

A blurred background image showing a close-up of a person's hands typing on a computer keyboard. The hands are positioned on the right side of the frame, with fingers moving over the keys. The background is dark and out of focus.

CURSO

SOFTWARE TESTING

UNIDAD 6 – Testing Agile



PRESENTACIÓN

En esta unidad repasaremos las características del testing Agile, profundizando en la metodología Scrum

TEMARIO

1. – Manifiesto Agile
2. – Explicación del modelo Scrum
3. – Herramienta Jira.
4. – Ejemplos.

Manifiesto Agile - Valores

Individuos e interacciones
sobre
procesos y
herramientas

Software que funciona
Sobre
documentación exhaustiva

Colaboración con el cliente
sobre
negociación de contratos

Responder al cambio
sobre
seguimiento a un plan

Manifiesto Agile - Valores

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblegan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los períodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

Manifiesto Agile

The TESTING Manifesto

we value:

- Testing throughout **over** at the end
- Preventing bugs **over** finding bugs
- Testing understanding **over** checking functionality
- Building the best system **over** breaking the system
- Team responsibility for quality **over** tester responsibility

- *Testing durante SOBRE testing al final: Testear desde el inicio por sobre el testing como una etapa al finalizar el sprint.*
- *Prevenir bugs SOBRE encontrar bugs: Incentivar la discusión y revisión grupal de las tareas, entender qué hay que hacer, cómo y porque, evitar suposiciones y trabajar en equipo con el desarrollador para que los bugs puedan ser detectados de antemano por sobre esperar a que el desarrollo este completo para revisar y estudiar el mismo.*
- *Entender lo que se esta testeando SOBRE verificar funcionalidad: La necesidad de entender que es lo que quiere el usuario, para que lo quiere y como él lo va a usar, para realizar pruebas que den valor agregado por sobre solo ver que las especificaciones se cumplen*
- *Construir el mejor sistema SOBRE romper el sistema: Enfocar el trabajo del equipo en mejorar lo que se esta haciendo por sobre solo intentar romperlo.*
- *El equipo es responsable de la calidad SOBRE el tester es responsable de la calidad: Que el equipo se comprometa con todas las actividades y trabaje para generar valor, que la responsabilidad sobre la calidad sea del equipo por sobre que la calidad recaiga sobre un solo rol como es el del tester.*

Scrum

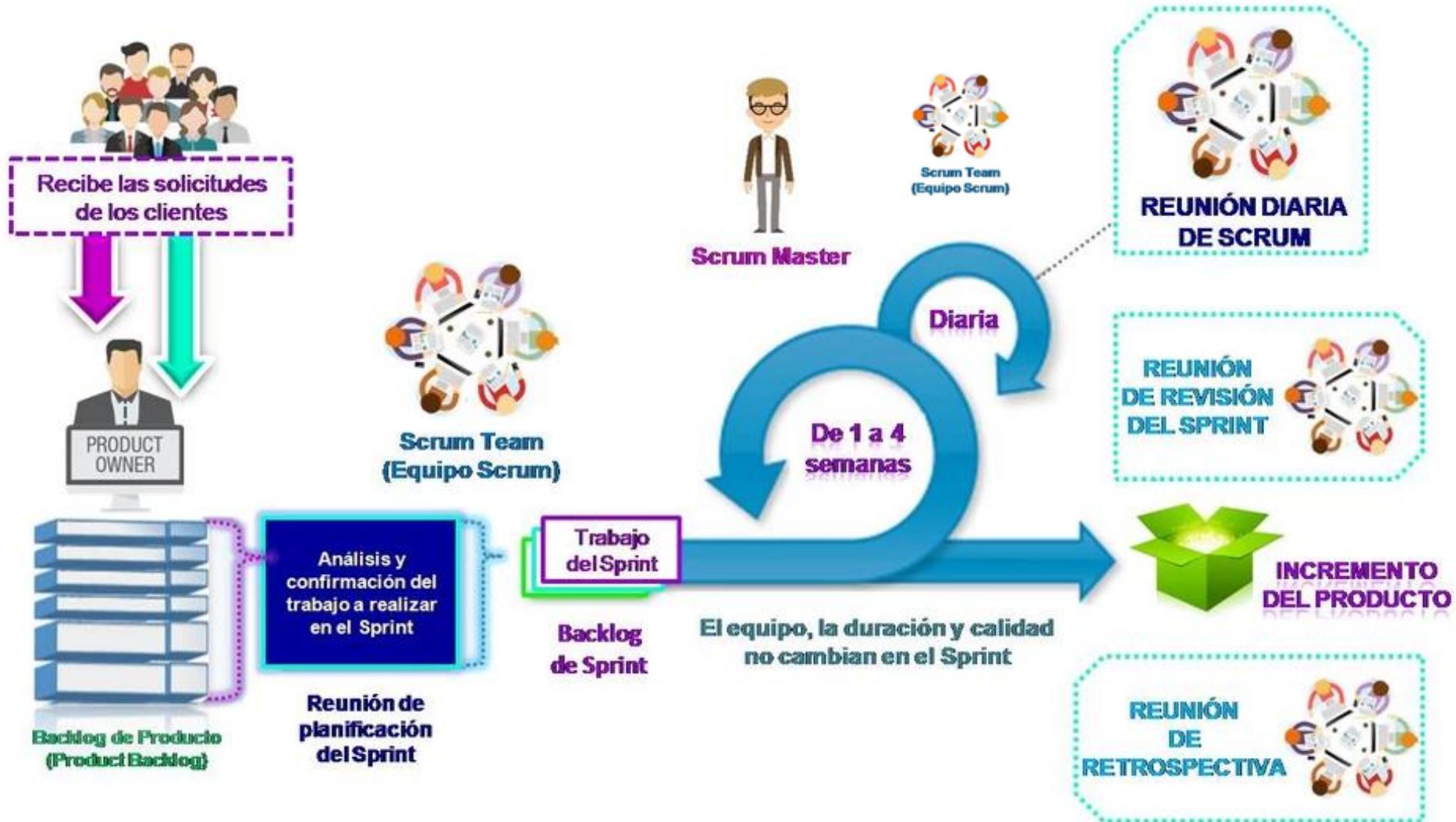
- Esta metodología, es un **marco de trabajo de procesos ágiles** que trabaja con el **ciclo de vida iterativo e incremental**, donde se va liberando el producto por partes de forma periódica, aplicando las buenas prácticas de trabajo colaborativo (en equipo).
- Con **Scrum** se realizan entregas regulares y parciales del producto final, todas ellas con una prioridad previamente establecida que nace según el beneficio que aporten al cliente, minimizando los riesgos que pueden surgir de desarrollos extremadamente largos. Es por tal motivo, que Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesitan obtener resultados de manera inmediata.
- Tres principios:
 - La **transparencia** de todo el entorno del proyecto. El hecho de que todos los implicados conozcan cada uno de los aspectos del proyecto favorece el ahorro de tiempo y costes.
 - La **inspección** continua es lo que permite que se esté siguiendo el camino hacia el objetivo deseado.
 - La **adaptación** permite que, si es necesario, se cambien los requerimientos del producto a entregar para que éste sea lo más competitivo posible.

