cadencia en el flujo de trabajo.

“Ir haciendo”: Encargar trabajo para ir avanzando algo no definido y así no tener paradas a las personas.

Desajustes de capacidad: Personas de gran talento asignadas a tareas rutinarias, y viceversa.

Errores: Retrabajo por bugs.

Imágenes creadas por dooder

3M (Lean, Sharing & Culture)

Análisis de causas y consecuencias:



- No cumplir con los plazos del proyecto por los tiempos del modelo de seguridad tradicional **(si el sprint mas común en las empresas es de 2 semanas**) como podes explicar a seguridad que su pentests tradicional no puede durar lo mismo que la totalidad del sprint?

- Pasajeros no deseados en producción generando un alto esfuerzo de regresiones y retrabajo por culpa de conocidos – conocidos.

- Ser noticia por un fallo de seguridad. No se trata solo si testeamos lo suficiente si no como se aprende y se elimina los desperdicios en el proceso de cada prueba, como impulsamos a las demás áreas a contar con un Security Champions.

- Por ejemplo, cuando las aplicaciones se prueban menos de tres veces al año, los defectos persisten más de 3.5 veces más que una organización que prueba de siete a doce veces al año.

Imágenes creadas por dooder

3M (Lean, Sharing & Culture)

Lecciones aprendidas

No importa de que área seas **“Sec, Dev o Ops”** nunca pierdas el Foco en la mejora continua, recomendamos a todas las organizaciones a que se vuelvan experimentales para impulsar la innovación al tiempo que reducen el riesgo.

El uso del "kata" de lean es clave para que esto se convierta en algo habitual, alejándose de una cultura de reuniones y planificación, hacia pequeñas y frecuentes mejoras incrementales. **(Quien no tuvo reuniones para planificar reuniones)**

Necesitas una cultura que estimule y recompense a los colaboradores a pensar como intra-emprendedores. A realizar pruebas y aprender en el camino.

.



Imágenes creadas por dooder

3M (Lean, Sharing & Culture)

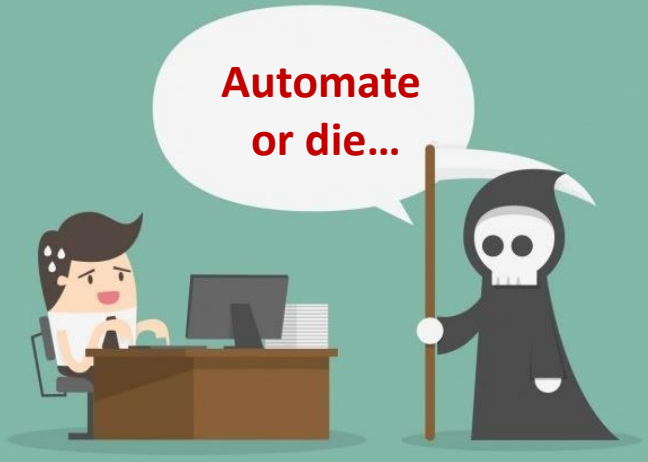
Lecciones aprendidas

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.



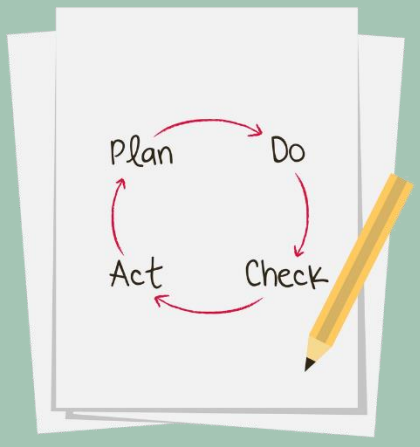
Imágenes creadas por dooder

3M (Lean, Sharing & Culture)

Lecciones aprendidas

Comenzamos por considerar 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, hacemos y verificamos, y luego actuamos en un ciclo continuo.

En nuestra experiencia, el *** do *** más importante para implementar la seguridad dentro del ciclo de vida de DevOps es la comunicación **(Sharing)**



El primer paso para comunicarse implica aprender otros idiomas, y con eso quiero decir que los equipos de seguridad deben aprender a hablar **"DevOps o Agile"** y comunicarse a través de las herramientas en las que ellos ya trabajan.



Imágenes creadas por dooder

CASO #5

Esta “el riesgo” y “el riesgo”….



Imágenes creadas por dooder

Esta “el riesgo” y “el riesgo”…. (Lean, Sharing & Culture)

Este caso en particular paso en uno de los grandes jugadores del mundo Retail.

Sabemos que para muchas empresas del rubro Retail / E-comerce el cumplimiento de PCI o SOX fueron grandes impulsores iniciales en sus estrategias de Gobierno de seguridad, gestión de riesgos y cumplimento, normalmente exigen pruebas de prácticas de seguridad y resultados de evaluaciones técnicas para demostrar que los productos y servicios de software que utilizan cumplen con sus estándares y requisitos.

¿buenas razones para que DevOps tenga seguridad y el pensamiento basado en riesgo, uno podría pensar?

Pero no, vimos en muchas implementaciones de DevOps del rubro que ese esfuerzo de seguridad solo se limitaba a los entornos alcanzados por los requisitos de cumplimiento y que normalmente eran entornos heredados, legacy y gestionados por equipos de Desarrollo Seguridad y Operaciones que no eran agiles en su cultura.

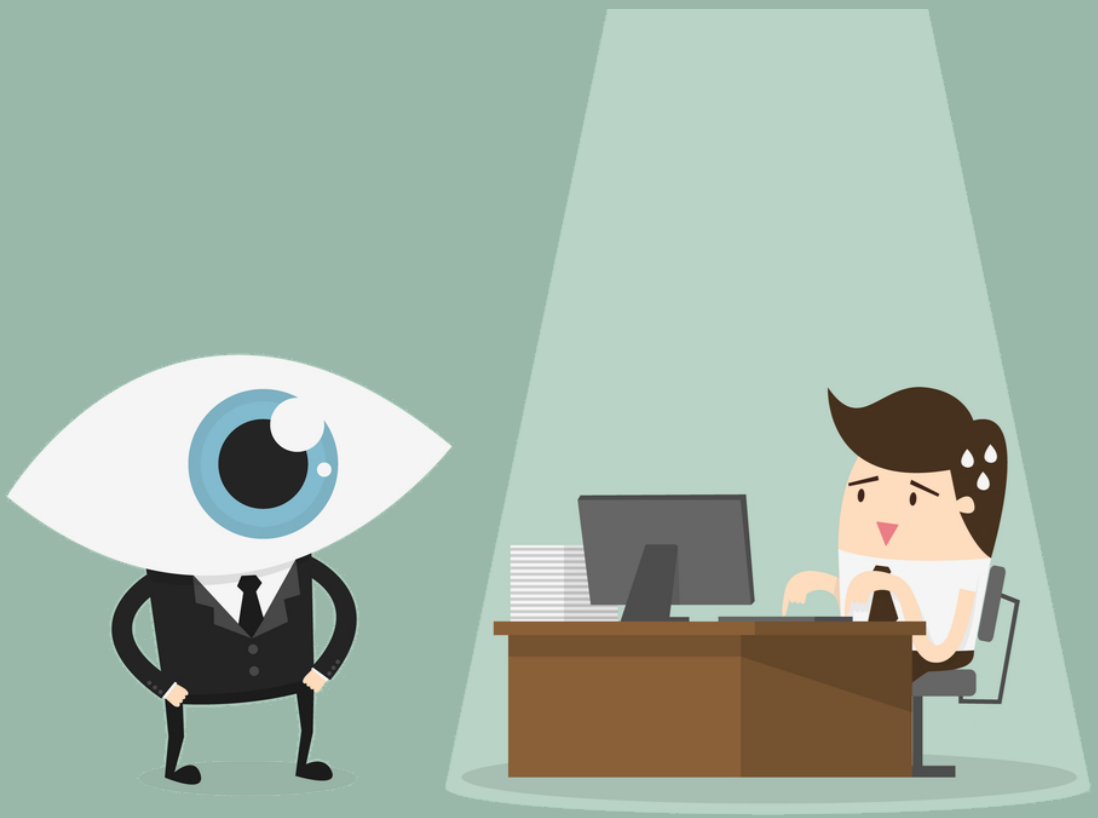


Imágenes creadas por dooder

Esta “el riesgo” y “el riesgo”…. (Lean, Sharing & Culture)

Típicamente veíamos todas las semanas los equipos de seguridad ir a los equipos de desarrollo y decir: **“Aquí hay un montón de errores. es muy importante que los arregles ahora. Por PCI o SOX“** usando el ya conocido muro del cumplimiento.

Pero en otras ocasiones siguiendo la misma postura y lógica cuando el equipo de seguridad visitaba a desarrollo para indicar los riesgos de los sistemas sin el respaldo de un requisito de cumplimiento básicamente le cerraban la puerta en las narices o dejaban de asistir a las reuniones. Lo cual era justo, en realidad, Seguridad solo les estaba dando trabajo extra para hacer sin contemplar sus agendas o objetivos.



Imágenes creadas por dooder

Esta “el riesgo” y “el riesgo”…. (Lean, Sharing & Culture)

Para los profesionales de la seguridad y desarrollo, esto puede sonar muy familiar. Es realmente un fastidio, porque puede parecer que realmente no estás progresando en el trabajo. Encontramos todo tipo de problemas de seguridad, pero eso es solo la mitad de la solución.

Por supuesto, nadie quiere ver el nombre de su empresa en los titulares debido a una brecha de seguridad.



“Nadie incluyendo desarrollo quiere una brecha o riesgo de seguridad, punto”.

Imágenes creadas por dooder

Esta “el riesgo” y “el riesgo”…. (Lean, Sharing & Culture)

Análisis de causas y consecuencias:



- Los equipos de seguridad pueden ganarse la reputación de "personas que dicen que no" cuando arrojan montones de errores de seguridad de software en el regazo de los desarrolladores y exigen que se corrijan.

- Comprender el punto de vista de la otra persona y las necesidades del negocio son esenciales para lograr los objetivos de mitigación de riesgos de un equipo de seguridad.

- En un mundo DevOps donde el software genera ingresos, el software seguro protege los ingresos. La seguridad debería dejar de ser un centro de costes a un motor de negocio.

- Es fundamental que los profesionales de la seguridad adopten un enfoque que sea curioso sobre otros equipos y el negocio (no solo cumplimiento). “Solo asociándonos con otros podemos asegurar la tecnología que construimos, compramos, vendemos y operamos”.

Imágenes creadas por dooder

Esta “el riesgo” y “el riesgo”…. (Lean, Sharing & Culture)

Lecciones aprendidas

En lugar de ir a los equipos de desarrolladores y decirles: "Aquí hay una gran cantidad de errores", diríamos: "Según las conversaciones que hemos tenido con ustedes, sobre su negocio, la arquitectura de su aplicación y cómo funciona, hemos creado para usted un perfil de riesgo”. Así es como creo que deberíamos abordarlo como grupo. Obtener la aceptación y priorizar conjuntamente los compromisos funcionales y de seguridad a un alto nivel **"ayuda a mantener una carga de trabajo uniforme en el equipo de implementación"**

Comunicarse claramente con el equipo de manera significativa con plazos negociados y razonables para la remediaciones. Ejemplo el 40 % no es un número que le guste a la gente de seguridad. A la gente de seguridad le gusta un número como el 95%. La cuestión es que si decimos **"vamos a intentar eliminar el 95 % de los errores en un sistema“** probablemente vamos a recibir la misma respuesta que recibíamos antes: los equipos de desarrollo dejarían de invitarnos a sus reuniones, dejarían de venir a nuestras reuniones e incluso podrían dejar de leer nuestros correos electrónicos.



Imágenes creadas por dooder

Esta “el riesgo” y “el riesgo”…. (Lean, Sharing & Culture)

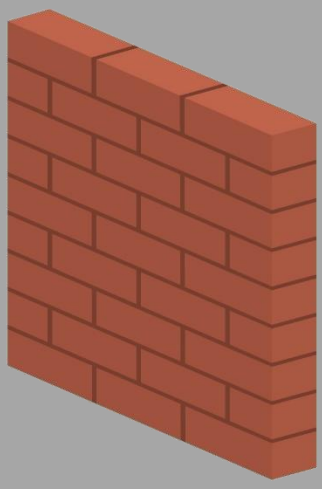
Lecciones aprendidas

La percepción de **“la gente que dice que no"** prevalece especialmente en organizaciones que tienen silos entre desarrollo, operaciones y seguridad. En esos ambientes, se cree que la única forma de cruzar la cerca y hacer las cosas es ir a la guerra con otros equipos.

En esa empresa vimos que para optimizar el proceso de seguridad. Tomaban una política de 300 páginas y reducían a 50 páginas. Luego trataban de empujarla por la garganta a la gente. El enfoque sugerido implico en organizar un taller de creación de políticas, invitar a los equipos de las partes interesadas y decirles: “Cotizamos en bolsa, y eso significa que debemos cumplir con SOX”. Esto es lo que tenemos que hacer. Hablemos al respecto y establezcamos prioridades en conjunto. Desde una perspectiva de seguridad, aquí está el listón que debemos cumplir.

DevSecOps promete romper los silos, **"Sin embargo, desde nuestra experiencia, la práctica en sí no tiene posibilidades de hacerlo si no existe primero una cultura de cooperación entre los equipos".**





Imágenes creadas por dooder

CASO #6

C – de compromiso no solo de C - Level



Imágenes creadas por dooder

C – de compromiso no solo de C – Level (Sharing & Culture)

En este caso lo vamos a evaluar de una manera más amplia ya que es muy común en la mayoría de las empresas en que estuvimos realizando diagnósticos de nivel de madurez de las prácticas de DevOps y DevSecOps.

Como lo dice el título del caso en muchas de las iniciativas de DevSecOps lograr el compromiso de los C-Levels es una de las tareas más complicadas al momento de justificar un programa integral de cambio cultural tan ambicioso como puede ser DevOps o DevSecOps.

En muchos casos se hizo casi imposible mostrar a los C-levels el amor que los equipos tienen por sus programas agiles y como quieren ayudar a realizar el objetivo original.

Por desgracia, algunos decisores al estar protegido por tantos niveles de ejecutivos egoístas, que as veces es difícil encontrar el correcto vector para comprometerlo.



“Si supieran que impacto masivo hubieran logrado en los equipo si vinieran y caminaran por el programa con nosotros”

Imágenes creadas por dooder

C – de compromiso no solo de C – Level (Sharing & Culture)

Vimos en muchas empresas planteos similares por partes de los participantes de los programas de DevOps/ DevSecOps

¿Si el programa existe para ayudar a la empresa a lograr el objetivo que ustedes establecieron, entonces ¿no es un mensaje poderoso cuando vienen y descubren por parte de la gente cómo está funcionando su programa?

“Ven y cuéntanos cuán importante es ese objetivo para tus objetivos estratégicos”.

Igualmente, si nos preguntamos cómo van las cosas, eso envía un poderoso mensaje cultural. "¡Las comunicaciones están abiertas!" Estoy llamando a tu puerta diciendo "Hola amigo, ¿cuáles son las noticias, ¿cómo estás?“

Creemos que siempre es interesante cuando interactúan los C-levels. El éxito puede depender de varios factores. Una transformación tecnológica puede y tocará todo el negocio, lo que significa que una gran parte de esa transformación tiene como la pieza mas importante las personas y su cultura.



Imágenes creadas por dooder

C – de compromiso no solo de C – Level (Sharing & Culture)

Análisis de causas y consecuencias:



- Uno de los errores mas comunes que vemos por parte de los C-levels es subestimar el poder de la cultura en estos procesos de cambio (Como lo dijo Peter Ducker – "La cultura come la estrategia en desayuno, almuerzo o cena“ Depende quien haya traducido).

- Ningún líder empresarial quiere incorporar el secretismo y la deshonestidad en sus culturas pero vimos que lograr la honestidad y transparencia en la cultura de desarrollo de software de una organización puede ser una tarea difícil. Diferentes personajes, egos, políticas y agendas no alineadas se interponen en el camino. No hay tecnología que pueda resolver eso"

- Por lo general, el tema de la seguridad de la aplicación lo presenta y promueve una persona de seguridad que, en última instancia, no va a ser el usuario principal de los sistemas o sufrirá el impacto relacionado a un incidente de seguridad “Un incidente es una inversión no planificada, y si no lo ves de esa manera como un líder, no estás obteniendo un retorno de la inversión que ya se hizo en tu nombre".

C – de compromiso no solo de C – Level (Sharing & Culture)

Lecciones aprendidas

Vimos en muchas empresas que lo Líderes audaces que quieren cambiar la cultura del desarrollo de software en una organización necesitan **controlar la narrativa del cambio.** "No se arregla mágicamente un programa completo o incluso cada uno de sus componentes, pero si estableciendo una narrativa común“

Si la narrativa esta siendo impulsadas directamente por el C-level, tenemos la confianza explícita. Con ese historial establecido, El respaldo en el nivel superior está allí.

Si estás siendo impulsada por alguien por debajo del C-level, podríamos necesitar un par de experimentos del tamaño de un invernadero en el medio del negocio. Podemos obtener algunas victorias en nuestro haber, aumentar el apetito de riesgo y lograr subir a bordo esos C-levels para el resto del viaje.



De todas maneras, el patrocinio ejecutivo es crucial para el éxito de cualquier programa de transformación.

CASO #7

Rápidos y muy furiosos





Rápidos y muy furiosos (Culture, Automation)

Automatización por demás sin control, donde la velocidad gana a la diciplina

En este caso de no éxito, luego de escuchar a varios clientes y conocidos, vemos un número importante de empresas que para superar a los competidores, deben ofrecer Software a mayor velocidad. Se concentran principalmente en los plazos en lugar de en la calidad (Seguridad) del producto. Lo único que les importa es ejecutar el código en producción.

El 70% creen que carecen de conocimientos adecuados sobre DevSecOps

El 52% de las empresas sacrifican la ciberseguridad por la rapidez



“DevSecOps está ganando popularidad a un ritmo vertiginoso, pero eso tiene un precio”.

El objetivo final del equipo de DevSecOps debería ser ofrecer un canal seguro y funcional. Y si prioriza la velocidad mientras crea esta funcionalidad e integra la seguridad en el proceso, puede perderse algún aspecto importante y generar riesgos, pero sobre todo frustración y enojo.

