



# Curso ISTQB® Foundation LEVEL AGILE TESTER



# Capítulo 3:

## Métodos, técnicas y herramientas de pruebas ágiles

### 3.1 Métodos de testing ágil

### 3.2 La evaluación de riesgos de calidad y Estimación de esfuerzo de prueba

### 3.3 Técnicas en proyectos ágiles

- Criterios de aceptación, cobertura adecuada y otra información para pruebas
- Aplicación del desarrollo guiado por pruebas de aceptación
- Diseño de pruebas de caja negra funcionales y no funcionales
- Pruebas exploratorias ("exploratory testing") y pruebas ágiles

### 3.4 Herramientas en un proyecto ágil





## 3.3 Técnicas en proyectos ágiles

### ***Técnicas de pruebas en proyectos ágiles***

Existen diversas técnicas de pruebas, las más utilizadas para los proyectos ágiles son las siguientes:

- Criterios de aceptación, cobertura y otra información
- Técnica de aplicación del desarrollo guiada por pruebas de aceptación
- Técnica de diseño de prueba de caja negra funcionales y no funcionales
- Pruebas exploratorias y pruebas ágiles

# Capítulo 3:

## Métodos, técnicas y herramientas de pruebas ágiles

### 3.1 Métodos de testing ágil

### 3.2 La evaluación de riesgos de calidad y Estimación de esfuerzo de prueba

### 3.3 Técnicas en proyectos ágiles

- Criterios de aceptación, cobertura adecuada y otra información para pruebas
- Aplicación del desarrollo guiado por pruebas de aceptación
- Diseño de pruebas de caja negra funcionales y no funcionales
- Pruebas exploratorias ("exploratory testing") y pruebas ágiles

### 3.4 Herramientas en un proyecto ágil





## 3.3 Técnicas en proyectos ágiles

### ***1. Técnica basada en Criterios de aceptación, cobertura adecuada y otras informaciones***

- Los proyectos ágiles definen sus requisitos iniciales en HU, las cuales se priorizan al inicio del proyecto.
- Las HU son breves y tienen un formato predefinido.
- Los requisitos NO FUNCIONALES como usabilidad y rendimiento, son importantes y se deben definir en historias de usuario únicas o relacionadas con otras historias funcionales.



## 3.3 Técnicas en proyectos ágiles

### ***Criterios de aceptación, cobertura adecuada y otra información***

#### **Bases de pruebas**

Las historias de usuario son una parte importante de la base de pruebas.

*Otras bases de pruebas que se pueden usar son:*

- Experiencia de proyectos anteriores
- Funciones, funcionalidades y características de calidad del sistema existentes
- Código, arquitectura y diseño
- Perfiles de usuario (contexto, configuraciones del sistema y comportamiento del usuario)
- Información sobre defectos procedentes de proyectos existentes y anteriores
- Una categorización de los defectos en una taxonomía de defectos
- Normativas aplicables (por ejemplo, [DO-178B] para software de aviónica)
- Riesgos de calidad



## 3.3 Técnicas en proyectos ágiles

### ***Criterios de aceptación, cobertura adecuada y otra información***

- Durante cada iteración:
  - Los desarrolladores crean código que implementa las funciones y funcionalidades descritas en las HU
  - Este código se verifica y valida a través de pruebas de aceptación.

**¡Para lograr estos los criterios de aceptación deben ser testeables!**





## 3.3 Técnicas en proyectos ágiles

### ***Criterios de aceptación, cobertura adecuada y otra información***

Los criterios de aceptación para ser testeables deben incluir lo siguiente:

- ***Comportamiento funcional:*** El comportamiento debe ser observable externamente con acciones de usuario que operan bajo ciertas configuraciones, por ejemplo, en forma de entradas de datos
- ***Características de calidad:*** ¿Cómo el sistema ejecuta el comportamiento especificado?. Las características también pueden denominarse atributos de calidad o requisitos no funcionales. Algunas características de calidad habituales son rendimiento, fiabilidad, usabilidad, etc.





## 3.3 Técnicas en proyectos ágiles

### ***Criterios de aceptación, cobertura adecuada y otra información***

- ***Escenarios (casos de uso)***: Una secuencia de acciones entre un actor externo (a menudo un usuario) y el sistema, para conseguir un objetivo específico o una tarea de negocio.
- ***Normas de negocio***: Las actividades que solo pueden realizarse en el sistema bajo ciertas condiciones definidas por procedimientos y restricciones externos (por ejemplo, los procedimientos utilizados por una compañía aseguradora para gestionar partes de seguros).
- ***Interfaces externas***: Las conexiones entre el sistema en fase de desarrollo y su contexto. Las interfaces externas pueden dividirse en varios tipos (interfaz de usuario, interfaz con otros sistemas, etc.).