Equipo Scrum

El equipo Scrum tiene unas características muy particulares. Es **multidisciplinar y autoorganizado**. Esto quiere decir que es capaz de gestionar su trabajo de manera eficaz y no depende de terceras partes para realizar su cometido. Por otra parte, todos los miembros mantienen una **actitud colaborativa**, ya que, el resultado del proyecto es **responsabilidad de todos**

- Por una parte, el **Product Owner** es el cliente o quien representa al cliente. Sabe qué producto final desea obtener y cómo debe ser.
- En segundo lugar, el **Scrum Master** es el líder del equipo. Dirige, organiza y da apoyo al resto del equipo, **es** el encargado de eliminar cualquier obstáculo que impida la consecución de los objetivos. Delega las tareas en sus colaboradores para que éstos se auto-organicen y alcancen un nivel de coordinación y colaboración exitoso.
- Por último, el **Development Team** es el grupo de profesionales que elaboran el producto que propone el dueño del producto, la cantidad ideal es entre 5 y 9 personas.

Scrum



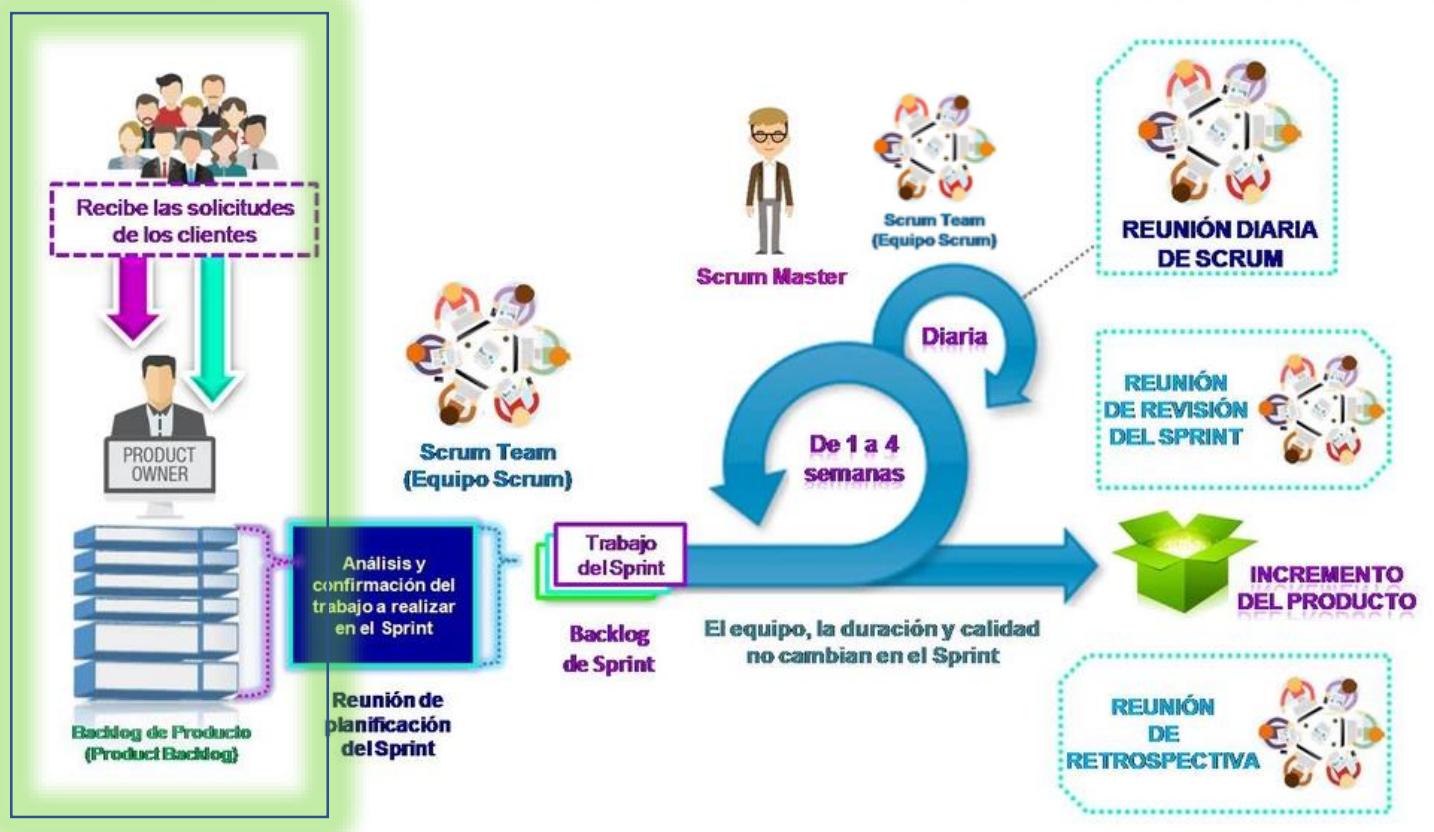
Eventos:

- Sprint (cada sprint tiene un objetivo x ej que los usuarios hagan login en la app)
- Reunión diaria
- Reunión de revisión (con p.owner)
- Reunión de retrospectiva

Artefactos:

- Product Backlog
- Sprint Backlog
- Incremento (elementos de backlog terminados)

