



Curso ISTQB® Foundation LEVEL AGILE TESTER



Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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



3.1 Métodos de testing ágil

Métodos de pruebas ágiles

Existen **prácticas** de pruebas que pueden realizarse en proyectos ágiles como en tradicionales, que pueden seguirse para conseguir productos de calidad.

- Escribir pruebas con antelación para expresar el comportamiento correcto
- Centrarse en la prevención, detección y eliminación temprana de defectos.
- Asegurar que se ejecutan los tipos de pruebas adecuados en el momento idóneo y en el nivel de prueba correcto

Nosotros como probadores **tenemos la función de orientar** sobre el uso de estas prácticas en todo el ciclo de desarrollo.

Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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



3.1 Métodos de testing ágil

Desarrollo guiado por pruebas

Los equipos ágiles utilizan tres técnicas complementarias , para llevar a cabo las pruebas en los distintos niveles de pruebas.

Cada una de estas técnicas es un ejemplo del principio fundamental de las pruebas: **Pruebas tempranas, ya que estas se definen antes de escribir el código.**

- Desarrollo basado en pruebas (TDD)
- Desarrollo guiado por pruebas de aceptación (ATDD)
- Desarrollo guiado por el comportamiento(BDD)

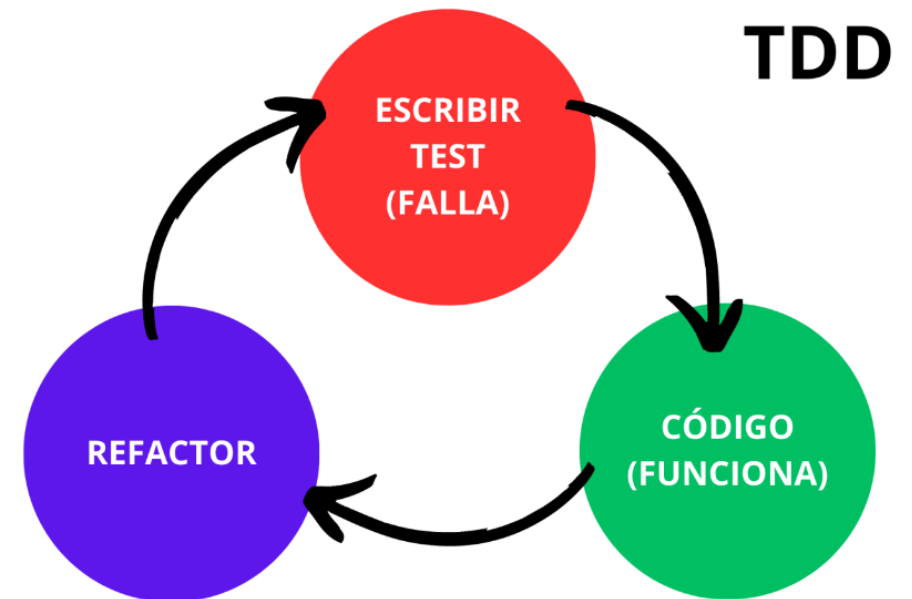
3.1 Métodos de testing ágil

Desarrollo guiado por pruebas- TDD

TDD- Test driven developmen, se utiliza para desarrollar código guiado por casos de prueba automatizados.

Las etapas del proceso de prueba es el siguiente:

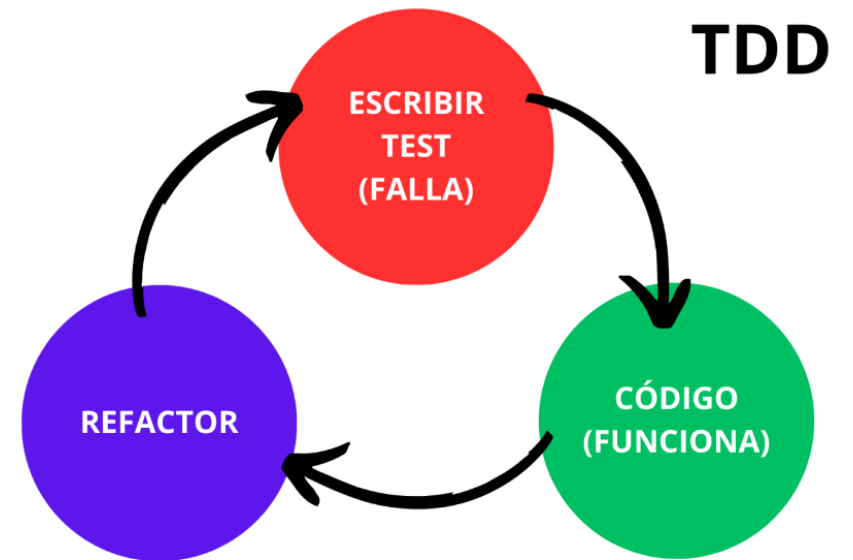
- Añadir una prueba que capte el concepto que tiene el programador del funcionamiento deseado de un pequeño trozo de código.
- Ejecutar la prueba, que debería fallar ya que el código no existe.
- Escribir el código y ejecutar la prueba hasta que esta pase.
- Refactorizar el código una vez pasada la prueba, volver a ejecutar la prueba para asegurar que sigue pasando contra el código refactorizado.
- Repetir este proceso para el siguiente pequeño trozo de código, ejecutando tanto las pruebas previas como las pruebas añadidas.



3.1 Métodos de testing ágil

Desarrollo guiado por pruebas- TDD

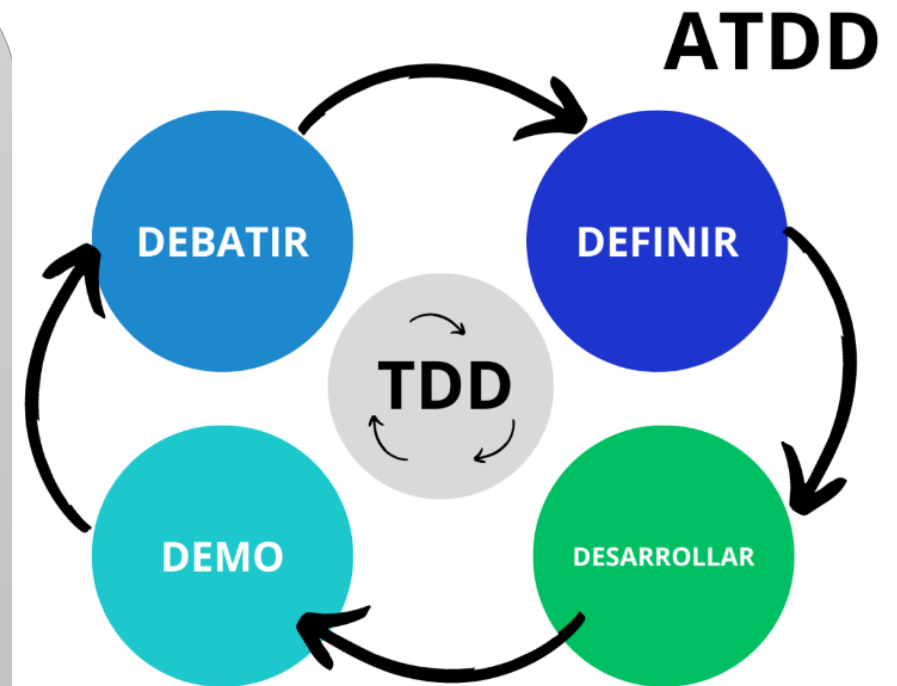
- Las pruebas escritas son principalmente de nivel unitario y están centradas en el código
- También pueden realizarse a nivel de sistema y de integración, pero el enfoque principal es unitario.
- Este modelo se hizo popular en XP.
- Este tipo de desarrollo ayuda a los desarrolladores a concentrarse en resultados esperados claramente definidos
- Las pruebas se automatizan y se utilizan en la integración continua



