



# Curso ISTQB® Foundation LEVEL AGILE TESTER



# 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





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **Aspectos de enfoques ágiles**

- Hay varios enfoques o metodologías ágiles que se están usando en las organizaciones.

La mayoría de los enfoques incluyen lo siguiente:

- Creación de historias de usuarios colaborativas
- Retrospectivas
- Integración continua
- Planificación de cada iteración y de entrega



# 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





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### *Aspectos de enfoques ágiles*

Hay varios enfoques de desarrollo ágil de software, cada uno implementa los valores y principios del Manifiesto Ágil de forma distinta.

Los enfoques que revisaremos son:

- Programación extrema (XP)
- Scrum
- Kanban

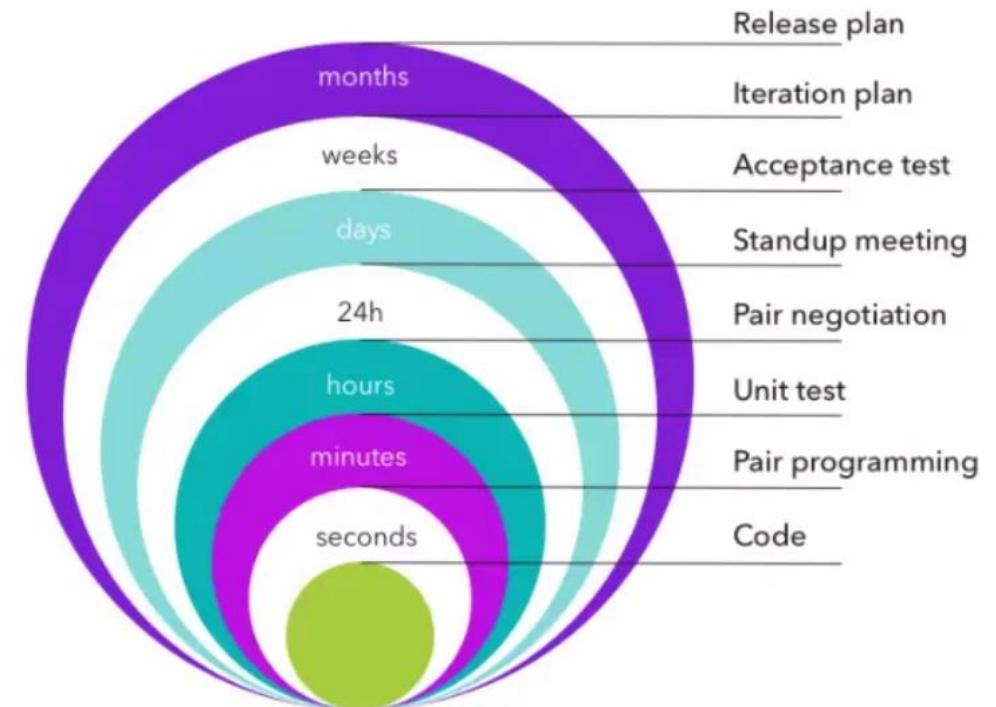




## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### ***Enfoque 1: Programación Extrema (XP)***

- Extreme Programming o XP es una metodología de desarrollo de software
- Se basa en un conjunto de prácticas útiles y técnicas probadas para implementar código de forma que su diseño, arquitectura y codificación permitan incorporar modificaciones y nuevas funcionalidades sin demasiado impacto a la calidad del mismo.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### *Programación Extrema (XP)- Quién lo creó?*

**Kent Beck** Es ingeniero de software estadounidense, creador de

- programación extrema (eXtreme Programming o XP) y
- el desarrollo guiado por pruebas (Test-Driven Development o TDD)
- framework de pruebas unitarias para Java conocido como JUnit.

Beck fue uno de los 17 firmantes originales del Manifiesto Ágil en 2001.

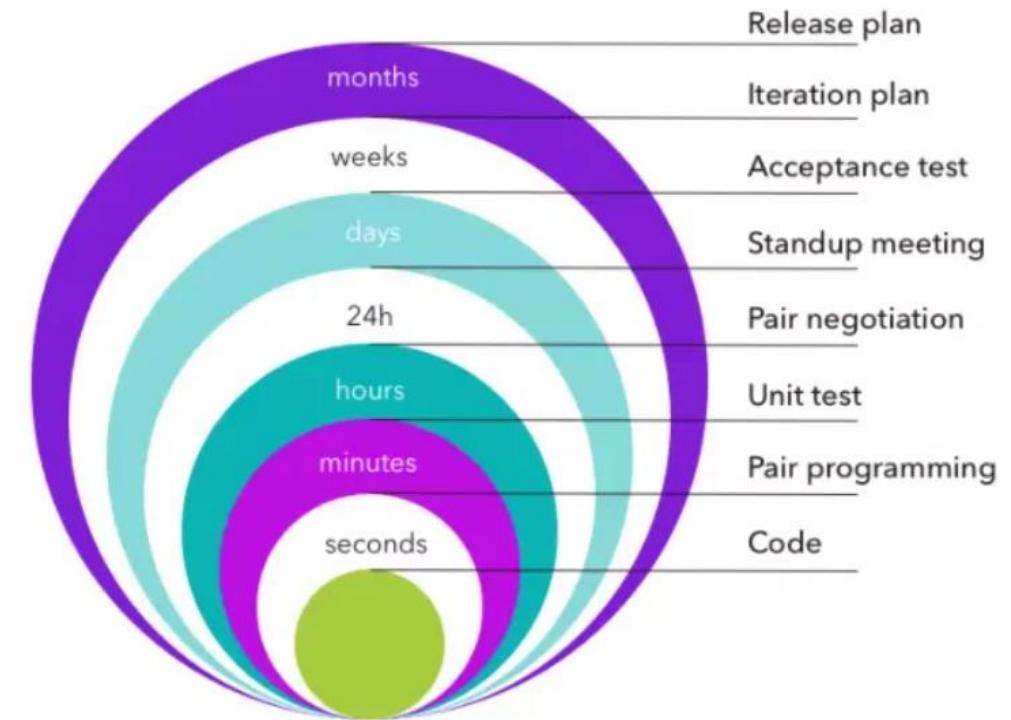
Kent vive en Medford, Oregon y actualmente trabaja para Facebook





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

- Esta metodología busca:
  - Reducir el riesgo del proyecto
  - Mejorar la respuesta ante cambios en el negocio
  - Mejorar la productividad a lo largo de toda la vida del software
  - Los equipos XP producen software de calidad a un ritmo constante y sostenible