Rápidos y muy furiosos (Culture, Automation)

Luego de analizar este caso de no éxito, que es más común de lo que las empresas y los vendedores de soluciones mágicas les gustaría en nuestra región, podríamos afirmar que el error más común, que también es el más fácil de solucionar es:

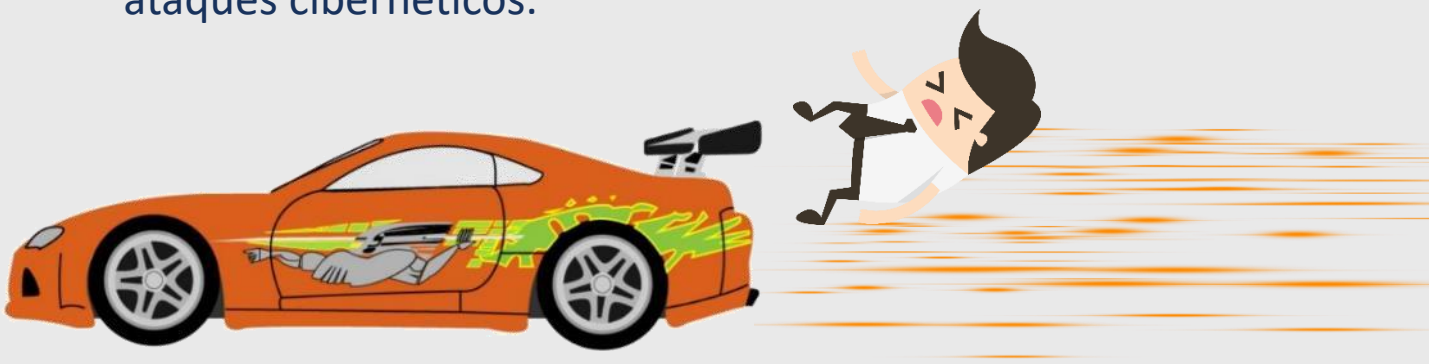
“La automatización sin estrategia”

Los esfuerzos de DevSecOps fracasan cuando la velocidad gana a la disciplina.

El deseo de ofrecer una funcionalidad mejorada hace que sea fácil pasar por alto los procesos de seguridad, incluidas las pruebas, la aplicación de parches y la supervisión.

La falta de segregación de funciones puede permitir un acceso excesivo a la producción y la implementación de cambios no autorizados.

El producto final podría tener errores y ser vulnerable a ataques cibernéticos.



Rápidos y muy furiosos (Culture, Automation)

Análisis de causas y consecuencias:

Aunque los líderes tecnológicos recurren cada vez más a las prácticas de DevSecOps, la mentalidad anterior a DevSecOps ha condicionado a los equipos a priorizar la velocidad sobre la seguridad y la calidad.

El despliegue de productos en el mercado y el tiempo de comercialización es la principal preocupación y un aspecto crucial del ciclo de desarrollo, no debería ser una razón para sacrificar la seguridad.

Se considera que simplemente activar escaneos dentro de los pipelines de manera automatizada es lograr DevSecOps, mientras que se presta muy poca atención a las etapas óptimas del SDLC.





Rápidos y muy furiosos (Culture, Automation)

Análisis de causas y consecuencias:

Las capacidades mejoradas de identificación de vulnerabilidades son efectivas, pero todos esos nuevos tickets de errores se acumulan más rápido de lo que los equipos de ingeniería pueden eliminar, lo que genera una percepción de conflicto entre innovación y seguridad .

No pasa mucho tiempo antes de que los desarrolladores comiencen a sentirse restringidos, frustrados y provoquen disturbios. Esto rápidamente pasa factura a la adopción de DevSecOps, ya que es imposible ir al grano sin tener desarrolladores de su lado.

Mantener a los equipos de desarrollo al margen mientras se diseña la automatización es una mala idea e impide el éxito de la adopción de DevSecOps a largo plazo.





Rápidos y muy furiosos (Culture, Automation)

Lecciones aprendidas

Desde hace tiempo se acepta que los avances en la entrega de software están ayudando a las empresas a definir quiénes son y cómo compiten. **Todas las empresas están bajo presión para innovar rápidamente y a veces sin estrategia.**

Aunque según CALMS, la automatización es esencial para la adopción exitosa de DevSecOps, **vemos un enfoque algo defectuoso en la práctica** como vimos en el primer E-book en el caso **Herramientitis aguda** .

Iniciar análisis dentro de los pipelines ya no es un problema, ya que casi todos los proveedores de seguridad ahora ofrecen integraciones nativas con los principales pipelines.

El enfoque correcto es automatizar las pruebas de seguridad sólo después de examinar cuidadosamente las necesidades de las aplicaciones y acordar el enfoque correcto con los equipos de desarrollo operaciones y seguridad.



Rápidos y muy furiosos (Culture, Automation)

Lecciones aprendidas

Aceptar que los errores ocurren. Los errores forman parte del proceso y DevSecOps reconoce que las organizaciones nunca mantendrán las vulnerabilidades fuera del ciclo de vida del Software, ni detendrán por completo a los delincuentes informáticos.

Integrar los principios de seguridad en todo el ciclo de vida y vincularlos a las estrategias generales de riesgo de la organización minimizará los problemas en el futuro.

Para ello, el sector debe reflexionar sobre el concepto de desplazamiento a la izquierda y garantizar que la seguridad se incorpora antes, no después. Como resultado, la seguridad se incluirá en todo el proceso del SDLC.



CASO #8

El traje nuevo del rey



Imágenes creadas por dooder

El traje nuevo del rey (Culture)

Intentar crear DevSecOps manteniendo intactas las estructuras antiguas es como “El traje nuevo del rey”



Como en el cuento de **Hans Christian Anderse**, pudimos observar en varias empresas que Scrum es el estándar de facto para el desarrollo de Software. A medida que abandonan Waterfall por Agile (Scrum), muchas quedan atrapadas en lo que denominamos Scrumfall

Lo mismo está pasando con DevSecOps, las organizaciones no logran adoptar de manera efectiva los principios Agile o de CALMS y como en el cuento, caen en las manos de los que afirman **vender la mejor ropa del mundo**. Pero, sin embargo, sólo las personas inteligentes pueden ver la ropa…

DevSecOps como Scrum se trata de patrones y nos brinda soluciones reutilizables para problemas comunes. **¿Qué pasa si no vemos los patrones?**

Fingimos verlos, tal como los estafadores y el pueblo finge ver la ropa ☹

Imágenes creadas por dooder

El traje nuevo del rey (Culture)

Podríamos reírnos de esto, pero es triste ver cómo DevSecOps se vende a los decisores como en el cuento, **afirmando que DevSecOps es la mejor ropa del mundo** y les hará tener éxito.

Así que los decisores toman un curso de dos días para aprender sobre DevSecOps, y **todo lo que han aprendido son fundamentos**, responsabilidades y qué eventos realizar, además del nombre de algunas ASTs. **(También se consigue en Wikipedia ☺)**

Luego se ponen el traje nuevo y hablan de transformaciones Ágiles en base a DevSecOps. **Y nadie en la corte del emperador se atreven a cuestionar lo que están haciendo, temerosos de parecer tontos**.

De hecho, todo el mundo puede ver la nueva ropa de DevSecOps. **En cambio, deberíamos entender los patrones,** tales como los que Scrum intenta y no lo logra hace tiempo.



Imágenes creadas por dooder

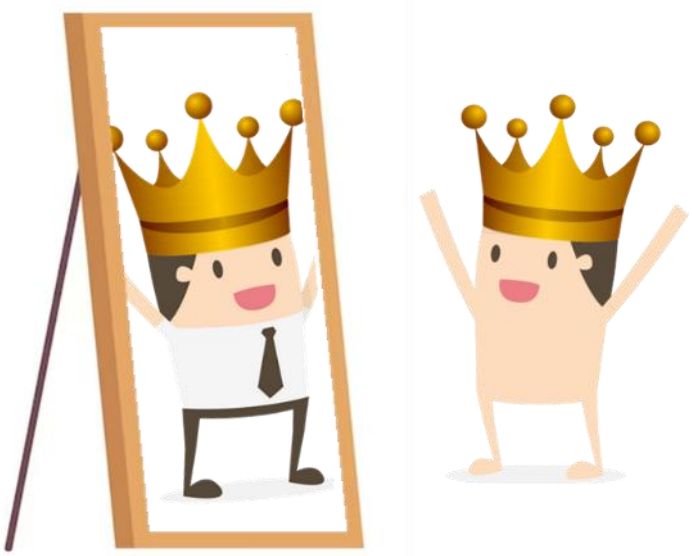
El traje nuevo del rey (Culture)

Análisis de causas y consecuencias:

Como se mencionó anteriormente, **incorporar DevSecOps a una organización bien establecida** ya sea con Waterfall, Scrum, Kamban, etc… **puede causar un choque cultural e inicialmente, cierto rechazo** por parte de las bases.

Vemos muchos equipos DevSecOps pomposos , pero que no reciben retroalimentación del usuario final y **no conoce si el traje fue a medida ya que no participa del Tailoring inicial**. No saben los riesgos reales de valor, usabilidad, viabilidad, seguridad y compliance. **Estan limitado a una visión de amenazas técnicas y punto.**

Empiezan DevSecOps sin cambiar la base, omitir elementos, no comprender sus ideas o patrones subyacentes que intenta promover, **solo encubre problemas y limita los beneficios de DevSecOps** potencialmente incluso volviéndolo inútil



Imágenes creadas por dooder

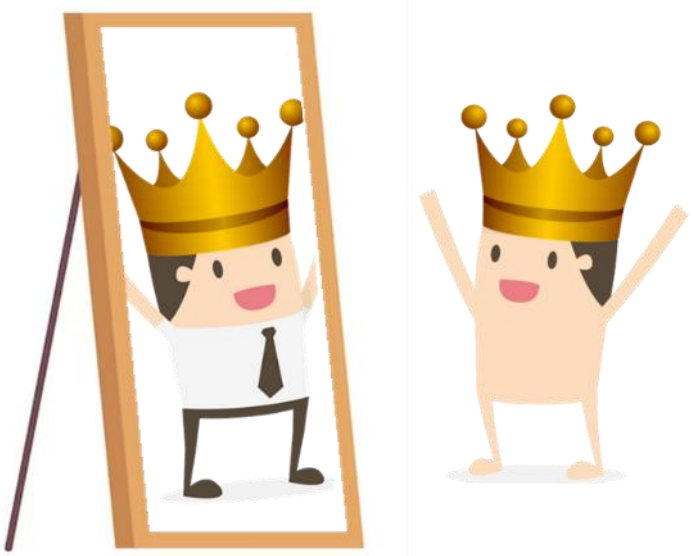
El traje nuevo del rey (Culture)

Análisis de causas y consecuencias:

Tanto CALMS como el Manifiesto Ágil pinta un cuadro muy atractivo. **El problema es que casi siempre se implementa en lugares de trabajo dedicados al resultado finales.** Cuando un director de proyecto se ve atrapado entre una promesa a un cliente y las prioridades de los DEVs... **¿Que puede malir sal?**

DevSecOps muchas veces queda en el medio de la tormenta, como el que demora el proceso de entrega. Sin embargo, **el Software mal estructurado y construido también tiene la capacidad de ralentizar** la entrega.

La seguridad es cada vez más importante, **muchas organizaciones todavía tienen un enfoque clásico** sobre cómo manejar la seguridad. ”Esto es para verificar la seguridad al final del ciclo de vida del desarrollo”.



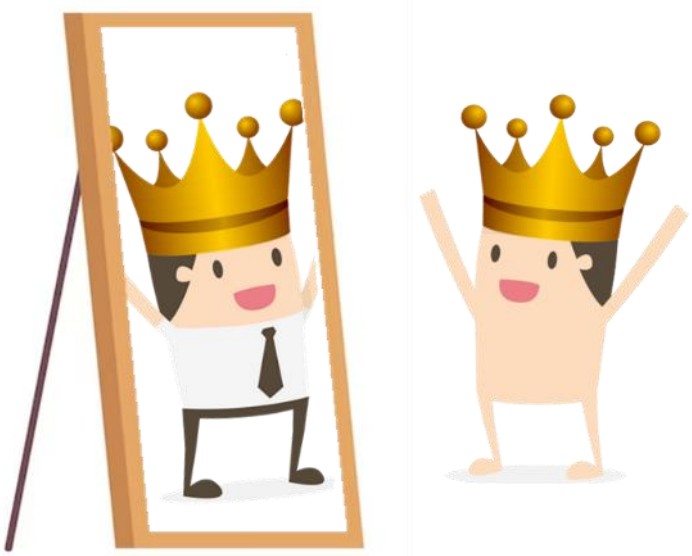
Imágenes creadas por dooder

El traje nuevo del rey (Culture)

Análisis de causas y consecuencias:

Lo que no funciona es un enfoque **“ Fake Agile ”** que cede completamente a este retroceso cultural. Lo que algunas organizaciones intentan hacer, en un intento equivocado de mantener intacta la vieja cultura, terminan con un **Frankenstein con pedazos de mini cascada, ceremonias Agiles, múltiples metodologías por células, etc…**

La grieta y el fanatismo cultural: Me han dicho que tener una historia de diseño es Waterfall . Me han dicho que los documentos de arquitectura son Waterfall, que realizar Pentest es Waterfall (Para los fanáticos del culto AGILE significa que **cualquier cosa más grande de lo que un desarrollador puede hacer en dos semanas es inviable.** Esto sucede a menudo cuando el equipo que crea el Software no se considera parte interesada.



Imágenes creadas por dooder

El traje nuevo del rey (Culture)

Lecciones aprendidas

No uses la ropa nueva de DevSecOsp sin un buen Tailoring. Atrévete a cuestionar DevSecOps y **desafiar el "statusquo".**

DevSecOps como Agile fomenta la inspección y la adaptación, pero debemos aplicar este patrón al propio DevSecOps. **(no existe una única ropa = Metodología)** de lo contrario, es posible que te encuentres desnudo y sintiéndote tonto cuando alguien te haga una pregunta simple, como donde se encuentra el modelado de amenazas.



Cualquier equipo Agile se expandirá para incluir todas las tareas necesarias en el equipo para poder seguir siendo Agile. Por lo tanto, **todo lo que impida el envío se absorbe o se elimina**."

Imágenes creadas por dooder

El traje nuevo del rey (Culture)

Lecciones aprendidas

Si estas considerando DevSecOps simplemente porque **"es el futuro"** o para mantenerte al día con tu competencia, en lugar de por el deseo de reconstruir y mejorar fundamentalmente sus procesos de negocio, **el éxito es muy poco probable.**

DevSecOps requiere tiempo y esfuerzo, requiere un liderazgo y una defensa sólida y debe medirse de una manera que sea compatible con los objetivos de su organización.

Esperemos que DevSecOps sea el niño que exclame que el rey está desnudo y no el nuevo SCRUMFALL.



Imágenes creadas por dooder

CASO #9

Planifinoaccion



Imágenes creadas por dooder

Planifinoaccion (Culture, LeanIT, Measuring)

El exceso de planificación puede bloquearnos debido a la parálisis del análisis.

En este caso que analizaremos, suele ser uno de los grandes **STOPER** de la industria, el cual vemos a menudo en muchas organizaciones que trabajan con desarrollo de Software, sin limitarse a un DEV, PO, QA, SEC o SM. Todos caen en algún momento en la trampa de analizar demasiado un tema y bloquearse.

Llamémoslo miedo, perfeccionismo, pereza o falta de concentración. Pasar todo el tiempo analizando y no actuar conduce a la **"parálisis del análisis".**

Con este contexto y problema típico **(heredado muchas veces a nivel cultural),** muchas empresas piensan en DevOps o DevSecOps como la solución.



Pero **¿cómo se implementa DevSecOps?,** si pensamos uno de los principales desafíos es que no existe un proceso estándar, ya que cada equipo tiene necesidades / objetivos diferentes y la gran cantidad de recursos y herramientas puede provocar un nuevo **"parálisis del análisis".**

Imágenes creadas por dooder

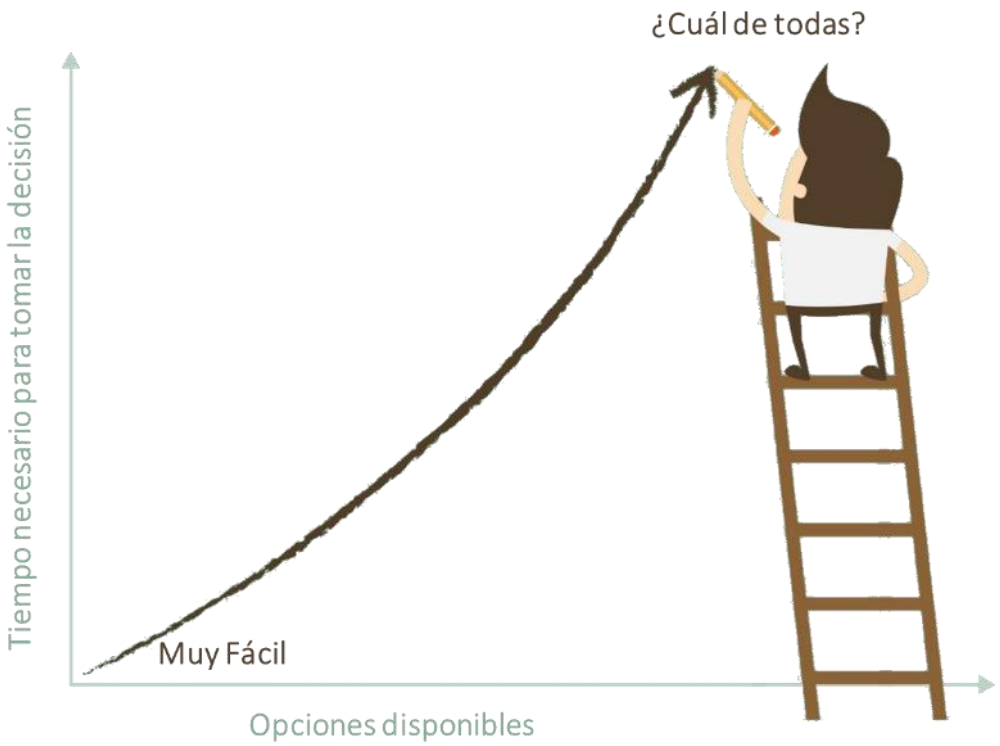
Planifinoaccion (Culture, LeanIT, Measuring)

¿De dónde son las raíces de la deuda procesal conocida como **“parálisis del análisis”?**

Probablemente haya muchas raíces de la deuda, pero creo que muchas de ellas proceden de la **mentalidad Waterfall**.