## 3.3 Técnicas en proyectos ágiles

### ***Criterios de aceptación, cobertura adecuada y otra información***

- **Restricciones:** Cualquier restricción de diseño e implementación que pueda restringir las opciones del desarrollador. Los dispositivos con software empotrado a menudo deben cumplir restricciones físicas tales como el tamaño, el peso y las conexiones de interfaz.
- **Definiciones de datos:** El cliente puede describir el formato, el tipo de datos, los valores permitidos y los valores por defecto para un elemento de datos en la composición de una estructura compleja de datos de negocio. (por ejemplo, entre Estados Unidos y códigos postales)



## 3.3 Técnicas en proyectos ágiles

### ***Criterios de aceptación, cobertura adecuada y otra información***

Además de la información entregada por las historias de usuario y los criterios de aceptación, hay puntos que se deben evaluar al realizar las pruebas:

- ¿Cómo el sistema debe trabajar y ser utilizado?
- ¿Qué interfaces de sistema están disponibles? ¿Se pueden utilizar para la prueba ?
- ¿Existen herramientas de apoyo, disponibles?
- ¿Tiene los conocimientos y habilidades suficientes el tester para ejecutar las pruebas necesarias ?

## 3.3 Técnicas en proyectos ágiles

### *Criterios de aceptación, cobertura adecuada y otra información*

- A menudo encontraremos que necesitamos más información, y debemos solicitarla al equipo.
- La información relevante es importante para determinar si una actividad específica puede considerarse hecha (terminada)

**IMPORTANTE**



## 3.3 Técnicas en proyectos ágiles

### ***Niveles de prueba y definición de “Done”***

***Cada nivel de pruebas tiene su propia definición de "hecho".***

#### ***“Hecho” en las Pruebas unitarias***

- 100% de cobertura de decisión cuando sea posible, con revisiones minuciosas de las rutas inviables.
- Se ha efectuado un análisis estático en todo el código
- No hay ningún defecto importante sin resolver (ordenado por prioridad y severidad)
- No se conoce deuda técnica inaceptable y conocida en el diseño y el código [Jones11]
- Todo el código, todas las pruebas unitarias y todos los resultados de las pruebas unitarias se han revisado
- Todas las pruebas unitarias están automatizadas
- Las características importantes están dentro de los límites acordados (por ejemplo, rendimiento)



## 3.3 Técnicas en proyectos ágiles

### ***Niveles de prueba y definición de “Done”***

***Cada nivel de pruebas tiene su propia definición de "hecho".***

#### ***“Hecho” en las Pruebas de integración***

- Se han probado todos los requisitos funcionales, incluyendo pruebas tanto positivas como negativas, con el número de pruebas en función del tamaño, la complejidad y los riesgos
- Se han probado todas las interfaces entre unidades
- Se han cubierto todos los riesgos de calidad en conformidad con el alcance acordado de las pruebas
- No hay defectos importantes sin resolver (priorizados por riesgo e importancia) Se han reportado todos los defectos encontrados
- Se han automatizado todas las pruebas de regresión, cuando sea posible, y todas las pruebas automatizadas están almacenadas en un repositorio común



## 3.3 Técnicas en proyectos ágiles

### ***Niveles de prueba y definición de “Done”***

#### ***“Hecho” en las Pruebas de sistema***

- Se han realizado pruebas de extremo a extremo de las historias de usuario, funcionalidades y funciones
- Se han cubierto todos los modelos de usuarios
- Se han cubierto las características de calidad del sistema más importantes (por ejemplo, rendimiento, robustez, fiabilidad)
- Se han hecho pruebas en un entorno parecido al de producción, incluyendo todo el hardware y el software para todas las configuraciones soportadas, en la medida de lo posible
- Se han cubierto todos los riesgos de calidad de conformidad con el alcance acordado de las pruebas
- Se han automatizado todas las pruebas de regresión, cuando sea posible, y todas las pruebas automatizadas están almacenadas en un repositorio común
- Se han reportado y posiblemente arreglado todos los defectos encontrados
- No hay defectos importantes sin resolver (priorizados por riesgo e importancia)





## 3.3 Técnicas en proyectos ágiles

### ***Historia de usuario y definición de “Done”***

#### ***¿Cuándo se indica que está terminada una historia de usuario?***

- Las historias de usuario seleccionadas para la iteración están completas, el equipo las ha entendido y cuentan con criterios de aceptación detallados y testeables
- Todos los elementos de la historia de usuario se han especificado y revisado, incluyendo las pruebas de aceptación de la historia de usuario
- El equipo ha identificado y estimado las tareas necesarias para implementar y probar las historias de usuario seleccionadas



## 3.3 Técnicas en proyectos ágiles

### ***Prestación y definición de “Done”***

#### ***¿Cuándo se indica que está terminada una prestación?***

La definición de "hecho" para las prestaciones, pueden abarcar varias historias de usuario o épicas y puede incluir:

- El cliente ha definido y aprobado todas las historias de usuario que integran la funcionalidad, con criterios de aceptación.
- El diseño está completo, sin deuda técnica conocida
- El código está completo, sin deuda técnica conocida ni refactorización inacabada
- Se han realizado pruebas unitarias y se ha logrado el nivel definido de cobertura
- Se han realizado las pruebas de integración y las pruebas de sistema para la funcionalidad según los criterios de cobertura definidos
- No hay defectos importantes pendientes de corregir
- La documentación de la funcionalidad está completa, lo que puede incluir notas de la entrega, manuales de usuario y funciones de ayuda on-line



## 3.3 Técnicas en proyectos ágiles

### ***Iteración y definición de “Done”***

#### ***¿Cuándo se indica que está terminada una Iteración?***

- Todas las funcionalidades de la iteración están listas y han sido probadas individualmente de conformidad con los criterios de nivel de la funcionalidad
- Todos los defectos no críticos que no pueden arreglarse dentro de las restricciones de la iteración se han añadido al backlog del producto y se han priorizado
- Se ha completado y probado la integración de todas las funcionalidades para la iteración
- Se ha escrito, revisado y aprobado la documentación

En este punto el software es potencialmente entregables porque la iteración se ha completado con éxito, pero no todas las iteraciones tienen como resultado una entrega.



