



# Curso ISTQB® Foundation LEVEL AGILE TESTER

Sep-Oct 2023



# Capítulo 1:

## Desarrollo ágil de software

### 1.1 Fundamentos del desarrollo ágil de software

- Desarrollo ágil de software y el Manifiesto Ágil
- Enfoque de equipo completo
- Feedback temprano y frecuente

### 1.2 Aspectos de enfoques ágiles

- Enfoques de desarrollo ágil de software
- Creación de historias de usuario colaborativas
- Retrospectivas
- Integración continua
- Planificación de la entrega y planificación de la iteración



# 1.1 Desarrollo ágil de software- Fundamentos

## *Fundamentos del desarrollo ágil*

- No se trabaja de la misma manera en un proyecto tradicional que en uno ágil.
- Los probadores deben entender los valores y principios que sustentan los proyectos ágiles
- Los probadores deben entender que forman parte de un enfoque de equipo completo junto con los desarrolladores y representantes del negocio.



# Capítulo 1:

## Desarrollo ágil de software

### 1.1 Fundamentos del desarrollo ágil de software

- Desarrollo ágil de software y el Manifiesto Ágil
- Enfoque de equipo completo
- Feedback temprano y frecuente

### 1.2 Aspectos de enfoques ágiles

- Enfoques de desarrollo ágil de software
- Creación de historias de usuario colaborativas
- Retrospectivas
- Integración continua
- Planificación de la entrega y planificación de la iteración



## 1.1 Desarrollo ágil de software- Fundamentos

### *Desarrollo ágil de software y el Manifiesto ágil*

*Los miembros de un proyecto ágil se comunican entre sí de forma temprana y con frecuencia, que ayuda a eliminar defectos lo antes posible y a desarrollar un producto de calidad.*



# 1.1 Desarrollo ágil de software- Fundamentos

## *Valores del Manifiesto ágil (1/5)*

**Individuos e interacciones** —————> por encima de —————> **procesos y herramientas.**  
**Software funcionando** —————> por encima de —————> **documentación extensiva.**  
**Colaboración con el cliente** —————> por encima de —————> **negociación contractual.**  
**Respuesta ante el cambio** —————> por encima de —————> **seguir un plan.**

\* Si bien los conceptos de la derecha tienen valor lo de la izquierda tienen un valor mayor en la metodología ágil



# 1.1 Desarrollo ágil de software- Fundamentos

## ***Valores del Manifiesto ágil (2/5)***

### **Individuos e Interacciones**

- El desarrollo ágil es centrado en las personas
- Los equipos de personas construyen un software, y la forma en la que los equipos pueden trabajar con la máxima efectividad es a través de la comunicación e interacción continua, más que basándose en herramientas o procesos.
- El equipo esta conformado por un numero de personas con diferentes tareas
- La interacción es cercana y frecuente
- Es mas efectivo confiar en la colaboración entre los miembros del equipo que en herramientas y procesos



# 1.1 Desarrollo ágil de software- Fundamentos

## ***Valores del Manifiesto ágil (3/5)***

### **Software Que Funcione**

- Desde el punto de vista del cliente, un software que funciona es mucho más útil y valioso que una documentación excesivamente detallada. (Documentación detallada toma un lugar secundario)
- Un software que funciona, aun con una funcionalidad reducida, está disponible mucho antes en el ciclo de vida del Desarrollo, ofreciendo una ventaja en termino de time to market. El desarrollo ágil crea una ventaja considerable en el tiempo de lanzamiento del software
- Un software que funciona permite una retroalimentación rápida
- El Desarrollo ágil es útil en entornos de negocio que cambian con rapidez donde los problemas y/o soluciones no están claros, o donde el negocio desea innovar en nuevas áreas.





# 1.1 Desarrollo ágil de software- Fundamentos

## *Valores del Manifiesto ágil (4/5)*

### Colaboración con el cliente

- A los clientes les resulta difícil describir los requisitos para un sistema o entender que es lo que necesitan.
- Colaborando directamente con el cliente se mejora la probabilidad de entender exactamente que necesita el cliente y sus expectativas
- Los contratos con los clientes pueden ser importantes , pero trabajar regularmente y en colaboración es probable que traiga más éxito
- La colaboración frecuente y cercana con el cliente aumenta la posibilidad de éxito del proyecto



# 1.1 Desarrollo ágil de software- Fundamentos

## ***Valores del Manifiesto ágil (5/5)***

### **Respuesta ante el cambio**

- Los proyectos de software están sujetos a cambios, y es inevitable que esto ocurra.
- El entorno en el que opera el negocio, la legislación, la actividad de los competidores, los avances tecnológicos y otros factores pueden afectar significativamente el proyecto.
- El proceso de desarrollo tiene que responder a estos cambios, por lo que tener flexibilidad en las prácticas de trabajo para responder ante el cambio es más importante que limitarse a cumplir estrictamente a un plan.