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

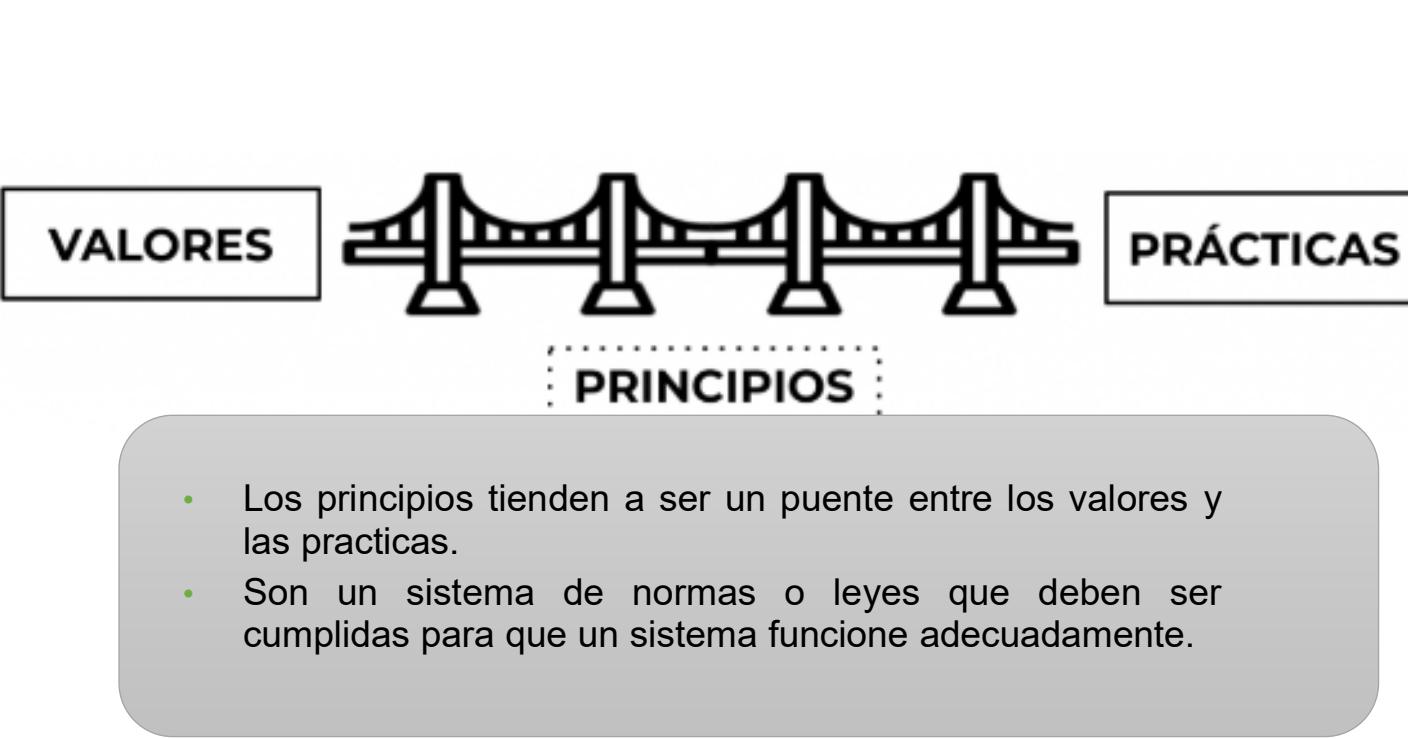
- Se caracteriza por:
  - Funciona con equipos de cualquier tamaño. Los valores y principios detrás de XP son aplicables a cualquier escala.
  - Se adapta a requisitos vagos o que cambian rápidamente. Pero también se puede aplicar a proyectos que los requisitos no son tan cambiantes.
  - Sus cortos ciclos de desarrollo dan como resultado feedback temprano, concretos y continuos.
  - Su aproximación a la planificación incremental.
  - Capacidad para programar de manera flexible la implementación de funcionalidades, respondiendo a las necesidades cambiantes del negocio.
  - Dependencia a pruebas automatizadas, desarrolladas con TDD (Desarrollo guiado por pruebas), para monitorear el progreso del desarrollo y permitir que el sistema evolucione detectando defectos lo más pronto posible.
  - Se basa en diseño evolutivo
  - Depende de la estrecha colaboración de personas activamente comprometidas con talento y excelencia.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

- Extreme Programming o XP tiene un enfoque determinado por valores, principio y prácticas.

- Los valores son criterios a gran escala que utilizamos para juzgar lo que vemos, pensamos y hacemos.
- Los valores aportan un propósito a las prácticas y deben ser explícitas para estas no se vuelvan rutinarias



- Las prácticas son las actividades que se realizan dentro del equipo de forma continuada en el día a día conforme a unas reglas claras y objetivas.
- Su ejecución da como resultado adquisición de nuevas habilidades y experiencias
- Las prácticas son la evidencia de los valores y pueden cambiar dependiendo de la situación.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **XP promueve 5 valores para guiar el desarrollo**

#### 1. Comunicación

- La comunicación es el valor más importante dentro de un equipo de desarrollo.
- Aunque algunos **problemas** son causados por falta de conocimiento, la mayoría surgen por **falta de comunicación**.
- Por lo tanto, cada uno es parte del equipo y **se comunican cara a cara** en la Daily.
- **Se trabajará juntos en todo**, desde la definición de los requisitos hasta la codificación y **se creará la mejor solución al problema entre todos**.

#### 2. Simplicidad

- La simplicidad es la esencia de XP, es el más intelectual de los valores de XP.
- Nos centraremos en hacer lo que se necesita y se nos pide, pero nada más. Esto maximizará el valor creado para la inversión hecha hasta la fecha.
- Se tomarán pequeños pasos que nos hagan avanzar a nuestra meta y se mitigarán los fallos según se den.
- **Crearemos algo de lo que estemos orgullosos de mantener a largo plazo a un coste razonable.**



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **XP promueve 5 valores para guiar el desarrollo**

#### 3. Feedback

- Feedback es una parte crítica de la comunicación.
- Los cambios son inevitables, estos generan la **necesidad de obtener un feedback**. Por lo que usaremos este feedback **para acercarnos más y más a nuestros objetivos** y en consecuencia los equipos de XP se esforzarán por obtener la mayor cantidad de feedback que puedan manejar lo más rápido posible.