3.1 Métodos de testing ágil

Desarrollo guiado por pruebas de Aceptación- ATDD

- El objetivo de ATDD es fomentar una colaboración permita una comprensión compartida de los requisitos del sistema, en forma de especificaciones escritas en un lenguaje sencillo. Después, las especificaciones se convierten en casos de pruebas de aceptación automatizadas.
- El desarrollo guiado por pruebas de aceptación define los criterios y las pruebas de aceptación durante la creación de historias de usuario
- ATDD permite a todas las partes interesadas conocer cómo tiene que comportarse el componente de software y qué necesitan los desarrolladores, probadores y representantes de negocio para asegurar este comportamiento.
- El desarrollo guiado por pruebas de aceptación crea pruebas reutilizables para las pruebas de regresión

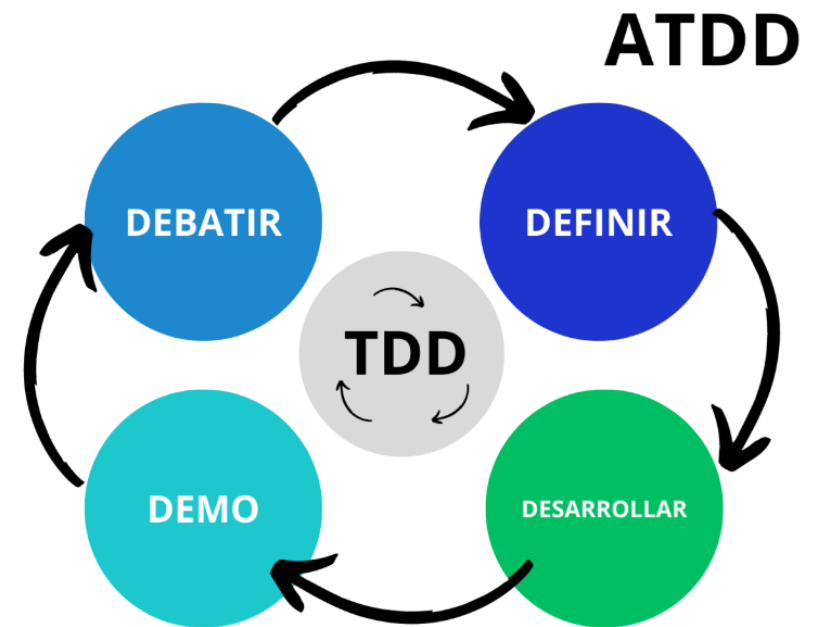


Acceptance Test Driven Development

3.1 Métodos de testing ágil

Desarrollo guiado por pruebas de Aceptación- ATDD

- **Importante:**
- Las pruebas son ejemplos, que describen las características específicas de la historia de usuario.
- Debido a que los ejemplos y las pruebas son las mismas, estos términos se utilizan a menudo como sinónimo
- Las pruebas se crean antes de implementar la HU, las cuales pueden ser manuales y automatizados. (De preferencia Automatizados)
- ATDD pone la automatización de las pruebas en primer plano. Al adoptarlo, los equipos utilizarán casos de prueba automatizados para impulsar el desarrollo del código de producción. (La automatización se realiza antes de desarrollar)

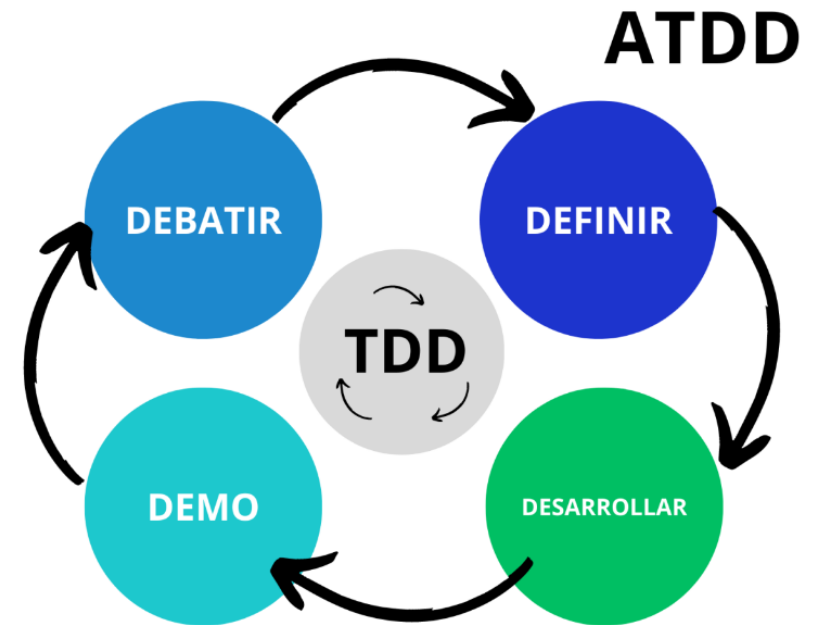


Acceptance Test Driven Development

3.1 Métodos de testing ágil

Desarrollo guiado por pruebas de Aceptación- ATDD

- El desarrollo guiado por pruebas de aceptación permite la rápida resolución de defectos y la validación del comportamiento de las funcionalidades.
- Ayuda a determinar si se cumplen los criterios de aceptación para la funcionalidad.
- Se ejecutan pruebas a nivel de sistema o aceptación.