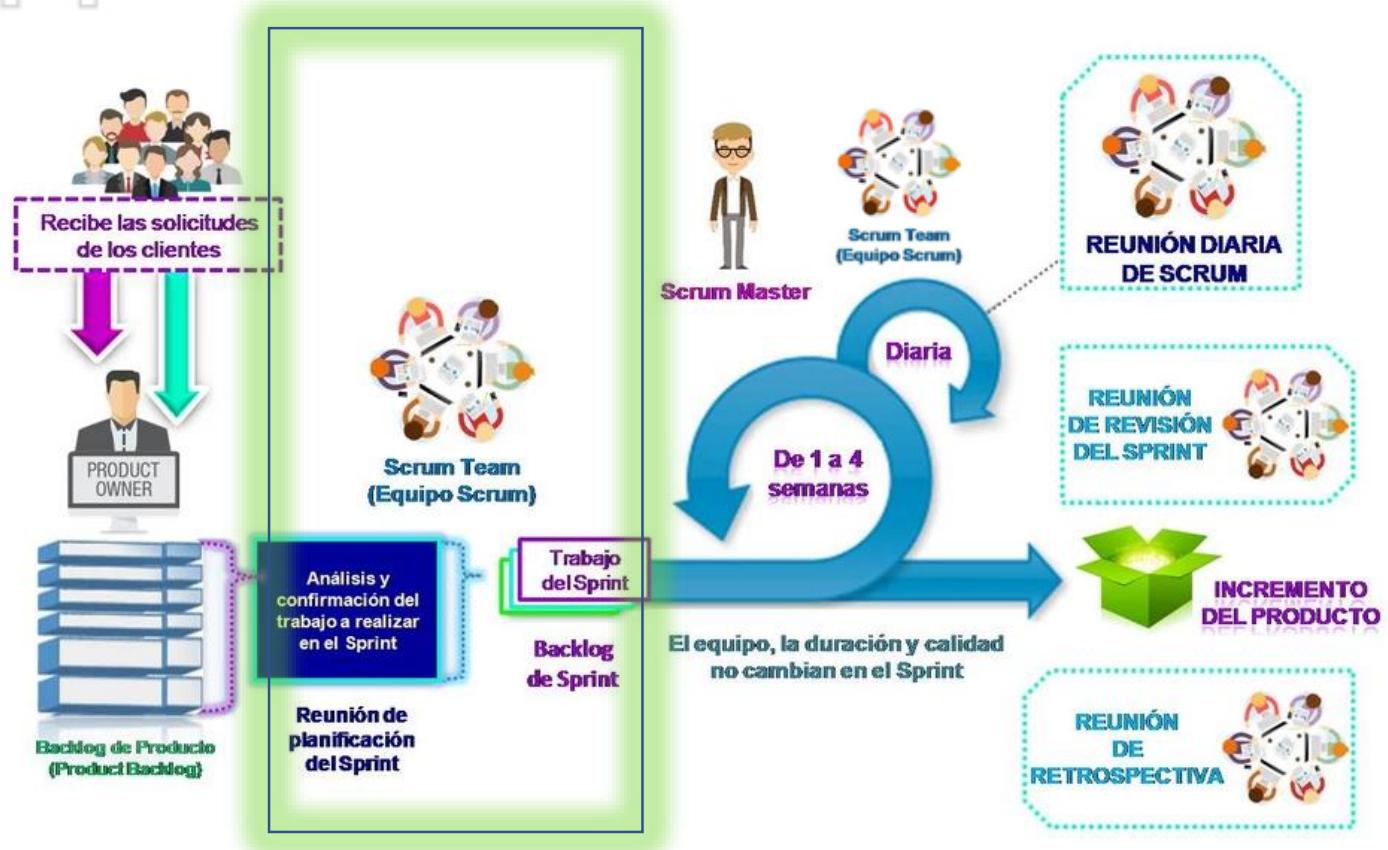
Scrum



El **Product Owner** representa las necesidades del cliente, se enfoca en los aspectos del negocio, incluyendo la prioridad en que el producto se va a ir desarrollando para que éste genere valor. También se encarga de las proyecciones y presupuesto. Es quien recibe de los clientes los requerimientos y quien arma el **backlog** del producto, con prioridades.

En el backlog van distintas tareas:

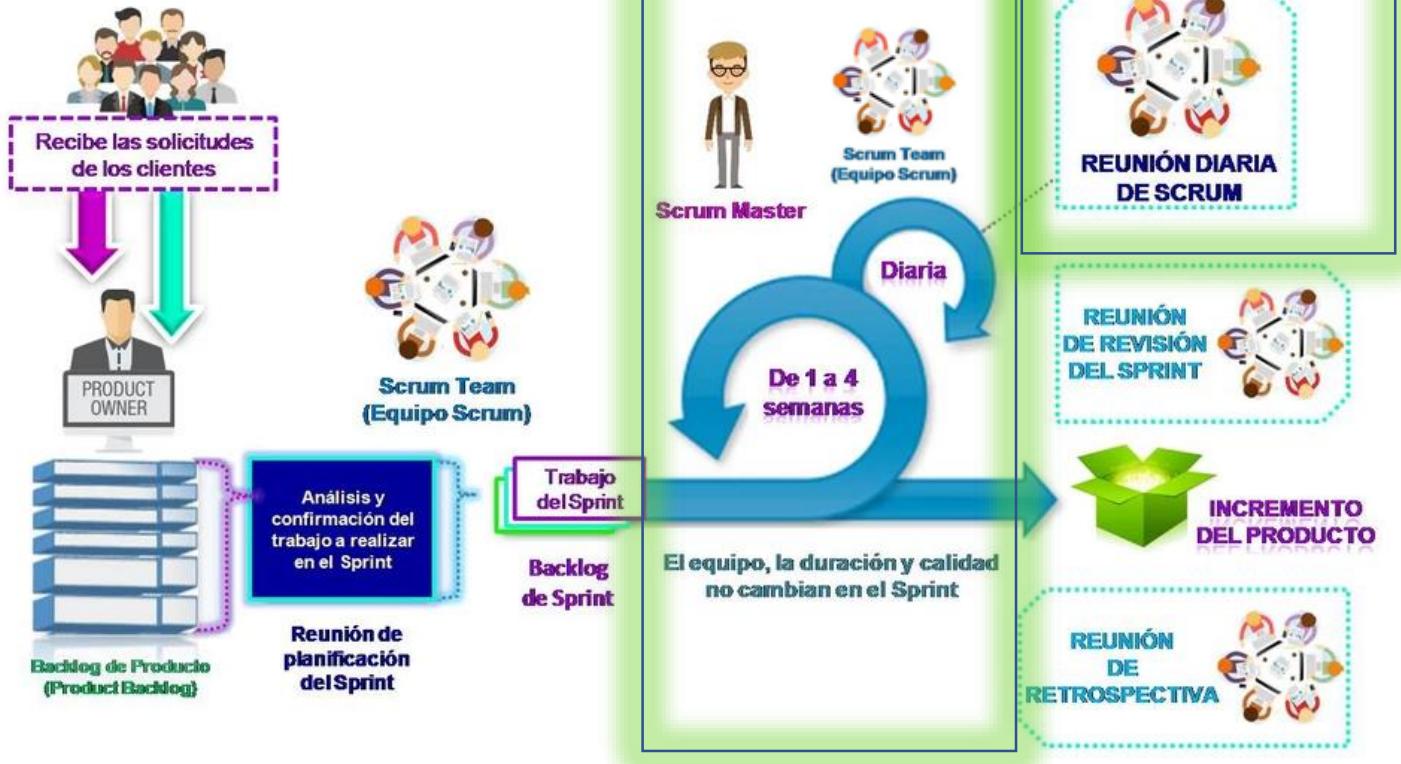
- Funcionalidad
- Bugs
- Actividades técnicas
- Actividades de investigación



Sprint Planning:

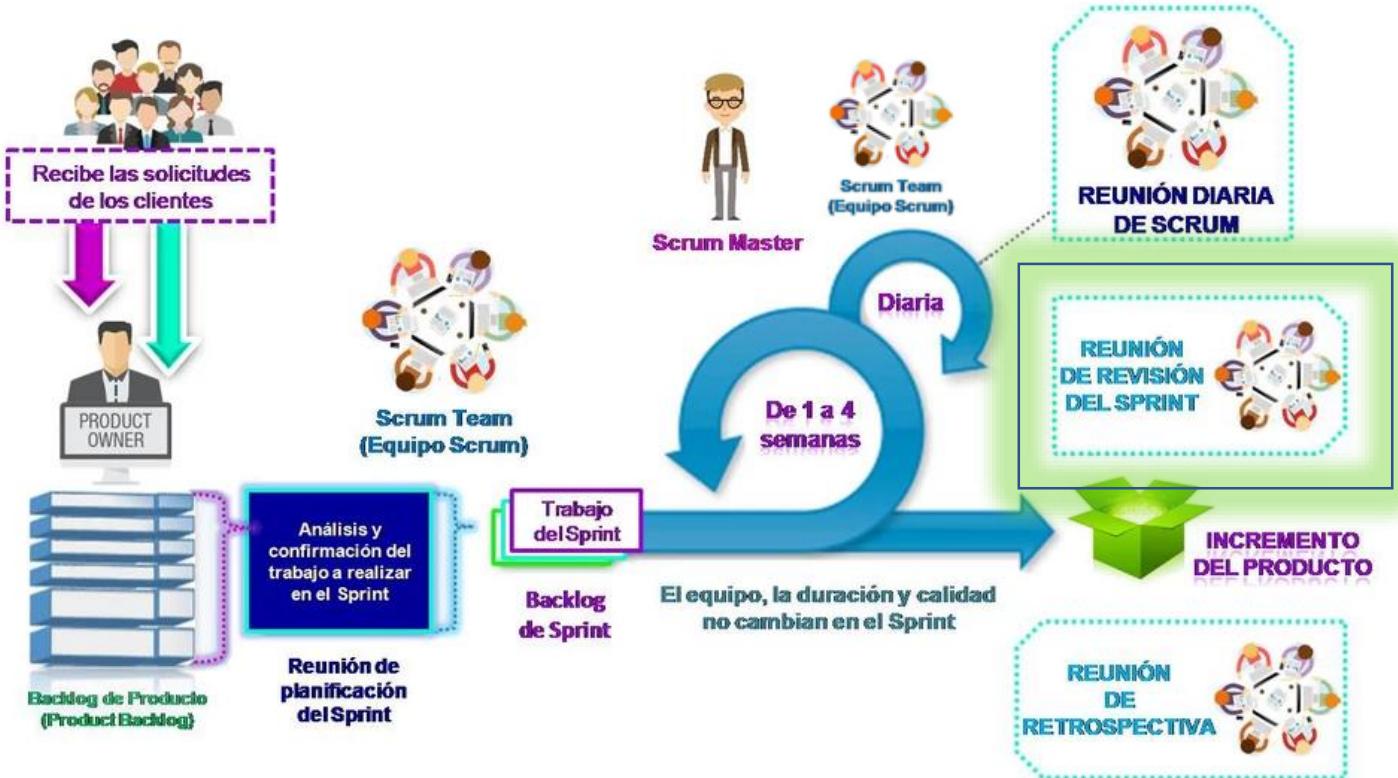
- El Product Owner le explica al equipo cuales son las tareas del backlog con mayor prioridad
- El Product Owner responde dudas acerca de el trabajo a realizar en cada actividad (tarea).
- El equipo estima utilizando puntos de historia.
- Se genera el Sprint Backlog (lista de actividades a realizar en el sprint)
- El sprint backlog debe estar priorizado y estimado.

Scrum



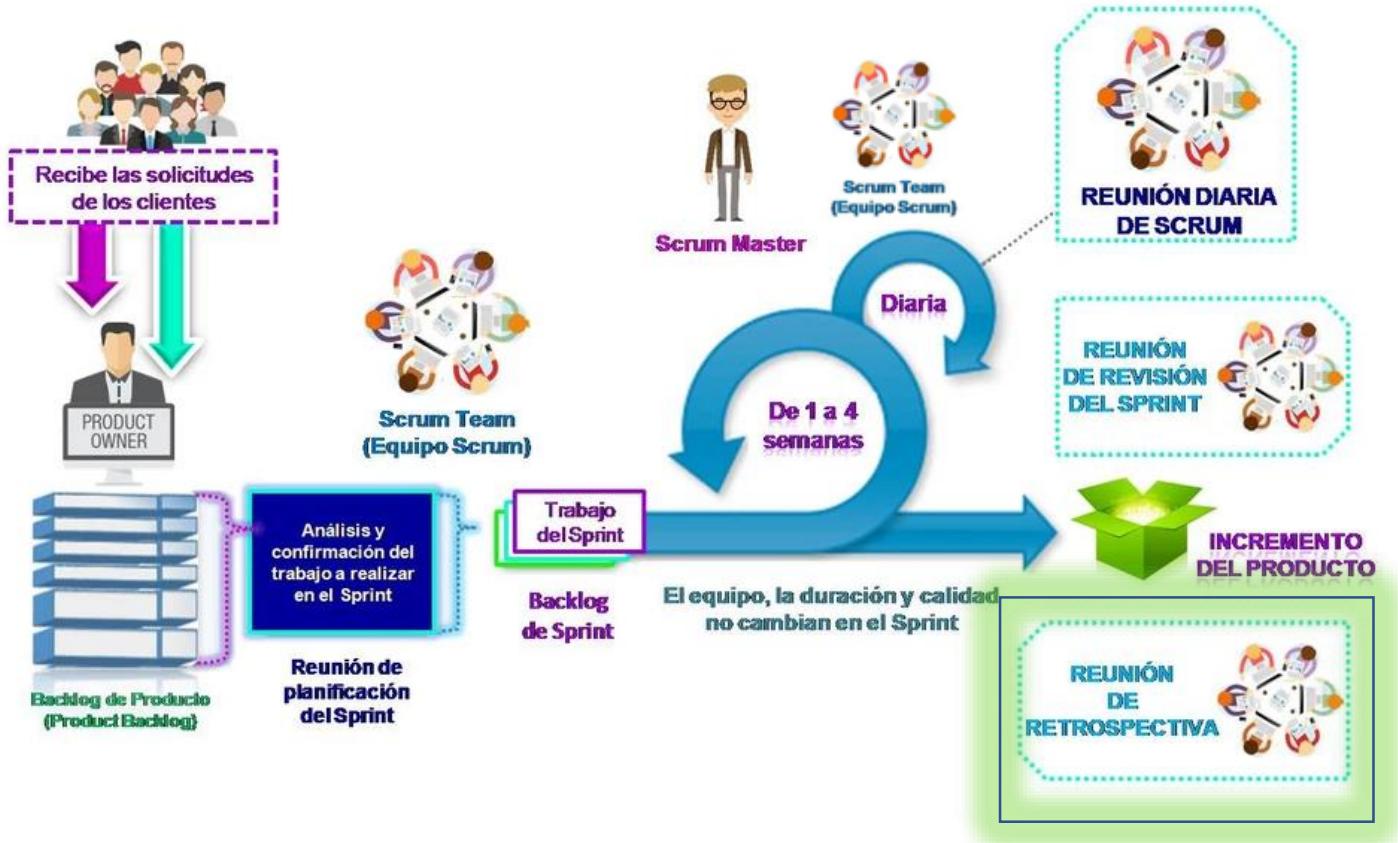
Sprint:

- Se realizan las tareas que componen el Sprint Backlog
- Duración entre 1 y 4 semanas.
- Daily Scrum: Reunión de 15 minutos que monitorea avances e impedimentos. ¿Qué hicimos ayer? ¿Qué haremos hoy? ¿Qué impedimentos tengo?