#### 4. Valentía

- La valentía es la fuerza de voluntad que puede desarrollar una persona, sin miedo al fracaso, **para superar ciertos impedimentos** y que a veces se manifiesta como la tendencia a la acción.
- La valentía como valor sólo es peligroso, pero junto con otros se hace poderoso.
- En XP se emplea para decir la verdad sobre el progreso y las estimaciones.
- No se teme a nada porque nunca se trabaja solo.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **XP promueve 5 valores para guiar el desarrollo**

#### 5. Respeto

- Si los miembros de un equipo no se preocupan por el resto ni el trabajo que hacen, XP no funcionará.
- Cada uno da y siente el respeto que ellos se merecen como un miembro valorado del equipo
- Cada uno aporta valor incluso si es simple entusiasmo.
- Los desarrolladores respetan la experiencia de los clientes y viceversa, la dirección respeta nuestro derecho a aceptar responsabilidad y recibe autoridad sobre nuestro propio trabajo.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### ***XP describe 14 principios como directrices adicionales***

- **Humanidad:** El software lo desarrollan las personas y es importante tener presente que los factores humanos son la clave para crear un software de calidad.
- **Economía:** El producto que se cree debe ser rentable tanto a corto como a largo plazo, debe producir beneficios.
- **Beneficio mutuo:** Uno de los principales principios de XP y de los más difícil de poner en práctica. Propone pensar siempre en el beneficio de todas las partes implicadas en un desarrollo.
- **Auto-similitud:** Buscar soluciones similares en diferentes contextos. Los patrones de la forma de trabajo deben repetirse adaptándolos a la situación en la que estemos.
- **Mejora:** Excelencia en el desarrollo a través de la mejora. Se busca perfeccionar, no lo perfecto.
- **Diversidad:** Dos ideas sobre un diseño presentan una oportunidad, no un conflicto.
- **Reflexión:** Los buenos equipos piensan cómo y por qué hacen el trabajo, no esconden errores.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### ***XP describe 14 principios como directrices adicionales***

- **Flujo:** Entrega Continua de Valor
- **Oportunidad:** Los problemas son oportunidades de cambio.
- **Redundancia:** Los problemas difíciles deben poder resolverse de diferentes maneras.
- **Fallo:** Un fallo no es desperdicio si me sirve para aprender algo.
- **Calidad:** No vas a ir más rápido aceptando rebajar la calidad.
- **Pequeños pasos:** ¿Qué es lo menos que puedes hacer que sea reconocible y que vaya en la dirección correcta?
- **Responsabilidad aceptada:** El que acepta algo se hace responsable de ello



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 1. Sentarse Juntos

- Disponer de un **espacio abierto** en el que todo el equipo pueda trabajar.
- Una forma de satisfacer la necesidad de espacio privado es **limitando el número de horas de trabajo**.
- El espacio de trabajo debe promover que florezca la comunicación, **elimina los cubículos y las barreras** que impidan **comunicarte** con las personas a tu alrededor.
- Cuantas **más reuniones cara a cara**, más humano y productivo se convierte el proyecto.

#### 2. Equipo Completo

- La gente necesita tener la **sensación de equipo**:
  - Todos somos un equipo
  - Estamos en esto juntos
  - Apoyamos el trabajo, el crecimiento y el aprendizaje de todos.
- **Los equipos son dinámicos**. Si se requiere un conjunto de habilidades o actitudes, trae a una persona con esas habilidades al equipo. Si a alguien no se le requiere más se libera y puede ir a otros lugares.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 3. Espacio de trabajo informativo

- Hacer que el **espacio de trabajo se empape del proyecto** en el que estamos trabajando.
- Colocar las tarjetas con **las historias de usuario en las paredes**, agrupadas por estado.
  - Proporcionan información rápidamente.
- Colocar también en un lugar visible distintas **gráficas de evolución del proyecto**
  - Información activa e importante

#### 4. Trabajo enérgico

- **Trabajar a pleno rendimiento**, pero solo tantas horas como se pueda ser productivo de forma sostenida.
- Con suficiente **cafeína** es muy fácil llegar al punto en el que el código que se escribe quita el valor al proyecto y (lo más peligroso) no darse cuenta de ello.
- Es posible aumentar la productividad **gestionando mejor mi tiempo**
  - Por ejemplo, declarar una franja de dos horas como tiempo de programación, desconectar los emails, los móviles y dedicarse sólo a eso.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 5. Programación en parejas

- Todo **el código** que va a producción **debe** ser escrito por dos personas sentadas frente al pc.
- Pair Programming es un **dialogo entre dos personas** que están simultáneamente analizando, diseñando, probando e intentando programar mejor.
- **Ventajas:**
  - Se mantienen centrados mutuamente
  - Se clarifican ideas
  - Se vencen bloqueos individuales
  - Se cumplen mejor los estándares del equipo
- **Consejos:**
  - Rotar las parejas
  - No juntar dos programadores novatos
  - No invadir el espacio personal del otro (monitores grandes)



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 6. Historias de usuarios

- Escribir las historias de usuario en tarjetas pequeñas, indicando un nombre, una descripción y una estimación de tiempo (Post it)
- Es más adecuado hablar de “historias” que de “requisitos” ( palabra con connotaciones de inmutabilidad y permanencia que no son compatibles con abrazar el cambio)
- Al estimar cada historia es fácil darse cuenta del coste de cada una de ellas y podemos combinarlas, priorizarlas.
- Importante colocar las tarjetas en la pared, en un sitio visible, no en un programa.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 7. Ciclo Semanal

- Reunión al comienzo de cada semana
  - Revisión del progreso hasta la fecha, incluyendo si el progreso de la semana previa se corresponde con lo previsto.
  - Hacer que el cliente escoja historias que sumen una semana de tamaño para la semana actual.
  - Fraccionar las historias en tareas, los miembros de equipo se apuntan a las tareas y las estiman.
- Comenzar la semana escribiendo las pruebas automáticas que se realizarán cuando las historias estén completas y pasar el resto de la semana implementándolas e integrándolas en el proyecto
- Al final de la semana las nuevas historias deben estar disponibles para ser desplegadas.
- La semana es un ciclo corto para poder hacer experimentos, por ejemplo, esta semana cambiaremos pareja cada una hora.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 8. Ciclo Trimestral

- Es conveniente hacer reuniones con un ciclo superior al semanal para ver la evolución del proyecto en conjunto.
- Cada 3 meses:
  - Identificar los cuello de botella, especialmente los que no están controlados por el equipo.
  - Planificar el tema o los temas para el cuatrimestre.
  - Escoger historias que sumen un trimestre para cumplir con los temas escogidos.
  - Centrarse en una visión general del cómo el proyecto encaja en la organización y el valor que añade.
- Los trimestres también son un buen ciclo para reflexionar sobre el equipo, proponer y evaluar experimentos que duran más de una semana.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 9. Holgura

- Trabajar con cierto colchón que suavice la tensión de cumplir compromisos imposibles.
- Es importante establecer una **atmosfera de confianza** en la que los que piden y los que desarrollen el producto se comunican de forma clara y honesta.
- Trabajar con plazos no realistas introduce errores inmanejables, mina la moral y afecta las relaciones.
- Cumplir con los compromisos.

