

Testing en Ambientes Agiles:

Dif entre los Tradi vs Agil: Como encaramos los casos

- Lo que debemos implementar tiene que estar claro bien tener
Se incluye a la gente de testing en la implementación
- Tod. el equipo debe funcionar como Agil sino se producen choques entre
Agentes integrados

Testing Agil: Cuestion de enfoque o cultural, sumado a una tendencia al automatismo

- El testing agil descansa en la idea de que el testing puede automatizarse

Manifiesto del Testing

Valoramos

- Probar durante sobre pruebas final
- Prevenir defectos sobre encontrar defectos
- Entender el producto y que se está probando sobre verificar una funcionalidad
La gente que hace testing se involucra y entiende el negocio
- Construir un mejor sistema sobre romper el sistema (combinaciones
y trabajo en equipo)
- Responsabilidad de todo el equipo por ~~la~~ calidad por sobre responsabilidad
de los testing (integrar el testing al equipo y no aislarlo)

El agilismo forma parte del backlog (Diseño, Análisis, Definir, Probar y luego despliegue)

* Un caso de prueba puede abarcar muchos criterios de aceptación

9 Principios Para testing en Agil

- Testing se mueve hacia adelante
- 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 (Automatizar en el testing, se sona de detalle en los VS)
- Las pruebas son parte del done
- De probar al final a Conducto las pruebas

Practicar Concretar / Materializar lo que dice el manifest!

- Pruebas de unidad e integración automatizadas
- Pruebas de regresión a nivel de sistema automatizadas
- Pruebas exploratorias
- TDD (test driven development) Automatiza el nivel unitario
- ATDD (Acceptance test driven development) Automatiza las pruebas de aceptación
- Control de versión de las pruebas con el código

Ver Practicas continuas (relacionadas con la automatización de los niveles de testing)

- CI: Integración Continua / Automatiza el nivel de integración
- CD: Entrega / delivery continuo / Automatiza nivel de sistema
- CD: Despliegue / Deployment Continuo / Automatiza nivel de aceptación

Hacer más testing no agrega calidad sino que evidencia la falta de calidad. Por eso es importante alinear primero Principios de calidad y de diseño.

ATDD (Desarrolla, conduce las pruebas de aceptación)

- Prueba de regresión: tratar a todos los ciclos como si fuera el ciclo con. Es decir que en cada ciclo ejecutara todas las pruebas aunque estas hayan pasado.

- Avance más automatiza el testing ya que la configuración del sistema tiene un costo y una dificultad ya que es manual (requiere un nivel de formación de testers más específico y la disponibilidad de ese perfil es distinta por la curva de aprendizaje)

• Pruebas de aceptación de alto nivel se corren con ejemplos elaborados. Es importante la participación del Product owner para poder desarrollar ejemplos específicos que permitan cubrir todas las fallas. La exploración de ejemplos se le tiene que dar mucha importancia. La corrección de defectos debe ser rápida.

Las pruebas exploratorias se utilizan para definir el universo de pruebas, conocer el producto, donde hacer énfasis en que probar.

El rol de los testers en el equipo ágil

- Es importante el vínculo entre el tester y el Product owner, por eso se insisten en integrar al tester al equipo.

es en detalle
cuando

Cuadrante de testing ágil (Pruebas orientadas a la tecnología o al negocio)

- Negocio: US, requerimientos funcionales, pruebas de aceptación, exploratorias
- Tecnología: No funcionales,

Pyramide de testing en agile

- Base fuerte hecha por Pruebas unitarias por el Programador
- Pruebas funcionales, de aceptación
- Pruebas no funcionales
- Pruebas manuales

Tradicional
(Usar bugs)

Agile
(Prevenir bugs)