Reunión de Revisión del Sprint:

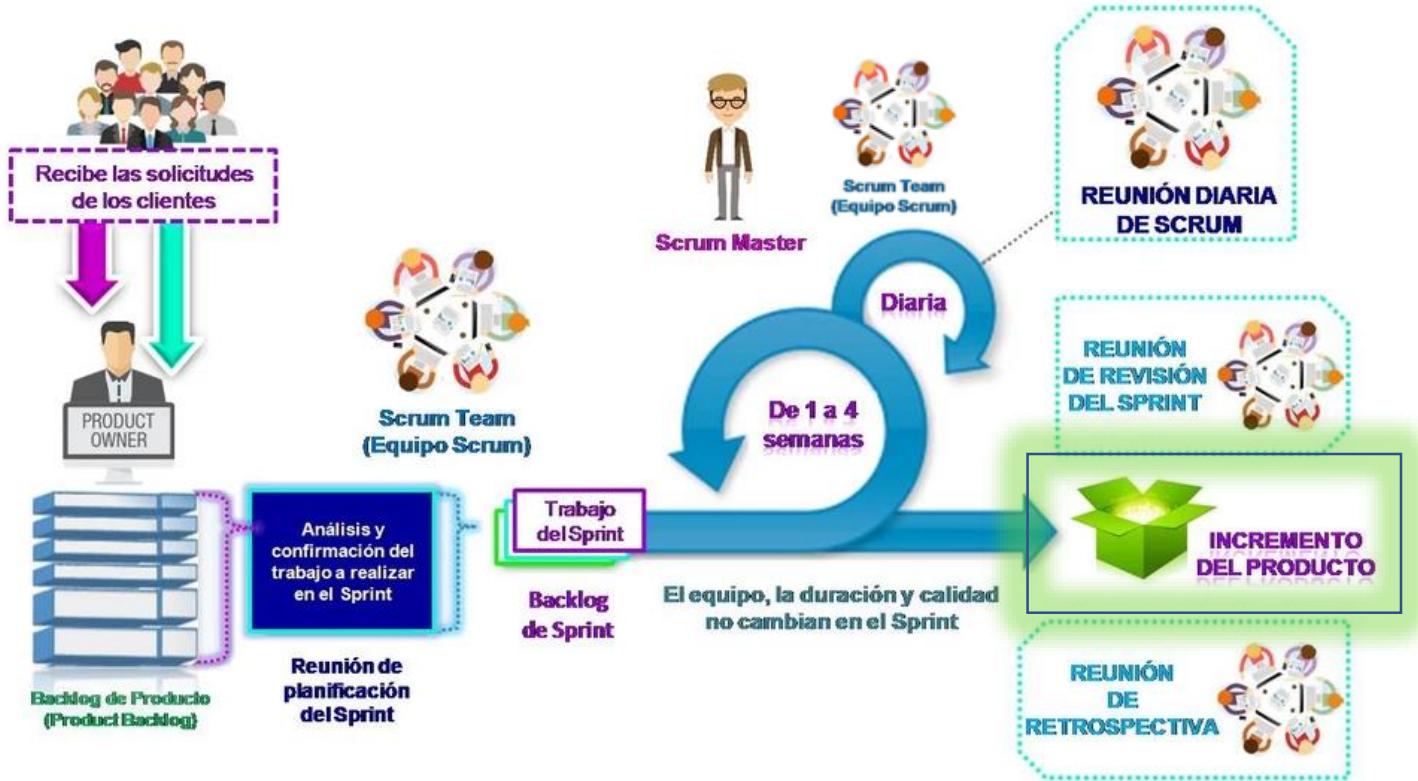
- Se le presenta al Product Owner el avance del proyecto, indicando las actividades Terminadas.
- El avance debe ser Tangible (server instalado, sistema ejecutando, componente corriendo, etc)
- Tienen a surgir nuevas ideas y funcionalidades para integrar al producto



Reunión de Retrospectiva del Sprint:

Reflexion:

- ¿Qué debemos mejorar?
- ¿Qué debemos seguir haciendo y que debemos empezar a hacer?
- ¿Cómo ser mas productivos?



Incremento del producto:

Al final del Sprint hay que obtener un producto liberable, aunque no se libere necesariamente.

Este producto tiene nueva funcionalidad

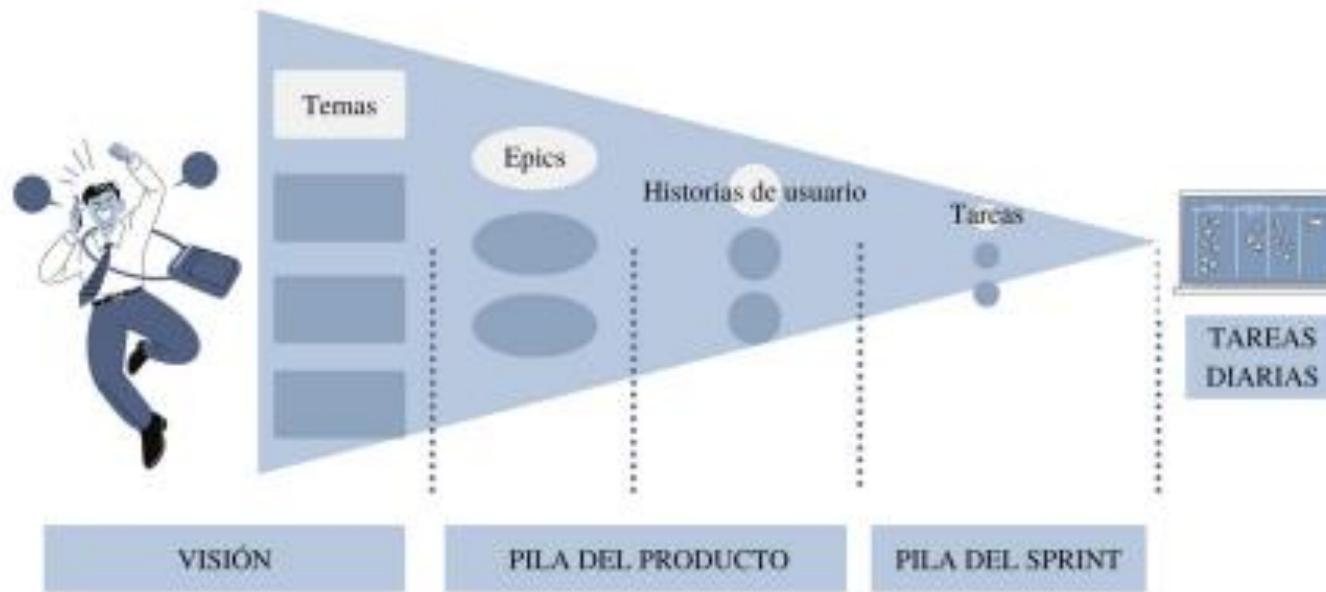
Backlog

Granulidad de los ítems

- Los ítems del Backlog de Producto no necesariamente deben tener una granulidad pareja. Es posible encontrar ítems tales como "es necesario contar con un módulo de control de stock y logística" o uno tan pequeño como "Modificar el color de fondo de los mensajes de error del sistema, de negro a rojo".
- Ítems de tan baja granulidad, suelen agruparse en un formato denominado "Historias de Usuario" mientras que los de alta granulidad, son los denominados Epics.
- Las Epics son historias que por su tamaño no pueden cumplirse dentro de un solo Sprint
- Una historia de usuario es aquella que puede escribirse con la siguiente frase:
 - Como [un usuario], puedo [una funcionalidad] para [un beneficio]*
 - Como usuario registrado, puedo cargar un voucher para calcular mi descuento en la compra.*