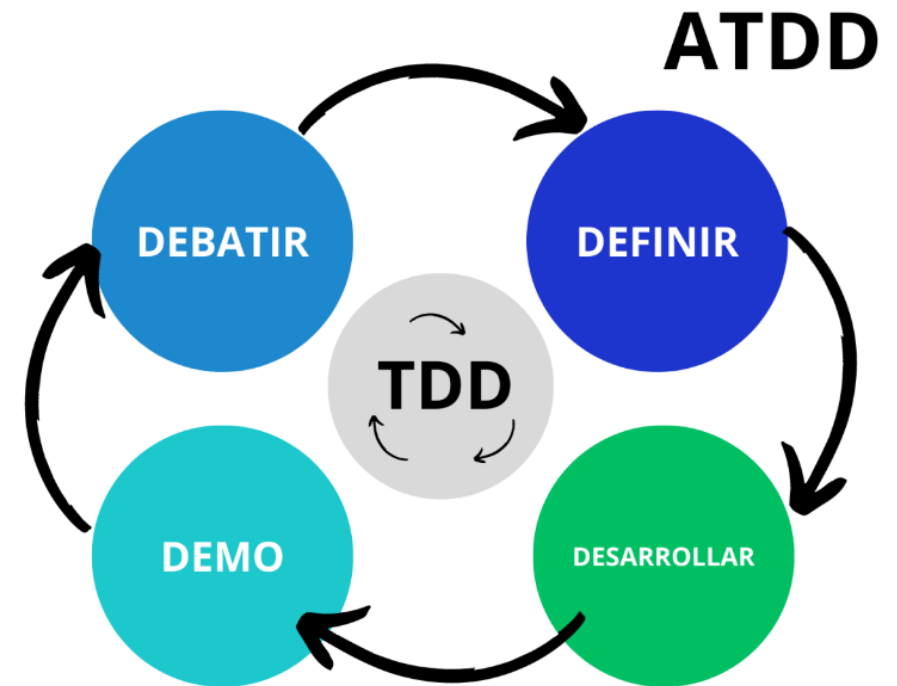


Acceptance Test Driven Development

3.1 Métodos de testing ágil

Desarrollo guiado por pruebas de Aceptación- ATDD

- 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

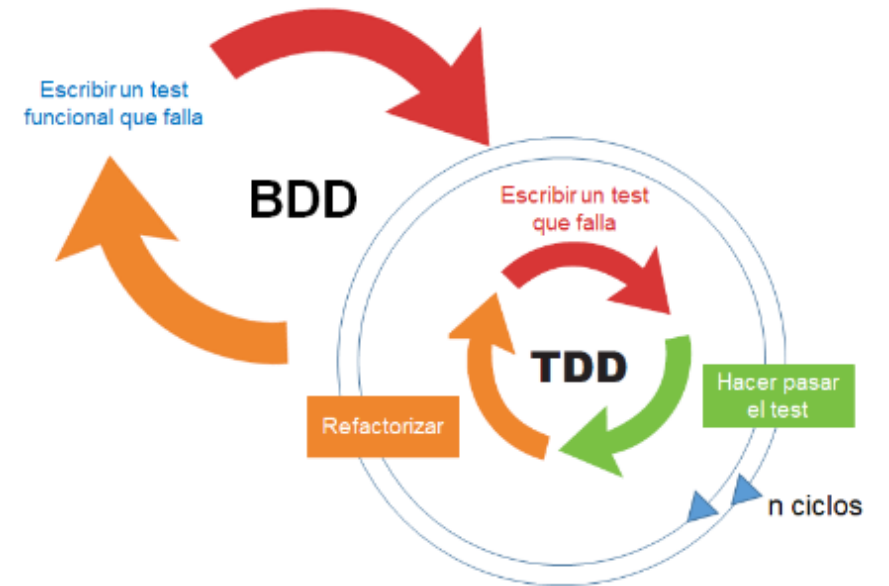


Acceptance Test Driven Development

3.1 Métodos de testing ágil

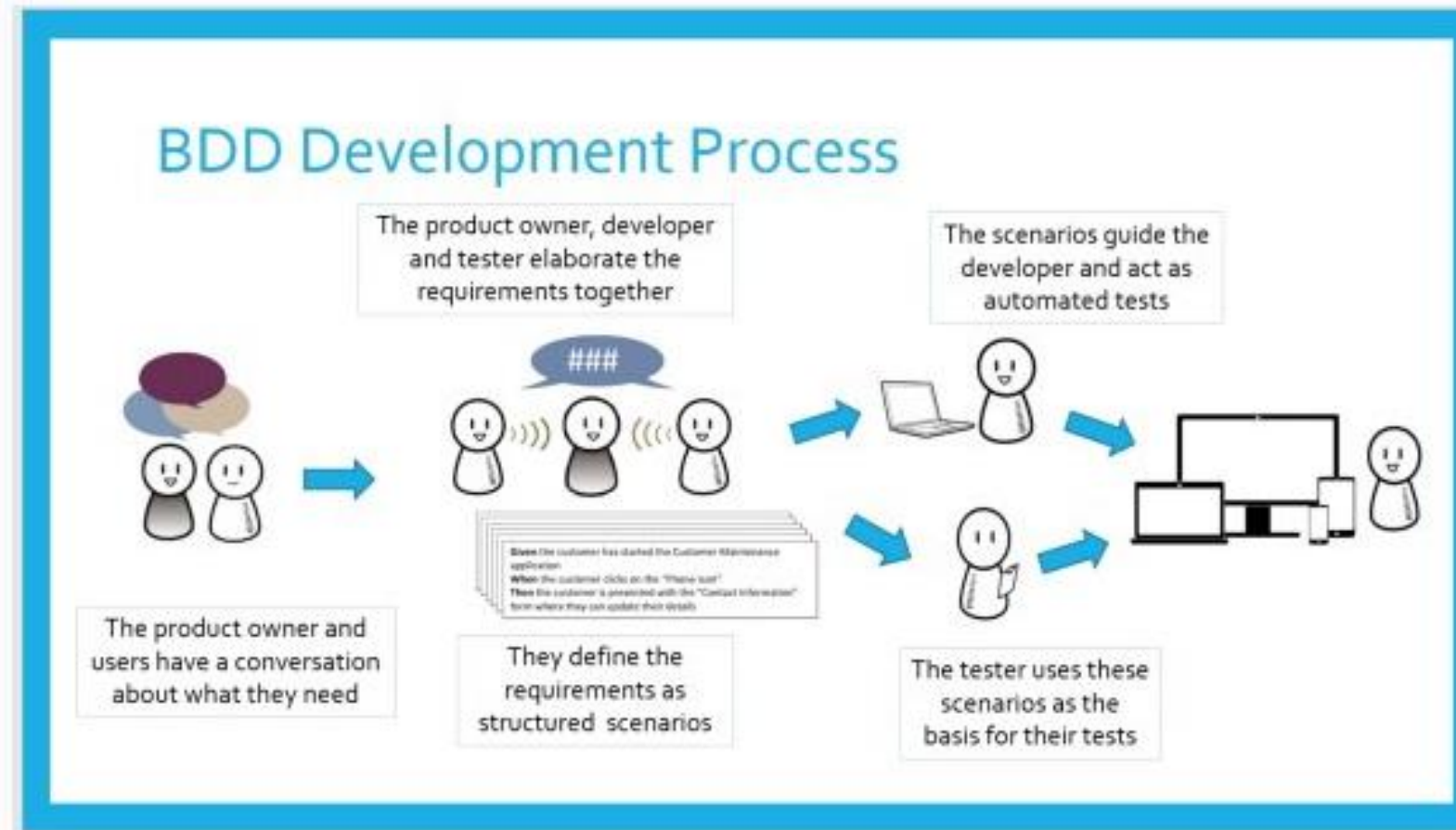
Desarrollo guiado por comportamiento- BDD

- Tanto en TDD como en BDD, las pruebas se deben definir antes del desarrollo, aunque en BDD, las pruebas se centran en el usuario y el comportamiento del sistema, a diferencia del TDD que se centra en funcionalidades.
- El desarrollo guiado por el comportamiento permite al desarrollador centrarse en probar el código basándose en el comportamiento esperado del software
- Las pruebas son más fáciles de entender para los demás miembros del equipo y partes interesadas dado que se basan en el comportamiento manifestado del software
- El desarrollo guiado por comportamiento describe casos de prueba en un lenguaje natural que los no programadores son capaces de leer:



3.1 Métodos de testing ágil

Desarrollo guiado por comportamiento- BDD



3.1 Métodos de testing ágil

Desarrollo guiado por comportamiento- BDD

BDD define los criterios de aceptación tomando como base el formato dado/cuando/entonces:

- Dado (Given) un contexto inicial,
- Cuando (When) se da un evento,
- Entonces (Then) se verifican ciertos resultados.

```
Feature: Búsqueda en Google
```

```
Como usuario web, quiero buscar en Google para poder responder mis dudas.
```

```
Scenario: Búsqueda simple en Google
```

```
Given un navegador web en la página de Google
```

```
When se introduce la palabra de búsqueda "pingüino"
```

```
Then se muestra el resultado de "pingüino"
```

3.1 Métodos de testing ágil

Desarrollo guiado por comportamiento- BDD

BDD define los criterios de aceptación tomando como base el formato dado/cuando/entonces:

- Dado (Given) un contexto inicial,
- Cuando (When) se da un evento,
- Entonces (Then) se verifican ciertos resultados.

Ejemplo 1:

El usuario cliente puede ingresar usuario y contraseña correcta

Dado, el cliente

Cuando , ingresa su usuario y contraseña

Entonces, ingresa al sistema

Ejemplo 2:

El usuario cliente tiene tres intentos para logearse, si no queda bloqueado

Dado, el usuario cliente

Cuando , ingresa su usuario y contraseña errónea tres veces

Entonces, el sistema bloquea a usuario

Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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



3.1 Métodos de testing ágil

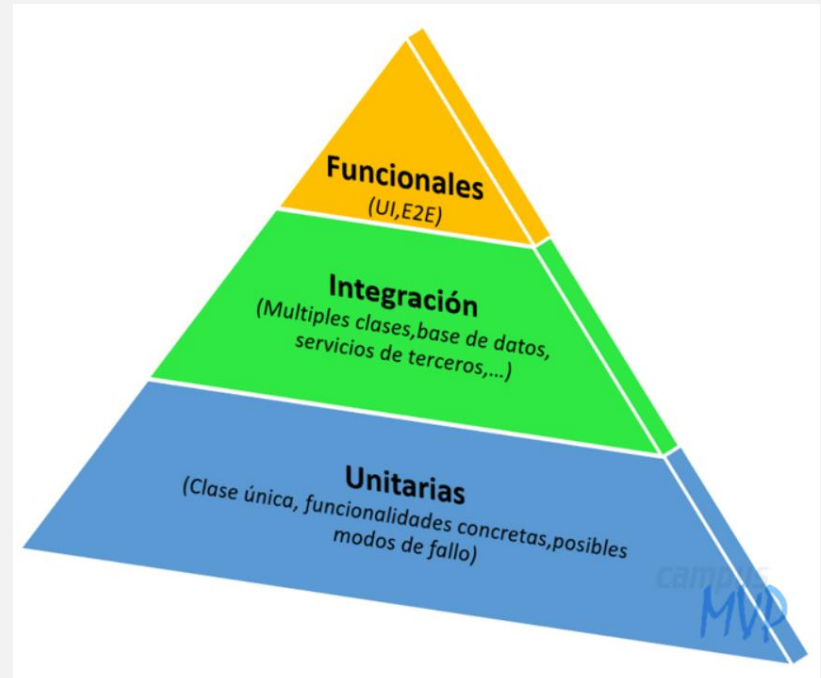
Piramide de pruebas

Piramide de Pruebas

Niveles de prueba típicos para sistemas de software son:

- Pruebas unitarias
- Pruebas de integración
- Pruebas del sistema
- Pruebas de aceptación

La pirámide de pruebas destaca por tener un número mayor de pruebas en los niveles inferiores (base de la pirámide) y, a medida que el desarrollo avanza hacia los niveles superiores, el número de pruebas se reduce .



