Creación Automática de Pipelines Para Devops Usando Líneas de Productos de Software

Robinson Cruz Delgado

Anteproyecto presentada(o) como requisito parcial para optar al título de:
Magister en Ingeniería de Software

Director(a):
Ph.D. Luisa Fernanda Rincón Pérez

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería
Departamento de Electrónica y Ciencias de la Computación
Cali, Colombia
23 de mayo de 2022

Ficha Resumen

Anteproyecto de Trabajo de Grado

Posible Título: Optimización de scripts para despliegue continuo con líneas de productos de software

1. Área de trabajo: Devops y líneas de producto
2. Tipo de proyecto (Aplicado, Innovación, Investigación): Aplicado
3. Estudiante: Robinson Cruz Delgado
4. Correo electrónico: robcruzd@javerianacali.edu.co
5. Dirección y teléfono: Cra 19f # 27 - 33 Sur Olaya Bogotá, 3135818403
6. Director: Luisa Fernanda Rincón Pérez
7. Vinculación del director:
8. Correo electrónico del director: lfrincon@javerianacali.edu.co
9. Co-Director (Si aplica): Jaime
10. Grupo o empresa que lo avala (Si aplica):
11. Otros grupos o empresas:
12. Palabras clave(al menos 5): Devops, Pipeline, Automation, Software Product Lines
13. Fecha de inicio:
14. Duración estimada (en meses): 6 meses
15. Resumen: Debe contener el tema a trabajar en el proyecto de grado, su importancia, la problemática que aborda, los objetivos propuestos, resultados esperado y posibles aplicaciones. Debe ser redactado en un solo párrafo y no contener espacios entre líneas ni sangría. Debe ser escrito con máximo 400 palabras

Índice

Estás acercándote más a definir el problema, pero todavía estás arrancando muy atrás en la historia y no explicas completamente la situación problematica, las soluciones que existen que no son adecuadas y las estrategias que estás planeando usar

1. Definición del problema

1.1. Planteamiento del problema

~~Debido al crecimiento exponencial que ha presentado~~ la industria del software en el mercado mundial en los últimos años, como se observa en (ReportLinker, 2022), en el cual se menciona que se espera que de $1.141 billones de dólares en 2021, se llegue a los $1.307 billones en el 2022. El software se ha convertido en un pilar irrefutable de las compañías de hoy en día, ya que, como se menciona en (Wilburn et al., 2018), la tecnología esta acelerando su habilidad para ayudar a los negocios a hacer más con menos y generar mejores resultados al mismo tiempo.

Sin embargo, la inclusión del software a las empresas ~~ha implicado~~ grandes retos e inconvenientes, ~~ya que, al ser este campo de desarrollo de software tan cambiante, en poco tiempo quedan obsoletas soluciones previas y se hace necesario estar continuamente trabajando en mantener los sistemas y desplegar nuevas versiones.~~ Por tanto, se presenta una lucha constante entre la complejidad de agilizar cada uno de los despliegues que los equipos de desarrollo implementan y los requerimientos que la operación de TI tiene para llevar cada cambio a producción. Sin contar, las necesidades que el equipo de seguridad y de QA tienen para aceptar la viabilidad de estos cambios.

Buscando agilizar el proceso de desarrollo y mejorar el camino entre desarrollo y producción con buenos estándares de calidad, surge la metodología de desarrollo DevOps, la cual se centra en cerrar la brecha entre desarrollo y la operación de TI, enfocándose en la comunicación y colaboración de los equipos, la integración continua, los controles de calidad y la entrega continua de una forma automatizada, utilizando un conjunto de prácticas de desarrollo, como lo definen en (Jabbari et al., 2016).

No obstante, aunque en cada paso del proceso de automatización para ~~devops~~ existen múltiples herramientas que se pueden usar, como se observa en (Leite et al., 2019). Este se ha convertido en un gran desafío para los ingenieros ~~devops~~, ya que como lo mencionan en (Bonda et al., 2021), no hay un estándar para la creación

de estas herramientas para devops y para la comunicación entre ellas, por lo que cuando se desea cambiar una de estas herramientas en un proyecto, existe una gran complejidad en la tarea y en ocasiones llega a ser imposible. Lo que dificulta la generación de los scripts para los pipelines, debido a que también varían estos según las herramientas a usar.

Adicionalmente, los pipelines de desarrollo tienen otro problema debido a los ambientes de desarrollo que se utilizan hoy en día en los proyectos de software, como los que brindan de ejemplo en (arti endava), en el cual, describe ambientes como desarrollo, compilación, pruebas y producción. Y también menciona la importancia que estos ambientes sean, en todo lo posible iguales. Sin embargo, esto no sucede habitualmente por costos y por el mismo trabajo que se adelanta por parte de los equipos de desarrollo. Ya que, muchas veces se debe cambiar las características de los ambientes de desarrollo, compilación y pruebas, por nuevas características que están en proceso de construcción.