#### 10. Construcción de 10 min

- 10 minutos para construir automáticamente todo el sistema y ejecutar todos los test.
- Todo el proceso del build y de paso de las pruebas debería estar automatizado.
- No comprobar solo las pruebas de la nueva parte añadida del sistema, sino todas las anteriores.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 11. Integración Continua

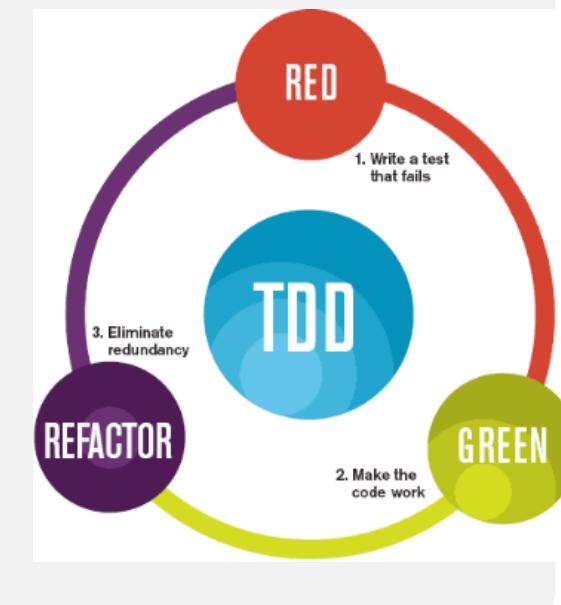
- No dejar pasar más de dos horas sin integrar los cambios que hemos programado.
- La programación en equipo es un problema de “divide , vencerás e integrarás”
- La **integración** es un **paso no predecible** que puesto costar más que el desarrollo
- **Integración síncrona:** cada pareja después de un par de horas sube sus cambios y espera a que se complete el build y se hayan pasado todas las pruebas sin ningún problema de regresión.
- **Integración asíncrona:** cada noche se hace build diario en el que se construye la nueva versión del sistema. Si se producen errores se notifica con alerta de mails.
- El sistema resultante debe ser un sistema listo para lanzarse sin demasiados problemas.

## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Las 13 Prácticas principales de XP

#### 12. Crear pruebas antes de programar (TDD)

- Antes de escribir cualquier código escribir las pruebas o aplicar un desarrollo orientado a pruebas (Test driven development)
- Ventajas:
  - Las pruebas sirven de especificación y aclaran el alcance del código que tenemos que escribir.
  - Cohesión y acoplamiento: si es difícil escribir el test, es una señal que tenemos un problema de diseño.
  - Confianza: Escribiendo código limpio que funciona y demostrando tus intenciones con las pruebas construye una relación de confianza con tus compañeros.
  - Ritmo: Es muy fácil perderse durante horas cuando se está programando. Con este enfoque, está claro lo que hay que hacer a continuación.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