En Waterfall, la idea era elaborar por adelantado todos los escenarios y resultados posibles para minimizar el riesgo. Irónicamente, esta **"parálisis por análisis"** extrema casi siempre llevaba a construir cosas equivocadas, ¡que era exactamente lo que se intentaba evitar!

Theodore Roosevelt dijo una vez: “En cualquier momento de decisión, lo mejor que puedes hacer es lo correcto. Lo peor que puedes hacer es no hacer nada”.



Imágenes creadas por dooder

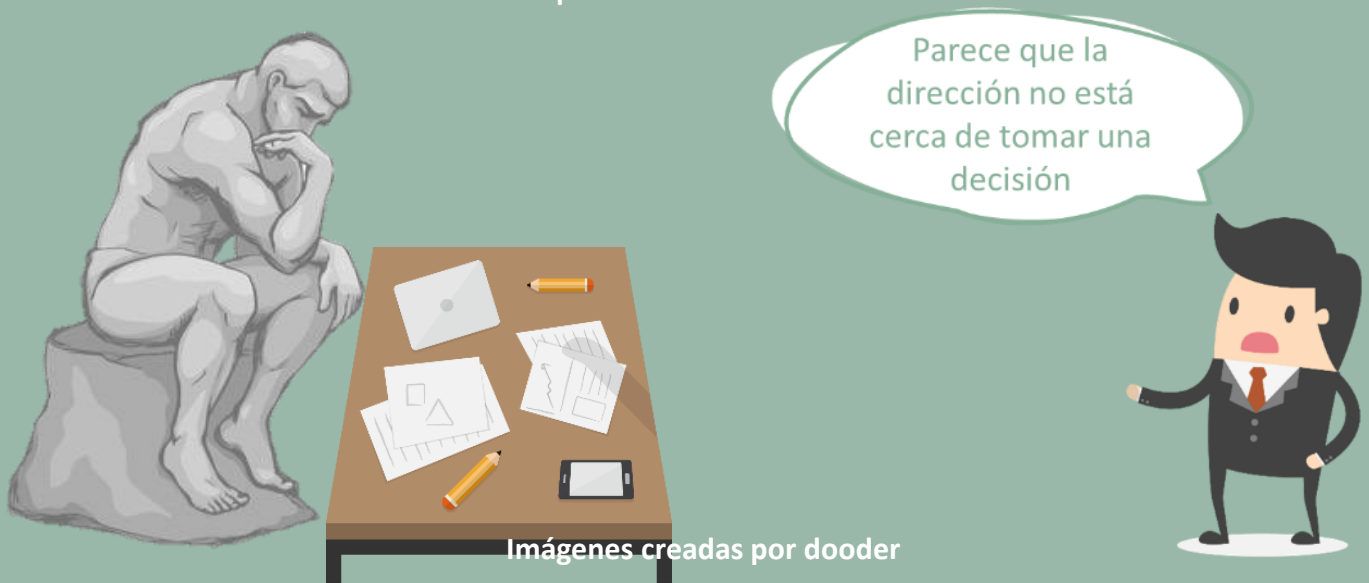
Planifinoaccion (Culture, LeanIT, Measuring)

Análisis de causas y consecuencias:

Comprender el todo, pero hacer algo a veces. El mayor enemigo del progreso iterativo es el deseo de comprender y darse cuenta de todo. Esto puede llevar a lo que a veces llamamos la “parálisis del análisis”, en la seguridad pasa mucho esto, donde se dedica tanto tiempo a analizar la situación que nunca se hace nada al respecto. (un ejemplo podría ser la adopción de la nube)

Comprender el panorama general es importante, pero también lo es progresar. El ecosistema cambia constantemente, por lo que la retroalimentación es esencial. El cambio ocurre constantemente, por lo que es muy importante buscar y utilizar la retroalimentación en todo momento y en todos los niveles y/o áreas.

Rápido no significa incompleto El hecho de que una iteración sea lo suficientemente pequeña para realizarse rápidamente no significa que no deba incluir todos los elementos necesarios para el éxito.



Imágenes creadas por dooder

Planifinoaccion (Culture, LeanIT, Measuring)

Análisis de causas y consecuencias:

Cualquier iteración debe producirse de acuerdo con el concepto de producto mínimo viable. Un producto mínimo viable es una versión del producto final que permite la máxima cantidad de aprendizaje validado con el menor esfuerzo. Eso también aplica a seguridad MVPS.

La parálisis del análisis puede afectar a cualquier tipo de empresa de Software, pero podría ser más frecuente en las reguladas. Esto se debe a que las empresas de Software reguladas tienen que cumplir con estándares y regulaciones estrictos.

También puede afectar los procesos de desarrollo de Software de diferentes maneras. Por ejemplo, una empresa de Software regulada podría tener que lidiar con cuestiones legales, éticas o de Seguridad que requieren un análisis y una documentación cuidadosa.



Imágenes creadas por dooder

Planifinoaccion (Culture, LeanIT, Measuring)

Análisis de causas y consecuencias:

Dedicar demasiado tiempo a planificar, diseñar e investigar la mejor solución, en lugar de codificarla y probarla. Esto puede resultar en incumplimiento de plazos, desperdicio de recursos y productos de baja calidad.

Otro ejemplo podria ser cuando un gerente de producto se siente abrumado por los datos y los comentarios de diversas fuentes y no puede priorizar las características o requisitos del producto. Esto puede generar una visión poco clara, un alcance impreciso, la insatisfacción del cliente o vulnerabilidades.

Intentar tomar una decisión sobre cómo proceder con la transformación digital en toda la organización es una tarea monumental. Se corre el riesgo de introducir muchos cambios variables, que a la vez que pueden volverse caóticos si no se gestionan bien.



Imágenes creadas por dooder

Planifinoaccion (Culture, LeanIT, Measuring)

Lecciones aprendidas

La parálisis del análisis es uno de los **mayores desafíos que enfrentan los líderes empresariales en la actualidad**. Una organización que gasta dinero en un proveedor de tecnología o servicios, comprensiblemente, quiere resultados.

Las posibilidades son casi infinitas. De hecho, son tan interminables que los equipos a menudo **se ven superados por la gran cantidad de herramientas, prácticas y protocolos disponibles para ellos**.

Puede ser abrumador y las empresas pueden verse tentadas a ser demasiado codiciosas al seleccionar sus prácticas y herramientas DevSecOps.

Esto puede provocar **fatiga en la toma de decisiones o parálisis del análisis.** En última instancia, el objetivo principal de DevSecOps a veces termina omitido o bloqueado



"El miedo es el camino hacia el lado oscuro. El miedo lleva a la ira. La ira lleva al odio. El odio conduce al sufrimiento". — Yoda



Imágenes creadas por dooder

Planifinoaccion (Culture, LeanIT, Measuring)

Lecciones aprendidas

Establecer prioridades en toda la organización no significa que se deba intentar “hervir el océano”. De hecho, deberíamos hacer lo contrario: empezar poco a poco y luego ampliar la estrategia una vez que se tenga éxito.

Estrategia para superar la parálisis del análisis

Establecer objetivos y plazos claros y realistas para la toma de decisiones.

Definir el alcance y los criterios de la decisión y atenerse a ellos.

Limitar la cantidad y las fuentes de información.

Buscar comentarios y consejos de expertos, pares o clientes.

Usar datos y evidencia para respaldar la decisión y documentar el fundamento.

Aceptar que no existe una solución perfecta y estar abierto al aprendizaje y la mejora continua.

Adoptar métodos de desarrollo ágiles e iterativos para ofrecer valor de forma rápida e incremental mientras se aprende de los errores.

CASO #10

Security Champion o al Champiñón



Imágenes creadas por dooder

Security Champion o al Champiñón (Culture, Sharing)

En este caso contare una experiencia muy cercana. “Lucho” veremos que si bien, los **Security Champion** han sido una práctica de seguridad probada y popular a lo largo de los años **no es ninguna solución milagrosa.**

Se que este enfoque está respaldado por estándares de la industria y organizaciones como SafeCode , OWASP SAMM y BSIMM . En consecuencia, no sorprende que una multitud de organizaciones hayan iniciado sus propios programas Security Champion.

Pero veo demasiadas empresas que **no cuentan con un gestor para liderar y facilitar** esta comunidad interna.

Vemos una y otra vez, que no hay personas que están **genuinamente interesadas en prácticas de seguridad, código limpio, etc..**

Lo que si vemos son los Security al Champiñón (si hongos) y se los comen crudos..



Imágenes creadas por dooder

Security Champion o al Champiñón (Culture, Sharing)

Durante mucho tiempo defendí este concepto como algo que toda organización debería implementar. Pero luego de implementar muchos programas de Security Champion y ver cómo se desarrollaron en otros lugares, me di cuenta de que este rol **no se adapta a todas las organizaciones de la misma manera.**

La idea de tener un Security Champion parece bastante lógica ¿no?, entonces, ¿por qué las iniciativas de Security Champion no resultan como uno planifico?

Si pensamos en los mismos números que de una encuesta de BSIMM que revelo que por cada **135 DEVs hay 1 SEC**. Dado que la base de BSIMM suelen ser empresas maduras esperaría que esta proporción sea incluso más baja.

A título personal y sin ofender a nadie, hoy en día creo que el programa de Security Champion es un parche necesario (Security al Champiñón) pero no resuelve el problema raíz de AppSec.



Imágenes creadas por dooder

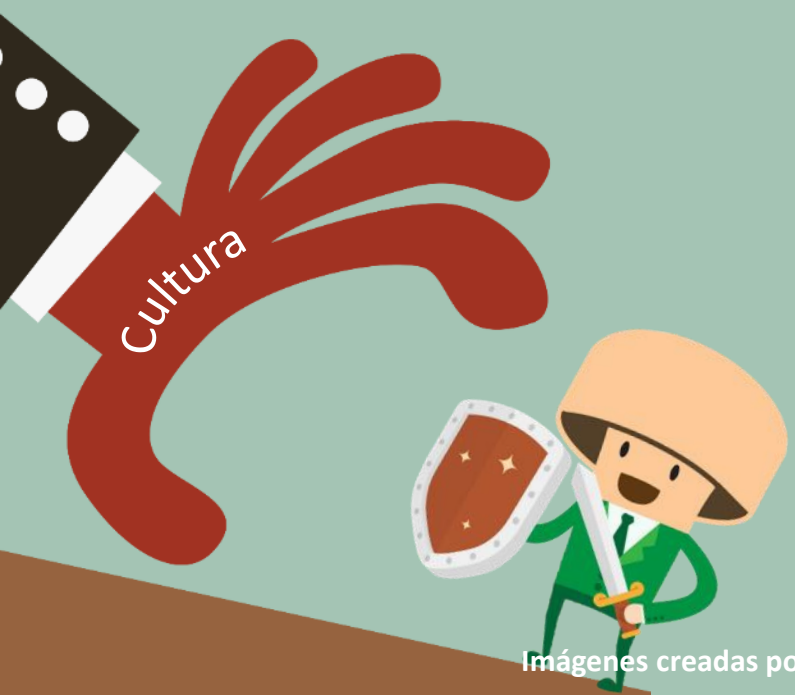
Security Champion o al Champiñón (Culture, Sharing)

Análisis de causas y consecuencias:

El desafío radica en el hecho de que **los profesionales de seguridad capaces de apoyar a los equipos de desarrollo**, normalmente parte de equipos de AppSec o ProdSec (Seguridad de Producto), **suelen ser escasos.**

La idea principal es preparar a uno o más colaboradores de cada célula, a través de capacitación especializada, talleres y participación regular en reuniones de la comunidad de practica de seguridad. **(Pero el Time to market es el amo y sr)**

También es un hecho que muchos de los proyectos en los que **participe involucraron principalmente a desarrolladores externos** y a menudo, eran estos desarrolladores externos los designados como campeones de seguridad.



Imágenes creadas por dooder

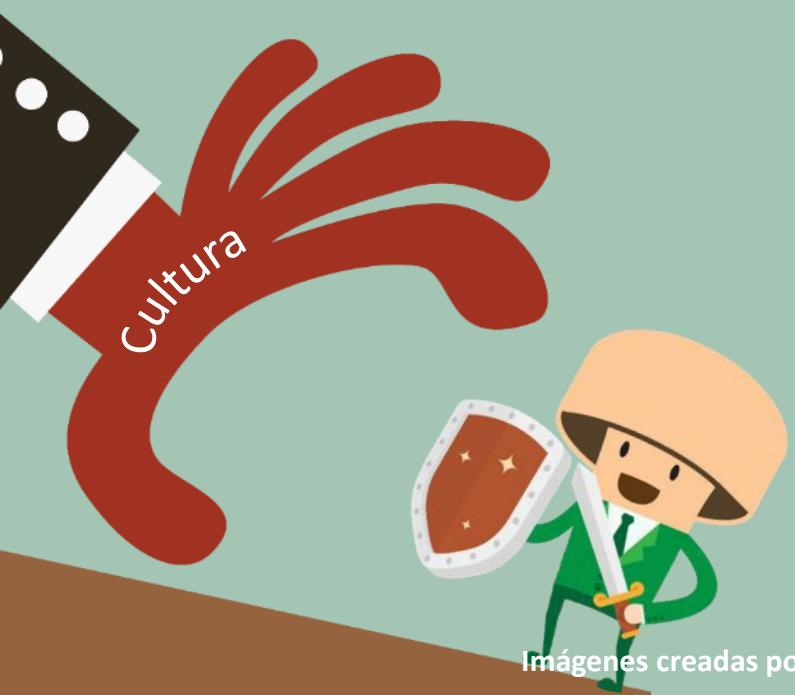
Security Champion o al Champiñón (Culture, Sharing)

Análisis de causas y consecuencias:

No todas las empresas son iguales y la eficacia depende de requisitos previos específicos que muchos parecen pasar por alto con tal de seguir la moda **(definen un Security Champion cuando ni siquiera tienen una política o criterio SecDev)**, haciendo que 'los voluntarios' para el rol con frecuencia se sientan **frustrados o desconectados.**

He observado que los gerentes de seguridad a menudo **malinterpretan el término " Desplazar a la izquierda " como "Desplazar hacia afuera"**, lo que implica que AppSec es responsabilidad única o principal de los equipos de desarrollo.

La falta de Patrocinio del máximo liderazgo: Si va a ejecutar un programa de "campeones de seguridad", comprometa recursos para su éxito.



Imágenes creadas por dooder

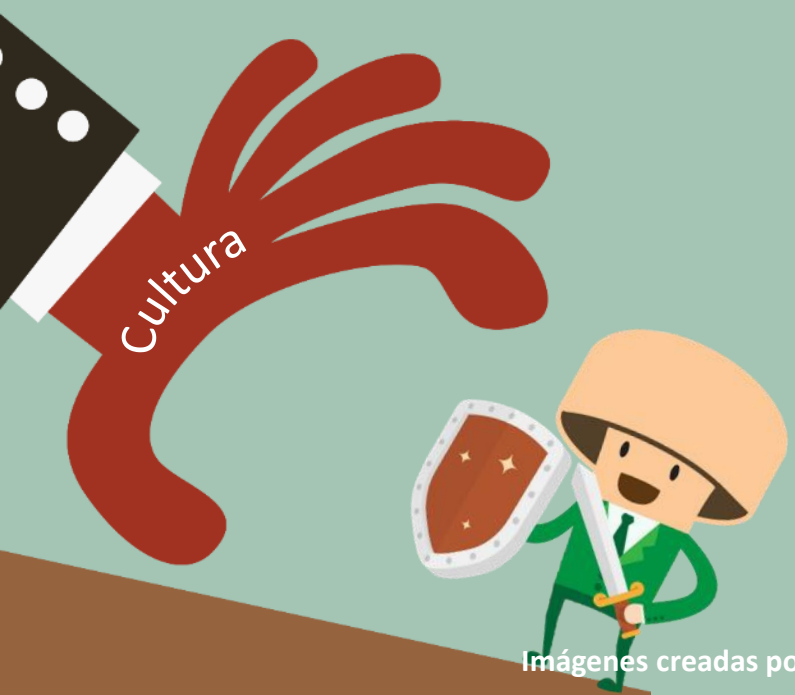
Security Champion o al Champiñón (Culture, Sharing)

Análisis de causas y consecuencias:

Sin él patrocinio, se puede sufrir un **conflicto de prioridades** con aquellos que quieran participar en el programa y sus responsabilidades diarias. Es necesario que se les permita asumir esta responsabilidad adicional con confianza.

La falta de cultura de los mandos medios:
A menudo resulta en el hecho de que la Seguridad no se considera una prioridad empresarial. En consecuencia, los campeones de Seguridad, terminan siendo aquellos con ancho de banda disponible **(generalmente los JR del equipo con influencia limitada dentro de la empresa).**

El último aspecto, aunque quizás menos crítico, pero igualmente digno de mención, es la resistencia que muestran muchas organizaciones cuando se trata de introducir un nuevo rol organizacional



Imágenes creadas por dooder

Security Champion o al Champiñón (Culture, Sharing)

Lecciones aprendidas

Llevo algunos años dedicándome a la Seguridad del Software. La mayoría de las veces estuve en el lado de SEC. Sin embargo, interactué lo suficiente con gente de ambos lados y según mi experiencia personal, **el principal desafío para lograr que los equipos de desarrollo adopten la Seguridad no se debe necesariamente a la ausencia de una función nueva o capacitación específica.**

A menudo, esto gira en torno a prioridades contradictorias y la excesiva complejidad de muchos requisitos de Seguridad no identificados al inicio como una responsabilidad colectiva. Al abordar estos problemas, fui descubriendo que ciertos enfoques son particularmente efectivos, independientemente de si tu organización utiliza Security Champions o no.

Pensemos en que **Seguridad es uno de los atributos de calidad del producto de Software,** según SQUARE o ISO 25000 deberíamos esperar que **los buenos desarrolladores harán lo correcto por sí solos**, que investigarán e implementarán las mejores prácticas para garantizar los atributos de calidad de lo que ellos construyen y como lo hace un arquitecto, deberían ser responsables.