# 1.1 Desarrollo ágil de software- Fundamentos

## *Principios del desarrollo ágil (1/4)*

### Principio 1

Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.

### Principio 2

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar venta competitiva.

### Principio 3

Entregamos software que funciona frecuentemente, entre dos semanas y dos meses, preferentemente el periodo de tiempo más corto posible.

### Principio 4

Los responsables del negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto

# 1.1 Desarrollo ágil de software- Fundamentos

## *Principios del desarrollo ágil (2/4)*

### Principio 5

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

### Principio 6

El método más eficiente y efectivo de comunicar la información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

### Principio 7

El software funcionando es la medida principal de progreso

### Principio 8

Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

## 1.1 Desarrollo ágil de software

### *Principios del desarrollo ágil (3/4)*

#### Principio 9

La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.

#### Principio 10

La simplicidad, o el arte de maximizar la cantidad de trabajo que no es necesario realizar, es esencial.

#### Principio 11

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-gestionados.

#### Principio 12

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para después ajustar y perfeccionar su comportamiento.



## 1.1 Desarrollo ágil de software- Fundamentos

### *Principios del desarrollo ágil (4/4)*

**Los diferentes métodos ágiles proveen descripciones específicas de como implementar estos principios.**



# Capítulo 1:

## Desarrollo ágil de software

### 1.1 Fundamentos del desarrollo ágil de software

- Desarrollo ágil de software y el Manifiesto Ágil
- Enfoque de equipo completo
- Feedback temprano y frecuente

### 1.2 Aspectos de enfoques ágiles

- Enfoques de desarrollo ágil de software
- Creación de historias de usuario colaborativas
- Retrospectivas
- Integración continua
- Planificación de la entrega y planificación de la iteración





## 1.1 Desarrollo ágil de software- Fundamentos

### ***Enfoque de equipo completo (1/3)***

#### **Un equipo interdisciplinario y auto organizado**

- El enfoque de equipo completo significa involucrar a todos quienes tengan el conocimiento y habilidades necesarias para asegurar el éxito del proyecto
- El equipo incluye representantes del cliente y de los interesados quienes definen las características del proyecto
- Equipos exitosos están compuestos entre tres y nueve personas, equipos relativamente pequeños
- Idealmente, todo el equipo comparte el mismo espacio de trabajo, dado que la co-ubicación facilita la comunicación y la interacción.
- Se realizan reuniones diarias involucrando a todos los miembros del equipo y son el lugar donde se comunica el progreso del trabajo y los obstáculos que bloquean el progreso.
- El enfoque de equipo completo promueve una dinámica de equipo más efectiva y eficiente.
- Todo el equipo es responsable de la calidad del proyecto
- La esencia de este enfoque, reside en los probadores, desarrolladores y representantes de negocios trabajan juntos en cada etapa.



## 1.1 Desarrollo ágil de software- Fundamentos

### ***Enfoque de equipo completo (2/3)***

#### **Un equipo interdisciplinario y auto organizado**

- Los probadores dan soporte y colaboran con los representantes del negocio para crear pruebas de aceptación adecuadas.
- Los probadores trabajan con los desarrolladores para establecer en conjunto la estrategia de pruebas y decidir sobre los enfoques de automatización de pruebas.
- Los probadores deben transferir y extender el conocimiento de pruebas a otros miembros del equipo e influenciar en el desarrollo del producto.
- Todo el equipo participa en las consultas o reuniones relativas a la presentación, el análisis o la estimación de las prestaciones del producto.
- “El Poder de tres” es el concepto de implicar a los probadores, desarrolladores y representantes del negocio en todas discusiones. (Cristpin08)

# 1.1 Desarrollo ágil de software- Fundamentos

## ***Enfoque de equipo completo (3/3)***

### **Ventajas de enfoque equipo completo**

- Mejora la comunicación y colaboración dentro del equipo
- Permite aprovechar y desarrollar las habilidades del equipo en beneficio del proyecto
- Hacer que la calidad sea de todos



# Capítulo 1:

## Desarrollo ágil de software

### 1.1 Fundamentos del desarrollo ágil de software

- Desarrollo ágil de software y el Manifiesto Ágil
- Enfoque de equipo completo
- Feedback temprano y frecuente

### 1.2 Aspectos de enfoques ágiles

- Enfoques de desarrollo ágil de software
- Creación de historias de usuario colaborativas
- Retrospectivas
- Integración continua
- Planificación de la entrega y planificación de la iteración



# 1.1 Desarrollo ágil de software- Fundamentos

## ***Feedback temprano y frecuente (1/3)***

Los proyectos ágiles tienen iteraciones cortas que permiten al equipo del proyecto recibir feedback temprano y de manera continua sobre la calidad del producto, durante todo el ciclo de desarrollo. Una forma de ofrecer esto es con Integración continua.