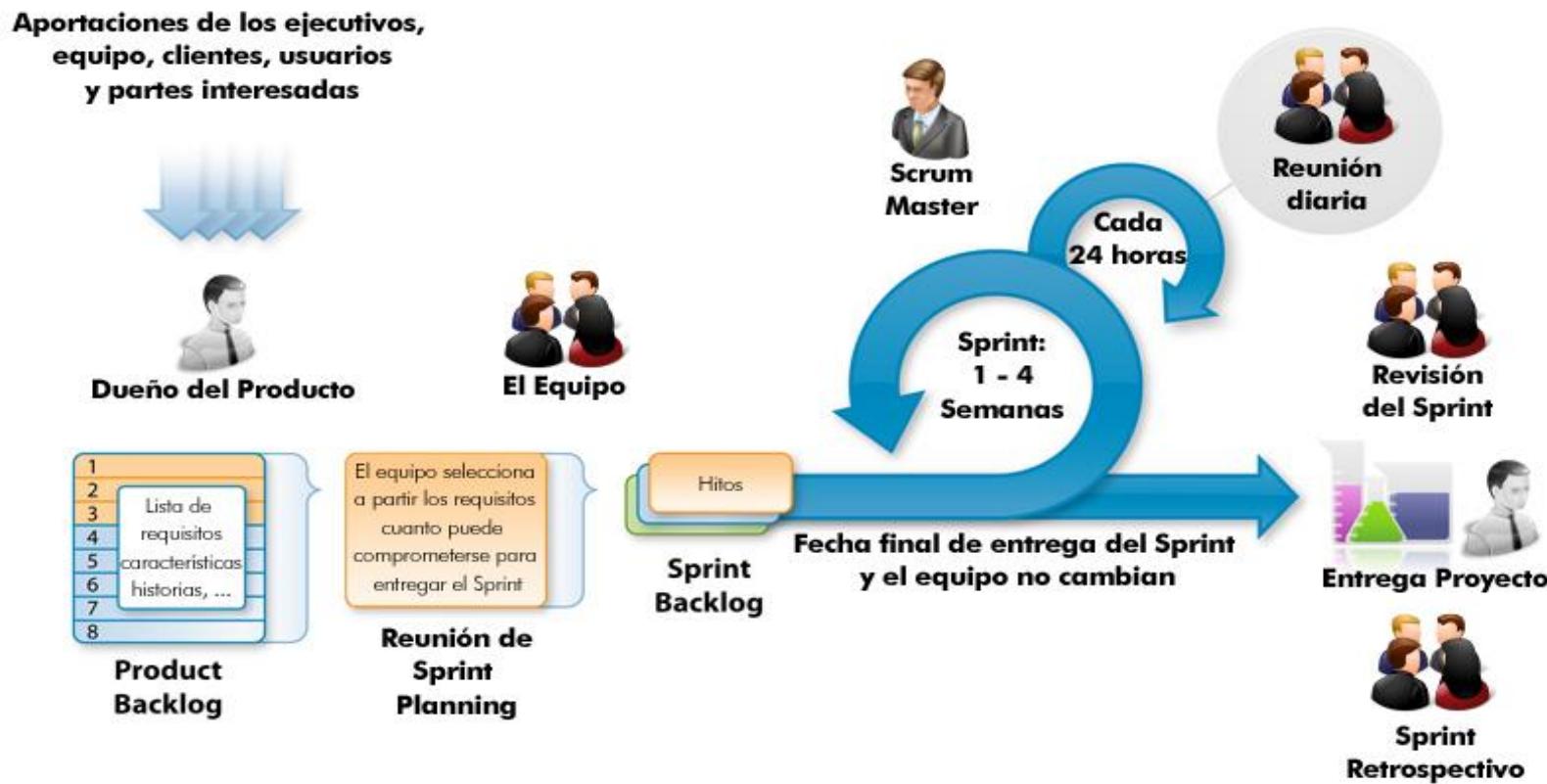
### Las 13 Prácticas principales de XP

#### 13. Diseño Incremental

- Diseño gradual, con pasos pequeños y seguros.
- El diseño no debe hacerse solo al principio, si no conforme vamos adquiriendo experiencia.
- Invertir cada día en el diseño del sistema, luchando por ver como la parte que estamos construyendo puede mejorarlo en su conjunto.
- Pensar todo el tiempo en el problema en su conjunto, buscando formas de diseñar mejor todo el sistema.
- Los equipos XP trabajan duro para crear condiciones bajo las que el coste de cambiar el software sea bajo.
- Una regla sencilla para mejorar el diseño es eliminar la duplicación.

## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

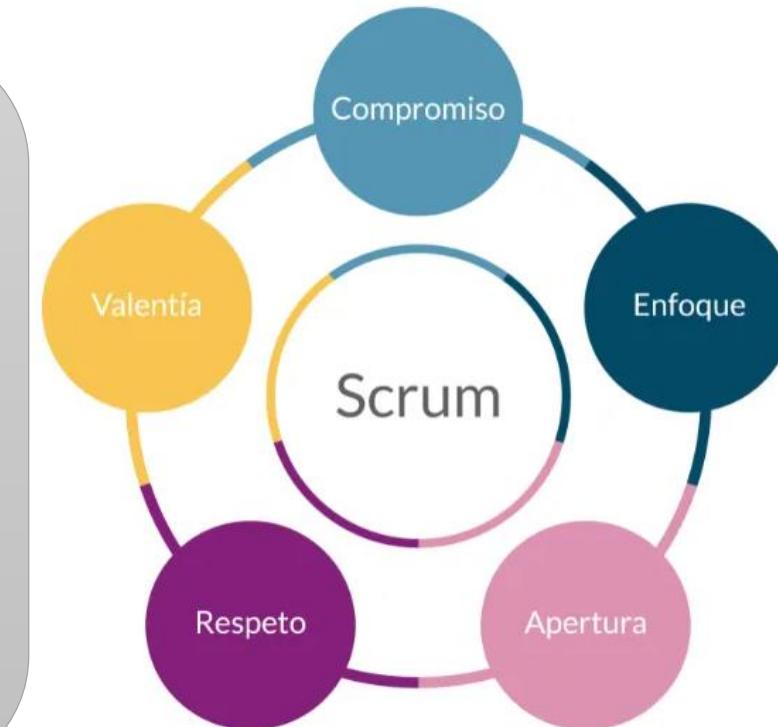
### Enfoque 2: Scrum



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Valores Scrum

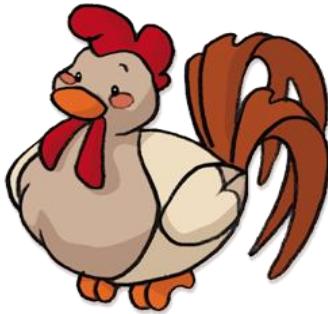
- Compromiso: Las personas se comprometen personalmente a alcanzar los objetivos del equipo Scrum
- Enfoque: El equipo está centrado en el trabajo de Sprint y en los objetivos como equipo Scrum
- Apertura: El equipo de Scrum y sus grupos de interés acuerdan ser abiertos sobre todo el trabajo y los retos que supone realizarlo.
- Respeto: Los miembros de Scrum Team se respetan entre sí por ser personas capaces e independientes. El respeto comienza con el reconocimiento de la humanidad fundamental de los demás.
- Valentía: Los miembros del equipo Scrum tienen el valor de hacer lo correcto y trabajar en problemas difíciles.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **Los roles en Scrum**



Una gallina y un cerdo paseaban por la carretera. La gallina dijo al cerdo: “Quieres abrir un restaurante conmigo”. El cerdo consideró la propuesta y respondió: “Sí, me gustaría. ¿Y cómo lo llamaríamos?”. La gallina respondió: “Huevos con beicon”.

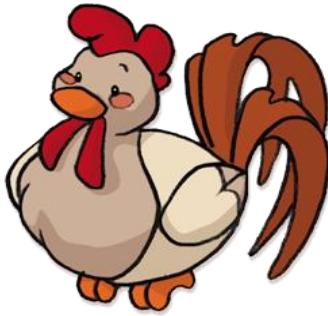
El cerdo se detuvo, hizo una pausa y contestó: “Pensándolo mejor, creo que no voy a abrir un restaurante contigo. Yo estaría realmente **comprometido**, mientras que tu estarías sólo **implicada**”.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **Los roles en Scrum**