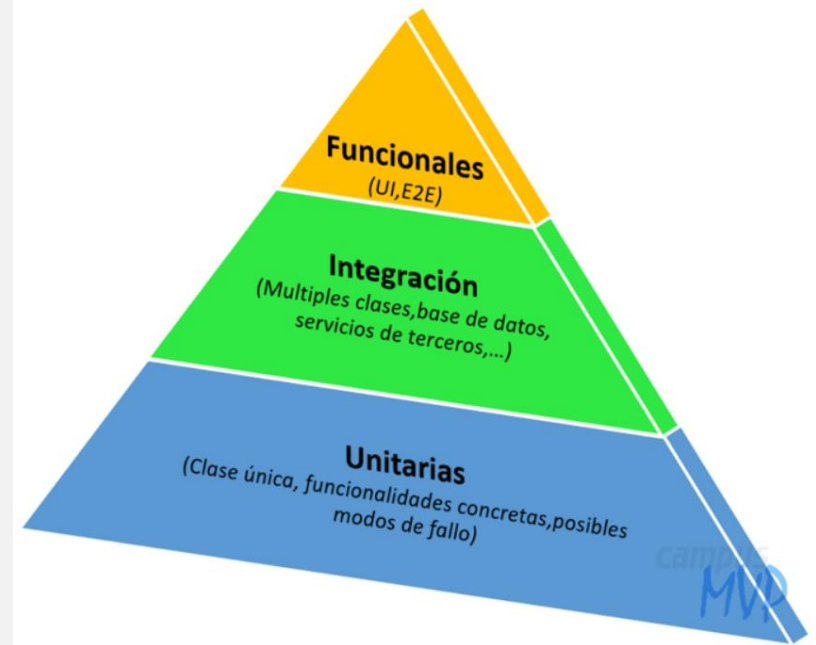
3.1 Métodos de testing ágil

Piramide de pruebas

Piramide de Pruebas

- Normalmente las pruebas de nivel unitario y de integración se automatizan y se crean empleando herramientas basadas en API .
- Las de sistemas y de aceptación se crean empleando herramientas GUI. (Interfaz gráfica usuario)

El concepto de la pirámide de pruebas se basa en el principio de aseguramiento de la calidad temprano y pruebas tempranas (es decir, eliminando los defectos lo antes posible en el ciclo de vida).



Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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

3.3 Técnicas en proyectos ágiles

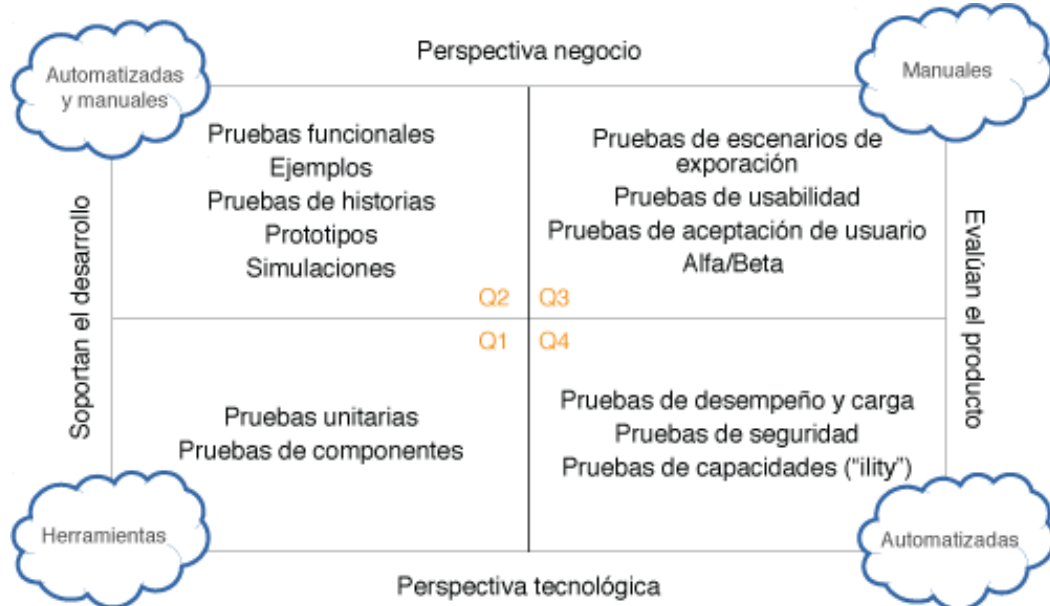
3.4 Herramientas en un proyecto ágil



3.1 Métodos de testing ágil

Cuadrante de pruebas

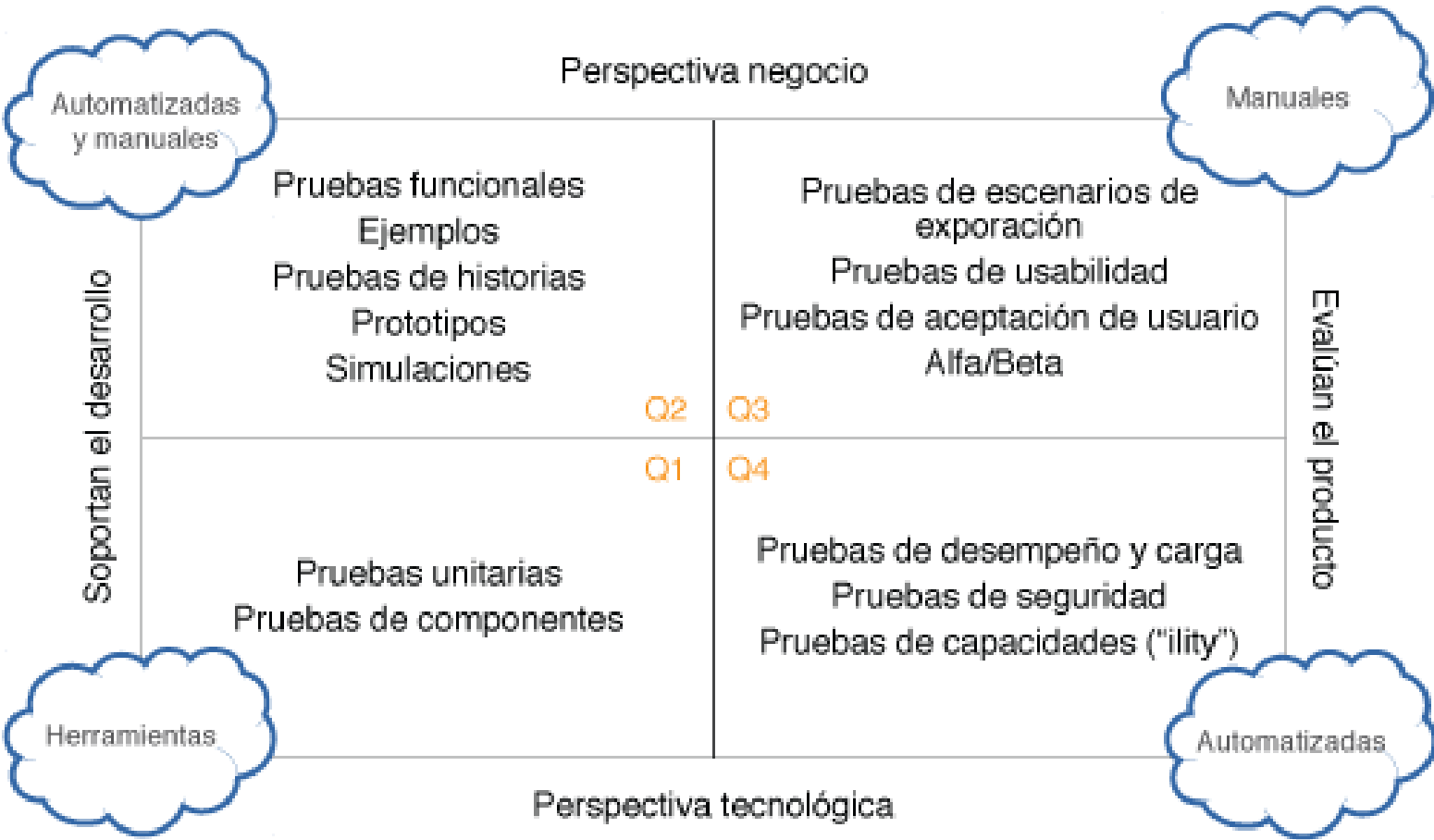
- Alinea los niveles de prueba con tipos de prueba adecuados en la metodología ágil
- Este es un modelo que permite asegurar que todos los tipos de pruebas importantes y niveles de prueba se incluyen en el ciclo de vida de desarrollo
- Ayuda a describir y diferenciar los tipos de pruebas para todos los interesados, incluyendo desarrolladores, probadores y representantes de las empresas
- Durante cualquier iteración dada, las pruebas pueden estar en uno o todos los cuadrantes que puedan ser requeridos