Imágenes creadas por dooder

Security Champion o al Champiñón (Culture, Sharing)

Lecciones aprendidas

Si bien no resuelve la causa raíz **(La falta de calidad en la construcción),** un programa de Security Champion con el apoyo adecuado tienen el potencial de ayudar a los equipos de seguridad e ingeniería a generar un mayor impacto.

Esta comunidad requiere liderazgo y curación: un facilitador o administrador que cree contenido atractivo y una estructura que fomente una verdadera asociación, aprovechando datos objetivos para escalar el programa.
En definitiva, al tratarse de un programa de desarrollo profesional y cambio cultural, los retos serán grandes y el camino muy largo.

No es un programa con un retorno inmediato, pero si es lo suficientemente consistente y recurrente como para que el tema de la seguridad sea cada vez más habitual en los equipos de ingeniería, cosecharemos buenos frutos y tendremos cada vez más profesionales de distintas áreas creando productos más fiables.



Imágenes creadas por dooder

CASO #11

Mas tuberías que Super Mário World





Imágenes creadas por dooc

Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

Pipelines homologados (Dorados) versus Pipelines secundarios (Dark)

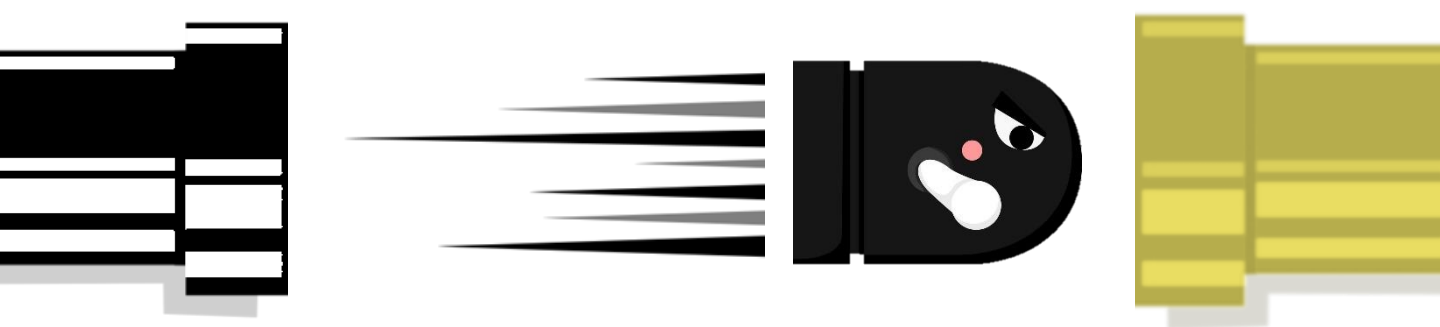
Si bien este caso lo vimos en una empresa de la industria del agro, el mismo es más común de los que nos gustaría y lo vemos de manera recurrente en empresas de tamaños e industrias diversas.

En muchos entornos DevOps, existe un proceso aprobado diseñado con medidas de seguridad integradas **(El famoso camino dorado hacia producción).**

Sin embargo, los desarrolladores a menudo recurren a procesos secundarios o flujos de trabajo alternativos para optimizar sus procesos.

Si estamos hablando sobre Dark Deploys.

A menudo, vemos que esta práctica de **pipelines secundarios introduce riesgos** de Seguridad ya que estos pipelines laterales (Dark) pueden ser débiles y no tener el mismo nivel de chequeo de Seguridad o puntos de control de Seguridad integrados que los homologados.



Imágenes creadas por dooder

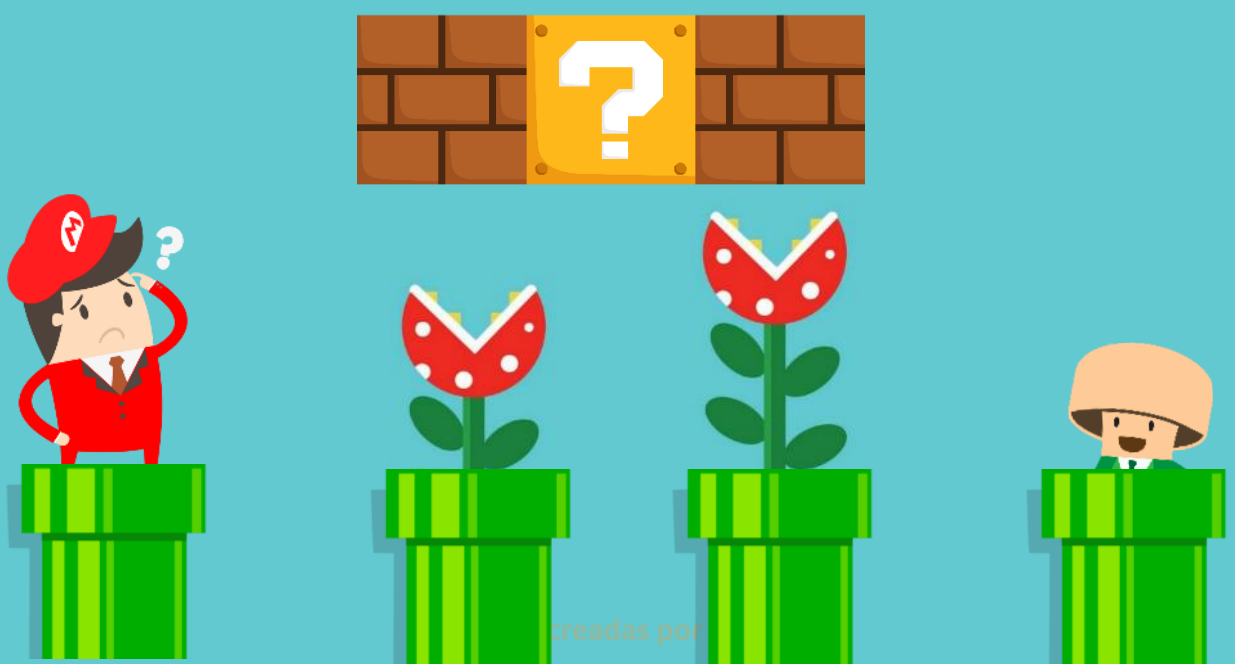
Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

Como pudimos observar en este caso y en varios típicamente la cadena de suministro de Software también **suelen incluir muchos activos de terceros que ingresan al pipeline** de una organización a través de repositorios de código fuente.

Esto también puede limitar la postura de riesgo de Seguridad de una organización a la madurez de Seguridad de otro proveedor.

La gestión de riesgos de Seguridad de la cadena de suministro de Software es una parte integral de este desafío.

Garantizar que las herramientas de Seguridad puedan integrarse sin problemas con diversas tecnologías y pipelines es crucial para brindar una cobertura adecuada independientemente del paso de un activo por una cadena de suministro aprobada o de terceros.



Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

Análisis de causas y consecuencias:

Uso no regulado de servicios de terceros

Si bien esto ofrece a los desarrolladores la libertad de elegir los servicios que desean utilizar, puede generar una serie de problemas de Seguridad, ya que los riesgos de Seguridad asociados con el uso de estos servicios son desconocidos o ignorados.

Por ejemplo, un desarrollador puede optar por utilizar un repositorio de código público y un registro de imágenes de contenedores que no sean propiedad de su empresa ni estén controlados por ella.

Es posible que este repositorio de código no tenga los mismos controles de Seguridad implementados que los repositorios de código internos de la empresa.

Como resultado, el código que se almacena en este repositorio público puede ser menos seguro y más vulnerable a ataques.



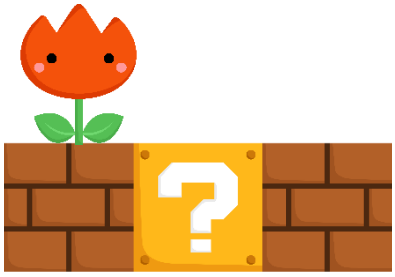
Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

Análisis de causas y consecuencias:

Diferentes equipos, diferentes herramientas
Se ve mucho esta problemática la cual es otro obstáculo en la integración, los distintos equipos y unidades de negocio utilizan herramientas distintas sujetas a distintos requisitos.

El desarrollo y las pruebas descentralizadas, junto con los distintos presupuestos y prioridades de los equipos, dificultan la aplicación de procesos y estándares de Seguridad uniformes.

Los equipos de Seguridad centralizados encargados de proteger y analizar el trabajo de los equipos distribuidos se **enfrentan a desafíos organizativos para lograr una cobertura de Seguridad uniforme.**





Imágenes creadas por dooder

Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

Análisis de causas y consecuencias:

Ejecución por pipeline envenenado (PPE)

PPE es un tipo de vulnerabilidad de Software que los atacantes pueden aprovechar para **inyectar código malicioso en un proceso en ejecución.** La misma puede aprovechar para tomar el control del proceso o provocar que el proceso falle. Las PPE se encuentran a menudo en aplicaciones tanto del lado del cliente como del lado del servidor que aceptan entradas que no son de confianza sin un saneamiento adecuado.

Controles de acceso insuficientes basados en el pipeline

Los controles de acceso insuficientes pueden ser perjudiciales para la Seguridad, ya que **pueden provocar una escalada de privilegios y una fuga de datos.** Un atacante con pocos privilegios puede obtener acceso con muchos privilegios o modificar código al que no se supone que se pueda acceder. Además, si no hay controles de acceso implementados, un atacante puede ver o manipular datos que no están destinados a él.



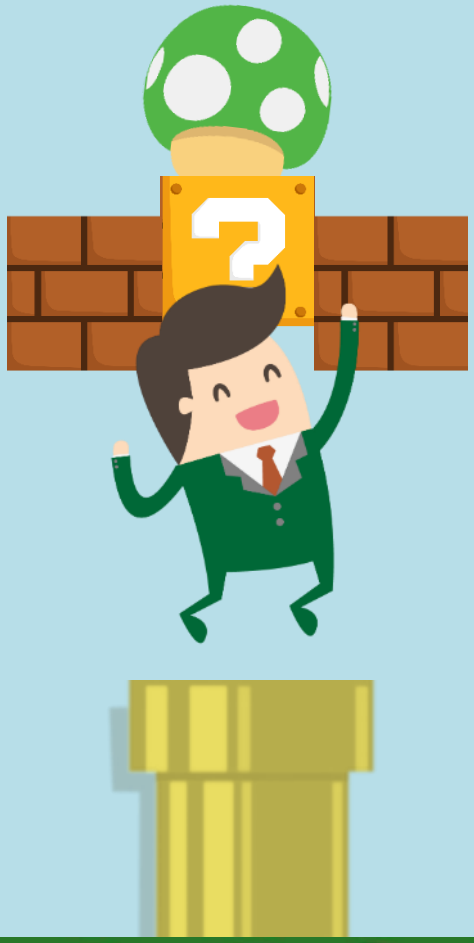
Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

Lecciones aprendidas

Para superar estos obstáculos, es fundamental implementar una variedad de metodologías de prueba que puedan escalarse a medida que el negocio crece y adaptarse a medida que evolucionan las tecnologías. Esto significa aprovechar las pruebas con herramientas ASTs (SAST, DAST, SCA, IAST, RASP) con mecanismos integrados para realizar pruebas y generar informes de riesgos a fin de maximizar el control.

Uso no regulado de servicios de terceros

Para ayudar a mitigar estos riesgos, las empresas deben **adoptar políticas y procedimientos que regulen el uso de servicios de terceros**. La aplicación de dichas políticas debe garantizar que solo se utilicen y actualicen periódicamente los servicios aprobados que satisfagan los controles de seguridad de la organización para hacer frente al cambiante panorama de amenazas.



Imágenes creadas por dooder

Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

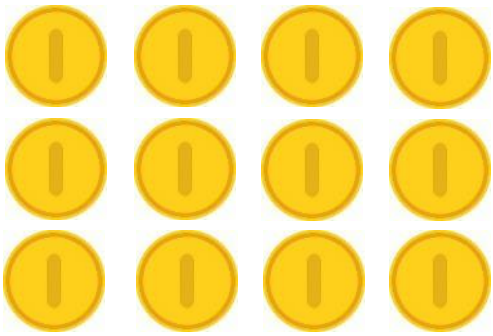
Lecciones aprendidas

Ejecución por pipeline envenenado (PPE)
Un enfoque eficaz para prevenir las PPE es **evitar el uso de entradas que no sean de confianza en aplicaciones del lado del Servidor y evitar el uso de salidas que no sean de confianza (generadas desde el servidor) en aplicaciones del lado del cliente.** Como práctica recomendada, todas las entradas que no sean de confianza deben validarse cuidadosamente.

En cuanto a las salidas que no son de confianza, los datos deben mostrarse como no confiables y usarse con precaución.

Aunque las vulnerabilidades de ejecución de tuberías envenenadas **pueden ser difíciles de explotar, plantean graves consecuencias si se explotan con éxito**.







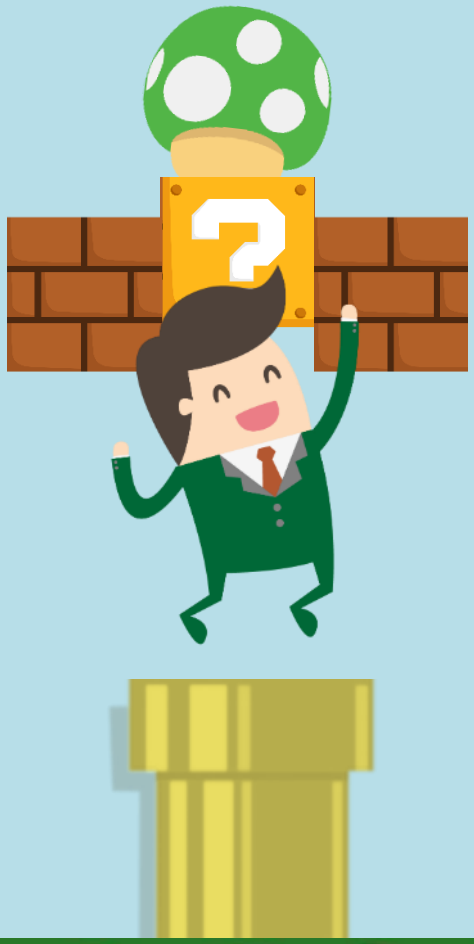
Imágenes creadas por dooder

Mas tuberias que Super Mário World (Culture, Automation, LeanIT, Measuring)

Lecciones aprendidas

Diferentes equipos, diferentes herramientas
Para hacer frente a estos desafíos, a menudo es necesario adoptar un enfoque de seguridad basado en plataformas que ofrezca **una visión unificada del riesgo en todos los proyectos**, equipos y regiones. Esto debe permitir a las organizaciones gestionar múltiples metodologías de prueba de manera eficiente.

Controles de acceso insuficientes basados en el pipeline
Para evitar esto, se recomienda **restringir las políticas de Seguridad y el acceso a los recursos a grupos limitados** para mejorar la Seguridad y la confianza.



Imágenes creadas por dooder

Mas tuberías que Super Mário World (Culture, Automation, LeanIT, Measuring)

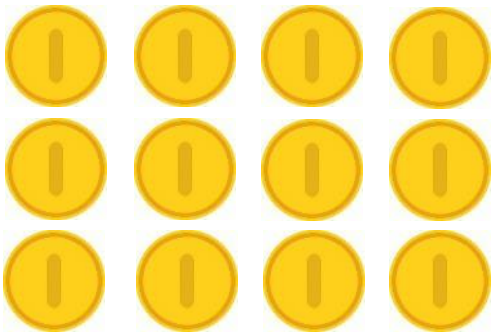
Lecciones aprendidas

2 R's Minimizadas (Riesgo y tiempo de revisión)
Mantener el código en un sistema de control de versiones central abre la puerta a ampliar la calidad del código con pruebas y formateo.

Esto garantiza que en cada solicitud de extracción, el código se valide con pruebas. Las solicitudes de extracción sean revisadas por el administrador y los cambios se fusionen estrictamente según el estado de aprobación. El enfoque también ayuda a identificar errores de código antes de la fusión y brinda una mejor descripción general de qué refactorizar y dónde.

La reducción del riesgo y la revisión hacen de CI/CD un enfoque prometedor para minimizar las fallas de Seguridad y configuración, lo que permite a los desarrolladores lanzar al mercado un producto seguro y confiable.







Imágenes creadas por dooder

CASO #12

DevSecOpoly - no todo se puede comprar



$ 270 $ 240 $ 220 $ 200 $ 180 $ 150 $ 290 $ 120 $ 150 $ 300 $ 320 $ 100 $ 330 $ 100 $ 360 $ 380 $ 270 START

Imágenes creadas por dooder

DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Este caso se remonta a una reunión con el director de una gran empresa de telecomunicaciones en Latinoamérica. Al finalizar nuestra conversación, este director al mejor estilo Mr. Monopoly dice:

“Necesito a alguien que implemente DevSecOps para ayer y la plata no es un problema"

“¿Y qué esperas exactamente que haga esta persona (Héroe o Mago), dado que recién conocimos su infraestructura y sistemas?”.

Si bien este caso no es el más común en todas las empresas, si lo vemos con mucha frecuencia la idea de que



“DevSecOps se puede comprar”
No, en absoluto.
Pero mucha gente estará feliz de vendérselo.

DevSecOps

DevSecOps

Imágenes creadas por dooder

DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Vemos que las empresas a menudo creen que pueden salir de su deuda técnica sin realizar cambios sustanciales. A menudo piensan que pueden comprar DevSecOps en una caja.

Bueno, si bien estamos en la época self-service (o pago por uso), desafortunadamente **la cultura no se puede comprar con tarjeta de crédito.**

Como vimos anteriormente en este E-book para que una organización adopte DevSecOps necesitará adoptar un cambio cultural, adoptar herramientas que permitan la automatización y cambiar los procesos para incluir tanto desarrollo como operaciones **(Si operaciones también incluye seguridad).**

Me gusta pensar que DevSecOps se puede explicar simplemente como operaciones trabajando junto con ingenieros de desarrollo para hacer las cosas más rápido (automatizada o no) y repetible sumando calidad **(Si, Seguridad es un atributo de calidad del producto de Software).**



Imágenes creadas por dooder

DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Dar el paso a DevSecOps puede ser una tarea desalentadora, ya que muchas organizaciones no saben por dónde empezar.

