

# ISW

martes 07-10-25

## Testing en ambientes ágiles

### Waterfall

Implementaciones híbridas son riesgosas.

NO es lo mismo ser ágil que hacer ágil

Lo que marca la diferencia del método tradicional y el ágil, es la mentalidad.

- Hacer sw es lo mismo, pero el de qué manera es lo distinto

El nivel de granularidad en ágil :

parece casi que estamos haciendo todo al mismo tiempo

Testing ágil descansa mucho en que testing se tiene que automatizar.

## Manifiesto de testing

Cuestiones culturales, actitudinales

- Pruebas durante **sobre** pruebas al final
- Prevenir defectos **sobre** encontrar defectos
- Entender lo que se está probando **sobre** verificar la funcionalidad
- Construir un mejor sistema **sobre** romperlo
- Responsabilidad de todo el equipo **sobre** el teste es responsable de la calidad

## El ciclo de vida del testing ágil vs tradicional

### Criterio de aceptación vs prueba de aceptación

- de un criterio pueden salir muchas pruebas de aceptación
- es una relación 1-n

### Enfoque tradicional (cascada) vs ágil

- las cosas que se hacen son las misma

## Principios para testing en proyectos ágiles

- Testing se mueve hacia adelante en el proyecto
- Testing no es una fase
- Todos hacen testing
- Reducir la latencia del feedback
- Las pruebas representan expectativas
- Mantener el código limpio, corregir los defectos rápido
- Reducir la sobrecarga de documentación de las pruebas
- Las pruebas son parte del "done":
- De probar al final a Conducido por Pruebas: *especificamos en las pruebas los detalles que sean necesarios, (los tomamos a los detalles de los requerimientos. (tdd)*

*scrum habla mucho de la persona en termino del proyecto. Cuando hablamos de testing, hablamos de producto*

## Practicas concretas (6):materializan el manifiesto ágil

- (1)Pruebas de unidad e integración automatizadas
- (2)Pruebas de Regresión a nivel de sistema automatizadas
- (3)Pruebas exploratorias
- (4)TDD
- (5)**ATDD**
- (6)Control de versión de las pruebas con el código

Gestión de configuración de sw es la base para que la ingeniería producto pueda funcionar

3 practicas continuas

- ci: integración continua
  - automatizamos el *nivel de integración* (1)
    - Antes de este, es el unitario, que la automatización la hace TDD (4)
- cd: delivery (entrega) continuo
  - automatizamos el *nivel de sistema* (1)
- cd: deployment (despliegue) continua
  - automatizamos el *nivel de aceptación* (1)

Calidad es cuestión de cultura y de hábito

- *la prueba de regresión volvía a probar todo lo que no paso y todo lo que si habia pasado. Las pruebas sin regresión, solo las pruebas que habían pasado.*
  - La pregunta de si hacer testing de regresión o no, es si se hace manual (por el tiempo que involucra). Si se hace automático, debería ser con regresión si o si.
- Hay una parte del testing que siempre tiene que ser manual.
- El nivel de formación de los que hacen testing automatizado es distinto que los que lo hacen manual. (testar automático (QA automatizó), es un profesional que debe saber programar. Manejar mínimamente BD, SQL. ) (un tester manual puede ser cualquiera)
- Automatización tiene un costo de conversión, a partir del cual converge para que convenga sobre el manual o no. Algunos dicen que 4, 5, 7, depende del producto de sw que se vaya a testear.

#### (5)ATDD

1. Los escenarios de ejemplos sirven para poder armar las pruebas. El PO es fundamental para esta parte. Da situaciones, algo concreto.
2. Armamos la historia de usuario

=====ciclo de testing=====

3. Armamos las pruebas de aceptación en base a los ejemplos , y las historias de usuario.
4. Exploracion de ejemplos, con el PO. El de tester estaría bueno que esté presente en estas reuniones, suele tener ideas que a los desarrolladores no se le ocurren

=====

5. Mas pruebas
6. Automatización de las pruebas
7. Programarlas y ejecutarlas
8. Pruebas exploratorias (también las puedo hacer cuando no conozco tanto el producto, al principio.) (o cuando aparecen requerimientos que no se entienden bien) (son manuales, no hay forma de automatizar si no sabes que hace)
9. -..

#### El rol del tester en los equipos ágiles

- Tester tiene que ser parte del equipo, se debería integrar lo más rápido posible

#### Cuadrantes del Testing ágil.

Pruebas orientadas al negocio (ux, de usabilidad, de aceptación de la historia)

Pruebas orientadas al tecnología (reqs no funcionales)  
Pruebas que guía el desarrollo  
Pruebas que critican el producto (a validar si es el producto correcto o no desde la perspectiva del usuario final)

### **Pirámide del Testing ágil.**

Empieza con una base gruesa de pruebas unitarias, luego las pruebas de negocio, y en la punta de la pirámide, las de flujo de trabajo (las que tienen que ser manuales si o si)

Pruebas End to End suelen ser manuales, que van desde la interfaz de usuario hasta.. (no llegué a escuchar, nt)

### **Niveles de prueba:**

- son siempre los mismos, no importa si el testing es tradicional o ágil.
- Prueba de aceptación vs la de sistema
  - lo unico que se diferencia es quien lo hace y cuando, porque al fin y al cabo lse testa lo mismo
  - la prueba de aceptación la hace PO, puede invitar otra gente. usualmente tiene más de un tipo de usuario.

### **Beneficios de la automatización del testing**

cobertura: cuando podemos probar sobre el 100% de los pruebas que hay que hacer