

UNIVERSIDAD PRIVADA DE TACNA
FACULTAD DE INGENIERÍA DE SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA SISTEMAS



PROYECTO TRABAJO FINAL
UNIDAD III

CURSO
CALIDAD Y PRUEBAS DE SOFTWARE Sec. A
DOCENTE

Mag. Patrick Cuadros Quiroga

INTEGRANTES

Tarqui Montalico, Risther Jaime - 2017057857

Limache Victorio, Víctor Piero - 2017057857

Catari Cabrera Yober Nain - 2017059289

Liendo Velasquez, Joaquín - 2017057857

Callata Flores, Rafael - 2017057857

Vargas Cusacani, Posi - 2009034478

TACNA - PERÚ

2020

Resumen

Los productos de software hacen parte de la actividad diaria de la mayoría de las compañías, debido a que la tecnología está presente en cada uno de sus procesos; es así como las empresas desarrolladoras de software tienen un gran reto en crear productos con confiabilidad y calidad. Aquí es donde entran las herramientas de gestión de pruebas estas nos ayudan a gestionar la información relativa a los casos de prueba, para planificar actividades de testing, para gestionar los informes resultantes después de pasar dichos test, etc. Esto es fundamental para cualquier proyecto.

Abstract

Software products are part of the daily activity of most companies, because technology is present in each of their processes; This is how software development companies have a great challenge in creating products with reliability and quality. This is where the test management tools come in. They help us to manage the information related to the test cases, to plan testing activities, to manage the resulting reports after passing these tests, etc. This is essential for any project.

PROYECTO TRABAJO FINAL

UNIDAD III

1. INTRODUCCION

Las empresas desarrolladoras de software que quieran construir sus productos con calidad, requieren un área dedicada a la realización de soporte y pruebas, que verifica la funcionalidad y datos de entrada como de respuesta de las transacciones realizadas a las bases de datos desde la aplicación que se está probando, así se hace necesario contar con una trazabilidad de las consultas realizadas a las bases de datos por el ingeniero de soporte y pruebas asignado.

De esta necesidad nace la idea de Sistema de Gestión para la ejecución de consultas en pruebas de productos de software, que permite realizar estas mismas actividades mejorando el rendimiento de la maquina en la que se realizan las pruebas y una auditoria o seguimiento posterior a las sentencias que fueron ejecutadas por el ingeniero de soporte y pruebas para la verificación de datos. Con este nuevo modelo se busca obtener ventajas operacionales como: mejor determinación y utilización de la capacidad disponible de los recursos físicos y humanos con los que cuenta la compañía, generando una asignación de recursos que conlleven a un equilibrio de las cargas de trabajo.

2. TITULO

Herramientas de Gestión de Pruebas.

3. AUTORES

- Tarqui Montalico, Risther Jaime
- Limache Victorio, Víctor Piero
- Catari Cabrera Yober Nain
- Liendo Velasquez, Joaquín
- Callata Flores, Rafael
- Vargas Cusacani, Posi

4. DESARROLLO

- **¿QUÉ SON LAS HERRAMIENTAS DE GESTIÓN DE PRUEBAS?**
Es la herramienta que proporciona soporte a la gestión de pruebas y control de parte del proceso de pruebas. A menudo tiene varias capacidades, tales como gestionar los productos de soporte de pruebas, planificación de pruebas, registro de resultados, seguimiento del proceso, gestión de incidencias y generación de informes de las pruebas.
- **Función de las herramientas de gestión de pruebas**

Las tareas de automatización de pruebas están cada vez más presentes en el mercado para automatización de pruebas de software. Existen una serie de herramientas de automatización, pero es poco probable que una sola herramienta pueda automatizar todas las tareas de prueba. La mayoría de las herramientas se centran en una tarea o grupo de tareas específicas, aunque algunas solo tratan un aspecto de la tarea.