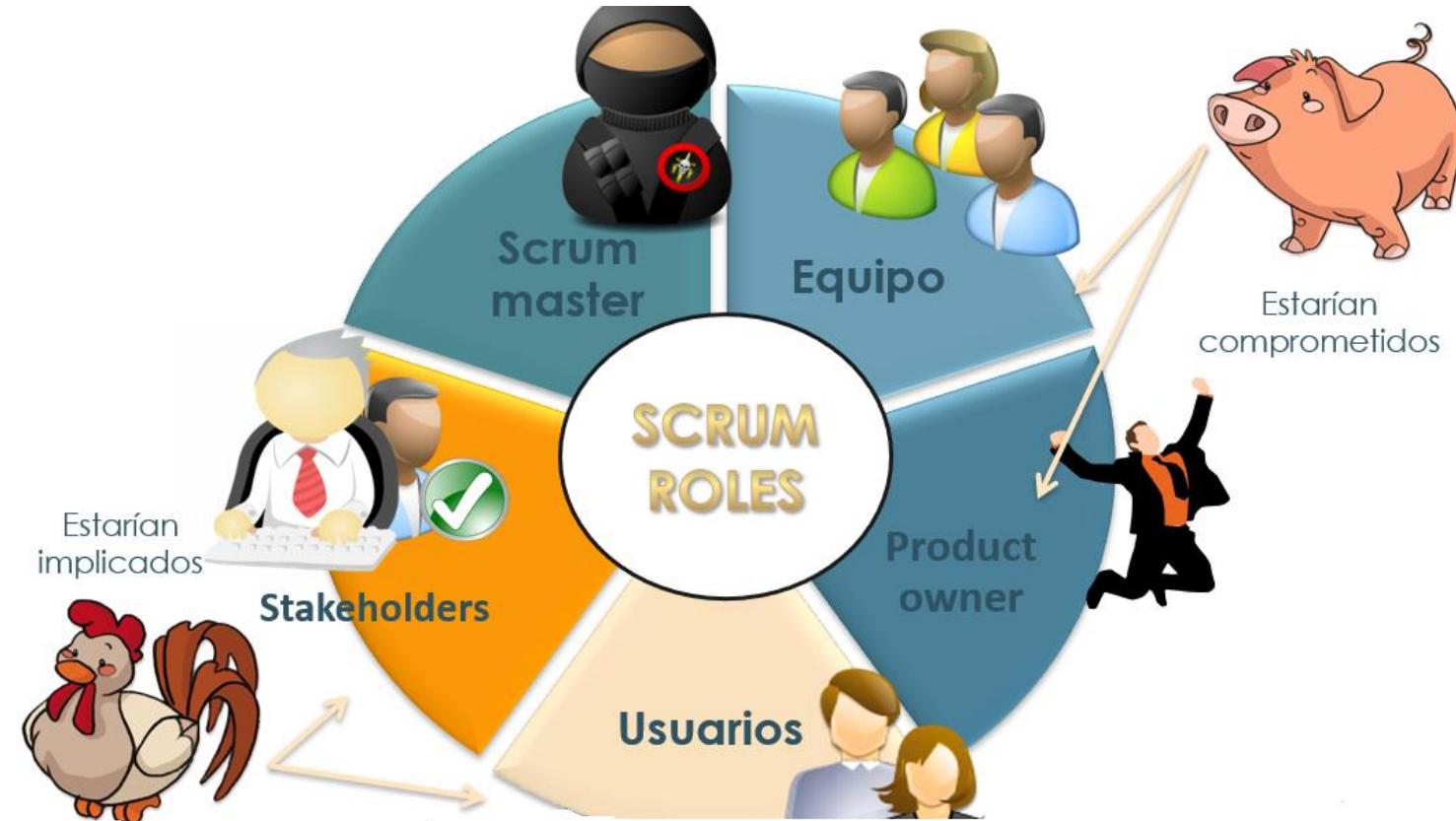
Una gallina y un cerdo paseaban por la carretera. La gallina dijo al cerdo: “Quieres abrir un restaurante conmigo”. El cerdo consideró la propuesta y respondió: “Sí, me gustaría. ¿Y cómo lo llamaríamos?”. La gallina respondió: “Huevos con beicon”.

El cerdo se detuvo, hizo una pausa y contestó: “Pensándolo mejor, creo que no voy a abrir un restaurante contigo. Yo estaría realmente **comprometido**, mientras que tu estarías sólo **implicada**”.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### *Los roles en Scrum*





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

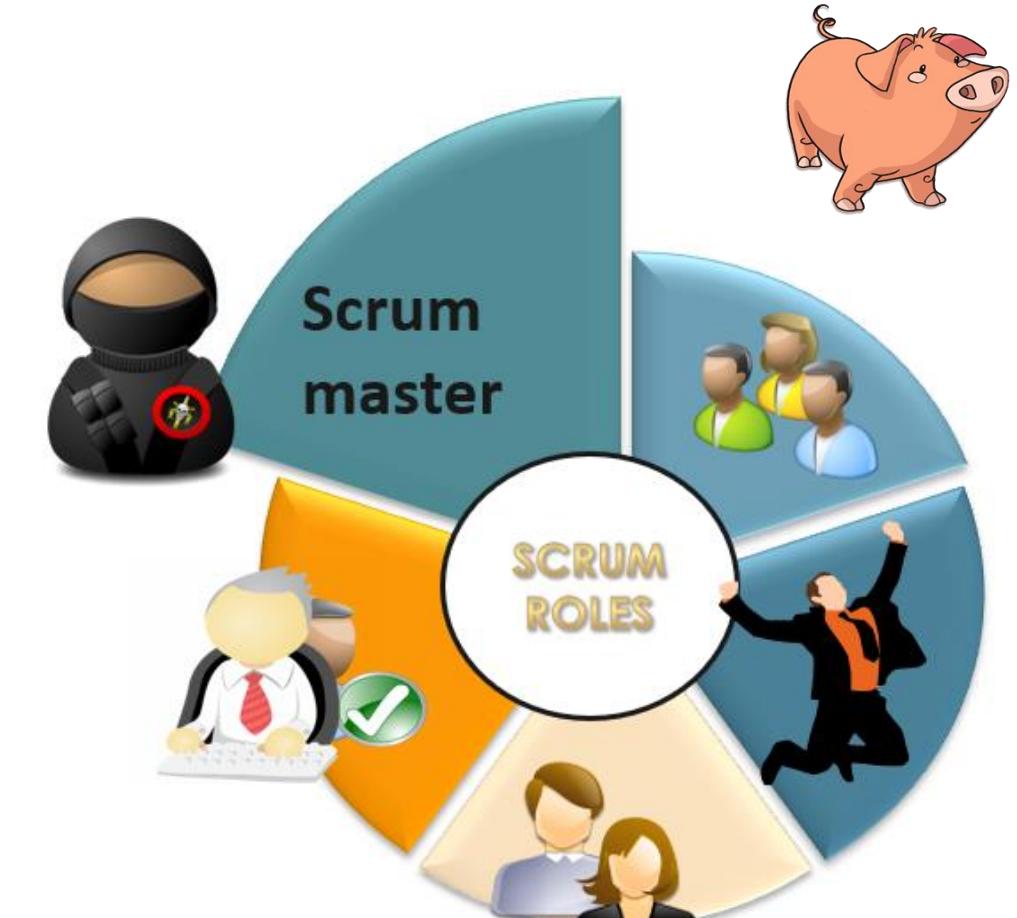
### **Los roles en Scrum**

#### **Scrum master**

Una persona con grandes conocimientos y experiencia en técnicas y métodos de desarrollo de software.

##### **RESPONSABILIDADES:**

- Garantizar la correcta aplicación de Scrum.
- Resolver los conflictos que entorpezcan el progreso del proyecto.
- Incentivar y motivar al Scrum Team, creando un clima de trabajo colaborativo.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **Los roles en Scrum**

#### **Equipo**

Es el equipo de desarrolladores multidisciplinario, formado por programadores, diseñadores, testers y demás, que de forma auto-organizada, será los encargados de desarrollar el producto.