Backlog

Requisitos ágiles



Backlog

Prioridad	ítem	Estimación	Estado	Criterios de aceptación
1	Como administrador quiero poder administrar las secciones del sistema para poder establecer el orden de visualización de las mismas	3	En curso	# No pueden cargarse secciones duplicadas # Probar establecer todas las secciones con el mismo número de orden # Probar sin asignar un número de orden
2	Como administrador quiero poder asignar un mismo motivo de consulta a varias secciones diferentes para tener que evitar volver a cargar datos que pueden ser reutilizados	5	Pendiente	# Una misma sección no puede contener un mismo motivo de consulta asignado más de una vez # Cuando se modifique un motivo de consulta debe verse reflejado en todas las secciones a las cuales haya sido asignado

Backlog

PRODUCT BACKLOG DEL TERCER SPRINT

ID	Historia de usuario	Estimación	Prioridad	Sprint
HU09	Como docente quiero registrar datos de horario para poder gestionar los horarios de la institución.	5	-	3
HU10	Como docente quiero modificar datos de horario para poder gestionar los horarios de la institución.	5	-	3
HU11	Como docente quiero buscar datos de horario para poder gestionar los horarios de la institución.	5	-	3
HU12	Como docente quiero cambiar estado de horario para poder gestionar los horarios de la institución.	5	-	3
HU13	Como docente quiero registrar datos de alumnos para poder gestionar los horarios de la institución.	3	-	3
HU14	Como docente quiero modificar datos de alumnos para poder gestionar los horarios de la institución.	3	-	3
HU15	Como docente quiero buscar datos de alumnos para poder gestionar los horarios de la institución.	3	-	3
HU16	Como docente quiero cambiar estado de alumnos para poder gestionar los horarios de la institución.	3	-	3
HU17	Como docente quiero registrar datos de matrícula para poder gestionar los horarios de la institución.	2	-	3
HU18	Como docente quiero modificar datos de matrícula para poder gestionar los horarios de la institución.	2	-	3
HU19	Como docente quiero buscar datos de matrícula para poder gestionar los horarios de la institución.	2	-	3
HU20	Como docente quiero cambiar estado de matrícula para poder gestionar los horarios de la institución.	2	-	3

Backlog

Sprint Backlog

Producto Web de Compra de Libros	Pendiente	En Progreso	Finalizado
Nº Sprint: 04	Historia #1	Tarea 1.2 Tarea 1.3	Tarea 1.1 Tarea 1.4
Objetivo del Sprint <i>El objetivo de este Sprint es que el usuario pueda completar una búsqueda de libros por autor y ordenarla por precio de compra ascendente y descendente, así como por año de publicación.</i>	Historia #2	Tarea 2.2 Tarea 2.1 Tarea 2.3	
	Historia #3 Tarea 3.5	Tarea 3.3 Tarea 3.4	Tarea 3.1 Tarea 3.2
	Tarea Técnica #04 Subtarea 4.1	Subtarea 4.2	
	Spike #1 Bug 001 Bug 005 Bug 012		

Casos de prueba basados en Historias de Usuario

COMO <rol>

QUIERO <eventos>

PARA <funcionalidades>

COMO USUARIO WEB

QUIERO CONSULTAR LA
TABLA DE PEDIDOS
PARA SABER EL ESTADO
DE TODOS MIS PEDIDOS

- El Product Backlog es nuestro medio de comunicación con el cliente, las historias no deben contener lenguaje técnico
- Las Historias de Usuario deben ser independientes entre si, cada una es una unidad que puede cumplirse independientemente de otra
- Una Historia de usuario escrita es menos importante que la conversación que provoca

Historias de Usuario

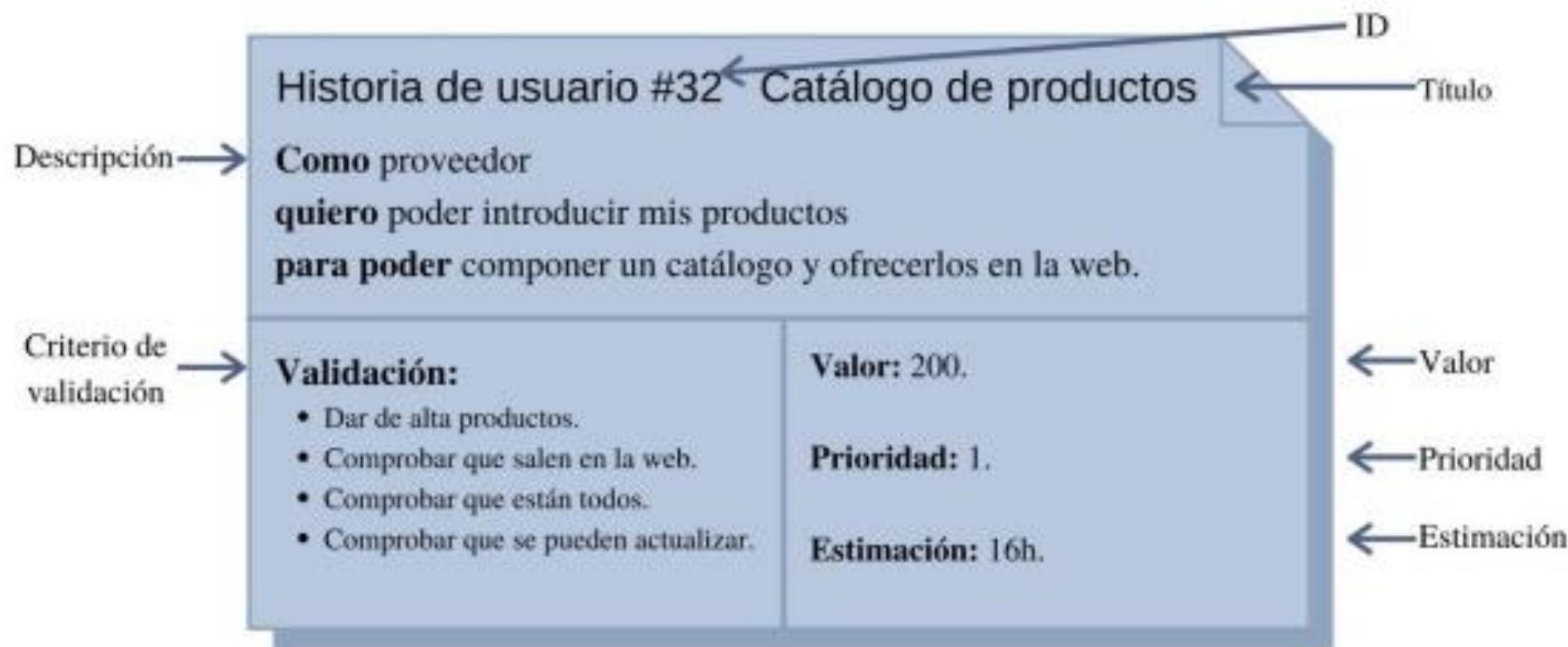
La Historia debe ser lo suficientemente pequeña para que entre en una tarjeta y que invite a una conversación.

