4.3 - Las Historias de Usuario

Las Historias de Usuario (User Story) son la forma recomendada en entornos ágiles para escribir los requisitos del producto desde la perspectiva del usuario. Dado que son afirmaciones breves, simples y fáciles de entender, resultan en una mejor comunicación entre los Stakeholders y el equipo Scrum.

Una de las principales ventajas de las Historias de Usuario es su estructura simplificada y orientada a la acción. Debido a que se centran en entender "¿quién solicita la funcionalidad?", "¿qué necesita/requiere?" y "¿por qué o para qué lo necesita?", las Historias de Usuario proporcionan una vista clara y contextualizada de lo que se espera, eliminando ambigüedades comunes en los requisitos tradicionales.

Los criterios de aceptación, que acompañan a cada Historia de Usuario, definen las condiciones específicas que deben cumplirse para que la historia se considere satisfactoriamente implementada o desarrollada. Actúan como una guía para el equipo, aterrizando las expectativas del Product Owner sobre el comportamiento o funcionalidad que debe tener esa historia. Mientras que, la Definición de Terminado (DoD) es la que establece cuándo una historia o tarea se considera completa en su totalidad.

Las Historias de Usuarios centran la atención en los usuarios finales, por lo que no utilizan un lenguaje técnico, con el fin de facilitar el entendimiento para los usuarios y asimismo ofrecer un contexto suficiente para guiar el esfuerzo de los Developers.

> Mike Cohn, uno de los pioneros en el desarrollo ágil, menciona que "Las historias de usuario permiten a los equipos trabajar en un lenguaje compartido, y eso hace que toda la experiencia de desarrollo sea más orientada al usuario y al valor. En lugar de ser una mera lista de tareas, las historias son llamadas a la acción para resolver desafíos específicos para los usuarios".

Estructura de una Historia de Usuario

Las Historias de Usuario suelen expresarse como una frase simple y breve que cumple con la siguiente estructura:

Como: Describe el ***Rol*** del Stakeholder que solicita (o usaría) la funcionalidad o requerimiento.

Quiero: Describe la ***necesidad o requerimiento*** del usuario, por lo general, es una frase corta.

Para: Describe el ***beneficio esperado*** por el Stakeholder una vez se desarrolle el requerimiento

Ejemplos de Historias de Usuario

> **Como:** Cliente, **Quiero:** pagar mi suscripción mensual vía sitio web por medio de transferencia bancaria o tarjeta de crédito, **Para:** evitar el desplazamiento hasta la sucursal de pago.

> **Como:** Supervisor de ventas, **Quiero:** consultar un listado de los pedidos de venta que han sido registrados y aún no han sido procesados, **Para:** Tener un reporte detallado a la mano.

> **Como:** Ejecutivo de cuenta, **Quiero:** consultar los datos de un cliente suministrándole al sistema únicamente su documento de identidad o código de cliente, **Para:** facilitar el proceso de búsqueda y reducir el tiempo de atención.

> **Como:** asesor de ventas, **Quiero:** disponer de una diadema inalámbrica, **Para:** poder levantarme regularmente de mi puesto y disminuir el estrés y el cansancio.

Componentes adicionales de una Historia de Usuario

Una Historia de Usuario puede tener componentes adicionales que permitan mejorar su entendimiento tanto para el usuario como para los Developers. Estos componentes también garantizan una ejecución precisa y acorde a las expectativas. Usualmente estos componentes son:

- Criterios de aceptación (detalle técnico)
- Estimaciones

- Prototipos, Diagramas, etc.

1. Criterios de aceptación (detalle técnico)

> Microsoft define a los criterios de aceptación como **las condiciones que un producto de software debe satisfacer para ser aceptado por un usuario, cliente o stakeholder.**

> Para Google, son **estándares pre-establecidos o requerimientos que un producto o proyecto debe satisfacer.**

Cada historia de usuario debe ir acompañada de sus criterios de aceptación. Estos criterios son esenciales para garantizar que todos en el equipo tengan el mismo entendimiento de lo que se espera. Ayudan a delimitar el alcance de una historia y a definir cómo se verificará que se ha cumplido con la necesidad o requerimiento descrito.

