Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 22
Pruebas de sistemas

IIC3745 – Testing

Rodrigo Saffie

rasaffie@uc.cl

23 de noviembre de 2020

A/B testing

- Ampliamente utilizado en sitios *web*.
- Se prueban dos versiones distintas de una página y se evalúa cual resulta mejor:

- Split

Alpha/Beta testing

- Pruebas de la aplicación simulando ambiente real de utilización.
- *Alpha testing*:
  - Usuarios seleccionados o desarrolladores prueban el producto en el sitio de desarrollo bajo la dirección de un encargado de las pruebas.
- *Beta testing*:
  - Número de usuarios reales controlado prueba libremente el producto en su propio ambiente de uso.

Ambiente de pruebas

- *Review apps*
- *Staging*
  - Ambiente de ejecución del sistema que trata de emular lo más posible al ambiente real (*production*)
  - La idea es probar el sistema de manera segura sin afectar los datos reales
  - Se necesita tener claras las dependencias del sistema (ej. DB / servicios externos) y aislarlas del ambiente de pruebas
    - Por ejemplo, en el ambiente de pruebas no se deberían enviar correos => ¿cómo probar correos sin enviarlos?
  - Útil para ejecutar pruebas de humo sobre *release candidates*

Pruebas de sistemas

- Pruebas de la aplicación en su ambiente de ejecución
- Validan los requisitos no funcionales, como por ejemplo:
  - Rendimiento
  - Seguridad
  - Escalabilidad
  - Resiliencia
  - Portabilidad
- Sirven para asegurar un *Service Level Agreement*
  - Auth0

Métricas

- Una métrica son datos procesados que expresan numéricamente el rendimiento sobre un criterio
  - Ejemplo: *Coverage*
- Sirven para:
  - tener respaldo cuantitativo sobre un criterio
  - comparar la efectividad de distintas estrategias

Etapas de una métrica

- **Formulación:** formalización de factores apropiados para representar el *software*
- **Recolección:** mecanismos para acumular datos a partir de la formulación
- **Análisis:** procesamiento de los valores recolectados para obtener información
- **Interpretación:** evaluación de la información para determinar mejoras
- **Retroalimentación:** recomendaciones derivadas de la interpretación

Métricas en *software*

- Ejemplos de métricas:
  - Tiempos de respuesta
  - Flujo (*throughput*): solicitudes por minuto (o segundo)
  - Uso de RAM / CPU
- New Relic
- Scout

Pruebas de rendimiento

- Se busca probar que el sistema cumpla con los requerimientos de desempeño
  - Por ejemplo, tiempos de respuesta, uso de RAM/CPU/*bandwidth*
- Ejemplos:
  - **Pruebas de carga**: asegurar el comportamiento del sistema bajo ciertas condiciones de uso
  - **Pruebas de estrés**: probar cómo responde el sistema dada una carga mayor para la cual fue diseñado
- Herramientas:
  - Loader
  - BlazeMeter
  - JMeter

Pruebas de escalabilidad

- Pruebas las políticas de escalabilidad para los sistemas bajo situaciones controladas
- Tipos de escalabilidad
  - **Vertical**: reemplazar los componentes por otros con mayor capacidad
  - **Horizontal**: aumentar la cantidad de componentes para que ejecuten el mismo proceso en paralelo

Pruebas de seguridad

- Ataques simulados para detectar fortalezas y debilidades de los sistemas (*white hat hackers*)
- *Bug bounty programs*: recompensas por reportar vulnerabilidades
  - Bugcrowd
  - GitHub
  - Google
- Se deben reportar con extremo cuidado
  - Si se reportan como *bug* funcional se corre el riesgo que otras personas se aprovechen de la vulnerabilidad

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 22
Pruebas de sistemas

IIC3745 – Testing

Rodrigo Saffie

rasaffie@uc.cl

23 de noviembre de 2020