### Desarrollo Tradicional

- Desarrollo secuencial
- La entrega tiene lugar después de la finalización, al final del proceso de desarrollo
- No existe una retroalimentación continua
- Las necesidades del cliente puede que no sean entendidos
- El desarrollo se completa en un bloque
- Procedimientos fijos



### Desarrollo Ágil

- Iteraciones cortas
- Retroalimentación temprana y continua
- Monitoreo constante de la calidad del producto
- Retroalimentación frecuente de cliente
- Características importantes primero
- Evaluación de los sprints para aumentar la velocidad y de mejora (Mejora continua)





## 1.1 Desarrollo ágil de software- Fundamentos

### ***Feedback temprano y frecuente (2/3)***

- Un punto importante del feedback frecuente del cliente es que a medida que el proyecto avanza, los equipos ágiles pueden incorporar la mayoría de los nuevos cambios al proceso de desarrollo del producto.
- El feedback temprano y frecuente ayuda al equipo a concentrarse en las prestaciones que tienen mayor valor de negocio o mayor riesgo asociado, y estas se entregan al cliente en el primer lugar.
- Ayuda a gestionar mejor al equipo, ya que la capacidad del equipo es transparente para todos.



## 1.1 Desarrollo ágil de software- Fundamentos

### ***Feedback temprano y frecuente***

#### **Ventajas**

- Evitar malentendidos de requisitos, que posiblemente no se hubieran detectado hasta más avanzado el ciclo de desarrollo cuando su reparación sería más cara.
- Aclarar requisitos de las necesidades del cliente, haciendo que estén disponibles cuanto antes para su uso por parte del cliente. De esta forma, el producto refleja mejor lo que quiere el cliente.
- Descubrir (mediante la integración continua), aislar y resolver problemas de calidad de forma temprana.
- Ofrecer información al equipo ágil sobre su productividad y capacidad para realizar entregas.
- Promover un impulso consistente dentro del proyecto.



## 1.1 Desarrollo ágil de software- Fundamentos

### ***Feedback temprano y frecuente***

#### **Ventajas**

- Evitar malentendidos de requisitos, que posiblemente no se hubieran detectado hasta más avanzado el ciclo de desarrollo cuando su reparación sería más cara.
- Aclarar requisitos de las necesidades del cliente, haciendo que estén disponibles cuanto antes para su uso por parte del cliente. De esta forma, el producto refleja mejor lo que quiere el cliente.
- Descubrir (mediante la integración continua), aislar y resolver problemas de calidad de forma temprana.
- Ofrecer información al equipo ágil sobre su productividad y capacidad para realizar entregas.
- Promover un impulso consistente dentro del proyecto.

# Capítulo 1:

## Desarrollo ágil de software

### 1.1 Fundamentos del desarrollo ágil de software - EJERCICIOS

- Desarrollo ágil de software y el Manifiesto Ágil
- Enfoque de equipo completo
- Feedback temprano y frecuente

### 1.2 Aspectos de enfoques ágiles

- Enfoques de desarrollo ágil de software
- Creación de historias de usuario colaborativas
- Retrospectivas
- Integración continua
- Planificación de la entrega y planificación de la iteración





## 1.1 Desarrollo ágil de software- Fundamentos

### *EJERCICIO 1*

De acuerdo con la declaración de valores del Manifiesto Agile, en lugar de depender de herramientas y procesos, los proyectos de Agile valoran cuál de lo siguiente:

1. La presencia de un contrato de trabajo claro acordado con el cliente antes de iniciar el desarrollo
2. Comunicación e interacción entre los miembros del equipo en todas las etapas del ciclo de desarrollo
3. La creación de documentación de soporte de usuario durante cada iteración de desarrollo
4. Tener un plan detallado antes de cada release del sistema que se sigue de cerca durante la liberación de la versión



## 1.1 Desarrollo ágil de software- Fundamentos

### *EJERCICIO 2*

Según el Manifiesto Agile, ¿Cuál de las siguientes afirmaciones se valora más que la "documentación completa"?

1. Planes simples
2. Historias de usuario
3. Software trabajando
4. Compilaciones automatizadas





## 1.1 Desarrollo ágil de software- Fundamentos

### *EJERCICIO 3*

El desarrollo ágil fomenta un enfoque de todo el equipo para el desarrollo de software. Con respecto al enfoque de todo el equipo, ¿Cuáles de las siguientes afirmaciones son verdaderas y cuáles son falsas?