Concretamente en Scrum, se los define como un conjunto de sentencias redactadas de tal manera que conduzcan a una respuesta clara de "aceptado/rechazado". **Detallan las especificaciones técnicas de cada Historia de usuario o Tarea.**

El Product Owner es el responsable de redactar los criterios de aceptación para cada Historia de Usuario en conjunto con el cliente y/o los interesados, para posteriormente confirmarlos con los Developers antes de iniciar el desarrollo. Esta confirmación se realiza durante la Planificación del Sprint.

Los criterios de aceptación ayudan a los Developers a entender el funcionamiento del producto, de manera que estimarán mejor el trabajo necesario, además de servir como guía para tomar decisiones más acertadas de acuerdo con lo que se espera de cada Historia de Usuario.

Un criterio de aceptación debe poder probarse o testearse. Cuando no es posible probar algún criterio de aceptación es un claro indicio de una mala redacción.

Se debe tener en cuenta que al probar una Historia de Usuario frente a un criterio de aceptación se obtiene una respuesta binaria: Cumple o No cumple; **no existe el concepto de cumplir parcialmente un criterio de aceptación.**

Algunos beneficios de contar con buenos criterios de aceptación:

- Permite que los Developers entiendan una funcionalidad desde la perspectiva del usuario.
- Reduce las ambigüedades al desarrollar las Historias de Usuario.
- Promueve la calidad del producto permitiendo que los Developers se adhieran a los criterios de aceptación antes de llegar a una demostración de resultados en la Revisión

del Sprint.
- Permiten confirmar si una Historia de Usuario se comporta de acuerdo con lo esperado.

Tip para el Developer

No esperes hasta la Planificación del Sprint para hacer preguntas. Si un criterio de aceptación no especifica algo, no lo asumas. Es mejor preguntar y obtener claridad.
Si durante el desarrollo encuentras que un criterio de aceptación no es viable o requiere mucho más trabajo del estimado, comunica esto al Scrum Master lo antes posible.

Una manera **opcional** de construir los criterios de aceptación es la siguiente:

Cuando (Rol) hace (Acción) consigue (Resultado / Comportamiento esperado)

Veamos un ejemplo:

> **Historia de Usuario:** *Como* cliente, *quiero* pagar mi suscripción mensual vía sitio web por medio de transferencia bancaria o tarjeta de crédito, *para* evitar el desplazamiento hasta la sucursal de pago.
>
> **Criterios de Aceptación:**
>
> - Cuando el cliente ingrese al portal de pagos del sitio web, visualiza una opción (botón) para iniciar el proceso de pago de sus suscripciones.
> - El cliente tiene la opción de seleccionar su método de pago preferido: transferencia bancaria o tarjeta de crédito.
> - Si el cliente selecciona transferencia bancaria, se le solicitan los datos bancarios y se presentan las instrucciones. Si elige tarjeta de crédito, se le solicitan los datos de la tarjeta.
> - Al ingresar la información del pago, el sistema valida que todos los campos estén completos y que los datos correspondan a formatos válidos (por ejemplo, número de tarjeta válido).
> - Una vez realizado el pago, el cliente recibe un mensaje de confirmación en la pantalla y, adicionalmente, un correo electrónico con el recibo de pago.
> - Si ocurre algún problema durante el proceso de pago (como una tarjeta declinada), el cliente recibe un mensaje claro sobre el error y las recomendaciones para solucionarlo.

Tip para el Product Owner

En la discusión de las historias de usuario, puedes emplear la técnica de las 3 C's: Card (Tarjeta), Conversation (Conversación) y Confirmation (Confirmación). Esta técnica ayuda a mantener la simplicidad, ya que la información se mantiene ligera (tarjeta), se clarifica a través del diálogo (conversación) y se verifica (confirmación).