3.1 Métodos de testing ágil

Cuadrante de pruebas

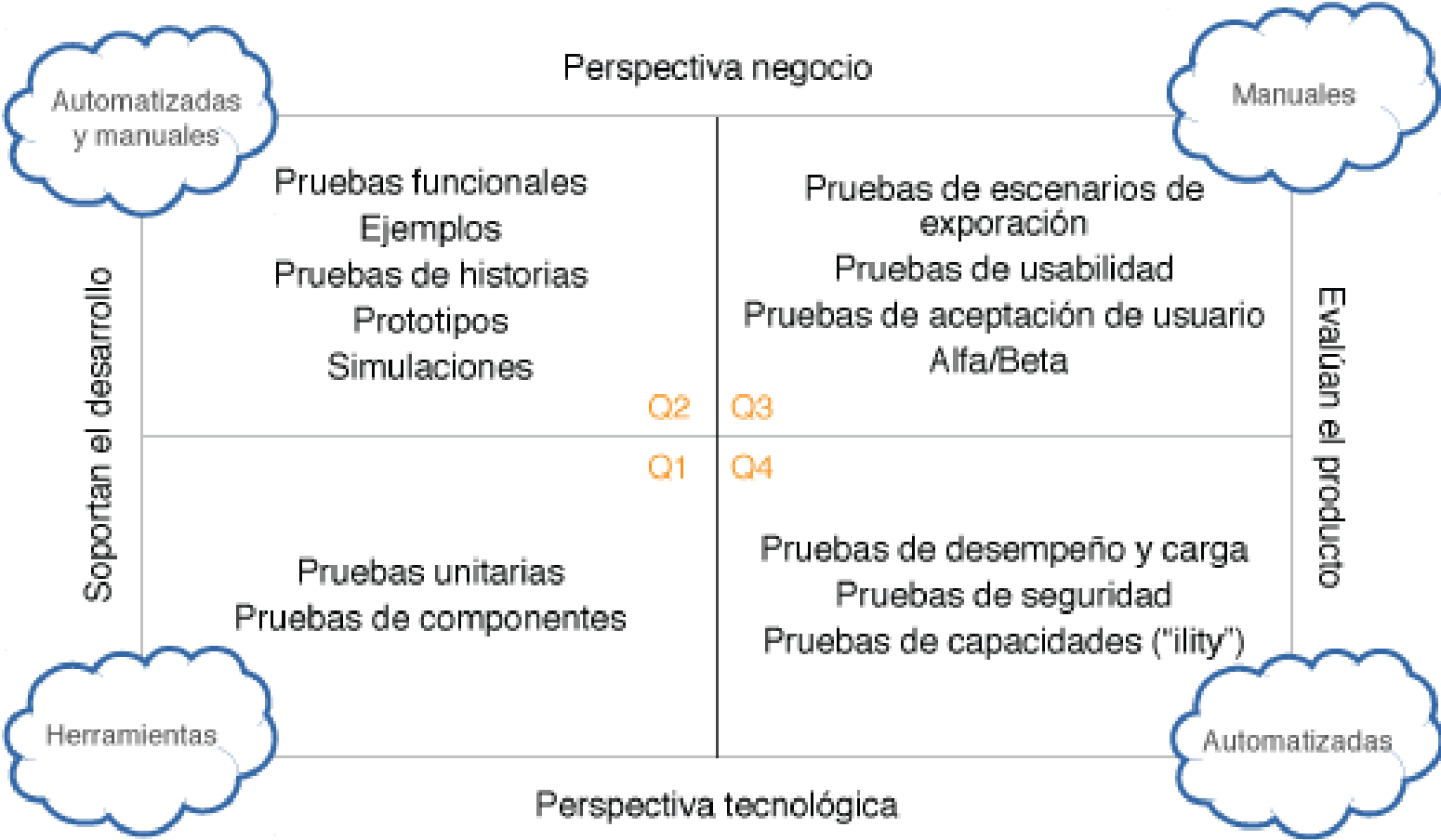
- Q1:
- Nivel de pruebas unitarias
 - Relativo a la tecnología y da soporte al equipo
 - Contiene pruebas unitarias, las cuales deben automatizarse e incluirse en el sistema de integración continua.



3.1 Métodos de testing ágil

Cuadrante de pruebas

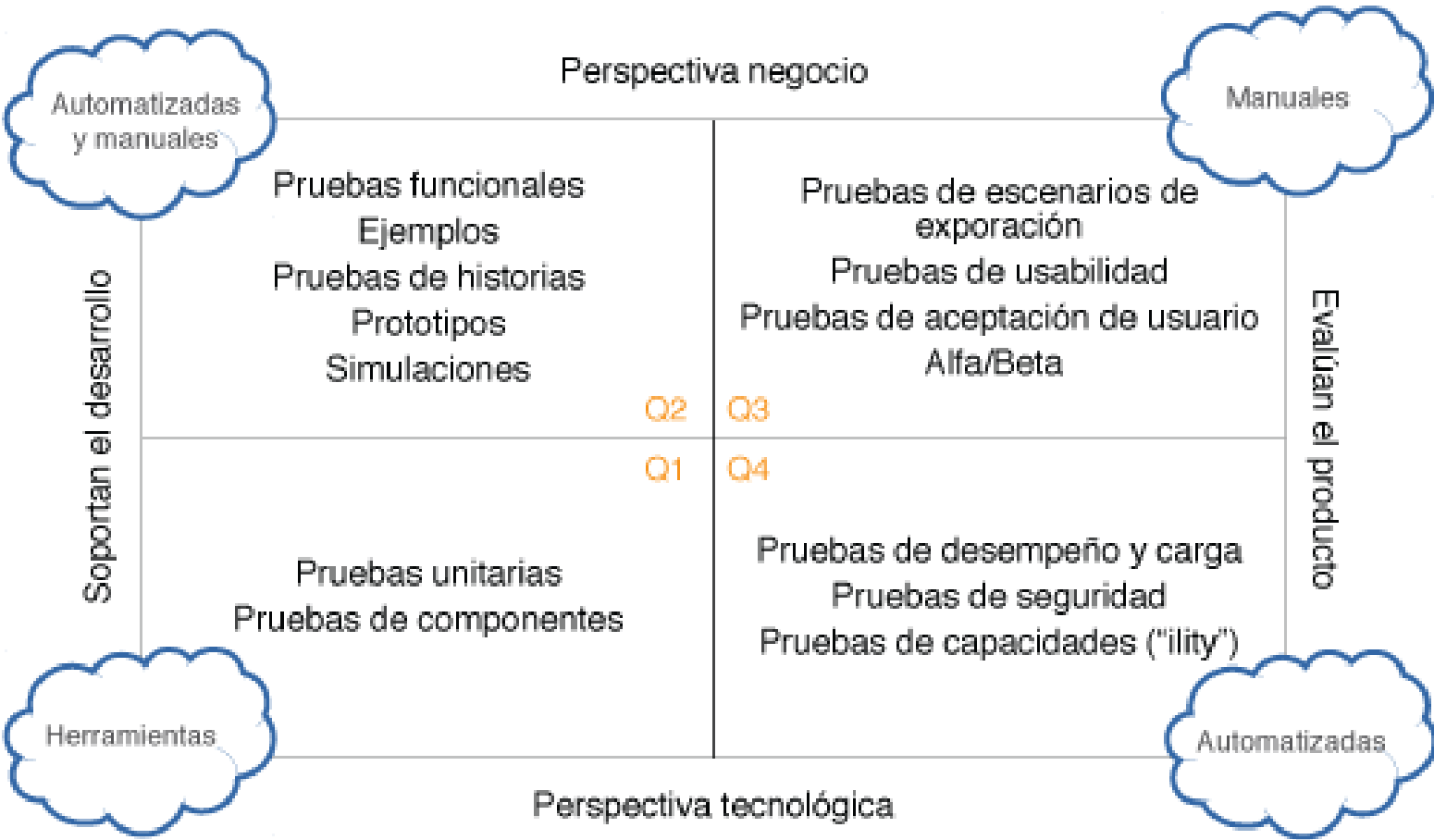
- Q2:
- Nivel de sistema
 - Orientadas al negocio y confirma el comportamiento del negocio.
 - Comprueban los criterios de aceptación
 - Pueden ser manuales o automatizadas
 - Normalmente se crean durante el desarrollo de las HU, mejorando la calidad de las historias
 - Útiles para crear juegos de prueba de regresión automatizadas