Basado en lo anterior, se puede concluir que uno de los problemas de Devops reside en la generación de los scripts de los pipelines que se utilizan para llevar las nuevas funcionalidades desde desarrollo a producción y posteriormente reutilizarlos en otros ambientes o proyectos. Esto porque, al ser tan variadas las herramientas, es complejo tener un pipeline definido que responda a todas ellas y cuando se necesita cambiar o copiar a otro proyecto, debe ser este igual para que funcione, Además, respecto a la variedad de ambientes de desarrollo, esto también dificulta la generación de pipelines, ya que al no ser iguales, el pipeline para cada uno de los ambientes debe variar y no se pueden reutilizar. Teniendo de esta manera que utilizar en los dos casos comúnmente prácticas de reúso oportunista, las cuales, al crecer la cantidad de pipelines clonados, aumentan los problemas técnicos de administración, mantenimiento y evolución, como lo describe (portugues)

1.2. Formulación del problema

Teniendo en cuenta la necesidad de mejora en la reutilización de código en los scripts de los pipelines, respecto a las herramientas y diferentes ambientes de desa-

rrollo, para la implementación de la metodología ~~devops~~, se plantean los siguientes interrogantes, cuyas respuestas se solucionen con el presente trabajo de grado.

¿Qué técnicas se han usado para mejorar el reúso de los scripts de los pipeline en Devops?

~~¿Cómo~~ podrían apoyar las líneas de productos de software el reúso de los scripts de los pipelines?

~~¿Cómo se podría evaluar la solución planteada en el presente proyecto?.~~

2. Objetivos del proyecto

2.1. Objetivo General

Crear automáticamente scripts de pipelines para Devops variando las herramientas a usar y el ambiente de desarrollo utilizando líneas de productos software.

2.2. Objetivos específicos

- Diseñar un software que genere automática scripts de Pipelines para Devops, varian las herramientas a usar el ambiente de desarrollo, utilizando líneas de productos de software.
- Implementar la solución diseñada en el objetivo anterior.
- ~~Evaluar la efectividad de la solución planteada en varios casos de prueba y probando su correcto funcionamiento.~~

3. Alcance

Este trabajo de grado tiene como objetivo encontrar una nueva alternativa para la creación de los scripts de los pipelines que tiene cada ambiente de desarrollo en un proyecto y las diferentes herramientas que se pueden utilizar, ~~apoyando de esta manera con el despliegue continuo del mismo~~. Para ello, se plantea el uso de las técnicas expuestas por las líneas de productos de software para el reúso de estos scripts, en tres ambientes de desarrollo y con máximo 3 herramientas en cada paso del pipeline.

La implementación se ~~validará~~ usando varios casos de prueba de una empresa real. Para así, ~~verificar~~ que tan factible es usar la herramienta en un ambiente de desarrollo real.

Debes tener en cuenta que validación y verificación no son sinónimos en el contexto de la ingeniería de software.
La validación se usa para recoger evidencia de algo, la verificación se usa para demnostrar formalmente alguna propiedad.
En particular, la validación sirve para demostrar que se cumplieron todos los requerimientos

Todavía se ve como una enumeración de cosas, le falta articulación para que parezca una unidad

4. Marco teórico de referencia y antecedentes

~~Para el proceso de recopilación de información se utilizó el motor de búsqueda de Google Scholar, mediante el cual, se obtuvo los siguientes documentos.~~

4.1. Bases Teóricas

~~En esta sección se describen los fundamentos teóricos que sustentan el trabajo de investigación o proyecto de grado, con sus respectivas citas bibliográficas (Es muy importante el manejo riguroso de las citas).~~

DevOps

~~Plantea~~ la unión de personas, procesos y tecnología para ofrecer valor a los clientes de forma constante, permitiendo que los roles que antes estaban aislados (desarrollo, operaciones de TI, ingeniería de la calidad y seguridad) se coordinen y colaboren para producir productos mejores y más confiables. Al adoptar una cultura DevOps, junto con prácticas y herramientas DevOps, los equipos adquieren la capacidad de responder mejor a las necesidades de los clientes, aumentar la confianza en las aplicaciones que crean y alcanzar los objetivos empresariales en menos tiempo. El propósito principal es mantener el proceso de desarrollo de software organizado y enfocado. Esto incluye integración continua, entrega/implementación continua (CI/CD), retroalimentación y operaciones continuas. En lugar de pruebas únicas o implementaciones programadas, cada función se realiza de forma continua.()

DevOps Pipeline

Es un conjunto de herramientas y procesos automatizados que permite a los equipos de desarrollo y a los equipos de operación de TI, trabajar conjuntamente para construir y desplegar código a producción. ~~Aunque los pipeline por lo general~~ son diferentes entre organizaciones y/o proyectos, estos por lo general incluyen automatización de la compilación, integración continua, entrega y despliegue continuo,

automatización de pruebas, validaciones y reportes. En algunos casos y donde se requiera, se puede permitir la intervención humana. ()

Integración Continua/Despliegue Continuo

el proceso de integración y despliegue continuo incorpora la automatización y la supervisión permanentes en todo el ciclo de vida de las aplicaciones, desde las etapas de integración y prueba, hasta las de distribución e implementación. ()

4.2. Estado del Arte

En la revisión de documentación realizada, se encontraron algunos trabajos previos relacionados con el tema de crear pipelines para Devops.