2. Estimaciones

Las estimaciones se refieren al tiempo o esfuerzo necesario para completar una Historia de Usuario o una tarea. Los equipos Scrum a menudo usan **puntos de historia, horas o días** para estimar la complejidad y el esfuerzo necesario.

Es muy importante realizar la estimación del trabajo requerido para desarrollar las Historias de Usuario y las tareas relacionadas, de esta manera se logran planificaciones más precisas.

La estimación **NO** es determinada por el cliente o el Product Owner, sino que son **los Developers en conjunto con el Scrum Master y el Product Owner,** quienes definen las estimaciones ya que ellos son quienes ejecutarán directamente el trabajo necesario.

□□ Tip para el Product Owner

Recuerda que las estimaciones son relativas y no deben interpretarse como compromisos rígidos. Sirven como guías basadas en la información y experiencia actual del equipo.

Para garantizar una estimación más precisa, se pueden considerar elementos como:

- Dificultad / complejidad: Hace referencia a qué tanto esfuerzo se requiere para ejecutar determinado trabajo. Se utilizan los Puntos de historia como unidad de medida.
- Duración: Hace referencia a cuánto tiempo se requiere para ejecutar determinado trabajo. Se puede utilizar horas o días como unidad de medida.
- Costo: Hace referencia al dinero requerido para la ejecución de un determinado trabajo, el cual se calcula a partir del trabajo de los developers más los recursos necesarios para su ejecución.

Uno de los problemas más comunes con la estimación de las Historias de Usuario, es sólo se estimar la dificultad (puntos de historia), o sólo estimar la duración (horas / días); por lo que es altamente recomendable estimar los dos. Muchos equipos prefieren unificar esto mediante una **relación entre los puntos de historia y la duración,** es decir que para cada punto de historia se asigna un tiempo determinado.

□□ Tip para el Scrum Master

Si sientes que las estimaciones consistentemente no reflejan el esfuerzo real de los Developers, aborda el tema en la retrospectiva del sprint. Analiza y ajusta el proceso de estimación en conjunto con el equipo según sea necesario.

El manejo que se da a los puntos de historia es **específico de cada equipo,** y es crucial entender que las estimaciones están íntimamente ligadas a la experiencia, conocimientos y dinámicas específicas de cada equipo, por lo que **NO** es recomendable comparar diferentes equipos basados en sus estimaciones: Incluso si dos equipos trabajan en el mismo producto o en productos similares, su ritmo de trabajo puede variar.

En Scrum el ritmo de trabajo también se conoce como la "velocidad" y se basa en cuántos puntos de historia, en promedio, un equipo puede completar en un Sprint. La velocidad no solo se basa en la capacidad de producción; también se ve influenciada por la comunicación dentro del equipo, las herramientas disponibles, las interrupciones externas, entre otros factores.

> **¿Por qué no comparar equipos basados en estimaciones?** Comparar equipos basados en sus puntos de historia o velocidad puede llevar a interpretaciones erróneas. Si un **equipo A completa 50 puntos** de historia en un Sprint y el **equipo B completa 30 puntos**, no significa necesariamente que el equipo A sea más productivo o eficiente. Puede que el equipo B esté manejando tareas más complejas o enfrentando desafíos que el equipo A no tiene. Comparar equipos de esta manera puede llevar a decisiones injustas y a presiones innecesarias, lo que puede afectar negativamente la moral y la productividad.

A continuación se muestra un ejemplo de ello:

Supongamos que tenemos dos equipos, Equipo A y Equipo B, ambos trabajando en diferentes funcionalidades del mismo producto.

El equipo A, con base en sus experiencias anteriores y en la comprensión que tienen sobre su capacidad y velocidad de trabajo, definió la siguiente tabla de estimaciones:

estimacion1
Image not found or type unknown

Por otro lado, el equipo B, que puede tener una composición diferente en términos de habilidades, experiencia y conocimientos técnicos, definió la siguiente tabla de estimaciones:

estimacion2
Image not found or type unknown

Durante la sesión de estimación, el Equipo A asigna 5 puntos a una historia de usuario específica, mientras que el Equipo B asigna 8 puntos a una historia similar. Esto no significa necesariamente que el Equipo B vea la historia como más complicada o que sean menos eficientes. Es posible que el Equipo B, basado en experiencias pasadas, considere ciertos riesgos o complejidades adicionales que el Equipo A no ha experimentado. Por otro lado, el Equipo A puede tener un experto en una herramienta o tecnología relevante que reduce la percepción de esfuerzo.

