Universidad Tecnológica Nacional

CUATRO PRINCIPIOS PARA VERSIONES DE SOFTWARE DE BAJO RIESGO

Calcaterra, Brian - González, Georgina - Senn, Juan - D'Alessandro, Matias - Zerpa, Roy - Soruco, Ezequiel

INTRODUCCION

Si usted no es un adicto a la adrenalina y no quiere malgastar el dinero de su empresa en lanzamientos de versiones de software problemáticas debería tener en cuenta los cuatros principios que propone Jez Humble

RESULTADOS

1

Principio 1: Las liberaciones de bajo riesgo son incrementales

Una organización madura tendrá sistemas compuestos por varios componentes o servicios interconectados, con dependencias entre esos componentes .Para ello, implemente componentes de forma independiente, en una configuración en paralelo. Si la nueva versión de su aplicación requiere una nueva versión de algún servicio, debe tener la nueva versión de ese servicio en funcionamiento, y probada antes de implementar la nueva versión de la aplicación que depende de ella.

2

Principio 2: desacoplar el despliegue y la liberación

Puede y debe instalar su versión de software en su entorno producción antes de ponerlo a disposición de los usuarios, de modo que pueda realizar "pruebas de humo" y cualquier otra tarea, de tal manera de que se pueda asegurar de que la implementación fue exitosa y, en particular, probar que los ajustes de configuración para el entorno de producción son correctos.

3

Principio 3: Reducir el tamaño de las entregas

Cuando reducimos el tamaño del lote, podemos realizar entregas con mayor frecuencia, ya que reducir el tamaño del lote reduce el tiempo del ciclo.
De esta manera implementamos en produccion mas frecuentemente y si hay un incidente, es mas facil de resolver ya que es menor la cantidad a corregir.

4

Principio 4: Optimizar para resiliencia

La capacidad de restaurar el sistema a un estado de referencia en un tiempo predecible es vital no solo cuando la implementación falla, sino también como parte de su estrategia de recuperación ante desastres.
El elemento más importante en la creación de sistemas resilente es el humano, se debe a que cuando se cae un servicio, es imperativo que todos sepan qué procedimientos seguir para diagnosticar el problema y hacer que el sistema vuelva a funcionar

METODOLOGÍA

LECTURA

ANÁLISIS

COMPRENSIÓN

Palabras Clave:

Resiliencia
Frecuente
Independiente
Costo

CONCLUSION

Con estas técnicas, puede reducir drásticamente el riesgo de liberar versiones de software a los usuarios. Debe tener en cuenta que vienen con un costo de desarrollo adicional y requieren una planificación inicial, por lo que se paga una cierta cantidad por adelantado para lograr esta reducción de riesgo. A menudo, este tipo de costos son difíciles de justificar,en parte porque las personas tienen una tendencia a darle poco valor a una recompensa que existe de alguna manera en el futuro.

Bibliografia

http://www.informit.com/articles/article.aspx?p=1833567
https://mysoftwar.wordpress.com/2011/04/12/diferentes-metodologias-de-implementacion-de-erp/