## 3.3 Técnicas en proyectos ágiles

### ***Entrega y definición de “Done”***

#### ***¿Cuándo se indica que está terminada una Entrega?***

- **Cobertura:** Se han cubierto mediante pruebas todos los elementos de la base de prueba relevantes para todos los contenidos de la entrega.
- **Calidad:** Se encuentren entre los límites aceptables
  - *La intensidad de los defectos* (por ejemplo, cuántos defectos se encuentran por día o por transacción)
  - *La densidad de los defectos* (por ejemplo, el número de defectos encontrados en comparación con el número de historias de usuario, el esfuerzo y/o los atributos de calidad)
  - *El número estimado de defectos pendientes están dentro de unos límites aceptables* (Riesgos, consecuencias de no corregir, etc)
- **Tiempo:** Si se ha alcanzado la fecha de entrega prefijada, deben tenerse en cuenta las consideraciones de negocio asociadas a la entrega.
- **Coste:** El coste del ciclo de vida estimado debe utilizarse para calcular el retorno de la inversión para el sistema entregado (es decir, el desarrollo calculado y el coste de mantenimiento debe ser significativamente menor que las ventas totales esperadas del producto).

# Capítulo 3:

## Métodos, técnicas y herramientas de pruebas ágiles

### 3.1 Métodos de testing ágil

### 3.2 La evaluación de riesgos de calidad y Estimación de esfuerzo de prueba

### 3.3 Técnicas en proyectos ágiles

- Criterios de aceptación, cobertura adecuada y otra información para pruebas
- Aplicación del desarrollo guiado por pruebas de aceptación
- Diseño de pruebas de caja negra funcionales y no funcionales
- Pruebas exploratorias ("exploratory testing") y pruebas ágiles

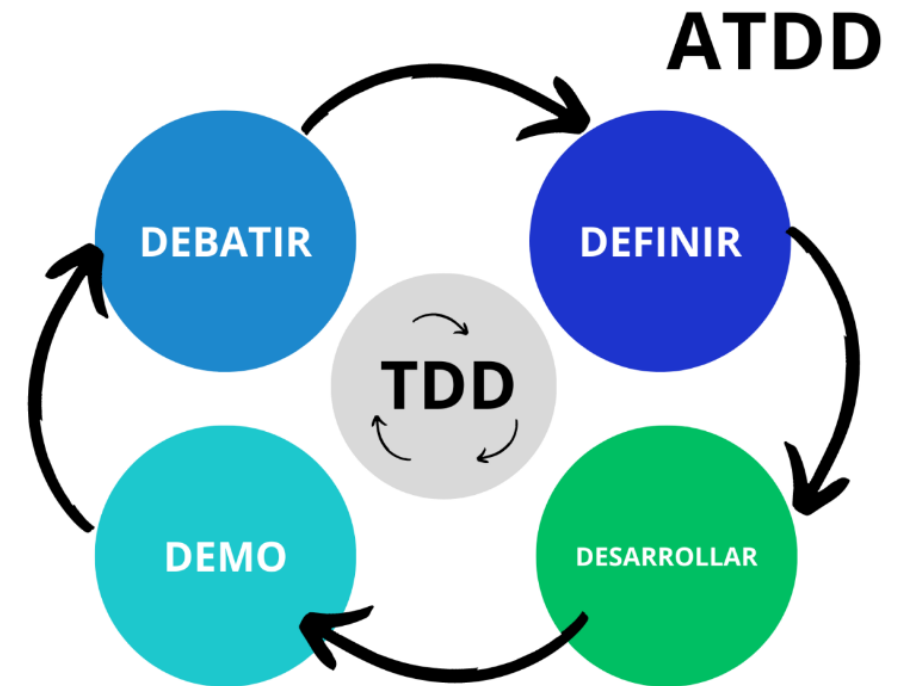
### 3.4 Herramientas en un proyecto ágil



## 3.3 Técnicas en proyectos ágiles

### *Aplicación del desarrollo guiado por prueba de aceptación*

- Etapas del desarrollo guiado por pruebas de aceptación:
  1. Se analiza, se debate y se escribe la HU por parte de desarrolladores, probadores y representantes del negocio. Todas las dudas, ambigüedades o errores se arreglan en esta etapa.
  2. Se crean las pruebas, las cuales las puede hacer el equipo en conjunto o el probador. El objetivo es convertir los criterios en pruebas automatizadas.
  3. Una persona independiente, como el representante del negocio, valida las pruebas.
  4. Se crean las pruebas
  5. Se escribe el código para que esas pruebas se aprueben (se corrigen defectos)
  6. Se genera el Demo del software desarrollado