Las herramientas de prueba se pueden categorizar según las funciones que realicen. Algunas designaciones de funciones típicas para herramientas son:

- Herramientas de adquisición de datos que adquieren datos para utilizar en las tareas de prueba. Los datos se pueden adquirir mediante la conversión, la extracción, la transformación o la captura de datos existentes, o mediante la generación de guiones de uso o especificaciones suplementarias.
 - Herramientas estáticas de medida que analizan información contenida en los modelos de diseño, el código fuente u otros orígenes fijos. El análisis produce información en el flujo lógico, el flujo de datos o la métrica de calidad, como la complejidad, el mantenimiento o las líneas de código.
 - Herramientas dinámicas de medida que realizan un análisis durante la ejecución del código. Las medidas incluyen la operación de tiempo de ejecución del código, como la memoria, la detección de errores y el rendimiento.
 - Simuladores o controladores que realizan tareas que, por cuestiones de tiempo, gastos o seguridad no están disponibles para las pruebas.
 - Herramientas de gestión de pruebas que ayudan en la planificación, el diseño, la implementación, la ejecución, la evaluación y la gestión de tareas de prueba o productos de trabajo.
- **Herramientas para realizar pruebas de software** Actualmente el número de herramientas para pruebas de software disponibles, tanto en el mercado como de manera gratuita (herramientas de código abierto), es muy amplio. En el siguiente listado os dejamos algunas de las más importantes.
- a) Herramientas Open Source
- Herramientas de gestión de pruebas
 - Bugzilla Testopia
 - FitNesse
 - qaManager
 - qaBook
 - RTH(open source)
 - Salome-tmf
 - Squash TM
 - Test Environment Toolkit

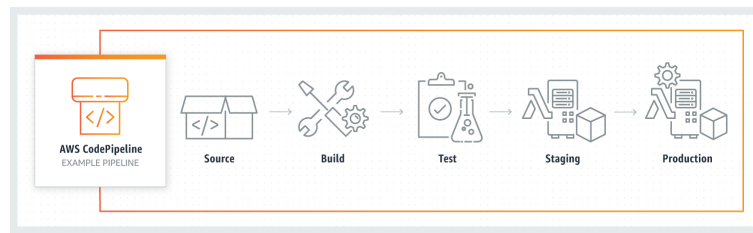
- TestLink
- Testitool
- XQual Studio
- Radi-testdir
- Data Generator
- Herramientas para pruebas funcionales
 - Selenium
 - Soapui
 - Watir (Pruebas de aplicaciones web en Ruby)
 - WatiN (Pruebas de aplicaciones web en Net)
 - Capedit
 - Canoo WebTest
 - Solex
 - mprimatur
 - SAMIE
 - ITP
 - WET
 - Webinject
- Herramientas para pruebas de carga y rendimiento
 - FunkLoad
 - FWPTT load testing
 - loadUI
 - jmeter

b) Herramientas comerciales

- Herramientas de gestión de pruebas
 - HP Quality Center/ALM
 - QA Complete
 - qaBook
 - T-Plan Professional
 - SMARTS5
 - QAS.Test Case Studio
 - PractiTest
 - SpiraTest
 - TestLog
 - ApTest Manager
 - Zephyr
- Herramientas para pruebas funcionales
 - QuickTest Pro
 - Rational Robot
 - Sahi
 - SoapTest

- Test Complete
- QA Wizard
- Squish
- VTest
- Internet Macros
- Herramientas para pruebas de carga y rendimiento
 - HP LoadRunner
 - LoadStorm
 - NeoLoad
 - WebLOAD Professional
 - Forecast
 - ANTS Advanced .NET Testing System
 - Webserver Stress Tool
 - Load Impact
- **¿QUÉ ES AWS CODEPIPELINE?**

Es un servicio de Amazon Web Services que nos ayuda a automatizar los procesos de lanzamiento para obtener actualizaciones rápidas y confiables. AWS Codepipeline compila, prueba e implementa tu código cada vez que hay un cambio, en función a los modelos de proceso de los releases que hagas.