Recientemente nos divertimos un poco realizando algunas **"evaluaciones de DevSecOps" del mercado para ver qué soluciones ofrecían. (como se dice, fuimos de pesca).**

Dimos diferentes respuestas: desde una organización que adopta plenamente DevSecOps hasta una que se encuentra al comienzo del viaje y con diferentes presupuestos.

Algunas de las evaluaciones proporcionaron un **valor real, al focalizarse sobre cultura y metodologías, mientras que otras simplemente nos ofrecieron una herramienta que prometía hacer realidad todos mis sueños** de DevSecOps (**El traje nuevo del rey**) como lo vimos en un caso anterior.



Imágenes creadas por dooder

DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Análisis de causas y consecuencias:

Organizaciones grandes y pequeñas han experimentado estos cambios, los desafíos encontrados en el camino y la luz al final del túnel. Cada organización requiere su propio conjunto de herramientas y procesos para adoptar DevSecOps. **El cambio cultural depende en gran medida de la dinámica existente de una organización.**

Comprar rojo y naranja como en el monopoly: En las pequeñas empresas, hemos descubierto que es necesario comenzar desde cero y por lo general, **comienza agregando las herramientas DevSecOps luego de una investigación de estas**. (estadísticamente las propiedades rojas y naranjas son las que más aterrizan jugadores)

En organizaciones grandes, el valor de **DevSecOps debe venderse al equipo ejecutivo y luego aplicarse de arriba hacia abajo**. Este es un desafío enorme para la mayoría de las grandes empresas porque es difícil cuantificar el valor de DevSecOps. **Es como comprar ferrocarriles solo quien tiene los 4 entiende su ROI (Retornos de inversión).**



Imágenes creadas por dooder

DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Análisis de causas y consecuencias:

Las herramientas son absolutamente esenciales para el viaje de DevSecOps. Por ejemplo, las herramientas pueden entregar, automatizar o monitorear tu entorno. Sin embargo, DevSecOps no es un producto y las herramientas por sí solas no permitirán los procesos necesarios para aprovechar todo el valor de DevSecOps. Las personas son lo que más importa y no se puede hacer DevSecOps sin desarrollar primero las personas, la mentalidad y la cultura.



DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Análisis de causas y consecuencias:

Normalmente este es un proceso arduo que **requiere mucho tiempo, gente y coordinación.** Pero con la emoción de pasar a DevSecOps, olvidamos que lo que más importa en DevOps es la comunicación, no la automatización.

La mayoría de las organizaciones, especialmente las de tipo globa **no están preparadas para comunicarse bien**. Están aislados por grupo, geografía y función. Eso crea puntos de fricción que ralentizan la comunicación y las herramientas no ayudarán en este caso. Sólo un cambio fundamental en la cultura hacia la apertura y la propiedad del código tendrá un impacto duradero y notable.

Debemos considerar todos los factores que hacen que un equipo de DevSecOps funcione bien: **colaboración, eliminación de silos, procesos definidos, propiedad y automatización. Todo esto, junto con la mejora continua/entrega continua.**



Imágenes creadas por dooder

DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Lecciones aprendidas

Decidir comprar herramientas es un gran paso en la dirección correct, aunque lo que es **más importante es definir primero el "por qué"** o el objetivo final detrás de las decisiones.

Las herramientas pueden permitir a una organización colaborar mejor, automatizar o realizar entregas continuas. Sin embargo, sin la mentalidad y la adopción adecuadas es posible que no se pueda lograr la plena capacidad de una herramienta.

Comprar una herramienta definitivamente puede ser una ganancia para un equipo, **pero comprar una herramienta no es comprar DevSecOps.**

Hacer que las herramientas y las mejores prácticas funcionen para el equipo y lograr objetivos a corto y largo plazo es donde surge el real valor.



Imágenes creadas por dooder

DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Lecciones aprendidas

El cambio es difícil para muchas organizaciones e individuos, además, no se produce de la noche a la mañana. Es importante comprender cómo cada uno procesa el cambio.

En el Proceso de 8 pasos de Kotter se trata de articular la necesidad de un cambio, **crear urgencia en torno al por qué** y luego comenzar poco a poco, encontrar / desarrollar defensores internos. Todo esto, antes de intentar demostrar victorias (en este caso), comprar una herramienta.

Si las personas de una organización no son conscientes de un problema o de que hay una forma mejor de trabajar, será difícil conseguir la aceptación necesaria y motivar a los miembros del equipo para que adopten nuevas ideas y pasen a la acción.



DevSecOpoly - no todo se puede comprar (Culture, Automation, Sharing)

Lecciones aprendidas

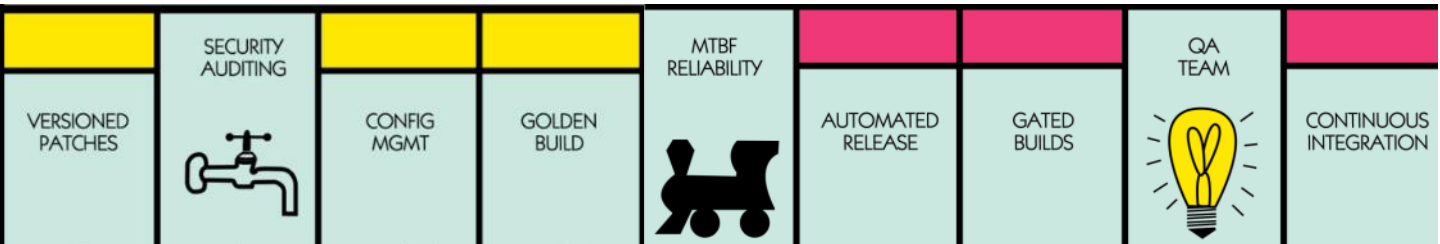
No es una cuestión de personas ni de recursos, se puede utilizar reclutadores y bolsas de trabajo para encontrar personas de DevSecOps. **Pero si no hay voluntad de cambiar la cultura, esas personas estarán destinadas al fracaso.**

Entonces, **¿podes comprar DevSecOps?**
Sí, hasta cierto punto podes...

Pero no son las herramientas en las que deberían centrarse tus compras, **sino en los pioneros y campeones que te alentarán a encontrar formas de trabajar nuevas y mejoradas.**

Creemos que el enfoque más inteligente para DevSecOps es adoptar un enfoque iterativo y de lotes pequeños.

Es mejor ver qué procesos y herramientas funcionan, observar la cultura en la que interactúan las personas y tomar nota de las cosas que pueden acelerar la transferencia de conocimientos y fomentar la colaboración. Con el tiempo, se puede ampliar esto a toda la organización.



CASO #13

The dark side of the pipeline



The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Este caso se basa en un relato que nos compartieron luego de nuestra última charla de la Ekoparty 2023, pero en el pasaje de Chris en algunas consultoras/software factory podemos validar su existencia.

Ojo que esto aplica para las empresas con Factory interna también…

Para las organizaciones de Software actuales, la Seguridad nunca ha sido una prioridad tan importante.

Por un lado, está la amenaza actual y creciente de ser pirateados por actores maliciosos como lo demuestra varios informes y estudios de tendencias.

Por el otro, hay **una ola de regulaciones de ciberseguridad por parte del gobierno,** para mitigar dichas vulnerabilidades de ciberseguridad.



Imágenes creadas por dooder

The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Las organizaciones de Software sienten la presión de ambos lados mientras trabajan para mejorar su postura de Seguridad de manera que también logren la auditoría y el cumplimiento de nuevas reglas.

Para afrontar estos desafíos, escuchamos mucho sobre la cadena de suministro de Software y la importancia de protegerla con enfoques DevSecOps y pipelines dorados hacia la producción

Como vimos en el caso “Mas tuberías que Super Mário World”, las organizaciones responsables ya dedican mucho tiempo y recursos a crear los pipelines dorados. Pero a pesar de todo ese esfuerzo, vemos que hay un riesgo oculto que a menudo se pasa por alto en la cadena de suministro:

Los cambios por fuera de los pipelines autorizados.



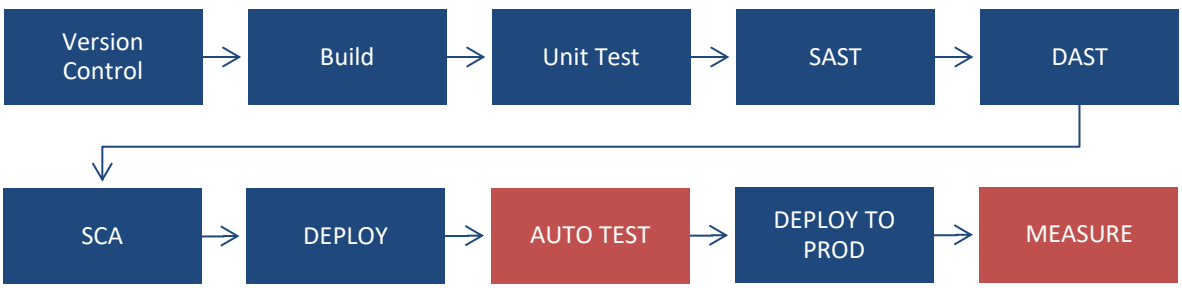
Imágenes creadas por dooder

The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Análisis de causas y consecuencias:

En las industrias reguladas, la gobernanza en desarrollo es un enfoque clave cuando se trata de asegurar el proceso de entrega de Software. A menudo se requiere que los equipos sigan pautas estrictas para garantizar la Seguridad y la integridad de sus sistemas.

Muchas organizaciones hacen esto con lo que a menudo se conoce como un "pipeline dorado", que **incorpora todos los controles de Seguridad necesarios, como controles de solicitudes de extracción, SAST, DAST, análisis de composición de software, etc**.



La mayoría de los equipos siguen estos canales dorados hasta producción para garantizar que el proceso de entrega de Software sea seguro y cumpla con las normas. Hay muchos beneficios al seguir este enfoque:

- A cada equipo le quita la carga de descubrir cómo cumplir.
- Permite que todas las partes interesadas colaboren y lleguen a un consenso de una vez por todas.
- Facilita la auditoría y la gestión de cambios.



Imágenes creadas por dooder

The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Análisis de causas y consecuencias:

Sin embargo, si bien los procesos DevSecOps como estos han sido eficaces para controlar y monitorear el proceso de entrega, no aseguran todo el ciclo de vida del desarrollo de Software. **Los que pasan al DARK SIDE,** pueden eludir estos controles e introducir cambios no aprobados directamente en producción, poniendo en riesgo a toda la organización.

Vemos que se implementan Pipelines para garantizar que los cambios realizados en un sistema sigan un proceso definido y se autoricen antes de la implementación.

Pero los controles de los pipelines tienen una limitación obvia cuando se trata de garantizar la seguridad y el cumplimiento de todo un sistema de Software. Solo pueden garantizar la seguridad y el cumplimiento de los cambios que ya se han realizado. No tienen en cuenta a los malos que acceden a producción sin seguir el Camino dorado.



Imágenes creadas por dooder

The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Análisis de causas y consecuencias:

Hay varias preguntas clave de Seguridad que no se pueden responder en el proceso:

¿Cómo descubrimos cargas de trabajo que no han pasado por nuestro proceso?
¿Qué pasa si un desarrollador interno tiene las claves de producción?
¿Qué sucede si nuestro proceso de implementación tiene fallas silenciosas?

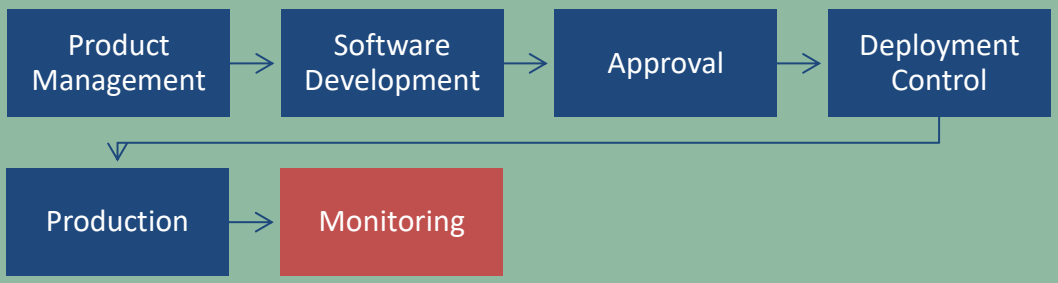


The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Lecciones aprendidas

Estas preguntas de Seguridad que no se pueden responder en el proceso se pueden responder en tiempo de ejecución. **Monitorear producción es la única manera** de estar seguro de lo que realmente se está ejecutando en los entornos y agrega una capa adicional de Seguridad a los pipelines dorados. También es bastante sencillo:

- Registrar cada implementación aprobada en proceso.
- Monitorear (y registrar) lo que se está ejecutando en producción.



Al monitorear y registrar el estado del entorno de producción, las organizaciones pueden detectar cualquier cambio no autorizado que haya pasado por alto el proceso de gestión de cambios.

Esto cierra el círculo de la Seguridad , garantiza el cumplimiento continuo en los entornos y mitiga en gran medida las amenazas y los riesgos.



Imágenes creadas por dooder

The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Lecciones aprendidas

Para muchos equipos, particularmente en industrias reguladas como las finanzas y la atención médica, la entrega de Software debe gestionarse de acuerdo con un proceso específico para gestionar los riesgos. Para garantizar que se siga el proceso para cada cambio, se agrega un proceso de gestión de cambios para la verificación antes de la implementación.

La gestión de cambios implementada como una puerta tiene muchos aspectos negativos:

- Aumenta el tiempo de espera para los cambios.
- Conduce a liberaciones más grandes y más riesgosas
- Requiere auditorías manuales para garantizar el cumplimiento



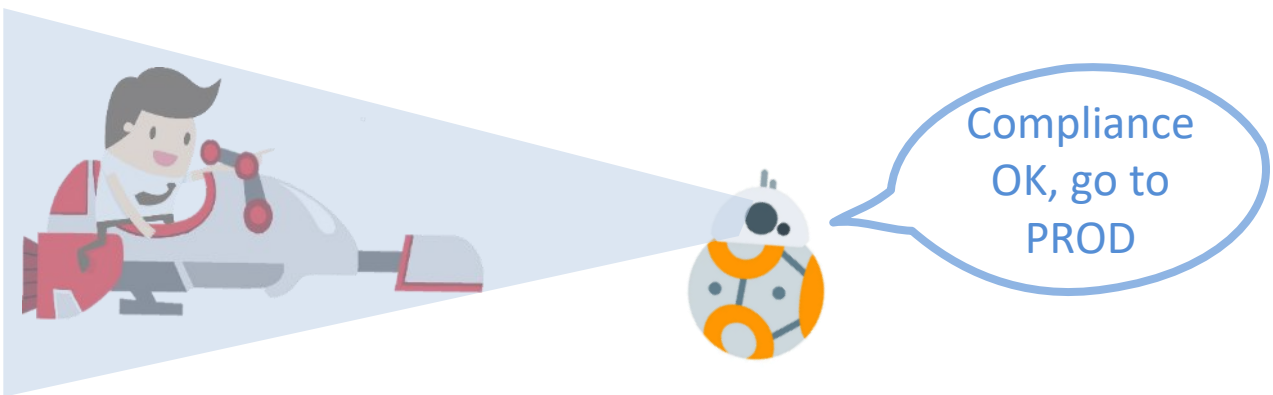
Imágenes creadas por dooder

The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Lecciones aprendidas

Pero adoptar un enfoque de pipeline dorado para el proceso de gestión de cambios se debe generar cumplimiento en cada cambio. Al registrar la evidencia de cada prueba, escaneo de seguridad, solicitud de extracción, etc. a medida que ocurren en el proceso, se pueden recopilar automáticamente pruebas de que se está siguiendo el proceso de entrega y se están cumpliendo los controles de riesgo.

Es más confiable, más rápido y elimina la necesidad de una puerta manual.



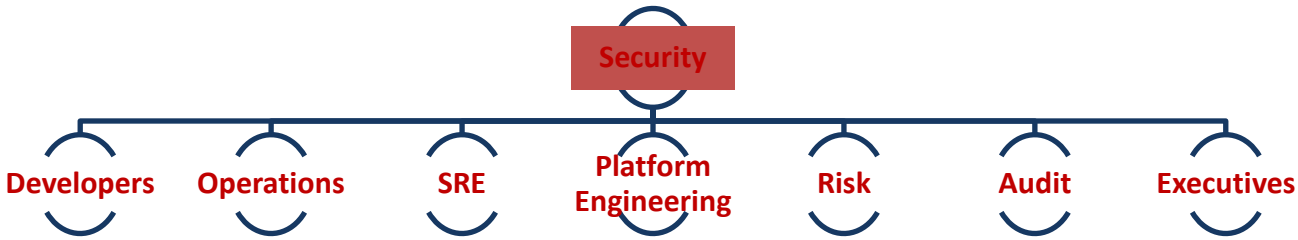
Este enfoque convierte los procesos subjetivos en una gestión de riesgos objetiva y de alta eficacia. También tiene el beneficio adicional de permitir **una auditoría automatizada y continua**. No es necesario dedicar tiempo a reunir la documentación para un registro de auditoría cuando ya se han capturado todas las pruebas en los procesos de producción y luego se han almacenado de forma segura.

Imágenes creadas por dooder

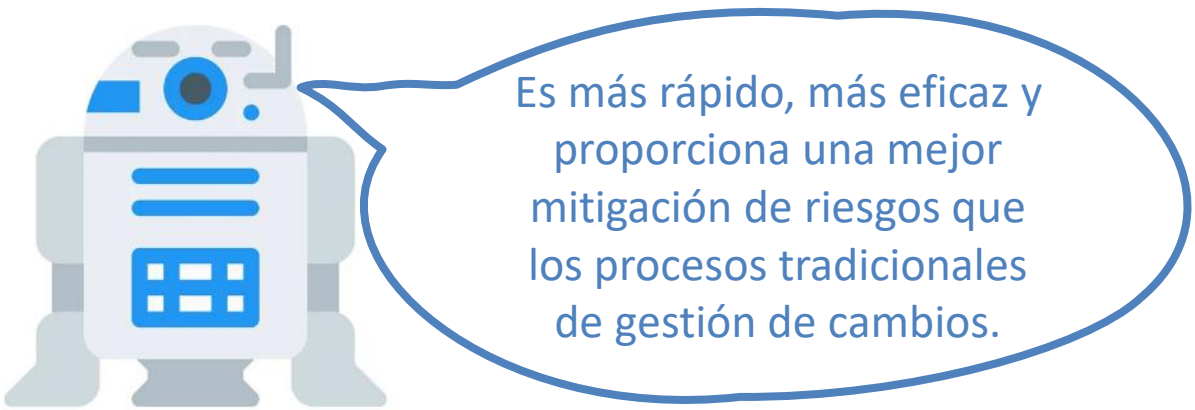
The dark side of the pipeline (Culture, Automation, LeanIT, Measuring)

Lecciones aprendidas

La Ingeniería de Gobernanza es un enfoque alternativo a los procesos tradicionales de gestión del cambio. **Reúne a los equipos de cumplimiento, riesgo, Seguridad y desarrollo para definir estándares y automatizar controles**, recopilación de evidencia y trabajo de aprobación. Deja a los humanos informados sobre la gobernanza y al mismo tiempo, proporciona un cumplimiento continuo y listo para auditorías para las entregas.