#### **RESPONSABILIDADES:**

- Transformar el Backlog, en cada iteración, en un incremento en la funcionalidad del producto.
- Planificar su propio trabajo para lograrlo.
- Son responsables en conjunto del éxito de cada iteración y del proyecto en su totalidad.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **Los roles en Scrum**

#### **Product Owner**

Persona que representa a todos los stakeholders y es la única autorizada para decidir qué funcionalidades y características funcionales tendrá el producto. Representa al cliente, usuarios del software y todas aquellas partes interesadas en el producto.

##### **RESPONSABILIDADES:**

- Saber "escuchar" a las partes interesadas en el producto y transmitirlas en "objetivos de valor para el producto" al equipo de Scrum
- Revisar el producto e ir adaptándole sus funcionalidades



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **Los roles en Scrum**

#### **Stakeholder y Usuarios**

- Los roles gallina en realidad no son parte del proceso pero deben tenerse en cuenta. Incluyen a los usuarios, y otros interesados (stakeholders). Es importante que esa gente participe y entregue feedback con respecto a fin de revisar y planear cada sprint.
- Stakeholders son la fuente más importante de la validación del proyecto. Tienen deseos y necesidades, y son la razón por cual el equipo está desarrollando el software.
- Los usuarios son considerados como las empresas o personas que utilizarán el resultado del proyecto o producto.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Artefactos Scrum

#### Backlog del producto

- El propietario gestiona una lista priorizada de los elementos planificados del producto.
- El backlog del producto evoluciona de sprint a sprint (refinamiento del backlog)

#### Backlog del Sprint

- Al inicio de cada sprint, el equipo Scrum selecciona una serie de elementos de máxima prioridad (Backlog del sprint) a partir del backlog del producto.

#### Sprint

- Scrum divide el proyecto en iteraciones (sprints) de longitud fija (2 a 4 semanas)



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### **Eventos Scrum**

#### Planificación del Sprint

- Una jornada de trabajo. El propietario del producto explica las prioridades.
- El equipo estima el esfuerzo de los requisitos y se elabora el Backlog del sprint.

#### Sprint

- Ciclo de desarrollo básico en el marco estándar del scrum, de duración recomendada inferior a un mes y nunca superior a 6 semanas.

#### Incremento

- Cada Sprint acaba siendo un producto potencialmente entregable, conocido como incremento.



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### Eventos Scrum

#### Scrum Diario

- 15 minutos máximo
- Responsabilidad de todo el equipo
- Cada miembro expone: lo que hizo ayer, lo que hará hoy y si tiene obstáculos.
- Se actualiza el backlog del sprint.

#### Revisión del Sprint

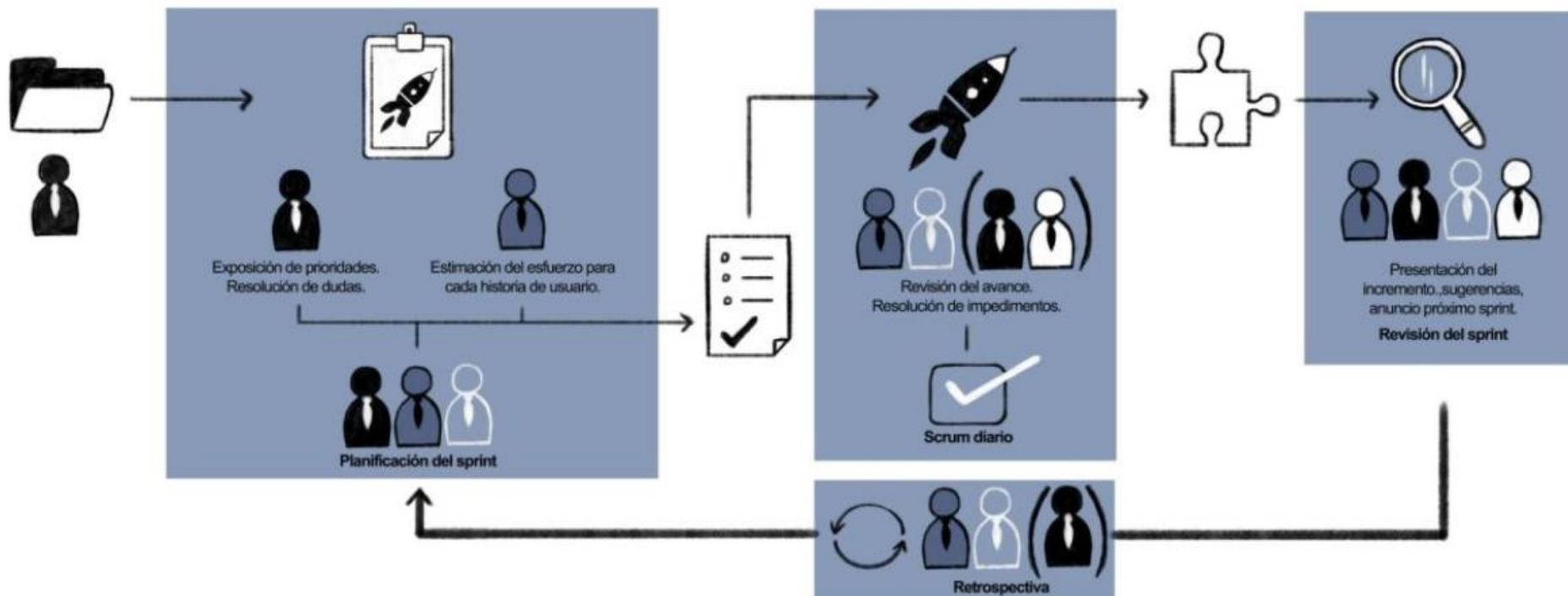
- Reunión informativa
- Máximo 4 horas
- Se revisa el incremento, se plantean sugerencias y se anuncia el próximo sprint.

#### Retrospectiva

- Equipo autoanaliza la forma de trabajo.
- Identifican fortalezas y debilidades, para fortalecer las primeras y determinar el plan de mejora de las segundas.

## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### LAS REGLAS DE SCRUM





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### KANBAN

Kanban, llega como gestión visual de proyectos, de la mano de la automotriz Toyota.