Hay que recordar que en un enfoque ágil de desarrollo no hay porque tenerlo todo perfectamente definido por adelantado antes de llevarlo al equipo. El equipo y el negocio pueden descubrir la necesidad mientras trabajan juntos, y esto puede acordarse a través de la conversación y colaboración alrededor de las Historias de Usuario.

La tarjeta conduce a una conversación entre el Product Owner y los equipos, esto debe suceder en persona tanto como sea posible, y finalmente debe incluir una especie de confirmación. En ese momento es cuando el equipo sabe realmente qué se debe hacer y en este momento es cuando entran los criterios de aceptación.



Historias de Usuario



Desglose de Historias de Usuario

Historia: Como Cliente quiero pagar los productos de mi carrito de compras para recibirlas en mi casa.

Como [rol del usuario], quiero [objetivo], para poder [beneficio].

- 1) Como cliente, quiero iniciar sesión con mi cuenta, de tal modo que no tengo que volver a ingresar mi información personal cada vez;
- 2) Como cliente, quiero revisar y confirmar mi pedido, para poder corregir los errores antes de pagar;
- 3) Como cliente, quiero pagar con una transferencia bancaria;
- 4) Como cliente quiero pagar con tarjeta de crédito
- 5) Como cliente, quiero recibir un correo de confirmación de mi pedido, para tener un comprobante de mi compra

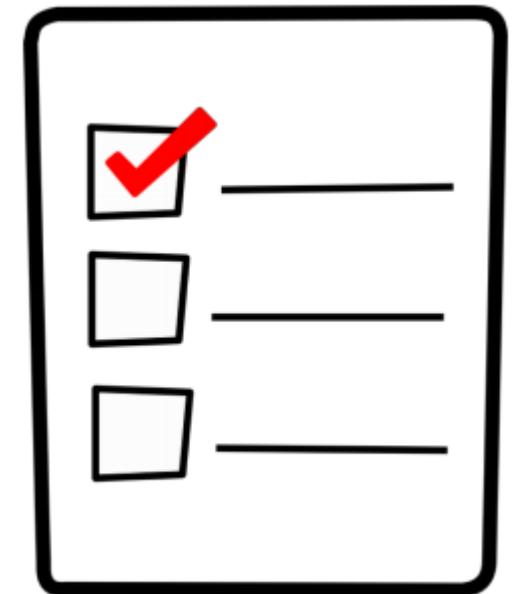
Criterio de Aceptación

Las Historias de Usuario plantean criterios de aceptación, que son escenarios sobre los cuales basar nuestras pruebas

Como vicepresidente de marketing, quiero seleccionar una temporada de vacaciones para revisar el rendimiento de campañas publicitarias pasadas para poder identificar las rentables.

Criterios de Aceptación:

- Asegurarse de que funcione con las principales fiestas minoristas: Navidad, Pascua, Día de la Madre, Día del Padre, Día del Trabajo, Año Nuevo.
- Las temporadas de vacaciones se pueden establecer de una fiesta a otra (como Día de Acción de Gracias a Navidad).
- Las temporadas de vacaciones se pueden establecer en un número de días antes de las vacaciones.



Los criterios de aceptación son binarios, se cumplen o no, teniendo formato de checklist

Criterio de Aceptación

- Específicos
- Medibles
- Alcanzables
- Relevantes
- Limitados en el tiempo

Se terminan de definir en el Sprint Plannig, surgen de la conversación

(Scenario) Escenario [número de escenario] [titulo del escenario]:

(Given) Dado que [contexto] y adicionalmente [contexto],

(When) Cuando [evento],

(Then) Entonces [resultado / comportamiento esperado].

Criterio de Aceptación

Historia de usuario:

Como cliente **quiero** retirar dinero del cajero **para** poder evitar ir al banco a hacer cola.

Criterios de aceptación

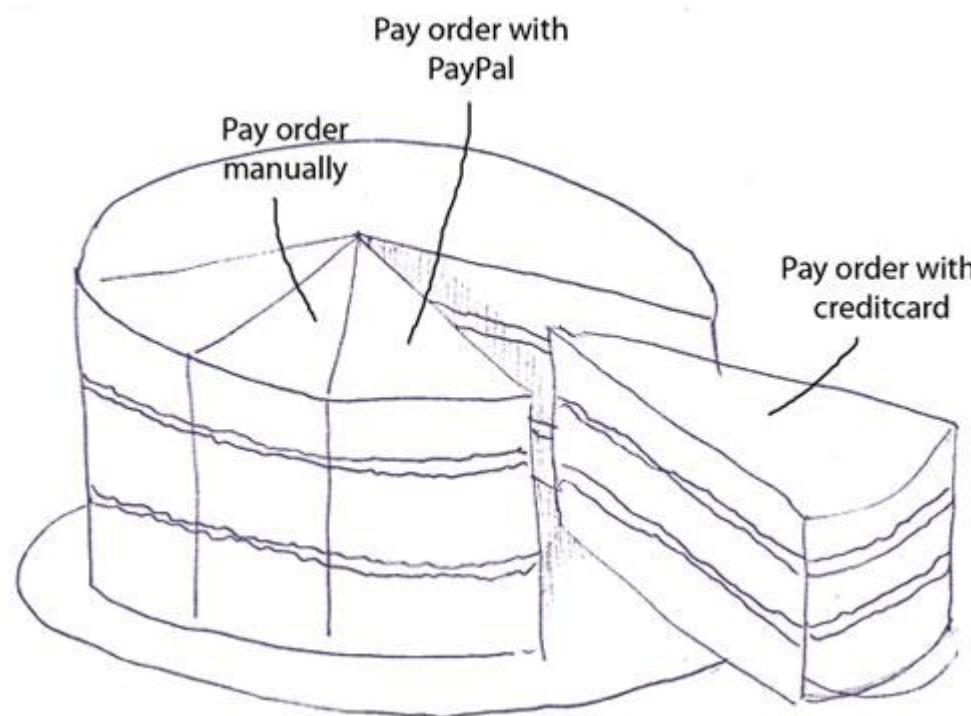
- Escenario 1: cuenta tiene crédito.