Con Governance Engineering, las organizaciones pueden identificar y abordar los cambios que no cumplen con las normas de inmediato en lugar de esperar el momento de la auditoría.



Imágenes creadas por dooder

CASO #14

Yo soy Groot...digo Pinocho.

¿De qué madera estas hecho?



Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Este caso lo sufren todas las industrias, pero sobre todo los reclutadores, gerentes de TI, entre otros y lo podemos denominar como "Fake Resume": candidatos o profesionales en puestos de DevSecOps que pueden ser más publicidad que sustancia.

El mercado de TI es muy difícil para los empleadores. Es competitivo, los costos aumentan, surgen nuevas demandas y tecnologías constantemente como DevSecOps. La institución STEM (Science, Technology, Engineering and Mathematics) no sigue el ritmo, todos estos aspectos crean un entorno similar al que vimos a fines de los años 90 y principios de los 2000, antes de que estallara la burbuja puntocom.

Todo lo anterior, excepto por una diferencia importante que estamos viendo ahora y esa diferencia no es buena:

La continua falsificación de credenciales presentadas por los candidatos en el mercado.

Name: Groot
Address: I am groot
Email: i am groot
Phone: i am groot



Objective: I AM GROOT
Skills: I AM GROOT. I AM GROOT.
Activities: I AM GROOT. I AM GROOT. I AM GROOT
Education: I AM GROOT. I AM GROOT
Awards: I AM GROOT. I AM GROOT
Personal Interests: I AM GROOT

Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

La popularidad de DevSecOps hace que algunos candidatos estén siendo un poco “digamos”, **creativos con sus currículums y perfiles de LinkedIn** con la esperanza de conseguir un puesto de trabajo.

Para mí y para mi equipo, esto se conoce como **la estafa del “Fake Resume”.**

Supongo que algunos de los que están leyendo este caso están sorprendidos. ¿se preguntarán, alguien podría fingir su experiencia y pasar una entrevista, siendo en realidad, ser contratado para un puesto para el que no estaba calificado?

En la industria de TI, una entrevista típica implica preguntas técnicas o ejemplos y hasta respuestas detalladas a problemas teóricos presentados... no es como si estuvieras contratando a un jefe administrativo que dice que puede hacer malabarismos con múltiples prioridades, pero en realidad no puede (sin ofender a los jefes administrativos desde ya).



Imágenes creadas por Linda Bouderbala

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

La primera vez que nos pasó fue hace unos 5 años con un contratista que había pasado la entrevista (culpa a medias, dado que estábamos urgidos para integrar a alguien al proyecto y encima, súmale el contexto Argentina…)

Cuando me enteré de que había fingido todo, me sorprendí y pensé... ¿en serio?. En ese momento, estos puestos difíciles de cubrir se basaban en alguna tecnología específica que se necesitaba un determinado proyecto y estas habilidades no se encontraban fácilmente.

Las tarifas eran altas, el grupo de candidatos pequeño y la demanda inmediata.

El contratista pasó la entrevista, se incorporó y se presentó en nuestro cliente para comenzar a trabajar. Dos o tres días después, el cliente llamó y dijo que este tipo no sabe lo que está haciendo y que lo confrontó y el contratista admitió que había embellecido su currículum y su experiencia: **Yo soy Groot...digo Pinocho.**



Imágenes creadas por Linda Bouderbala

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Análisis de causas y consecuencias:

En ese momento lo atribuí a un individuo deshonesto que intentaba maximizar su poder adquisitivo cuando no tenía las habilidades necesarias, pero luego comenzamos a ver qué ocurría cada vez más. El modus operandi es siempre el mismo:

- **Una tecnología de vanguardia.**
- **Habilidades de alta demanda.**
- **Gerentes o empleadores desesperados.**
- **Tendencia a reducir los estándares de las entrevistas (desesperación por cubrir puestos).**
- **El candidato potencial está por debajo del precio del mercado.**
- **Contratista, no candidato a empleo de tiempo completo.**

Desde que ocurrió el primer caso, hemos visto que esto sucede cada vez más. No pasa una semana sin que se presente un caso y también vemos que ocurren más casos de este tipo en la segunda mitad del año.



Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Análisis de causas y consecuencias:

Pero si lo analizamos en profundidad, podemos también identificar que **existen empresas que sacan provecho de esta demanda.** El perfil de estas empresas y cómo operan se está volviendo más conocido.

Ofrecen a los candidatos un **bootcamp** donde reciben una capacitación básica en la tecnología, lo suficiente como para estar algo familiarizados con ellas y **luego estas empresas comienzan a comercializarlos como candidatos experimentados** a un precio cercano al del mercado a cambio de una comisión de su salario durante un año.

Por lo general, se comercializarán con otras empresas de personal en lugar de clientes directos (eliminando una capa de responsabilidad y responsabilidad directa).

Afortunadamente la misma industria se está dando cuenta que el modelo no funciona.

Lo único que me da pena es que muy buenos perfiles estan quedando en el camino frustrados con las falsas promesas de estos modelos.



Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Análisis de causas y consecuencias:

Sin duda, **existen estrategias legítimas para hacer la transición a una carrera en DevSecOps desde distintos ámbitos de TI**, pero exagerar o inventar directamente habilidades y experiencia relevantes en DevSecOps no es una de ellas.

Aun así, **puede resultar tentador para algunas personas debido a los altos salarios que se cobran** por puestos como ingeniero de DevOps, DevSecOps, SREs y otros puestos relacionados con DevOps aumentan aún más el atractivo de estos puestos. Siendo realistas si simplemente **agregamos la palabra "DevSecOps" al perfil de LinkedIn** o a otras plataformas probablemente te hará aparecer en los resultados de búsqueda de los reclutadores.



Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Análisis de causas y consecuencias:

Esto presenta un desafío potencial para los reclutadores y gerentes de contratación que buscan cubrir puestos vacantes. ¿Cómo se puede detectar a las personas que están fingiendo durante el proceso de selección y contratación?, o ¿cómo se puede saber si alguien no está necesariamente fingiendo, sino exagerando, o simplemente tiene una percepción distorsionada de cómo sus antecedentes coinciden con el puesto vacante de DevSecOps?

Planteamos estas preguntas a varios líderes de TI que han estado en el negocio varias veces. Estos son sus consejos sobre cómo identificar a los candidatos que pueden parecer y sonar bien, pero que pueden ser más publicidad que sustancia.



Imágenes creadas por dooder

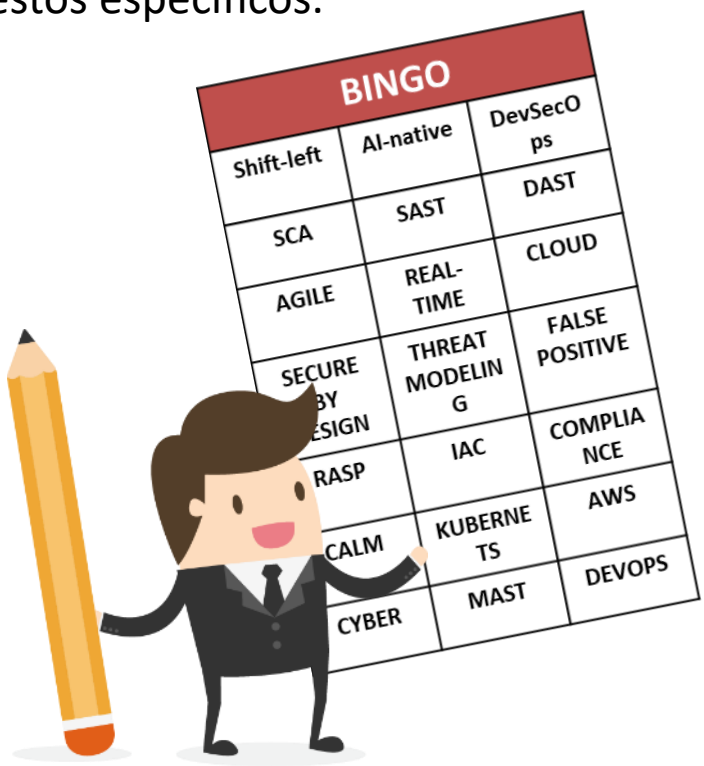
Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Lecciones aprendidas

Hay que tener cuidado con los candidatos que juegan al bingo de palabras claves.

Es una práctica inteligente a la hora de buscar trabajo adaptar los curriculum a puestos específicos.

Pero existe una diferencia entre destacar habilidades y experiencia legítimas relevantes para un puesto de trabajo y simplemente llenar un currículum con palabras clave como DevOps, automatización, CI/CD, Agile, gestión de configuración, orquestación, etc… con la esperanza de obtener un resultado.



Así que cuando ese currículum perfecto llegue a tu escritorio, está bien e incluso es recomendable, preguntarte si no será demasiado perfecto.

Hay una forma sencilla de ver con qué frecuencia alguien cambia su perfil y para esto, podes poner la URL de su perfil de LinkedIn en la **Wayback Machine de Internet** para ver cómo ha cambiado con el tiempo.

Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Lecciones aprendidas

Plantearse las preguntas, las pruebas y el material de detección adecuados

Nuestros expertos coincidieron en general en que descubrir a un impostor de DevSecOps puede ser similar a descubrir a alguien que está exagerando sus conocimientos de programación o cualquier otra habilidad de TI.

Podes pedirles a los candidatos que realicen desafíos o pruebas relacionados con DevOps y DevSecOps de la misma manera en que probablemente verificarías la fluidez de un desarrollador con un lenguaje o marco en particular.

De manera similar, las buenas preguntas en una entrevista telefónica o en persona deberían ayudar a detectar a los exagerados, incluidos aquellos que simplemente "engordaron" su currículum con palabras clave en lugar de incluir trabajo previo sustancial.



Imágenes creadas por doucet

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Lecciones aprendidas

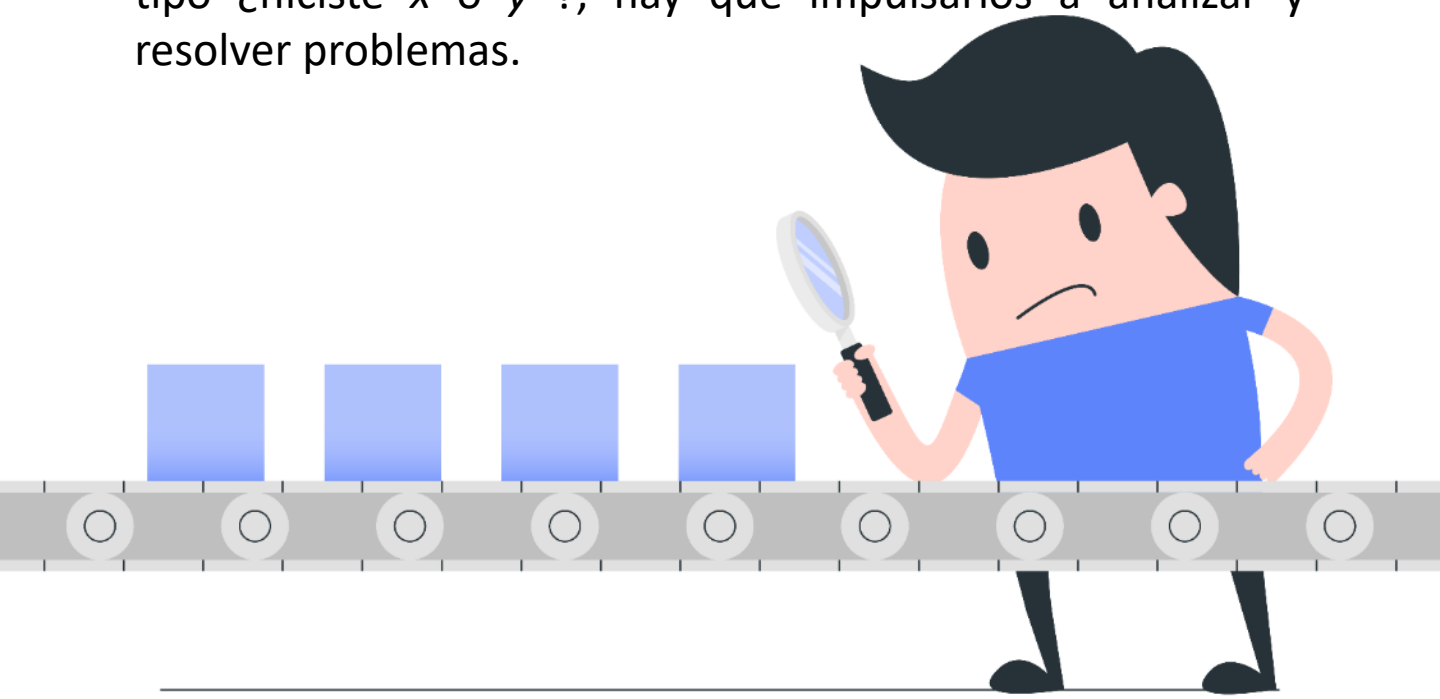
Desconfiar del perfeccionista

Aceptar el fracaso es una parte fundamental de la cultura DevOps. Los ingenieros de DevSecOps "se dan cuenta de que el fracaso es una oportunidad para aprender y mejorar”

Podemos preguntar a los candidatos cómo aplicarían DevSecOps a un sistema o proceso legacy existente.

Aconsejan centrarse en la resolución de problemas como una habilidad fundamental en el proceso de contratación.

También hay que seguir investigando hasta que tropiecen un poco, no se pueden hacer preguntas de alto nivel o del tipo ¿hiciste *x* o *y* ?, hay que impulsarlos a analizar y resolver problemas.



Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho? (culture)

Lecciones aprendidas

Tener cuidado con el enfoque en las herramientas

Quizás una de las mayores señales de alerta de que podrías estar tratando con un impostor de DevSecOps es que pueden hablar largo y tendido, incluso elocuentemente, sobre herramientas y habilidades técnicas, pero se quedan en blanco cuando mencionas cuestiones de cultura, trabajo en equipo y más.

DevSecOps tiene tanto que ver con la cultura, la visibilidad y la transparencia como con las habilidades técnicas, por lo que cualquiera que diga que es un ingeniero de DevOps y no hable de esos aspectos y solo enumere sus calificaciones técnicas tal vez solo esté tratando de pasar los filtros al incluir 'DevOps' en su currículum".

Las herramientas y las habilidades técnicas son sin duda importantes en los equipos de DevOps, pero las habilidades interpersonales son igualmente fundamentales.





Imágenes creadas por dooder

Yo soy Groot...digo Pinocho. ¿De qué madera estas hecho?



Bonus Track

Responsabilidades clave de un ingeniero de DevSecOps

Integración de seguridad: Incorporar prácticas y herramientas de Seguridad en el ciclo de vida del desarrollo de Software para garantizar que la Seguridad sea una prioridad desde el principio y durante todas las etapas del desarrollo.

Automatización de procesos de seguridad: Automatizar las pruebas de Seguridad y los controles de cumplimiento para integrarlos sin problemas en los procesos de integración continua y despliegue continuo (CI/CD), haciendo que los controles de Seguridad sean eficientes y consistentes.

Evaluación y mitigación de riesgos: Identificar, evaluar y abordar de forma proactiva los riesgos de seguridad y las vulnerabilidades dentro del Software y la infraestructura, garantizando que los problemas potenciales se resuelvan antes de que puedan ser explotados.

Colaboración y formación: Trabajar en estrecha colaboración con los equipos de desarrollo y operaciones para fomentar una cultura de Seguridad. Esto suele implicar educar a los miembros del equipo sobre las mejores prácticas de seguridad y garantizar que todos sean conscientes de su función en el mantenimiento de la Seguridad.

Respuesta al incidente: Responder y gestionar incidentes de Seguridad, incluida la implementación de medidas para prevenir que vuelvan a ocurrir. Esto implica tanto una respuesta inmediata a las amenazas como un análisis continuo para mejorar las estrategias de respuesta.

Cumplimiento y gobernanza: Garantizar que todos los procesos de desarrollo e implementación de Software cumplan con las políticas, estándares y regulaciones de seguridad pertinentes, protegiendo así a la organización de problemas legales y regulatorios.

Cobertura de extremo a extremo : Encontrar herramientas de seguridad que cubran toda el stack de desarrollo, incluido entornos multi cloud, contenedores, Kubernetes, etc. Los silos o puntos ciegos generarán fricción y confusión que, a su vez, convertirán a los equipos de seguridad en el chivo expiatorio de cualquier problema que surja.

Imágenes creadas por dooder

Yo soy Groot...digo Pinocho.
¿De qué madera estas hecho?



Bonus Track

Habilidades clave para un ingeniero DevSecOps

Codificación y scripting: Competencia en lenguajes de programación como Python, Java o Ruby, y lenguajes de scripting para tareas de automatización.

Comprensión de las herramientas y prácticas de DevOps: Familiaridad con herramientas CI/CD (como Jenkins, GitLab CI o Travis CI), orquestación de contenedores (como Kubernetes) y herramientas de infraestructura como código (IaC) (como Terraform o Ansible).

Conocimientos de ciberseguridad: Un sólido conocimiento de los principios, prácticas y herramientas de Seguridad. Esto incluye conocimientos sobre modelado de amenazas, gestión de riesgos, protocolos de Seguridad, tecnologías de cifrado y evaluación de vulnerabilidades.

Seguridad en la nube: Habilidades en la protección de entornos de nube, incluida experiencia con proveedores de servicios de nube como AWS, Azure o Google Cloud Platform y comprensión de herramientas y prácticas de Seguridad nativas de la nube.