Acceptance Test Driven Development

## 3.3 Técnicas en proyectos ágiles

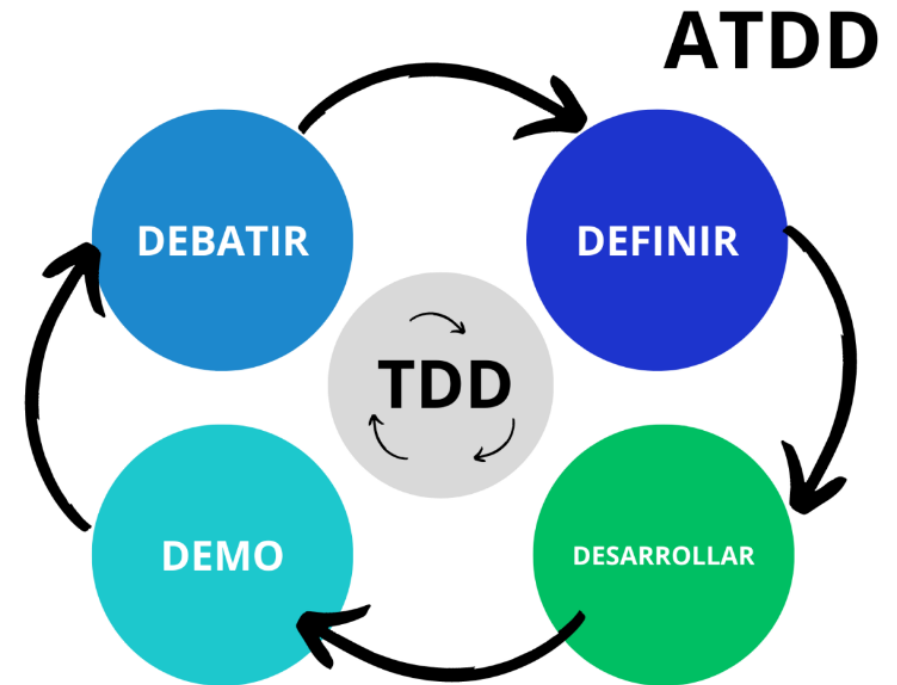
### *Aplicación del desarrollo guiado por prueba de aceptación*

- **Primero se crean Pruebas positivas**  
Por lo general, las primeras pruebas describen el comportamiento estándar sin excepción o condiciones de error. Incluyen el orden de actividades que van a ser ejecutadas y si todo va según lo planeado
- **Después las pruebas negativas**  
El equipo también debe cubrir excepciones, manejo de errores y características de calidad no funcionales (por ejemplo, rendimiento, facilidad de uso)

#### *Ejemplo:*

Una aplicación requiere una entrada válida edad entre 18 y 65 años ( ambos inclusive )

- Pruebas positivas: Entrada de 18 , 30 o 65 como la edad
- Pruebas negativas: Entrada de 17 , -5 o ABC como la edad



Acceptance Test Driven Development



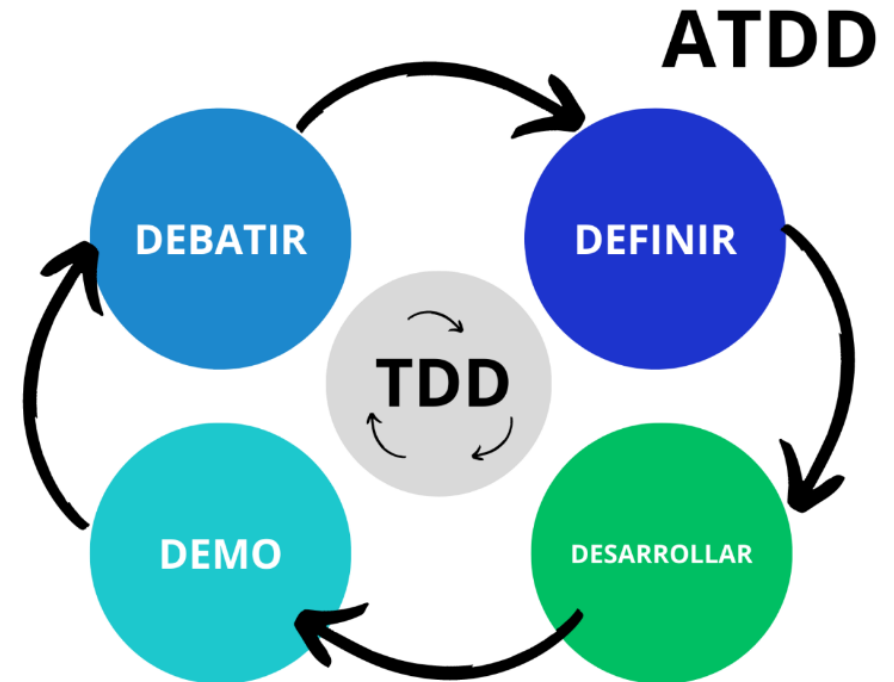
## 3.3 Técnicas en proyectos ágiles

### ***Aplicación del desarrollo guiado por prueba de aceptación***

- Las pruebas se formulan en lenguaje natural, por lo que todas las partes interesadas sean capaces de entenderlas
- Esto también incluye las condiciones previas necesarias, los aportes y los resultados esperados

#### **Normas prácticas para la creación de los ejemplos:**

- Todas las características de la historia de usuario se cubren
- Una característica sólo se describe por un ejemplo (Prueba)
- Los ejemplos no añaden nuevas características, significa que no hay ejemplo para cualquier aspecto no está documentado en la historia de usuario