- I. La experiencia de los probadores significa que tienen la responsabilidad última de la calidad.
- II. Se debe alentar a los desarrolladores a hablar directamente con los representantes de negocios cuando no tengan claro los requisitos.
- III. Los probadores deben consultar a los desarrolladores al escribir pruebas automatizadas.
- IV. Los representantes empresariales deben escribir pruebas de aceptación independientemente del resto del equipo.
- V. Los representantes de negocios no deben tener ningún papel en las discusiones en las que el equipo de desarrollo está estimando el tamaño de las historias de usuario.

- 1. i) iii) y iv) son verdaderos; (ii) y (v) son falsos
- 2. ii) y iii) son verdaderos; (i), (iv) y (v) son falsos
- 3. (iv) y (v) son verdaderos; (i), (ii) y (iii) son falsos
- 4. (ii), (iv) y (v) son verdaderos; (i) y (iii) son falsos



## 1.1 Desarrollo ágil de software- Fundamentos

### *EJERCICIO 3*

El desarrollo ágil fomenta un enfoque de todo el equipo para el desarrollo de software. Con respecto al enfoque de todo el equipo, ¿Cuáles de las siguientes afirmaciones son verdaderas y cuáles son falsas?

- I. La experiencia de los probadores significa que tienen la responsabilidad última de la calidad.
- II. Se debe alentar a los desarrolladores a hablar directamente con los representantes de negocios cuando no tengan claro los requisitos.
- III. Los probadores deben consultar a los desarrolladores al escribir pruebas automatizadas.
- IV. Los representantes empresariales deben escribir pruebas de aceptación independientemente del resto del equipo.
- V. Los representantes de negocios no deben tener ningún papel en las discusiones en las que el equipo de desarrollo está estimando el tamaño de las historias de usuario.

- 1. i) iii) y iv) son verdaderos; (ii) y (v) son falsos
- 2. ii) y iii) son verdaderos; (i), (iv) y (v) son falsos
- 3. (iv) y (v) son verdaderos; (i), (ii) y (iii) son falsos
- 4. (ii), (iv) y (v) son verdaderos; (i) y (iii) son falsos



## 1.1 Desarrollo ágil de software- Fundamentos

### *EJERCICIO 4*

¿Cuál de las siguientes opciones es una ventaja de tener a todo el equipo responsable de la calidad?

1. Los probadores ahora son responsables de la calidad en todo el proyecto, por lo que los desarrolladores pueden concentrarse en escribir código
2. Los desarrolladores son responsables de escribir pruebas automatizadas y mantenerlas actualizadas en el servidor de compilación de integración continua para que los probadores puedan centrarse en las actividades de pruebas exploratorias
3. Todos los miembros del equipo son expertos en todos los roles en el equipo de Agile, por lo que cualquier otro miembro del equipo puede asumir fácilmente el trabajo de otro miembro del equipo si están enfermos.
4. Los límites del rol se rompen y los miembros del equipo contribuyen al éxito del proyecto aprovechando sus conjuntos de habilidades y perspectivas únicos



## 1.1 Desarrollo ágil de software- Fundamentos

### *EJERCICIO 5*

¿Cuál de las siguientes OPCIONES describe un beneficio asociado con la retroalimentación temprana?

1. Los clientes no tienen que escribir pruebas de aceptación, debido a que constantemente están proporcionando retroalimentación al equipo de desarrollo
2. Probadores pasan menos tiempo escribiendo pruebas generales, debido a que están en contacto regular con los clientes
3. La densidad de defecto aumenta porque los desarrolladores entienden mejor lo que el cliente quiere
4. Mediante la integración continua, los problemas de calidad se pueden detectar antes cuando son más fáciles de solucionar



## 1.1 Desarrollo ágil de software- Fundamentos

### *EJERCICIO 6*

¿Cuáles de las siguientes afirmaciones son verdaderas?

- I. La retroalimentación temprana da a los desarrolladores más tiempo para desarrollar características del sistema.
- II. La retroalimentación temprana da al equipo la confianza de que los requisitos son correctos.
- III. La retroalimentación temprana reduce la cantidad de tiempo necesario para las pruebas del sistema.
- IV. La retroalimentación temprana reduce el re-trabajo en iteraciones posteriores del proyecto.

- 1. i) y iv) son verdaderos; (ii) y (iii) son falsos
- 2. ii) y iii) son verdaderos; (i) y (iv) son falsos
- 3. (ii) y (iv) son verdaderos; (i) y (iii) son falsos.
- 4. (i) y (iii) son verdaderos; (ii) y (iv) son falsos



## 1.1 Desarrollo ágil de software- Fundamentos

---

### *Respuestas*

*P1. 2*

*P2. 3*

*P3. 2*

*P4.4*

*P5. 4*

*P6. 3*





# Curso ISTQB® Foundation LEVEL AGILE TESTER

Sep-Oct 2023