Administración de redes y sistemas: Comprensión de las arquitecturas de red, los protocolos y el diseño de redes seguras. El conocimiento de la administración de sistemas también es fundamental para proteger la infraestructura subyacente.

Conocimientos regulatorios y de cumplimiento: Conocimiento de los requisitos legales y reglamentarios relacionados con la Seguridad de la información, como GDPR, HIPAA y SOC 2.

Capacidad de análisis y resolución de problemas: Capacidad para identificar problemas de Seguridad y vulnerabilidades y desarrollar soluciones o mitigaciones efectivas y eficientes.

Comunicación y colaboración: Sólidas habilidades de comunicación para colaborar eficazmente con los equipos de desarrollo, operaciones y negocios, y para promover una cultura de concienciación sobre Seguridad.

Aprendizaje continuo: Compromiso de mantenerse actualizado con las últimas tendencias, amenazas y tecnologías de Seguridad, y la voluntad de aprender y adaptarse continuamente.

Respuesta al incidente: Habilidades en el manejo de violaciones e incidentes de Seguridad, incluida la capacidad de liderar o participar en esfuerzos de respuesta a incidentes.

Conocimiento de la arquitectura de aplicaciones: es fundamental dominar los patrones y técnicas clave a la hora de diseñar y desarrollar aplicaciones. Es un plus conocer las herramientas de orquestación.

Familiaridad con la automatización: en la vertiginosa industria actual, la automatización es fundamental, ya que optimiza los procesos operativos y de Seguridad, incluida la respuesta y mitigación de incidentes.

Imágenes creadas por dooder

Síntomas Comunes de falla en DevSecOps



Imágenes creadas por dooder

Síntomas comunes de falla en DevSecOps

Los síntomas comunes de DevSecOps fallidos o faltantes incluyen:

Culpa y evasión: Se culpa al equipo de Seguridad por ralentizar los proyectos y los equipos de proyecto con frecuencia "eluden" la Seguridad.

Revisiones de Seguridad lentas e ineficaces: Las revisiones de Seguridad de nuevas aplicaciones tardan semanas y producen poca información procesable.

Mala experiencia del usuario: La experiencia del usuario para la autenticación y autorización es incómoda e inconsistente.

Complejidad inmanejable: La lista de vulnerabilidades es abrumadoramente grande, compleja y está creciendo rápidamente.



Brechas en el perfil de riesgo: **Gran** parte de tu stack de tecnología tiene vulnerabilidades de Seguridad no descubiertas.

Imágenes creadas por dooder

Errores típicos y diagnósticos



Imágenes creadas por dooder

Errores típicos

Esperaron demasiado para iniciar:

Demasiadas empresas postergan la implementación de iniciativas DevSecOps hasta que "sospecharon o validaron" una violación de datos vinculada a componentes de código para responder con una campaña propia de Seguridad de datos en toda la empresa.

No hacer que la Seguridad sea una prioridad:

Los desarrolladores de Software generalmente son conscientes de la necesidad de una mayor seguridad de Software pero simplemente no actúan de acuerdo con esa conciencia, dado que "no tienen tiempo para gastar" en temas de DevSecOps.

Centrarse en la velocidad en lugar de la Seguridad y la calidad:

Otra razón principal por la que las compañías luchan con los despliegues de DevSecOps es que sacrifican la Seguridad y la calidad por la velocidad. Sí, sacar productos al mercado rápidamente es una base fundamental para el éxito de la organización.



Imágenes creadas por dooder

Errores típicos

No involucrar al equipo de Seguridad:

DevOps implica más que simplemente unir los equipos de desarrollo y operaciones, es un proceso continuo de desarrollo y automatización de Software que incluye Seguridad, auditoría y cumplimiento. Muchas empresas cometen el error de no seguir sus prácticas de Seguridad con mucha antelación.

No prepararse para el cambio cultural:

Una vez que se tiene las herramientas adecuadas para las prácticas de DevOps, probablemente enfrentaremos un nuevo desafío fundamental y es tratar de hacer que los equipos utilicen las herramientas para un desarrollo más rápido, pruebas automatizadas, entrega continua y monitoreo. Pero, ¿Tu cultura de desarrollo o de operaciones está lista para los cambios?

Hacerlo complicado por default:

La Seguridad es vista por muchos como un problema que solo puede ser resuelto por consultores altamente calificados y remunerados (Ojala fuera así). Si bien existe una necesidad absoluta de estas habilidades al revisar arquitecturas y realizar auditorías, no se requiere este nivel de experiencia cuando se trata de controles de seguridad (Conocidos – conocidos) se pueden automatizar.



Imágenes creadas por dooder

Errores típicos

¿Seguridad? ¿Dónde está Seguridad?

En muchas implementaciones DevOps que vimos, un error típico fue ausencia de consideración de la integración de las herramientas de seguridad actualmente existentes para SAST, DAST y otros niveles de escaneo. La parte de **”Integración"** se consideró para DevOps, pero no se extendió a la Seguridad. Por lo tanto, se perdió una importante integración con herramientas empresariales ampliamente utilizadas. Aunque esto fue realmente un fallo de comunicación, las herramientas que se utilizaban estaban muy bien documentadas y es importante considerar el ecosistema holístico de las empresas al seleccionar nuevas herramientas.

¿Quién va a ser el propietario?

Después de considerar el tipo de herramienta que necesitaríamos para permitir una entrega más rápida de Software, a veces nos damos cuenta de que existen algunos solapamientos en la propiedad de las herramientas, especialmente en lo que respecta a la Seguridad.

En concreto, para el análisis y la gestión de dependencias estaba claro que habría que compartir responsabilidades entre los equipos. Es importante definir un buen modelo de responsabilidad compartida de las herramientas, ya que la gestión de una nueva herramienta va a formar parte del trabajo para equipos ya muy ocupados.



Imágenes creadas por dooder

Errores típicos

La colaboración debe ser voluntaria

Un día mientras hablábamos con uno de nuestros clientes de los progresos y de las mejoras, el mismo dijo: "¡La colaboración es difícil!, tengo que trabajar mucho para que la gente colabore. En un abrir y cerrar de ojos, vuelven a trabajar de forma aislada haciendo malas suposiciones que conducen a malas decisiones y malos resultados para nuestros clientes. **Lucho contra el aislamiento todos los días**". Ese fue un momento de luz, aunque sabíamos que nuestro mejor momento era cuando dejábamos que las ideas se convirtieran en algo tangible a través de la colaboración, ahora también teníamos que impulsar intencionadamente la colaboración entre equipos.

La práctica no siempre logra la perfección.

Sólo la práctica perfecta hace la perfección. A medida que se introducían nuevos conceptos en las organizaciones, nos dimos cuenta de que aunque los equipos tenían en cuenta estos puntos durante la implementacion inicial, las implementaciones posteriores eran cada vez menos alineadas con las normas que se pusieron en marcha.

Al principio, eran estrellas del rock, pero, con el paso del tiempo el equipo cambia, lo que introduce nuevos individuos que no tienen el mismo nivel de exposición que los miembros originales del equipo. Por ello, los conceptos que antes se aplicaban en todo el equipo se van quedando atrás.



Imágenes creadas por dooder

Como evitar o mitigar?

Cambio de mentalidad: El software se ha convertido en la piedra angular de la innovación y el crecimiento económico, incluso expandiéndose a industrias tradicionalmente no tan tecnológicas. Como tal, es una prioridad para los profesionales de TI que se elimine la fricción falsa entre seguridad y desarrollo.

Requisitos tecnológicos cambiantes: Claramente existe la necesidad de que la Seguridad de las aplicaciones se convierta en un aspecto integral del proceso DevOps, pero ahora que hemos superado la barrera mental necesitamos trabajar en la tecnología para que pueda funcionar conjuntamente sin problemas.

Integrar para "fallar rápidamente“: Inspeccionar la calidad del producto lo antes posible es parte de la filosofía de DevOps. Se debería comenzar cuando el desarrollador comience a codificar y la integración de la Seguridad debería en el IDE para escanear el código a medida que se escribe. Esto no solo alerta a los desarrolladores sobre los problemas a medida que se presentan, sino que también tiene un efecto educativo que puede capacitar al desarrollador para codificar de forma segura desde el principio.

“Como nos gusta decir, el código sale seguro desde los dedos de los desarrolladores”

Imágenes creadas por dooder

Como evitar o mitigar?

Asegúrarse de que no haya falsas alarmas en el proceso: Como la industria ha aprendido, una tecnología que informa demasiados falsos positivos será ignorada y no será adoptada. Se pierde demasiado tiempo valioso. Eso puede ser tolerable si el problema de Seguridad es real, pero es intolerable si el hallazgo es falso positivo.

Construir campeones de Seguridad: La mayoría de los equipos de Seguridad son demasiado pequeños para involucrar activamente a todos los equipos de desarrollo. Los campeones de Seguridad son una excelente manera de extender su alcance. La capacitación de uno o dos desarrolladores de cada equipo de aplicaciones hace posible que la Seguridad esté siempre en la sala y la capacitación es beneficiosa tanto para la empresa como para el colaborador. Sin embargo, elegir al campeón adecuado es clave. Podes buscar a los influenciadores.

Mantener la visibilidad operacional: La Seguridad no debe detenerse después de la implementación. Al igual que con otros aspectos de DevOps, una solución bien diseñada debe tener un ciclo de retroalimentación cerrado desde su puesta en producción, en caso de problemas de Seguridad.

Imágenes creadas por dooder

Como evitar o mitigar?

Pensamiento basado en riesgo: Insistir en que el equipo de desarrollo se apropie del problema de la Seguridad del producto y acepte que el papel de la Seguridad no es el guardián, sino de un asesor.

Conciencia tranquila: Crear una cultura en la que sea más importante hacer lo correcto en todo momento, aunque eso signifique que alguien pierda sus objetivos métricos trimestrales. Esto puede significar que tengas que cambiar una política, hacer una excepción o enfrentarte a tu jefe o al departamento legal.

Bi modal o velocidad: Cree diferentes caminos para diferentes personas del equipo y acelerar a los que tienen prácticas DevOps maduras. Invertir lo mínimo para proporcionar el apoyo tradicional de Seguridad de las aplicaciones a los equipos con menor madurez de DevOps, pero insistir en que se pongan en el camino de la madurez de DevOps antes de hacer más y conectar a esos equipos con el grupo adecuado para que inicien este camino de DevOps.

Imágenes creadas por dooder

Como evitar o mitigar?

La tecnología no debe ser una caja negra: En muchas empresas vimos que la resiliencia necesitaba muchas manos a la obra y un interés en aprender más sobre el aspecto técnico de la plataforma. Se enfrentaban al hecho de que la tecnología era una caja negra. Para evitar esto, podemos establecer una **"hoja de ruta de las mejoras tecnológicas"** para la plataforma. Además, establecer revisiones periódicas y sesiones de priorización para mejorar gradualmente la transparencia.

CI/CD: Optimizar la selección de las herramientas para aquellas que permitan una rápida y fácil integración en el pipeline CI/CD, en lugar de la integración IDE. Tambien, sugerimos crear recetas y plantillas para permitir una fácil integración CI/CD de las herramientas de escaneo de ciberseguridad.

Imágenes creadas por dooder

Bonus Track

TOP 10 de los STOPERS en la adopción de DevSecOps



Imágenes creadas por dooder

TOP 10 de los STOPERS en la adopción de DevSecOps

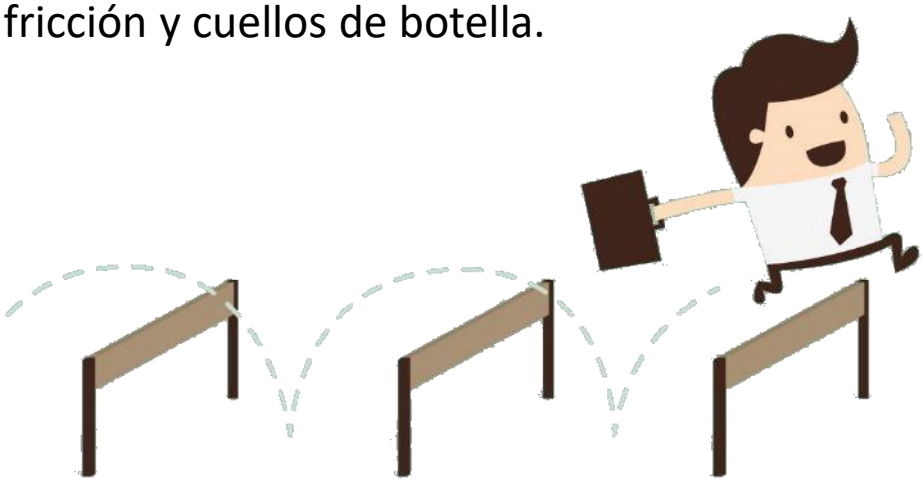
Entornos inconsistentes: Cuando los entornos de desarrollo, pruebas y producción están configurados de forma diferente, se producen errores.

Madurez del SDLC: Muchas organizaciones todavía tienen procesos en cascada o procesos ágiles de tipo "water-scrum-fall", lo que limita su capacidad.

Proceso de gestión de cambios heredado: En muchas organizaciones, existen barreras y revisiones manuales de cambios para obtener aprobación que tienden a ralentizar y los cambios.

Falta de madurez operativa: Los desarrolladores tienen un acceso limitado a las herramientas de supervisión del rendimiento de las aplicaciones, por lo que carecen de información sobre cómo funciona.

Prácticas de prueba obsoletas: Las pruebas manuales simplemente no pueden escalar y seguir el ritmo de entrega a la velocidad ágil y DevOps, también son una fuente común de fricción y cuellos de botella.



Imágenes creadas por dooder

TOP 10 de los STOPERS en la adopción de DevSecOps

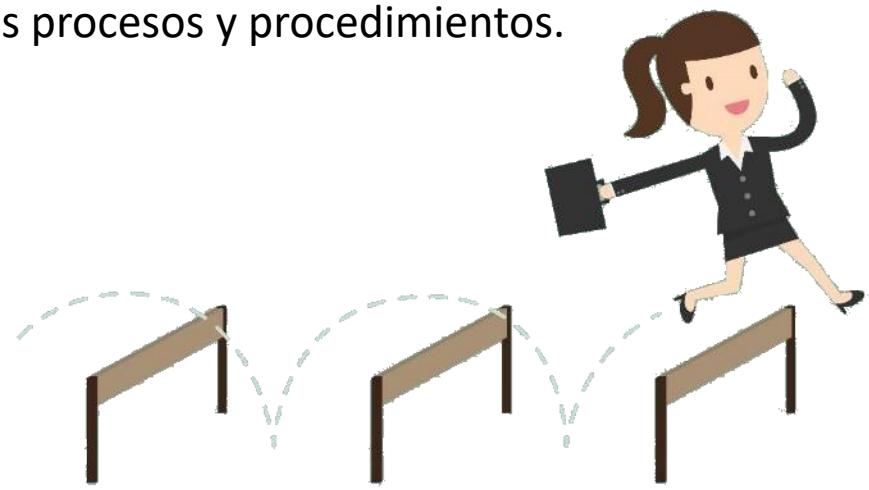
Intervención manual: Todo el trabajo manual que se realiza lleva tiempo, es propenso a errores y ralentiza la entrega.

Automatizar el desperdicio: Si el equipo automatiza el proceso existente, probablemente esté automatizando un proceso ineficiente. Hay que mejorar el proceso antes de automatizarlo.

Incentivos competitivos o desalineados y falta de propiedad compartida: A menudo, los silos tienen sus propias metas, objetivos y KPI, si no están bien diseñados a menudo esto crea situaciones en las que los equipos no comparten los mismos objetivos comunes.

Dependencia de esfuerzos heroicos: En el Proyecto Phoenix, "Brent" era el héroe vital para todos los proyectos. En realidad, era un cuello de botella que ralentizaba a los equipos porque dependían de él.

La gobernanza como una idea de último momento: A menudo, las transformaciones no tienen en cuenta cómo deben cambiar otros procesos y procedimientos.



Imágenes creadas por dooder

Conclusiones



Imágenes creadas por dooder

Conclusiones

En nuestra experiencia los **“Casos de no éxito”** más comunes están relacionadas con la cultura, la colaboración y la adopción / cambio.

Hacer cambios sin tener ningún medio de medir la eficacia no es mejor que hacer disparos al azar en la oscuridad con la esperanza de golpear algo... cualquier cosa... y luego llamar a ese golpe un **"éxito".**

Nadie tiene la receta perfecta para la cultura DevSecOps ideal, pero si nuestros años de consultoría no ha enseñado algunas prácticas recomendadas para comenzar.

“Desarrolle su grito de guerra para la transformación de DevSecOps”



Conclusiones

Cuando hablamos de incluir Seguridad y las pruebas en el mundo DevOps (DevSecOps), lo primero que nos viene a la mente es **“DevSecOps es un esfuerzo de equipo”** y cuando los equipos trabajan juntos, todos ganan.

Desarrollo: Al integrar y automatizar los controles y pruebas de Seguridad en todo el SDLC, los equipos de desarrollo optimizan el desarrollo y crean un Software seguro y de alta calidad.

Seguridad: Cuando los equipos de Seguridad trabajan con el equipo de desarrollo para llevar las pruebas de Seguridad lo mas temprano en el SDLC, los costos y el esfuerzo disminuyen y las tasas de cumplimiento aumentan.

Operaciones: Los equipos de operaciones pueden evitar vulnerabilidades de las aplicaciones en producción cuando tienen una visión temprana de los componentes utilizados en las aplicaciones y los riesgos asociados a esos, implementado en todo caso controles compensatorios.



Conclusiones

Es importante recordar que las recomendaciones brindadas en este e-book no deben ser actividades únicas, sino procesos continuos.

Debido a la naturaleza dinámica y cambiante de DevOps, lo ideal es que las actividades de seguridad se realicen de manera continua.

Estas mejores prácticas deberían ser gestionadas por un grupo de expertos con experiencia en adopción de seguridad en el ciclo de vida de desarrollo.

Te gustaría saber más sobre cómo Cloud Legion puede ayudarte en sus planes de DevSecOps Visítanos en https://cloudlegion.com.ar/



AUTORES



Imágenes creadas por dooder

Luciano Moreira



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

https://twitter.com/Luciano_m_cruz

Me gusta describirme como un buscador de soluciones en todos los ámbitos laborales y personales. 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, Security AI), Auditor Líder ISO 27001:2013, 27018, 27017, 20000, 22301 y 9001:2015.
Cofundador y Tribe leader de DevSecOps Latinoamerica, Ex 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 varios institutos.

