

## Testing en Ambientes Ágiles:

Dif entre los Tradi vs Ágil: (con encarar las cosas)

- Lo que debemos implementar tiene que estar claro bien temprano
- Se incluye a la gente de testing en la implementación
- Tod. el equipo debe funcionar como Ágil sin se producen choques entre Agilistas e integrantes

Testing Ágil (Cuestión de entrega o cultural, sumando a una坦akia que autoriza)

- El testing ágil descansa en la idea de que el testing puede autorizarse

## Manifiesto del Testing

### Valores

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

El agilismo forma parte del backlog (Visión, Análisis, Definir, Probar, Llegar, Despliegues)

\* Un caso de prueba. Puede abarcar muchos criterios de aceptación

## 9 Principios para testing en Agile

- Testing se move hacia adelante
- Testing no es una fase
- Todos hacen testing
- Reducir la latencia del feedback
- Las pruebas representan efectividad
- Mantener el código limpio, corregir los defectos rápidamente
- Reducir la sobrecarga de documentación de las Pruebas (Automatizar en el testing)
- Las Pruebas son parte del done se especifica en los VS
- De Probar al final a Conducido por Pruebas

## Practicar (concretar / Materializar lo que dice el principio.)

- 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 (desarrollo inducido por Pruebas de aceptación)
- Control de versión de las Pruebas con el código

## Ver Practicar continua (relacionado 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 valor sino que evidencia la falta de calidad.  
Por eso es importante aplicar los 9 principios de calidad y de diseño.

- ATDD (Desarrollo guiado por pruebas de aceptación)

- Prueba de regresión: tratar a todos los ciclos como si fuera el ciclo cap. Es decir que en cada ciclo ejecutara todos los Pruebas que se han pasado.

- A veces no se automatiza el testing ya que la configuración del automata tiene un costo y una dificultad ya que es manual (requiere un nivel de formación de testers más elevado), 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 filas.

La ejecución 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 prueba, conocer el producto, donde hacer enfasis en que hacer.

El rol de los testers en los equipos ágiles

- Es importante el vínculo entre el tester y el product owner, por eso se insiste en integrar al tester al equipo

Considerar de testing ágil (Pruebas orientadas, la tecnología o el negocio)

- Negocio: US, requerimientos funcionales, Pruebas de aceptación, exploratorias

- Tecnología: No funcionales,

### Pirámide de testing en agile

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

Traditional  
(Buscar bugs)

Agile  
(Prevenir bugs)