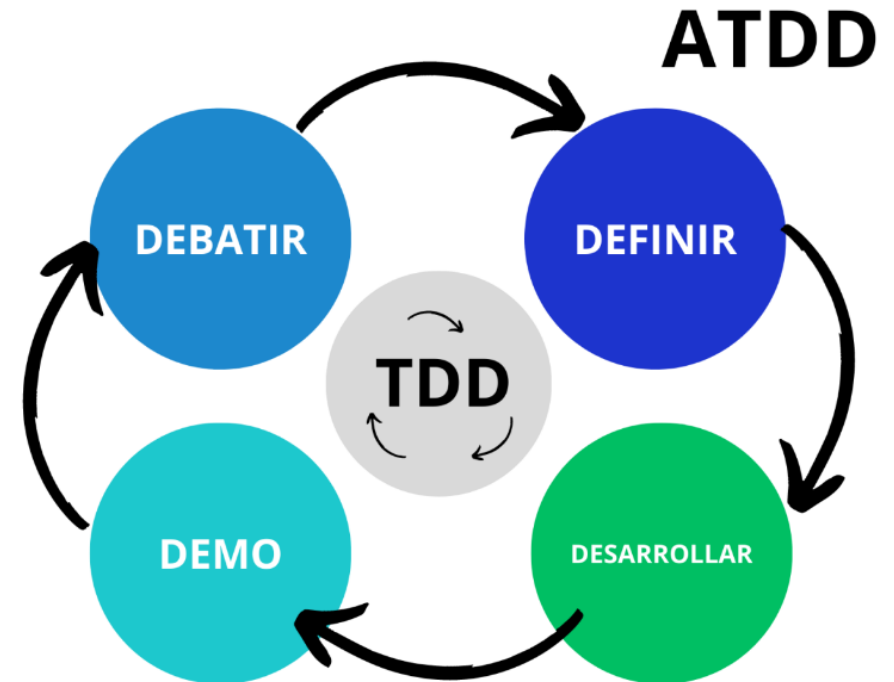


Acceptance Test Driven Development

## 3.3 Técnicas en proyectos ágiles

### *Aplicación del desarrollo guiado por prueba de aceptación*

- **Ejemplo:**
- Se está revisando una historia de usuario donde se quiere comprar libros y agregarlos al carrito.
- El equipo hace preguntas de la siguiente forma:
  - ¿Qué hace el sistema si el libro que se quiere añadir al carrito ya está dentro de él?
  - ¿Qué sucede si se ha agotado el libro en el almacén?
  - ¿Se le indica al usuario que el libro ha sido añadido al carrito de la compra?
- Y se crean pruebas/ejemplos de tipo:
  - Añadir libro X en stock produce: El libro X ha sido añadido al carrito
  - Libro X está contenido en el carrito
  - Libro X ya no está en catálogo de libros



Acceptance Test Driven Development

# Capítulo 3:

## Métodos, técnicas y herramientas de pruebas ágiles

### 3.1 Métodos de testing ágil

### 3.2 La evaluación de riesgos de calidad y Estimación de esfuerzo de prueba

### 3.3 Técnicas en proyectos ágiles

- Criterios de aceptación, cobertura adecuada y otra información para pruebas
- Aplicación del desarrollo guiado por pruebas de aceptación
- Diseño de pruebas de caja negra funcionales y no funcionales
- Pruebas exploratorias ("exploratory testing") y pruebas ágiles

### 3.4 Herramientas en un proyecto ágil

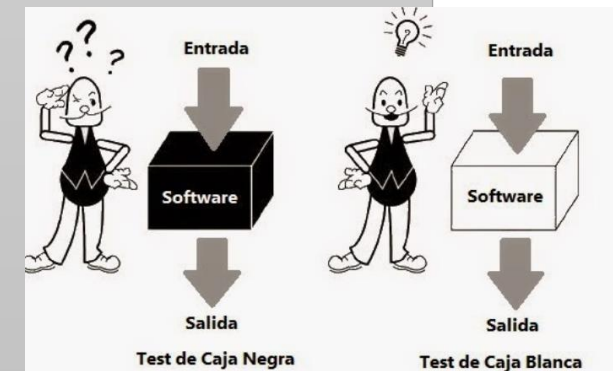


## 3.3 Técnicas en proyectos ágiles

### 3. Diseño caja negra funcionales y no funcionales

- En las pruebas ágiles, los probadores crean las pruebas en paralelo a las actividades de programación de los desarrolladores.
- Los probadores crean pruebas en base a las historias de usuario y a sus criterios de aceptación.
- Se pueden aplicar técnicas de diseño de pruebas de caja negra tradicionales:
  - partición de equivalencias,
  - el análisis de valores límite,
  - las tablas de decisión
  - las pruebas de transición de estado

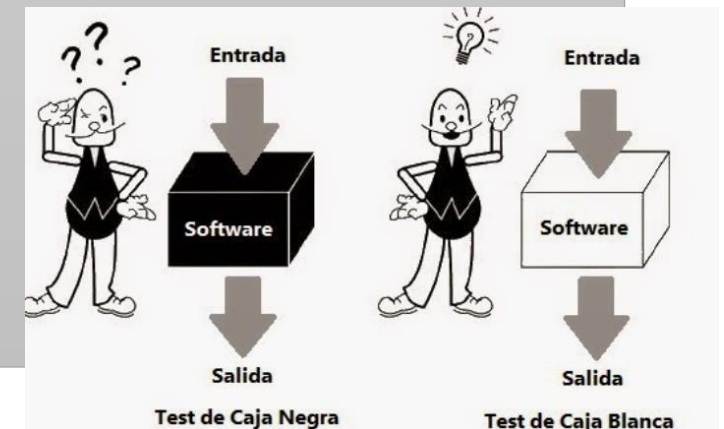
*Algunas pruebas, como pruebas exploratorias y otras pruebas basadas experiencia, se crean más tarde, durante la ejecución de la prueba*