Creating Azure DevOps Pipelines for Web Application

~~Esta es una tésis realizada por~~ Eetu Koskelainen en la Universidad de Tampere de Ciencias Aplicadas, en la cual, el autor provee instrucciones para crear pipelines automáticamente para aplicaciones web cliente-servidor basicas usando Github, Microsoft Azure y Pulumi para crear infraestructura como código. Sin embargo, en este proyecto se centra en estas tecnologías mencionadas y deja sin brindar una solución a otras tecnologías para devops como AWS o gestores de repositorios como Gitlab. (Koskelainen, 2021)

Comparing DevOps procedures from the context of a systems engineer

~~En este artículo se presentan~~ dos arquitecturas Devops, una montada con Azure Devops y otra con AWS. Las cuales, se comparan usando los tiempos de construcción, de procesamientos y de fallas. Lo anterior, utilizando unos proyectos ya funcionando con estas tecnologías en la empresa Valamis. Sin embargo, este proyecto se enfoca más en la comparación entre las arquitecturas, que en la generación de diversos pipelines dependiendo la tecnología utilizada.(Priyadarsini et al., 2020)

Fast Delivery, Continuously Build, Testing and Deployment with DevOps Pipeline Techniques on Cloud

~~Este artículo presenta~~ la implementación de pipeline para ~~Devops~~ en una organización, sobre la nube de Azure. Lo anterior, basado en los requerimientos del proyecto. Como resultado describen la solución a grandes desafíos culturales y de infraestructura que se debieron enfrentar, teniendo en cuenta, que en este último la solución se centró en la infraestructura como código que provee la nube.(Jumani et al., 2020)

Implementation of a DevOps Pipeline for Serverless Applications

~~Este artículo presenta~~ la implementación de un pipeline automatizado de Devops, para integración continua, entrega continua y prácticas de monitoreo. Todo lo anterior mediante un enfoque serverless, lo cual generó que 18 de las 27 prácticas implementadas fueran influenciadas por las características específicas de Serverless del proyecto. (Ivanov and Smolander, 2018)

Building Lean Continuous Integration and Delivery Pipelines by Applying DevOps Principles: A Case Study at Varidesk

~~En este artículo se presenta~~ la implementación de devops en una de las aplicaciones web de la empresa Varidesk. En este caso, mencionan dos desafíos a los que se enfrentaron, siendo el primero los largos tiempos de espera para que las compilaciones/lanzamientos se pongan en cola y se completen y el segundo, la falta de soporte en algunas herramientas, desde una perspectiva de nubes. Resolviendo estos con prácticas de contenedorización, infraestructura como código y orquestación. (Debroy et al., 2018)

Esta sección se cierra presentando la comparación de los trabajos que presentan soluciones al problema que planteaste en la primera parte del documento

5. Referencias Bibliográficas

Referencias

Bonda, D. T., Ailuri, V. R., and Ghazi, A. N. (2021). Tools integration challenges faced during devops implementation.

Debroy, V., Miller, S., and Brimble, L. (2018). Building lean continuous integration and delivery pipelines by applying devops principles: A case study at varidesk. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.

Ivanov, V. and Smolander, K. (2018). Implementation of a devops pipeline for serverless applications. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11271 LNCS:48–64.

Jabbari, R., Ali, N. B., Petersen, K., and Tanveer, B. (2016). What is devops? a systematic mapping study on definitions and practices.

Jumani, A. K., Shaikh, A. A., Khan, M. O., and Siddique, W. A. (2020). Fast delivery, continuously build, testing and deployment with devops pipeline techniques on cloud article in. *Indian Journal of Science and Technology*, 13:552–575.

Koskelainen, E. (2021). Creating azure devops pipelines for web application - theseus.

Leite, L., Rocha, C., Kon, F., Milojicic, D., and Meirelles, P. (2019). A survey of devops concepts and challenges. *ACM Computing Surveys*, 52:35.

Priyadarsini, K., Fantin, E., Raj, I., Begum, A. Y., and Shanmugasundaram, V. (2020). Comparing devops procedures from the context of a systems engineer. *Materials Today: Proceedings*.

ReportLinker (2022). Software products global market report 2022.

Wilburn, K. M., Edward, S., and Wilburn, H. R. (2018). The impact of technology on business and society. *Global Journal of Business Research*, 12:23–39.