Origen japonés Kan "visual"  
Ban "tarjeta y/o tablero"

"tablero visual"



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### KANBAN

- Es un sistema que controla el flujo de actividades a través de tarjetas, las cuales son utilizadas para indicar el estatus de cada una de ellas.
- Cada tarjeta Kanban acompaña a una actividad durante todo el proceso, hasta que esta, es empujada hasta la etapa de CONCLUIDA.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### KANBAN

#### ¿Para qué sirve?

- Balancear la demanda con la capacidad
- Limitar el trabajo en proceso, mejorar el flujo, descubrir los problemas y lograr un ritmo sostenible
- Controlar el trabajo (no a la gente), coordinar y sincronizar, descubrir los cuellos de botella y tomar decisiones
- Para que los equipos se auto-organicen
- Lograr una cultura de optimización

#### Beneficios

- Obtener reglas simples que permitan optimizar el trabajo
- Experimentar nuevas formas de trabajo
- Incrementar la colaboración
- Mejorar el ritmo de trabajo
- Visualizar el trabajo de todas las personas



## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### KANBAN

#### LAS TRES REGLAS

- Mostrar el proceso
- Limitar el trabajo en curso - WIP
- Optimizar el flujo de trabajo

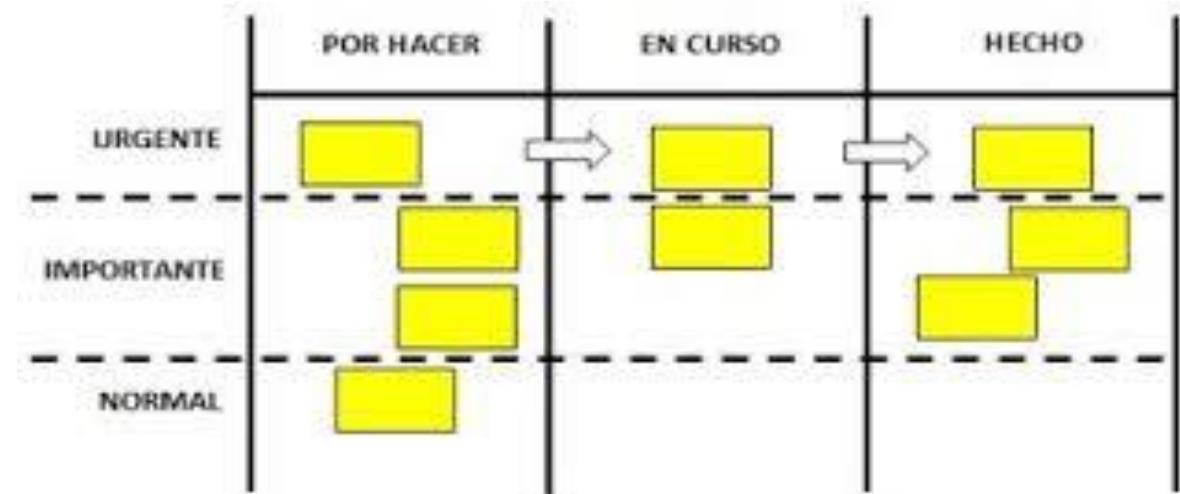


## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### KANBAN

#### REGLA 1: MOSTRAR EL PROCESO

- Consiste en la visualización de todo el proceso de desarrollo, mediante un tablero físico.
- El objetivo de mostrar el proceso, consiste en: Entender mejor el proceso de trabajo actual.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### KANBAN

#### REGLA 2: LIMITAR EL TRABAJO EN CURSO (WIP)

- Limitar el trabajo que se está desarrollando, sirve para evitar saturación tanto en el proceso, como en las personas.
- Cuando exista una cantidad elevada de tareas EN PROCESO, es posible, delegar o solicitar el apoyo de un tercero para concluir la actividad.





## 1.2 Desarrollo ágil de software- Aspectos de los Enfoques Ágiles

### KANBAN

#### REGLA 3: OPTIMIZAR EL FLUJO DE TRABAJO

- Este punto se refiere a la mejora continua de las actividades realizadas, y a la optimización que se puede dar paulatinamente a todo el flujo de trabajo.
- Una vez que las personas asimilan el flujo de una actividad, resulta sencillo que se generen ideas para mejorarlo





## 1.1 Desarrollo ágil de software-Fundamentos

### EJERCICIO 1

¿Cuál de las siguientes afirmaciones es cierta sobre Kanban, Scrum y XP?

1. Kanban se utiliza principalmente para la transición de XP a Scrum
2. Tanto XP como Scrum tienen como objetivo proponer buenas prácticas para el desarrollo
3. Scrum propone un marco para la organización de equipos y la gestión de proyectos
4. XP y Kanban se basan en un flujo de gestión



## 1.1 Desarrollo ágil de software-Fundamentos

### EJERCICIO 2

¿Cuáles de los siguientes son los cinco valores que XP adopta para guiar el desarrollo?

1. Comunicación, simplicidad, retroalimentación, coraje y respeto
2. Comunicación, oportunidad, control, calidad y confianza
3. Comunicación, simplicidad, retroalimentación, control y transparencia
4. Comunicación, honestidad, retroalimentación, coraje y respeto



## 1.1 Desarrollo ágil de software-Fundamentos

### EJERCICIO 3

**¿La reunión diaria de stand-up incluye?**

1. Las partes interesadas y los desarrolladores
2. Sólo los desarrolladores y gerentes
3. Todos los miembros del equipo agilen, incluidos los probadores
4. Probador y el cable de prueba



## 1.1 Desarrollo ágil de software-Fundamentos

### EJERCICIO 4

Los equipos ágiles utilizan el tablero Kanban para gestionar WIP. ¿WIP es una abreviatura para?

1. Trabajo integrando personas
2. Trabajo en progreso
3. Plan de inventario de trabajo
4. Parámetro de entrada de trabajo



## 1.1 Desarrollo ágil de software- Fundamentos

### EJERCICIO 5

Asocie los siguientes enfoques de desarrollo ágil de software con sus descripciones correspondientes:

- i. Kanban
  - ii. Programación extrema
  - iii. Scrum
- 
- a. Abarca cinco valores para guiar el desarrollo: comunicación, simplicidad, retroalimentación, coraje y respeto.
  - b. Divide un proyecto en iteraciones (llamadas sprints) de longitud fija (generalmente de dos a cuatro semanas). Cada sprint resulta en un producto potencialmente liberable / enviable.
  - c. El objetivo general de este enfoque es visualizar y optimizar el flujo de trabajo dentro de una cadena de valor añadido.
- 
1. i-c, ii- a, iii-b
  2. i-a, ii- c, iii-b
  3. i-a, ii- b, iii-c
  4. i-b, ii- a, iii-c



# Curso ISTQB® Foundation LEVEL AGILE TESTER

