**Oscar David Mesa Calderon**

# Diferencias entre DevOps y S-SDLC.

Las diferencias ms notables se dan en aspectos del ciclo de vida de desarrollo de software de ambos. Mas específicamente entre la interacción Equipo de Desarrollo con Equipo de Operaciones, donde Devops se centra entregar iteraciones frecuentes de manera reiterada con una proceso estándar y confiable, SDLC sacrifica las repeticiones de iteraciones que son menos frecuentes, para ganar tiempo de entrega final e implementación independiente de las fallas que se puedan presentar, las cuales se solución en la marcha de la implementación. Basado en [1]

De fondo SDLC con el factor de “Seguridad” pretende mitigar la posibilidad de fallos en la implementación, enfatizando en la fase previa al desarrollo para tener una idea unificada entre ambos equipos de desarrollo y operación. No es una solución al completo, pero ayuda a tener una coherencia entre ambos. A diferencia de DevOps que implementa una continua relación entre ambos equipos y cliente lo que exige mayor tiempo de las partes involucradas, pero garantiza productos de calidad y certeros. Lo que produce carias entregas del producto en forma de versiones de manera “simultanea” gracias a la sincronización de ambos equipos de trabajo. Basado en [1]

# Diferencias entre SQUARE y SREP.

La principal diferencia es en las etapas donde se ejecuta los procesos de seguridad y el enfoque de dichos procesos. Por un lado, SQUARE se enfoca en las etapas tempranas donde realiza un análisis enfocado a la seguridad con aspectos de patrones de ataque, vulnerabilidades, amenazas, malos casos de uso o abusos. Por medio de 9 pasos: Acuerdo definiciones, Identificación de metas, Desarrollo de artefactos, Evaluación Riesgos, Calificar Requisitos. Priorizar requisitos y Revisión por partes. En contraparte SREP se realiza durando cada fase como un proceso extra donde se sacan requisitos de seguridad donde se explican atributos de seguridad y evaluación de los mismo. Por medio de una serie de pasos semejantes a los anteriores sin embargo se diferencia en que no parte de identificar metas de seguridad sino de identificar activos críticos y vulnerables para poder llegar a objetivos de seguridad para garantizar los activos. Basado en [2]

# Diferencias entre DSDM y ASD.

La principal diferencia es en la cantidad de productos y su condición y tiempos en que se entregara. ASD propone una entrega final que se va construyendo por módulos los cuales pueden llegar a ser funcionales o no por si solos. Implementando la colaboración entre el grupo de trabajo para generar un aprendizaje en cada módulo que atribuye retroalimentación para el siguiente. Una vez se entrega el ultimo modulo el trabajo se da por concluido. Por otra parte, DSDM trabaja por módulos funcionales a los cuales se les realiza mantenimiento mientras se construye el siguiente. Gracias a este mantenimiento la retroalimentación es mayor y hace que los prototipos incrementes sus funcionalidades a medida que trascurre la metodología. Hay que mencionar que funciona con restricciones de tiempos muy apretadas. Basado en [3]

# Diferencias entre CbyC y CLASP.

La principal diferencia es el momento de validación y las pruebas que se hace a los productos. Por un lado, CbyC, hace retroalimentación y validación a medida que se realizan entregas. Lo que implica prototipos sucesivos que son participes no solo en las diferentes ent5rgas si no también en la definición de cada iteración para ser aprobada por el cliente. En cuanto a pruebas se realiza validación de seguridad a amenazas según el producto. En cambio, CLASP realiza validación y revisión en cada etapa, las cuales de dividen por actividades y roles encargados, con el objetivo de asociar posibles fallos preestablecidos en 5 niveles a las distintas actividades. Con ellos entregar producto “seguros” a los cuales se realizarán prueba de penetración, pruebas de seguridad basada en riesgos y casos de abuso. Basado en [4]

# Diferencias Control Interno y Auditoria Informática

La principal diferencia es en momento y objetos sobre los que son implementados. El control interno informático se encarga de analizar diariamente las actividades del sistema con el objetivo de velar por el cumplimiento de procedimientos estándar y normas en un marco de requerimientos legales. Mientras que la Auditoria Informática recoge, agrupa y evalúa los sistemas informáticos para determinar si están salvaguardando los activos. Por medio de un proceso constante de revisión en el Diseño, Realización e implementación de las aplicaciones informáticas. Basado en [5]

# Bibliografía

[1] <https://www.bitwiseglobal.com/blogs/comparing-devops-vs-traditional-sdlc/>

[2] <http://web.usbmed.edu.co/usbmed/fing/v3n1/v3n1a2.pdf>

[3]<https://www.researchgate.net/publication/334000094_ANALISIS_COMPARATIVO_DE_METODOLOGIAS_AGILES_DE_DESARROLLO_DE_SOFTWARE_UNA_REVISION_BIBLIOGRAFICA>

[4]<https://repository.libertadores.edu.co/bitstream/handle/11371/736/TrujilloSilvaDianaMarcela.pdf?sequence=2>

[5] <https://es.calameo.com/read/002090061e8f4ceab8b46>