## 3.3 Técnicas en proyectos ágiles

### 3. Diseño caja negra funcionales y no funcionales

- En muchas situaciones, los requisitos no funcionales pueden documentarse como historias de usuario.
- Las técnicas de diseño de pruebas de caja negra (como el análisis de valores límite) también pueden utilizarse para crear pruebas para características de calidad no funcionales.
- La historia de usuario puede contener requisitos de rendimiento o fiabilidad. Por ejemplo, una ejecución dada no puede exceder un límite de tiempo o un número de operaciones puede fallar menos de un cierto número de veces.



# Capítulo 3:

## Métodos, técnicas y herramientas de pruebas ágiles

### 3.1 Métodos de testing ágil

### 3.2 La evaluación de riesgos de calidad y Estimación de esfuerzo de prueba

### 3.3 Técnicas en proyectos ágiles

- Criterios de aceptación, cobertura adecuada y otra información para pruebas
- Aplicación del desarrollo guiado por pruebas de aceptación
- Diseño de pruebas de caja negra funcionales y no funcionales
- Pruebas exploratorias ("exploratory testing") y pruebas ágiles

### 3.4 Herramientas en un proyecto ágil





## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

- El testing exploratorio es importante en los proyectos ágiles debido al tiempo limitado que está disponible para el análisis de la prueba y el limitado nivel de detalle de las historias de usuario
- Para lograr mejores resultados, la prueba exploratoria debe combinarse con otros enfoques basados en la experiencia como parte de una estrategia de prueba reactiva
- También otras estrategias de prueba se pueden combinar :
  - pruebas basadas en riesgo analíticos
  - pruebas basados en requisitos analíticos
  - pruebas basadas en modelos
  - pruebas de regresión





## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

- En las pruebas exploratorias, el diseño de pruebas y la ejecución de pruebas tienen lugar a la vez, guiadas por un contrato de pruebas elaborado.
- **Un contrato de pruebas (carta de pruebas)** establece las condiciones de prueba a cubrir durante sesiones de pruebas limitadas en el tiempo.
- Durante las pruebas exploratorias, los resultados de las últimas pruebas guían la siguiente prueba.
- Pueden utilizarse las mismas técnicas de caja negra y caja blanca para diseñar las pruebas que las utilizadas para las pruebas prediseñadas.



## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

**Un contrato de pruebas puede incluir la siguiente información:**

- Actor: usuario previsto del sistema
- Propósito: el tema del contrato, incluyendo qué objetivo particular quiere alcanzar el actor, es decir, las condiciones de prueba
- Preparación: qué tiene que haber para poder iniciar la ejecución de la prueba
- Prioridad: importancia relativa de este contrato, en base a la prioridad de la historia de usuario asociada o del nivel de riesgo
- Referencias: especificaciones (por ejemplo, historia de usuario), riesgos, u otras fuentes de información
- Datos: los datos necesarios para llevar a cabo el contrato



## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

- Actividades: una lista de ideas de lo que el actor puede querer hacer con el sistema (por ejemplo, "Acceder al sistema como superusuario") y qué sería interesante probar (pruebas tanto positivas como negativas)
- Notas de oráculo: cómo evaluar el producto para establecer los resultados correctos (por ejemplo, capturar qué pasa en la pantalla y compararlo con lo que está escrito en el manual de usuario).
- Variaciones: acciones y evaluaciones alternativas para complementar las ideas descritas bajo actividades



## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

#### ***Sesiones de pruebas exploratorias:***

Por sesión se entiende un periodo ininterrumpido de pruebas que podría durar de 60 a 120 minutos.

- Las sesiones de pruebas incluyen lo siguiente:
- Sesión de reconocimiento (para aprender cómo funciona)
- Sesión de análisis (evaluación de la funcionalidad o características)
- Cobertura exhaustiva (casos extremos, escenarios, interacciones)



## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

***En las pruebas exploratorias, lo más importante es la habilidad del probador para hacer las preguntas sobre que probar:***

- ¿Qué es lo más importante a averiguar del sistema?
- ¿De qué forma puede fallar el sistema?
- ¿Qué sucede si...?
- ¿Qué debería suceder cuando...?
- ¿Se cumplen las necesidades, requisitos y expectativas del cliente?
- ¿Se puede instalar el sistema (y retirar si procede) en todas las rutas de actualización soportadas?



## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

#### ***Importante!***

- Durante la ejecución de la prueba, el probador utiliza la creatividad, la intuición, el conocimiento y las habilidades, para encontrar posibles problemas en el software
- El probador también debe poseer una sólida comprensión del software bajo prueba, así como el conocimiento de dominio, del uso del software y cómo determinar si el software falla
- Puede aplicarse una serie de heurísticas para guiar las pruebas:
  - Valores límites
  - CRUD (Crear, leer, actualizar, borrar)
  - Variaciones de la configuración
  - Interrupciones ( por ejemplo, desconexión, caída o reinicio)



## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

#### ***Documentación en las pruebas exploratorias:***

***Es importante documentar las pruebas exploratorias en la máxima medida posible.***

#### ***Algunos ejemplos de lo que es importante documentar:***

- Cobertura de pruebas: qué entrada se ha utilizado, cuánto se ha cubierto y cuánto queda pendiente de probar
- Notas de evaluación: observaciones durante las pruebas, ¿el sistema y la funcionalidad objeto de la prueba parecen estables?, ¿se ha encontrado algún defecto?, ¿qué se prevé como pasos siguientes según las observaciones actuales?, y cualquier otra lista de ideas.
- Lista de riesgos/estrategia: ¿Qué riesgos se han cubierto y qué riesgos siguen pendientes de cubrir entre los más importantes?, ¿se seguirá la estrategia inicial?, ¿necesita algún cambio?





## 3.3 Técnicas en proyectos ágiles

### 4. Pruebas exploratorias y ágiles

#### ***Documentación en las pruebas exploratorias:***

- Problemas, preguntas y anomalías: cualquier comportamiento imprevisto, cualquier pregunta sobre la eficiencia del enfoque, cualquier inquietud sobre las ideas/intentos de la prueba, el entorno de prueba, los datos de prueba, mala interpretación de la función, de script de prueba o del sistema bajo prueba.
- Comportamiento real: registro de comportamiento real del sistema que debe guardarse (por ejemplo, vídeo, impresiones de pantalla, ficheros de datos de salida)

***La información registrada debe capturarse y/o resumirse mediante algún tipo de herramienta de gestión de estados.***



# Capítulo 3:

## Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

3.2 La evaluación de riesgos de calidad y Estimación de esfuerzo de prueba

3.3 Técnicas en proyectos ágiles

3.4 Herramientas en un proyecto ágil





## 3.4 Herramientas en proyectos ágiles

### *Herramientas en un proyecto ágil*

#### Generalidades

A veces las herramientas se utilizan de manera diferente en metodologías ágiles y tradicionales.

En metodologías tradicionales se prefieren herramientas individuales como de gestión de pruebas, gestión de requisitos y gestión de defectos

- En equipos ágiles prefieren:
  - Herramientas más integrales como gestión del ciclo de vida de las aplicaciones o gestión de tareas, ya que utilizan funciones como tableros de tareas, burndown charts e historias de los usuarios
  - Herramientas de gestión de la configuración por dada la gran cantidad de pruebas automatizadas en todos los niveles y la necesidad de almacenar y gestionar los artefactos de prueba automatizados asociados



## 3.4 Herramientas en proyectos ágiles

### ***Herramientas de gestión y seguimiento de tareas***

#### Gestión y seguimiento de tareas

Para seguimiento de tareas se puede ocupar:

- tableros físicos de historias /tareas (por ejemplo, pizarra, tablero de corcho)
- software de gestión del ciclo de vida de las aplicaciones y gestión de tareas, incluyendo tableros de tareas electrónicos.

Estas herramientas se utilizan para gestionar y realizar un seguimiento de las historias de usuario, pruebas y otras tareas en todos los sprints

## 3.4 Herramientas en proyectos ágiles

### ***Herramientas de gestión y seguimiento de tareas***

#### Gestión y seguimiento de tareas

Estas herramientas sirven para los siguientes fines :

- Registrar las historias con el desarrollo relevante - y las tareas de prueba
- Recoger las estimaciones individuales de los miembros del equipo, el cálculo del esfuerzo total necesario para la implementación de la historia
- Combinando tanto el desarrollo y prueba de las tareas de la misma historia, para proporcionar una imagen completa de los esfuerzos necesarios para que los equipos puedan implementar la historia
- Registrar el estado de las tareas después de cada actualización de los desarrolladores y testers
- Representación visual (a través de métricas, gráficos y cuadros de mando ) de la situación real de cada historia de usuario , de la iteración y el lanzamiento
- Integración con herramientas de gestión de la configuración
  - registro automático de código de registro de entrada y construcción para tareas
  - actualizaciones de estado automatizados para tareas

## 3.4 Herramientas en proyectos ágiles

### ***Herramientas de comunicación y para compartir información***

#### **Herramientas de Comunicación e información**

- Normalmente equipos comparten información en
  - correo electrónico
  - los documentos
  - la comunicación verbal
- Los equipos ágiles a menudo utilizan tres tipos de herramientas para fomentar y compartir la información:
  - Wikis
  - mensajería instantánea
  - Escritorios compartidos

## 3.4 Herramientas en proyectos ágiles

### ***Herramientas de comunicación y para compartir información***

#### Comunicación e información

Las wikis permiten a los equipos construir y compartir una base de conocimiento on-line en base a varios aspectos del proyecto, incluidos los siguientes:

- Diagramas, grupos de debate, imágenes relacionadas con características relevantes, prototipos u otra información
- Herramientas y/o técnicas de desarrollo y pruebas que sean útiles para otros miembros del equipo
- Métricas, gráficos y cuadros de mando sobre el estado del producto
- Conversaciones entre miembros del equipo parecidas a la mensajería instantánea y al correo electrónico, compartiéndolo de esta forma con todos los miembros del equipo