Cuento con más de 19 años de experiencia en IT, de los cuales los últimos 15 años trabaje en una docena de proyectos en todas las capas de seguridad y infraestructura hibridas. Durante los últimos años con foco en seguridad en desarrollo y arquitectura de seguridad en entornos de computación en la nube.

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 más si me acuerdo de mis principios donde teníamos que desplegar maquinas físicas, armar clústeres, “virtualización si teníamos suerte”.

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

Cofundador de DevSecOps Argentina, tengo mas de 13 años de experiencia en IT, de los cuales los últimos 8 años en las áreas de Infraestructuras hibridas, cloud, DevOps y automatizacion de herramientas.

Participe en muchos proyectos de infraestructura, networking, migración e implementación en nubes publicas y privadas, 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 OpenSource puede tranquilamente coexistir con tecnologías propietarias sin ningún tipo de limitaciones. Entusiasta de DevOps y infraestructuras agiles.

Martin Ambort



https://www.linkedin.com/in/martin-ambort-08a7a8b3

https://twitter.com/mambort2014

Soy una persona simple, fan de la tecnología que hacen la vida más fácil y un entusiasta del movimiento de DevOps y la Calidad del producto de software. Creo que compartir el conocimiento es una de las maneras más eficaces para construir una sociedad más justa, coherente y evolucionada, yo creo que el conocimiento debe ser libre y que todos deben tener acceso a el.

Tengo un profundo respeto por la naturaleza, Me gusta la música de calidad como, también la literatura y sobre todo películas y series.
Soy Arquitecto de Seguridad y Calidad - por la auto enseñanza y mi foco se centra en las tecnologías modernas. En los últimos años me he especializado en la automatización de la infraestructura, especialmente la automatización definida por código, la calidad del producto de software y la seguridad en todo el proceso y ciclo de vida de desarrollo.
Consultor, Auditor y instructor experto en TI con un enfoque profundo en seguridad y cumplimiento. Cuento con más de 15 años de experiencia en la gestión de redes informáticas, servidores y seguridad.
DevSecOps chapter Líder en Cloud Legion
Referente en DevSecOps en la región
DevSecOps Engenieer (DSOE)
Cybersecurity Team of the Year en los premios Cybersecurity Excellence Awards del 2019
Scrum Master
Auditor Líder ISO 27001:2013, ISO 25000, ISO 9001:2015
Analista de Sistemas de Computación

Links y referencias



Imágenes creadas por dooder

Links y referencias

- https://cloudlegion.com.ar
- https://devsecops-latam.org
- https://www.devsecops.org
- https://www.ruggedsoftware.org
- https://en.wikipedia.org/wiki/DevOps
- https://devsecops.org
- https://en.wikipedia.org/wiki/Muri_(Japanese_term)
- https://en.wikipedia.org/wiki/Muda_(Japanese_term)
- https://en.wikipedia.org/wiki/Mura_(Japanese_term)
- https://dojo.target.com
- https://itrevolution.com/devops-dojo-captial-one
- http://dearauditor.org
- https://riesenfeld.com.mx/wp-content/uploads/2021/09/BSC_para_DevSecOps.pdf

CLOUD LEGION

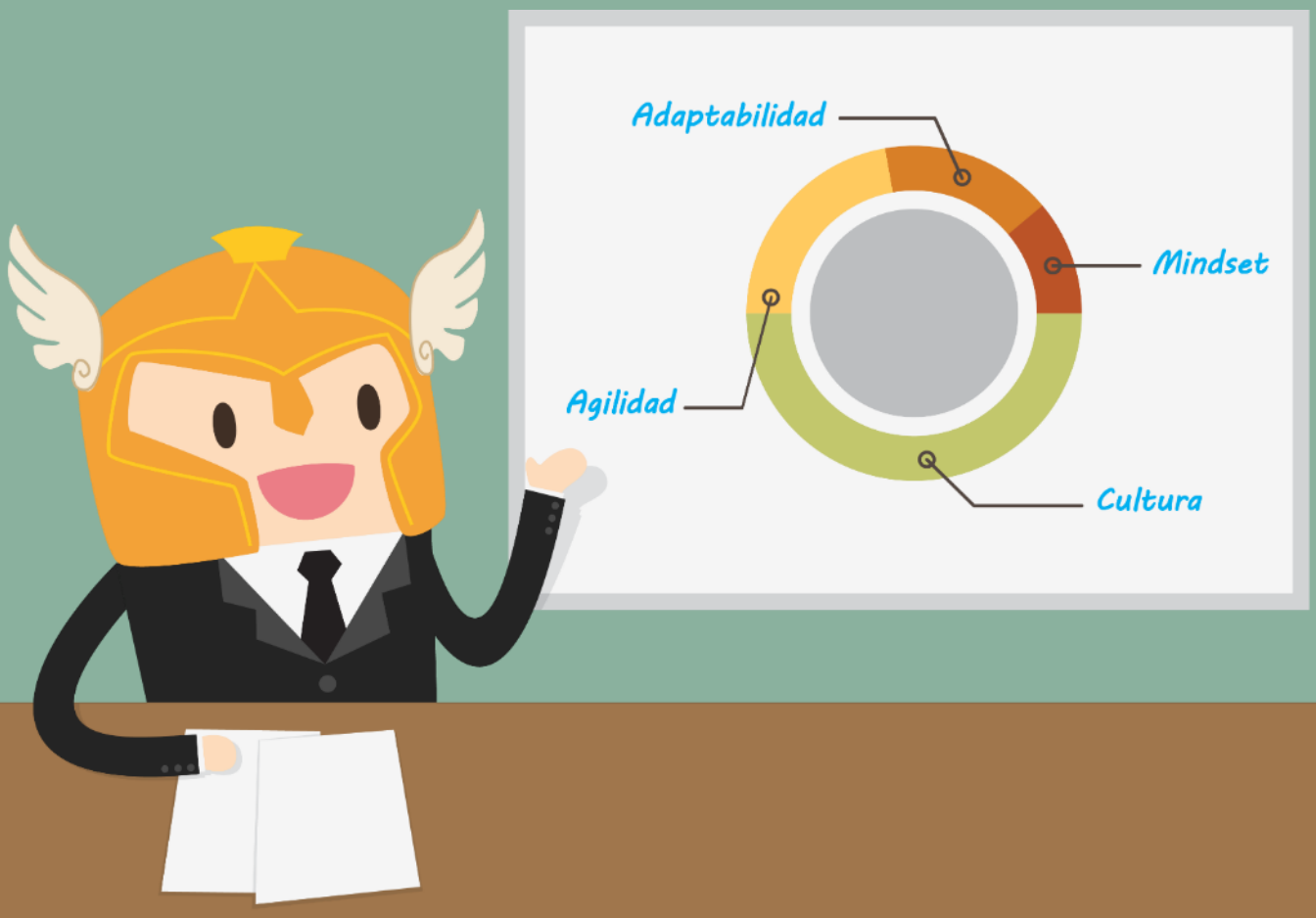
Somos mas que una consultora, somos una legión



Somos...

Una empresa multidisciplinaria centrada en la innovación y protección de la información de las empresas.
CLOUD LEGION trabaja con empresas y organizaciones tanto públicas como privadas de todos los sectores de la industria para identificar, cumplir, asegurar y administrar su activo más crítico: Los datos.

Nuestra Misión
Ayudar a organizaciones a tener éxito en un mundo digital, a mejorar su posición competitiva y su eficiencia mediante el uso de la tecnología informática de forma segura, en un entorno altamente competitivo, constantemente cambiante y cada vez más regulado.



Nuestra filosofía

Estas son las cinco "C" que definen los valores principales que guían nuestra Legion:

COMPROMISO

Enfoque total en las necesidades del cliente: Implementamos servicios y proyectos, pero sabemos que todas las organizaciones tienen necesidades distintas y particulares. Por esta razón, cada una de nuestras implementaciones está pensada en detalle y buscamos la forma en hacer que esta se enfoque en las necesidades verdaderas de nuestros clientes.

CREATIVIDAD

Soluciones simples: Nuestras soluciones buscan siempre ser faciles e intuitivas, difícilmente requieren de instrucciones o manuales. No tratamos nuestros proyectos o servicios como una mercancía más. Sabemos que todo proyecto o servicio tiene una razón de ser, y esta razón es la de suplir una necesidad para nuestros clientes de forma simple.

COMPETENCIA

Ser buenos no es suficiente: En Cloud Legion sentimos que la transformación digital tambien es nuestra responsabilidad y no solo de nuestros clientes. Intentamos siempre innovar y romper paradigmas en los temas y barreras tecnológicas en esta área. Somos un grupo académico, de investigadores que buscan siempre nuevas soluciones en tecnología y seguridad de la información.

Nuestra filosofía

CALIDAD

¿Qué es la calidad en Cloud Legion? Brindar soluciones de alta calidad para obtener la satisfacción de nuestros clientes, superando sus expectativas, y bajo un sistema de gestión de calidad, según los estándares establecidos por ISO. En Cloud Legion trabajamos todos los días para brindar soluciones informáticas de alta calidad, orientados a la mejora continua de los procesos, focalizados en el servicio al cliente.



COMUNIDADES

“DevSecOps Latinoamérica fue fundado en el año 2017, "Originalmente como DevSecOps Argentina" por el equipo da Cloud Legion como respuesta a las tendencias y necesidades de la comunidad local. Hoy por hoy luego de un camino recorrido y mucho apoyo de la comunidad de profesionales de la región nos orgullece ampliar nuestro alcance y llamarla DevSecOps Latinoamérica”

DEVSECOPS Argentina by Cloud Legion

2017

2020

DEVSECOPS Latinoamérica

Logros y reconocimientos

CLOUD LEGION

GLOBEE AWARDS 2024 GLOBEE AWARDS SILVER WINNER

CYBERSECURITY

Business Philanthropist of the Year (Cybersecurity)

2023 CHAMPION CYBERSECURITY AWARENESS MONTH

2023 Nuevamente somos **"Cybersecurity Awareness Month champion"** by NCSA & CISA of the U.S. Department of Homeland Security.

GLOBEE AWARDS 2023 GLOBEE AWARDS BRONZE WINNER

CYBERSECURITY

Chief Technology Officer of the Year (Cybersecurity)

TOP CYBERSECURITY COMPANY Clutch ARGENTINA 2023

TOP CLOUD CONSULTING COMPANY Clutch BUENOS AIRES 2023

TÜVRheinland CERTIFICADO

Sistema de Gestión
ISO 9001:2015
ISO/IEC 27001:2013

www.tuv.com
ID 9000006672

Certificamos **ISO 9001:2015 y ISO/IEC 27001:2013** en todos los procesos de la empresa.

2022 CHAMPION CYBERSECURITY AWARENESS MONTH

Somos **"Cybersecurity Awareness Month champion"** by NCSA & CISA of the U.S. Department of Homeland Security.

Info Security Products Guide 2020 GLOBAL EXCELLENCE BRONZE

Fuimos nombrados ganadores en la 16ta edición de los Premios a la excelencia **global** 2020 de Info Security PG´s **(Silicon Valley, USA)**

NETWORK PRODUCTS GUIDE IT World Awards 2020 BRONZE

NETWORK PRODUCTS GUIDE IT World Awards 2020 GOLD

Fuimos nombrados ganadores CTO of the Year | IT Software & Security Software en la 15th Annual IT World Awards 2020 de Network Products Guide **(Silicon Valley, USA)**

Logros y reconocimientos

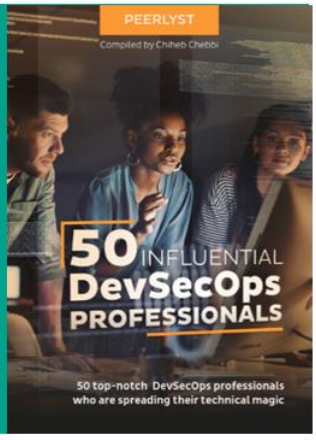
2019 to date Nuestro CDSO Chief DevSecOps strategy Officer fue reconocido como **Ambassador of DevOps Institute (Global Association)**

2021 GLOBEE AWARDS BRONZE WINNER
CYBER SECURITY GLOBAL EXCELLENCE AWARDS

Ganamos el premio Cybersecurity Excellence Awards 2019 como **Best Cybersecurity Startup y Cybersecurity Team of the Year (Global Association)**

Nuestro CDSO - Chief DevSecOps strategy Officer otro año es seleccionado como **MVP en Microsoft Azure, Developer Technologies**

2019 Christian Ibiri & Luciano Moreira, fueron seleccionados por la gente de peerlyst, **"50 Influential DevSecOps Professionals" (Global Association)**



Fuimos finalistas del premio Cybersecurity Excellence Awards 2017 en la categoría Best Cybersecurity Startup

Ganamos el premio Cybersecurity Excellence Awards 2018 en la categoría Best Cybersecurity Startup

Cloud Legion es la primer consultora en Latinoamérica en lograr ser **CSA Global Consultancy de Cloud Security Alliance (American continent)**

Nuestro CDSO - Chief DevSecOps strategy Officer fue elegido Cybersecurity Consultant of the Year en los premios Cybersecurity Excellence Awards 2016, 2017 y 2018.

Nuestro Equipo



Martin Ambort
Chief Information Security & Privacy Officer



Christian Ibiri
Co-CEO - Chief Delivery & Operating Officer



Luciano Moreira
Co-CEO & Chief Transformation & Strategy Officer



Agustina Manzanzani
Head of People and Culture

BOARD OF ADVISORS



Jim Reavis
Cloud Security Strategy



Federico Pacheco
Security Education & Research



Julio Balderrama
Compliance Strategy

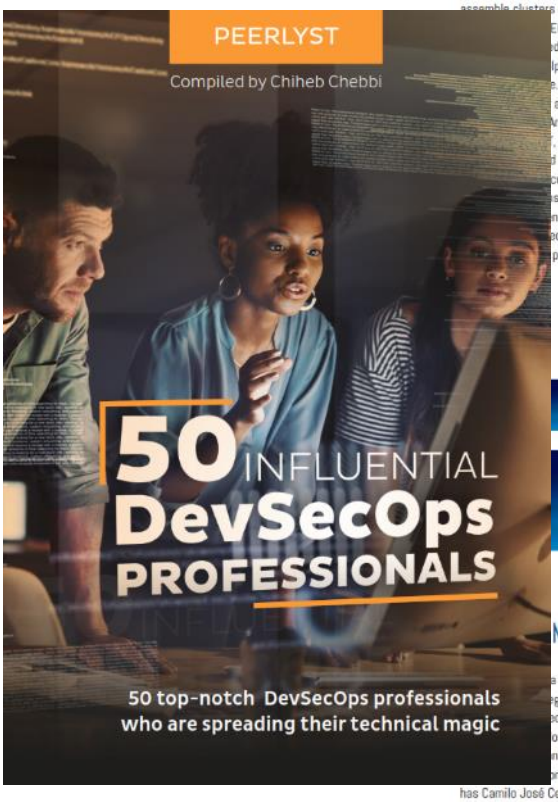


Santiago Cavanna
Security Digital Transformation Strategy

Somos referentes en DevOps y DevSecOps

DEVSECOPS Argentina
by Cloud Legion





AMBASSADOR
DevOps INSTITUTE
Human of DevOps



Somos referentes en Seguridad y gestión de la nube

Somos la primera consultora en la región en calificar como **CSA Global Consultant** y contar con un **Certified CSA STAR Auditor**

El Programa de consultoría global de CSA (CSA GCP) es un marco que permite a los clientes de la nube trabajar con una red de profesionales y organizaciones de seguridad de confianza que ofrecen servicios profesionales cualificados basados en las mejores prácticas de CSA. Estos proveedores tienen un amplio conocimiento de los retos a los que se enfrentan las organizaciones al pasar a la nube.

https://cloudsecurityalliance.org/global-consultancy/registry/

Nuestro CDSO - Chief DevSecOps strategy Officer es **MVP en Azure**

Somos referentes en Cumplimiento Normativo

Contamos con profesionales, certificado y acreditados por diferentes entidades en las principales normas de la industria. Expertos en Sistemas de Gestión, Cumplimiento Normativo ISO y Cumplimiento Legal

Nos respaldan el 100% de empresas certificadas a la primera,

y un 95% clientes que repiten



CSA STAR LEAD AUDITOR

TISAX®

WORLD COMPLIANCE ASSOCIATION
MIEMBRO ASOCIADO

CQI | IRCA

ISO 27017 CERTIFIED
Cloud Security

ISO 27018 CERTIFIED
Privacy Protection

bsi.

onetrust Privacy PROFESSIONAL

TÜVRheinland CERTIFIED

GENERAL LAW FOUNDATION
CertiProf
Professional Knowledge
DATA PROTECTION
LGPDF™



Socios Tecnológicos





























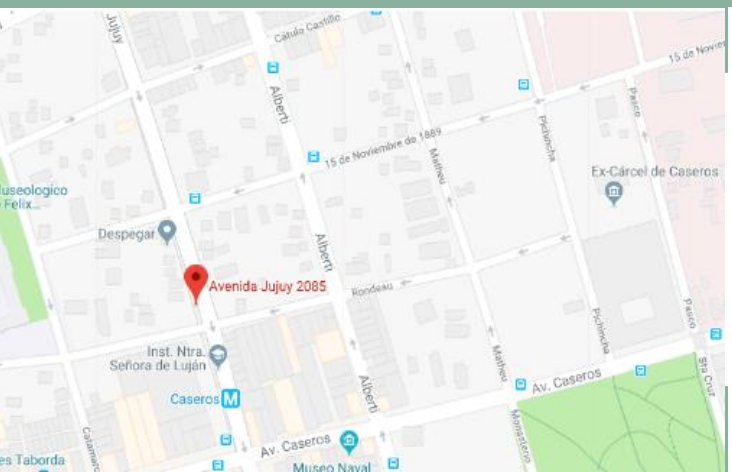






Imágenes creadas por dooder

CC BY SA



Av. Jujuy 2085 2 "C",
Parque Patricios, Capital Federal, Buenos Aires

+54 9 11 3233-1856

info@cloudlegion.com.ar

https://cloudlegion.com.ar