Dado que la cuenta tiene crédito **Y que** la tarjeta es válida **Y que** el cajero tiene dinero disponible. **Cuando** el cliente pide dinero **Entonces** la cuenta es debitada **Y** el dinero es entregado al cliente **Y** el cliente recupera su tarjeta

- Escenario 2: la cuenta excede el límite negativo acordado con el banco.

Dado que la cuenta excede el límite negativo acordado con el banco **Y que** la tarjeta es válida **Cuando** el cliente pide dinero **Entonces** el cajero muestra un mensaje negando el pedido **Y** el dinero no es entregado al cliente **Y** el cliente recupera su tarjeta

Desglose de Historias de Usuario Horizontal VS Vertical



Horizontal: Implica desglosar las historias por el tipo de trabajo que se necesita hacer, o las capas o componentes que están involucrados. Es decir, el trabajo que se tiene que hacer para la UI, la base de datos, algunos componentes, el front-end y las pruebas se dividen en ítems técnicos, pero los ítems individuales no dan como resultado un software funcional ni demostrable y, en consecuencia con valor para el negocio. Solo la combinación de todo el trabajo genera valor para el negocio. Pero si se genera valor cuando todas las tareas están completas y esto alimenta cuellos de botella.

Vertical: La funcionalidad no se dividirá en capas técnicas o tareas, sino en capas funcionales (partes de una torta)

Desglose de Historias de Usuario Horizontal VS Vertical



Una organización vertical de las tareas anticipa el feedback

Tareas

- Las tareas forman parte del Backlog del Sprint. Estas últimas se identifican durante la reunión de planificación del sprint, al descomponer las historias de usuario en tareas. Así se convierten en parte del Backlog de sprint.
- Una historia de usuario suele ser una funcionalidad, algo visible para los usuarios finales. Desarrollar una historia de usuario es algo complejo, y normalmente requerirá la colaboración de varios miembros del equipo: un programador, alguien que la pruebe e incluso puede que un diseñador o un analista.
- Una tarea, por otra parte, es típicamente algo como diseñar, codificar, testar, probar, automatizar, etc. Al respecto parece que las tareas tienden a ser cosas hechas por una persona.
- Es decir, las historias de usuario contienen varios tipos de trabajo (por ejemplo, programación, pruebas, diseño de la base de datos, diseño de interfaz de usuario, análisis, etc.) mientras que las tareas se limitan a un solo tipo de trabajo.

Caso de Uso / Historia de Usuario / Requisito Funcional

- Un caso de uso captura la funcionalidad deseada desde la perspectiva de los usuarios (actores) y cómo interactúan con el sistema. Los casos de uso se centran en cambio en la especificación de requisitos. Las historias de usuario están más cerca de la captura de requisitos, extraer las necesidades del usuario. Podríamos decir que una historia de usuario dice qué quiere el cliente o usuario, y un caso de uso entra en cómo lo quiere. las Historias de Usuario son acerca de necesidades, mientras que los Casos de Uso ayudan a explicar el comportamiento que el sistema debe tener para ser exitoso.
- Igual que en los requisitos funcionales, las historias de usuario dicen el qué pero no el cómo se desarrollará la funcionalidad. Las historias no deben tener el nivel de detalle que tiene la especificación de un requisito funcional
- Para los requerimientos tradicionales los Analistas de Negocio usan técnicas como entrevistas, encuestas u otras que extraigan respuestas en reacción a las acciones o preguntas, para luego documentarlas en los instrumentos que usen en la organización, el Analista obtiene las respuestas las documenta y cambia a la siguiente cosa, En Agil la Historia de Usuario es una discusión y un esfuerzo colaborativo. La manera en que son escritas ayudan a empujar una conversación, terminando en cosas como ¿porque la necesitan realmente?, ¿cuales son los detalles?, ¿que sabemos y que no sabemos? ¿cuales son los matices que la componen?

Actividades de QA

Product Backlog: Conocer el backlog y las funcionalidades que tendrá el producto (portal, app, etc) y estas deben estar priorizadas, identificar cuáles insumos le hacen falta a todo el equipo para empezar a desarrollar las HU's, identificar incongruencias entre lo que ya está definido y lo presentado en los diseños de UX/UI.



Actividades de QA

Sprint Planning: Participar en la definición detallada de los criterios de aceptación de cada HU's, comentar las dudas que se tengan sobre las HU's e identificar validaciones que no se hayan tenido en cuenta, validar que el diseño presentado por UX/UI finalmente si cumpla con lo definido en las HU's, identificar los tipos de usuario y demás datos que necesitemos para ejecutar los diferentes escenarios de prueba, identificar si es necesario generar un documento de consideraciones generales (Diseño, validaciones de campos, caracteres permitidos,etc) para tener en cuenta durante todo el desarrollo, generar un caso de prueba por cada HU's, comentarla como una tarea y socializarla con los desarrolladores, participar en la votación de puntos para cada HU's (Estimación).

Al final de este evento estará claro lo que se va a desarrollar y todo el equipo tendrá listos los insumos necesarios para comentar con el Sprint, por nuestra parte los QA podremos comenzar con el diseño de casos de prueba

Actividades de QA

Daily: ¿Qué hice ayer? ¿Qué voy a hacer hoy? Y ¿Qué impedimentos tengo?

Informar como va la ejecución del proceso de pruebas. (teniendo en cuenta versiones, ambientes, etc).

Comentar nuevos casos o validaciones que hayamos identificado y que deban ser contempladas en el desarrollo del proyecto. Con esta pequeña reunión realizada conjuntamente con todo el equipo, estaremos todos alineados y conoceremos día a día nuestro avance con el Sprint, además es la oportunidad de comentar los inconvenientes que surjan para poder solucionarlos lo más rápido posible y no afectar el avance del equipo



Actividades de QA

- **Review:** Presentar la Review, socializar los incidentes que se hayan presentado en el sprint (Indicadores), socializar la automatización de pruebas que se haya realizado (Si aplica), tomar nota de las observaciones o conclusiones que se presenten en la ceremonia para poder gestionarlas posteriormente.
- Muchas veces el cliente puede tener dudas o confusiones frente a lo que está viendo desarrollado del producto, y nosotros como QA podremos ayudar a resolver esas dudas. Es importante que tomemos evidencia del producto funcional (Vídeos o imágenes) antes de cada review y las tengamos presentes por si surge algún inconveniente que no permita hacer la demostración del mismo. También es relevante que socialicemos como fue el ciclo de ejecución de pruebas y sus resultados

Actividades de QA

Retrospectiva: Identificar posibilidades de mejora en cuanto a personas, relaciones, procesos y herramientas, para implementar en el próximo sprint. Buscar mejores formas de trabajar. Hablar sobre la información que salió en la Review si es necesario.



PREGUNTAS





UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Fin de a Unidad 6.