3.1 Métodos de testing ágil

Cuadrante de pruebas

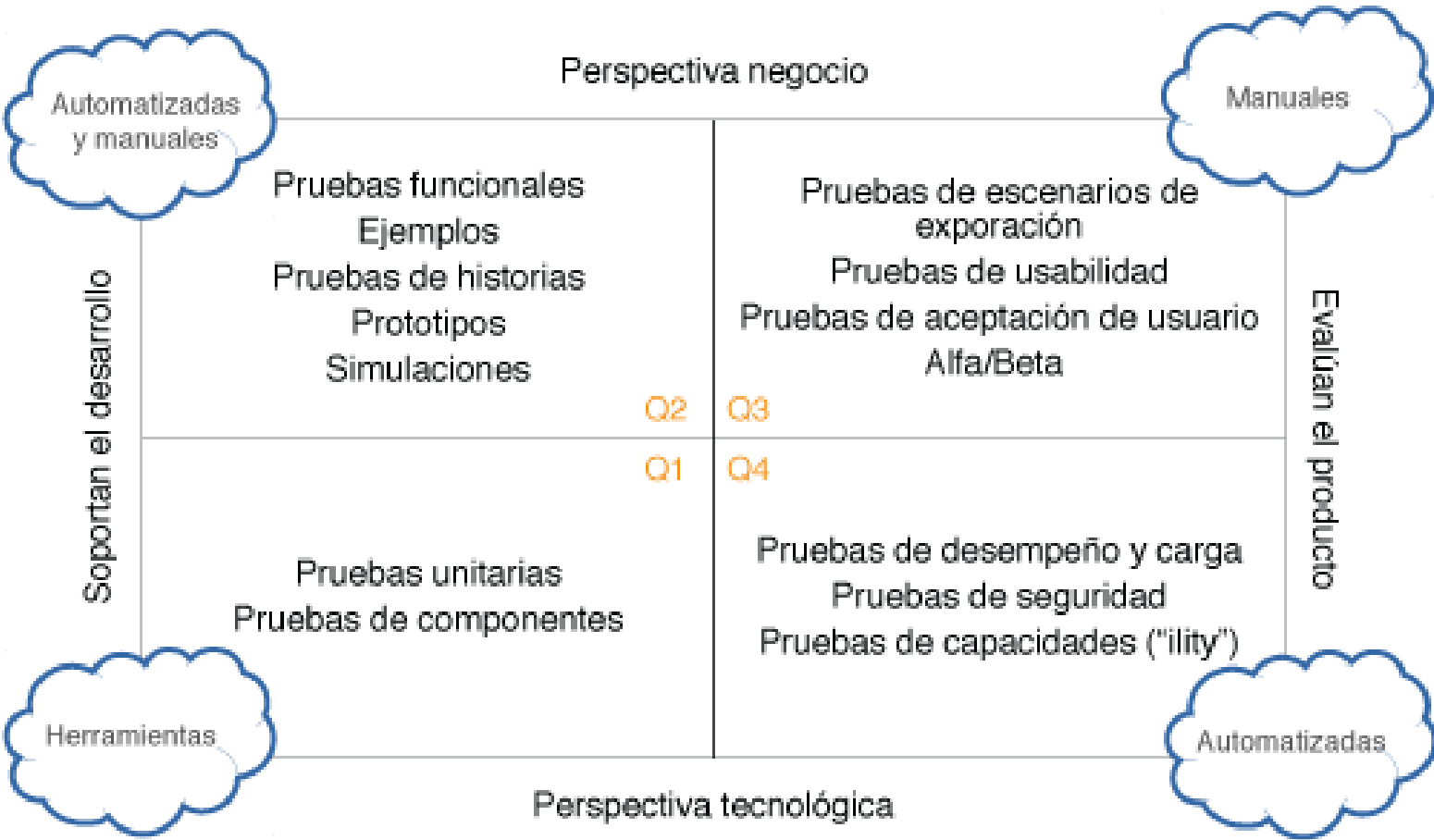
- Q3:
- Nivel de sistema o aceptación usuaria
 - Relativo al negocio
 - Contiene pruebas que evalúan el producto mediante escenarios y datos realistas.
 - Normalmente son manuales
 - Están orientadas al usuario



3.1 Métodos de testing ágil

Cuadrante de pruebas

- Q4:
- Nivel de sistema o aceptación operativa
 - Relativo a la tecnología y contiene pruebas que evalúan el producto.
 - A menudo son automatizadas



Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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

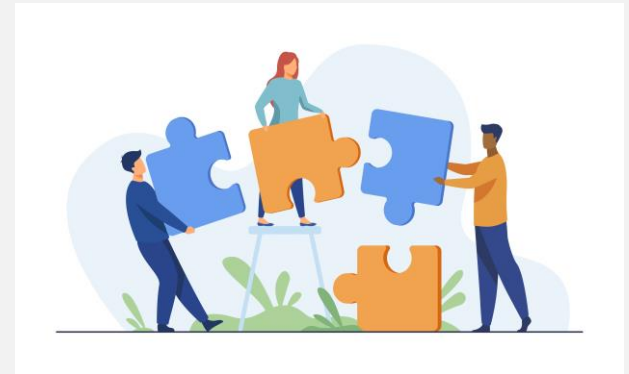


3.1 Métodos de testing ágil

Funciones de probador

Funciones del probador

- En este punto revisaremos las funciones del probador en el principio de trabajo en equipo, etapas y prácticas de pruebas relacionadas a la metodología Scrum.
 1. Principio de Trabajo en equipo
 2. Sprint Cero
 3. Integración
 4. Planificación de pruebas
 5. Rol del probador en distintas prácticas de pruebas ágiles



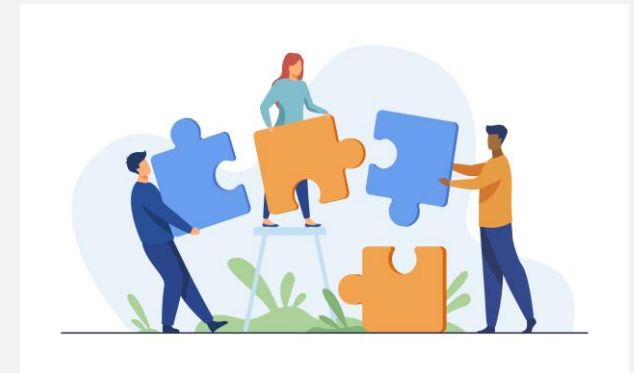
3.1 Métodos de testing ágil

Funciones de probador

1. Rol del probador en el principio de Trabajo en equipo

La colaboración de los desarrolladores, probadores y representantes de las empresas es un principio del fundamento ágil (enfoque de todo el equipo). Las Mejores Prácticas organizativas y de comportamiento en Equipos de Scrum son:

- **Multifuncional:**
 - Todos los miembros del equipo trabajan juntos sobre la estrategia de pruebas, la planificación, la especificación, la ejecución, la evaluación de las pruebas y la comunicación de los resultados de estas.
 - Cada miembro del equipo aporta un conjunto diferente de habilidades para el equipo
- **Auto-organización:**
 - Un equipo ideal incluye testers, no sólo a los desarrolladores

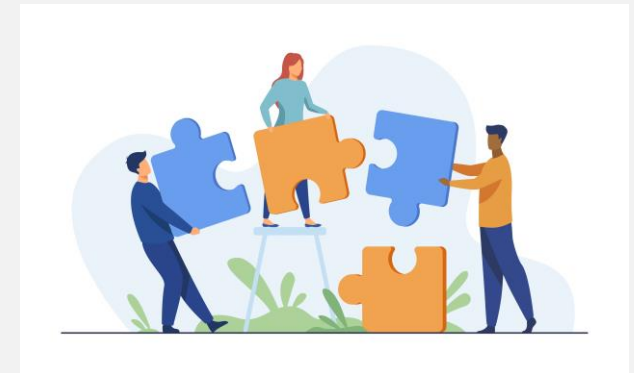


3.1 Métodos de testing ágil

Funciones de probador

Rol del probador en el principio de Trabajo en equipo

- **Co-ubicado:**
 - Probadores , desarrolladores y el dueño del producto se sientan juntos
- **Colaborativo:**
 - Probadores trabajan junto con sus compañeros de equipo , con otros equipos, los grupos de interés, el dueño del producto y el Scrum Master
- **Empowered (Empoderados):**
 - Todas las decisiones técnicas sobre el diseño y las pruebas son hechas por el equipo en su conjunto (desarrolladores, probadores y Scrum Master) en colaboración con el propietario del producto y otros equipos en los casos que sean necesario



3.1 Métodos de testing ágil

Funciones de probador

Rol del probador en el principio de Trabajo en equipo

- **Comprometido:**
 - El probador está comprometido a cuestionar y evaluar el comportamiento y las características del producto con respecto a las expectativas y necesidades de los clientes y usuarios
- **Transparente:**
 - Desarrollo y progreso de las pruebas son visibles en el tablero de tareas ágil
- **Creíble:**
 - Con el fin de ganarse la confianza de las partes interesadas, los probadores deben garantizar la credibilidad de la estrategia de prueba, su aplicación y la ejecución. Este punto se realiza facilitando información a todas las partes interesadas sobre el proceso de prueba.

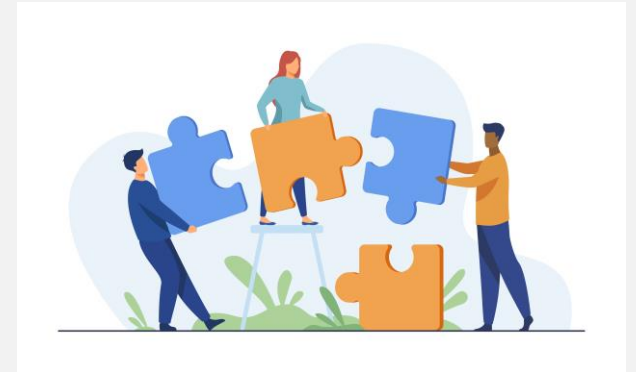


3.1 Métodos de testing ágil

Funciones de probador

Rol del probador en el principio de Trabajo en equipo

- **Abierto a la Feedback:**
 - Las retrospectivas permiten a los equipos a aprender de los éxitos y de los fracasos: las actividades que constituyen un obstáculo para el éxito no son continuos y las que conducen al éxito se intensifican.
- **Flexible:**
 - Las pruebas, como todas las demás actividades en proyectos ágiles, deben ser capaces de responder a los cambios.



Estas buenas prácticas maximizan la probabilidad de éxito en las pruebas Scrum.

3.1 Métodos de testing ágil

Funciones de probador

2. Sprint Cero

- **¿ Qué es el sprint cero?**

Es la primera iteración del proyecto, donde se llevan a cabo las actividades de preparación.

En esta iteración se establece la dirección de lo que tienen que conseguir las pruebas y como tienen que conseguirlo a lo largo de los sprints.



3.1 Métodos de testing ágil

Funciones de probador

¿Qué hace un probador en el sprint Cero?

- **El probador colabora con el equipo en las siguientes actividades:**
 - Identificar el alcance del proyecto (es decir, el backlog del producto)
 - Crear una arquitectura de sistema inicial y prototipos de alto nivel
 - Planificar, obtener e instalar las herramientas necesarias (por ejemplo, para la gestión de pruebas, la gestión de defectos, la automatización de las pruebas y la integración continua)
 - Crear una estrategia de pruebas inicial para todos los niveles de las pruebas, que aborde (entre otros temas), el alcance de las pruebas, los riesgos técnicos, los tipos de pruebas y los objetivos de cobertura



3.1 Métodos de testing ágil

Funciones de probador

¿Qué hace un probador en el sprint Cero?

- Llevar a cabo un análisis inicial de riesgos de calidad
- Definir las métricas de las pruebas para medir el proceso de prueba, el progreso de las pruebas en el proyecto y la calidad del producto
- Especificar la definición de "hecho"
- Crear el tablero de tareas
- Definir cuándo continuar o cuando detener las pruebas antes de entregar el sistema al cliente



3.1 Métodos de testing ágil

Funciones de probador

3. Integración

- En los proyectos ágiles, el objetivo es entregar valor al cliente de una forma continua (preferiblemente en todos los sprints).
- La estrategia de integración debe tener en cuenta tanto el diseño como las pruebas.
- **Para poder implementar una estrategia de pruebas continua en las funcionalidades y características entregadas, es importante definir todas las dependencias entre las funciones y funcionalidades subyacentes.**



3.1 Métodos de testing ágil

Funciones de probador

4. Planificación de pruebas

- La planificación de las pruebas debe empezar durante la sesión de planificación de la entrega y actualizarse durante cada sprint (Iteración).
- La planificación de las pruebas para la entrega y para cada sprint debe abordar los aspectos revisados en el capítulo 1. *
- La planificación del sprint da lugar a un conjunto de tareas a incluir en el tablero de tareas, donde cada tarea debe tener una duración de uno o dos días de trabajo.
- Debe hacerse un seguimiento de todos los problemas referidos a las pruebas para mantener un flujo de pruebas constante .



3.1 Métodos de testing ágil

Funciones de probador

Recordatorio de Memoria

Los tester entregan valor en la planificación de la entrega:

- Definición de historias de usuario testeables, incluyendo criterios de aceptación
- Participación en los análisis de proyecto y análisis de riesgos de calidad
- Estimación del esfuerzo de pruebas asociado a las historias de usuario
- Definición de los niveles de pruebas necesarios
- Planificación de las pruebas para la entrega



3.1 Métodos de testing ágil

Funciones de probador

Recordatorio de Memoria

Los tester entregan valor en la planificación de la iteración:

- Participar en el análisis detallado de riesgos
- Determinar su estabilidad
- Definir pruebas de aceptación
- Dividir historias de usuario en tareas, especialmente tareas de pruebas
- la estimación de esfuerzo de pruebas para todas las tareas de prueba
- la identificación de los aspectos funcionales y no funcionales
- el apoyo a la automatización en múltiples niveles de la prueba



3.1 Métodos de testing ágil

Funciones de probador

5. Rol del probador en distintas prácticas de pruebas ágiles

Algunas prácticas útiles para los probadores en las metodologías ágiles son las siguientes:

- **Por parejas:** Dos miembros del equipo (por ejemplo, un probador y un desarrollador, dos probadores o un probador y un propietario del producto) se sientan juntos para llevar a cabo pruebas u otra tarea del sprint.
- **Diseño de pruebas incrementales:** Los casos de pruebas se construyen gradualmente a partir de historias de usuario y otras bases de pruebas, empezando con pruebas sencillas y pasando a pruebas más complejas.
- **Mapas mentales:** Los mapas mentales son una herramienta útil a la hora de hacer pruebas. Por ejemplo, los probadores pueden usar mapas mentales para identificar qué sesiones de prueba llevar a cabo, para mostrar las estrategias de pruebas y para describir los datos de prueba.

Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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

- Evaluación riesgos de calidad
- Estimación de esfuerzo de prueba

3.3 Técnicas en proyectos ágiles

3.4 Herramientas en un proyecto ágil



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

Riesgos de Calidad y estimación de esfuerzo de prueba

- **Un objetivo común de las pruebas en todos los proyectos, ágiles o tradicionales, es reducir el riesgo de problemas de calidad del producto a nivel aceptable antes de la entrega.**
- Se pueden utilizar el mismo tipo de técnicas que se utilizan en los proyectos tradicionales para identificar los riesgos de calidad , por ejemplo:
 - evaluar el nivel asociado de riesgo
 - estimar el esfuerzo requerido para reducir esos riesgos de manera suficiente
 - mitigar esos riesgos a través del diseño, la implementación y la ejecución de pruebas.
- Debido a la brevedad de las iteraciones y del ritmo de los cambios en proyectos ágiles, se requieren adaptaciones a estas técnicas.

Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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

- Evaluación riesgos de calidad
- Estimación de esfuerzo de prueba

3.3 Técnicas en proyectos ágiles

3.4 Herramientas en un proyecto ágil





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

Evaluación riesgos de calidad en proyectos ágiles

- **Uno de los retos de pruebas es la correcta selección, asignación y priorización de las condiciones de prueba. Esto incluye:**
 - Determinar el esfuerzo a dedicar para cubrir cada condición con pruebas
 - Secuencias las pruebas con fin de optimizar la eficacia y eficiencia del trabajo de pruebas a realizar.
- En base a la evaluación de riesgos de calidad, los probadores deben definir un número aceptable de casos de prueba a ejecutar.

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

Evaluación riesgos de calidad en proyectos ágiles

- **¿Qué es el riesgo?**

El riesgo es la posibilidad de un resultado o evento, negativo o no deseado

- **¿Qué es el nivel de riesgo?**

El nivel de riesgo se establece evaluando la probabilidad de que el riesgo suceda y su impacto

- **¿Qué es el riesgo de calidad o de producto?**

Cuando el efecto principal del problema potencial se refiere a la calidad del producto

- **¿Qué es el riesgo de proyecto o de planificación?**

Cuando el efecto principal del problema potencial se refiere al éxito del proyecto

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

Evaluación riesgos de calidad en proyectos ágiles

En los proyectos ágiles, el análisis de riesgos de calidad se lleva a cabo en dos momentos.

- **Planificación de la entrega:**
 - los representantes de negocio que conocen las funcionalidades en la entrega proporcionan una perspectiva de alto nivel de los riesgos, y todo el equipo, incluidos los probadores, pueden ayudar a identificar y evaluar los riesgos.
- **Planificación de la iteración:**
 - todo el equipo identifica y evalúa los riesgos de calidad.

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

Evaluación riesgos de calidad en proyectos ágiles

Ejemplos riesgos de calidad:

- Cálculos incorrectos en informes (un riesgo funcional asociado a la exactitud)
- Respuesta lenta a entradas del usuario (un riesgo no funcional asociado a la eficiencia y al tiempo de respuesta)
- Dificultad para comprender pantallas y campos (un riesgo no funcional asociado a la usabilidad y a la facilidad de comprensión).

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

Evaluación riesgos de calidad en proyectos ágiles

¿Cómo deben enfrentarse los riesgos?

- Una iteración empieza con la planificación de la iteración, que culmina en tareas estimadas en un tablero de tareas.
- Estas tareas pueden priorizarse en parte en base al nivel de los riesgos de calidad asociados a las mismas.
- Las tareas asociadas a riesgos más altos deben empezar antes y suponer un esfuerzo de prueba mayor.
- Las tareas asociadas a riesgos más bajos deben empezar más tarde y suponer un esfuerzo de prueba menor.

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

Evaluación riesgos de calidad en proyectos ágiles

Ejemplo en una planificación de iteración:

1. Reunir a los miembros del equipo, incluidos los probadores.
2. Enumerar todos los elementos del backlog para la iteración actual (por ejemplo, en un tablero de tareas).
3. Identificar los riesgos de calidad asociados a cada elemento, teniendo en cuenta todas las características de calidad relevantes.
4. Evaluar cada riesgo identificado, lo que incluye dos actividades: categorizar el riesgo y determinar su nivel de riesgo en base al impacto y a la probabilidad de los defectos.
5. Determinar el alcance de las pruebas de manera proporcional al nivel de riesgo.
6. Seleccionar la técnica o técnicas de pruebas adecuadas para mitigar cada riesgo, en función del nivel de riesgo y de la característica de calidad relevante.

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

Evaluación riesgos de calidad en proyectos ágiles

Con el análisis de riesgo ya realizado en la planificación de la iteración:

- Tenemos que diseñar, implementar y ejecutar pruebas para mitigar los riesgos.
 - Esto debe incluir todas las funcionalidades, comportamientos, características de calidad y atributos que afectan a la satisfacción del cliente, del usuario y de las partes interesadas.
-
- A lo largo del proyecto, el equipo debe conocer cualquier información adicional que pueda alterar el conjunto de riesgos y/o el nivel de riesgo asociado a riesgos de calidad conocidos.
 - Debe realizarse un ajuste periódico de análisis de calidad y ajustes a las pruebas.
 - Los riesgos de calidad pueden mitigarse antes de empezar a ejecutar las pruebas. (ejemplo en historias de usuarios)

Capítulo 3:

Métodos, técnicas y herramientas de pruebas ágiles

3.1 Métodos de testing ágil

- Desarrollo Basado en Pruebas TDD, Desarrollo Basado en Pruebas de Aceptación
- La pirámide de pruebas
- Los cuadrantes del testing, niveles de pruebas y tipos de pruebas
- El rol del tester

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

- Evaluación riesgos de calidad
- Estimación de esfuerzo de prueba

3.3 Técnicas en proyectos ágiles

3.4 Herramientas en un proyecto ágil



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

Estimación del esfuerzo de prueba

Estimación del esfuerzo de prueba en base al contenido y al riesgo

- Durante la planificación de la entrega, el equipo ágil estima el esfuerzo necesario para completar la entrega.
- Esta estimación aborda también el esfuerzo de las pruebas .
- Técnica más utilizada, es la estimación de poker.
 - Pueden ocupar como medida número de puntos de historia, los días de esfuerzo o cualquier otra unidad con la que el equipo realice los cálculos.

¡OJO! Una estimación alta normalmente significa que la historia no se ha entendido bien o debería dividirse en varias historias más pequeñas.



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

Estimación del esfuerzo de prueba

Estimación del esfuerzo de prueba en base al contenido y al riesgo

- Normalmente se discute en la planificación del backlog el esfuerzo en base a:
 - Esfuerzo de desarrollo
 - Esfuerzo de pruebas
 - Complejidad de la historia
 - Alcance de las pruebas

Pero se debe incluir el nivel de riesgo + la prioridad indicada por el PO antes de iniciar la estimación de poquer.



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

Estimación del esfuerzo de prueba

Estimación del esfuerzo de prueba en base al contenido y al riesgo

Ejemplo de una secuencia de actividades para una sesión de planificación de póker :

1. Historia de usuario es explicada por el dueño del producto o cliente
2. Todos los miembros del equipo presentes escogen una tarjeta con un valor de estimación y mantienen la tarjeta cubierta
3. Las tarjetas se muestran simultáneamente y se comparan los valores de la estimación
4. Cuando los valores de la estimación son iguales, el valor se convierte en la estimación
5. Cuando los valores de la estimación son desiguales, se inicia una discusión, por lo general los que tienen la más alta y la más baja darán su razonamiento
6. Luego se “juega” otra ronda hasta que se encuentre un consenso



El consentimiento puede ser alcanzado de manera diferente :

- El equipo llega a un consenso o está cerca de llegar
- Uso de la promedios, mediana, valor más alto , etc.



Curso ISTQB® Foundation LEVEL AGILE TESTER