- **¿QUÉ ES AZURE DEVOPS?** Es una plataforma de SaaS (software como servicio) de Microsoft que nos proporciona una cadena de herramientas DevOps de punto a punto para desarrollar e implementar software. También proporciona alojamiento Git privado ilimitado, compilación en la nube para la integración continua, planificación ágil y administración de versiones para la entrega continua a la nube y en las instalaciones. Incluye amplio soporte IDE.



■ COMPARATIVA ENTRE AMBOS: VENTAJAS

AWS CodePipeline	Azure DevOps
Fácil de instalar	Completo, flexible e intuitivo
Servicio administrado	Gran ecosistema extendido
Integración con GitHub	Integración con Azure
Ejecución en paralelo	Integración con GitHub
Despliegue automático	Soporte Open Source y gratis para Stakeholders

■ COMPARATIVA ENTRE AMBOS: DESVENTAJAS

AWS CodePipeline	Azure DevOps
No hay pizarras de proyecto	Sigue siendo dependiente de C#
No tiene integración con las herramientas de 365 de "Power"	Pobre integración de Jenkins
Más básico que Azure DevOps	No tiene todos los requerimientos de herramientas de administración
No puede configurar repositorios de código localmente	Capacidad de equipos multifuncionales no es visible

■ COMPAÑÍAS QUE USAN ESTAS HERRAMIENTAS:

AWS CodePipeline	Azure DevOps
CRED	Via Varejo
Swingvy	Microsoft
Affirm	QRPoint
HENNGE	Queue-it

■ HERRAMIENTAS QUE PUEDEN SER INTEGRADAS:

AWS CodePipeline	Azure DevOps
GitHub	GitHub
Jenkins	Git
Amazon EC2	Docker
Amazon S3	Slack

■ ALGUNAS BUENAS ALTERNATIVAS A AMBOS:

- a) AWS CodeDeploy
- b) Jenkins
- c) AWS CodeBuild

- d) TeamCity
- e) Bamboo

5. CONCLUSIONES

a) CodePipeLine

- CodePipeline es una herramienta de entrega continua e integración continua muy flexible.
- Facilita un poco la implementación en el entorno de AWS. Hay una gran disponibilidad de datos no estructurados.
- Se usa bastante para gestionar la CI/CD (Integración continua e integración delivery)

b) Azure DevOps

- Es muy fácil de configurar y usar si tiene alguna experiencia con procesos ágiles.
- Las barreras de entrada iniciales son extremadamente bajas, ya que los primeros 5 usuarios pueden aprovechar la herramienta de forma gratuita. Encontré la característica / funcionalidad general más fácil de usar y más accesible que herramientas similares.
- Si ya es un usuario de git, esto se integra directamente con los repositorios de git, lo que facilita la transición.
- La herramienta también está integrada con muchos otros productos de Microsoft, por lo que si tiene una tienda centrada en Microsoft, puede aprovechar el ecosistema más amplio.

6. RECOMENDACIONES Aquí una serie de aspectos a tener en cuenta al momento de elegir una herramienta para la gestión de prueba.

- La categoría de defectos
- El lenguaje de programación y entorno de desarrollo
- El proceso de configuración y gestión de datos de prueba
- El control de versiones y CI (Integración Continua)
- Los reportes
- Las plataformas compatibles y etiquetado

7. BIBLIOGRAFIA

- <https://stackshare.io/stackups/aws-codepipeline-vs-azure-devops#:~:text=AWS%20CodePi>
- <https://www.trustradius.com/products/aws-codepipeline/reviews>
- <https://www.trustradius.com/compare-products/aws-codepipeline-vs-azure-devops>
- <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html>
- Fischer G.,” The Software Technology of the 21st Century: From Software Reuse to Collaborative Software Design”
- Pérez B.,” Gestión de las Pruebas Funcionales”