## 3.4 Herramientas en proyectos ágiles

### ***Herramientas de comunicación y para compartir información***

#### Comunicación e información

Los sistemas de mensajería instantánea, audio conferencia y herramientas de chat con vídeo ofrecen las siguientes ventajas:

- Permiten una comunicación directa en tiempo real entre los miembros del equipo, especialmente en el caso de equipos deslocalizados
- Implican a los equipos deslocalizados en las reuniones de seguimiento
- Reducen la factura telefónica al utilizar tecnología de voz sobre IP, eliminando las restricciones de coste que podrían limitar la comunicación entre los miembros del equipo en ubicaciones distribuidas

 Google Chat

## 3.4 Herramientas en proyectos ágiles

### ***Herramientas de comunicación y para compartir información***

#### Comunicación e información

Herramienta de escritorio compartido

- En los equipos deslocalizados pueden realizarse demostraciones de producto, revisiones de código e incluso trabajo por pares
- publicarse en la wiki del equipo



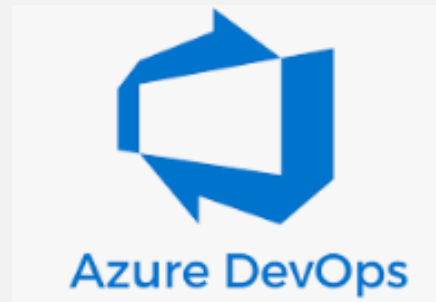
## 3.4 Herramientas en proyectos ágiles

### *Herramientas de construcción y distribución de software*

#### Herramientas de Construcción y distribución de software

La construcción diaria y el despliegue del software constituyen una práctica clave en los equipos ágiles.

Para ello es necesario el uso de herramientas de integración continua y herramientas de distribución de versiones.



## 3.4 Herramientas en proyectos ágiles

### ***Herramientas de gestión de la configuración***

#### Herramientas de Gestión de la configuración

Se pueden utilizar herramientas de gestión de la configuración sirven para

- almacenar código fuente
- pruebas automatizadas
- almacenan pruebas manuales
- otros productos de trabajo de pruebas en el mismo repositorio que el código fuente del producto.

*Esto proporciona trazabilidad para saber entre qué versiones del software se han probado y con qué versiones específicas de las pruebas, y permite el cambio rápido sin perder información histórica.*





## 3.4 Herramientas en proyectos ágiles

### *Herramientas de diseño, implementación y ejecución de pruebas*

#### Herramientas para pruebas

- **Herramientas de diseño de pruebas:** El uso de herramientas como los mapas mentales han ganado popularidad para diseñar y definir pruebas rápidamente para una nueva funcionalidad
- **Herramientas de gestión de casos de prueba:** El tipo de herramientas de gestión de casos de prueba utilizado en los procesos ágiles puede formar parte de la herramienta de gestión del ciclo de vida de la aplicación o de la herramienta de gestión de tareas de todo el equipo
- **Herramientas de preparación de datos de prueba y herramientas de generación de datos de prueba:** Las herramientas que generan datos para poblar la base de datos de una aplicación son muy beneficiosas cuando se requieren muchos datos y combinaciones de datos para probar la aplicación.

## 3.4 Herramientas en proyectos ágiles

### *Herramientas de diseño, implementación y ejecución de pruebas*

#### Herramientas para pruebas

- **Herramientas de carga de datos de prueba:** Una vez generados los datos para las pruebas, deben cargarse en la aplicación. La entrada manual de datos a menudo lleva mucho tiempo y es propensa a cometer errores, no obstante hay herramientas de carga de datos disponibles para conseguir que el proceso sea fiable y eficiente. De hecho, muchas herramientas generadoras de datos llevan incorporado un componente de carga de datos. En otros casos, también puede recurrirse a la carga masiva a través de los sistemas de gestión de bases de datos
- **Herramientas de ejecución de pruebas automatizadas:** Hay herramientas de ejecución de pruebas que están más alineadas con las pruebas ágiles. Hay herramientas específicas de código abierto y comerciales que permiten soportar enfoques de probar primero, como son el desarrollo guiado por el comportamiento, el desarrollo guiado por pruebas y el desarrollo guiado por pruebas de aceptación.
- **Herramienta de pruebas exploratorias:** Las herramientas que capturan y registran actividades llevadas a cabo en la aplicación durante una sesión de pruebas exploratorias son beneficiosas para el probador y el desarrollador, ya que registran las acciones realizadas





## 3.4 Herramientas en proyectos ágiles

### *Herramientas Cloud y de Virtualización*

#### Herramientas Cloud y virtualización

**La virtualización** incluye la creación de máquinas virtuales (no físicos) como emulación de hardware, un sistema operativo, almacenamiento de datos o recursos de la red . Esto permite combinar o dividir los recursos informáticos de manera transparente.

Mediante el uso de la virtualización , los equipos pueden tener un mayor número de recursos (por ejemplo, servidor) para el desarrollo y las pruebas disponibles .

**El tiempo de espera se puede reducir , por ejemplo, recursos hasta ahora bloqueados exclusivamente**

Algunas herramientas de gestión de pruebas actualmente utilizan tecnologías de virtualización para realizar la instantánea de los servidores en el momento en que se detecta un fallo, lo que permite a los probadores compartir la instantánea con los desarrolladores que están investigando el fallo.





# Curso ISTQB® Foundation LEVEL AGILE TESTER