Si bien ambos equipos trabajan en el mismo producto, tienen distintas experiencias, habilidades y contextos que influyen directamente en cómo ven y estiman el trabajo.

□□ Tip para el Developer

Primero la calidad: No sacrifiques la calidad de tu trabajo para cumplir con una estimación. Si una historia de usuario necesita más tiempo del estimado para ser completada adecuadamente, es mejor discutir esto con el Scrum Master para ajustar el plan que entregar un producto inferior.

3. Prototipos, Diagramas, etc

Los prototipos, diagramas o esquemas de una Historia de Usuario nos sirven para mostrar un acercamiento al resultado que se espera obtener cuando los Developers ejecuten su trabajo, de esta manera se puede probar el producto antes de construirlo para identificar posibles errores y generar un mejor entendimiento para los Developers.

> **Prototipo:** Es una representación preliminar de un producto, que muestra cómo se verá y funcionará, sin necesidad de que esté completamente funcional. Puede ser desde un boceto en papel hasta una versión interactiva digital.

> **Diagrama:** Son representaciones gráficas que muestran las relaciones entre diferentes componentes, actores o flujos de un sistema o proceso.

Estos elementos visuales sirven como herramientas de comunicación entre el equipo, el Product Owner y los stakeholders, y pueden ser vitales para reducir la ambigüedad y mejorar la calidad del producto final.

Tip para el equipo

Un prototipo no tiene que ser perfecto ni estar completo: La idea es que sea una representación aproximada que pueda modificarse fácilmente en función de los comentarios recibidos.

Beneficios de usar prototipos:

- Permiten a los Developers desarrollar su trabajo en torno a una guia clara sobre las expectativas del cliente/usuario final.
- Permiten iterar y hacer cambios con rapidez antes de entrar en el desarrollo.
- Constituyen un gran apoyo para lograr mejores estimaciones de trabajo.
- Son más fáciles de abordar con los usuarios finales.
- Permite que el usuario se sienta involucrado en construcción del producto, ya que puede "verlo por adelantado".
- Se reduce el riesgo o la incertidumbre sobre el resultado de producto esperado.

Tip para el equipo

No todos los elementos o características necesitarán un prototipo o diagrama. La decisión de cuándo usarlos dependerá de la complejidad de la historia, los recursos disponibles y las necesidades del equipo y los stakeholders. Es importante recordar que, aunque estas herramientas son valiosas, deben usarse de manera estratégica para maximizar su valor sin desperdiciar esfuerzos en detalles innecesarios.

Épicas

Una épica es una gran pieza de trabajo que se puede desglosar en un conjunto de tareas más pequeñas (historias de usuario) que estan relacionadas entre sí. Puede considerarse como un **contenedor para historias de usuario que están relacionadas** y que, una vez completadas, cumplirán un objetivo o función más grande y más complejo.

Algunas ideas de las que se podrían definir las épicas son: • Módulos. • Componentes. • Hitos. • Entregables. • Funcionalidades.

> Se puede decir que una épica es un nivel de agrupación por encima de las historias de usuario que permite clasificarlas por funcionalidades, módulos, subsistemas, etc.

Las épicas:

- Son más grandes y complejas que las historias de usuario individuales.
- Se pueden descomponer en múltiples historias de usuario que, una vez completadas, satisfarán la épica en su totalidad.
- A menudo se requieren de múltiples sprints o ciclos de desarrollo para completarlas.

> El propósito de identificar épicas es ayudar a gestionar y organizar el trabajo del Product Backlog de manera más efectiva. Se trata de simplificar la complejidad y proporcionar una visión clara del objetivo final, permitiendo a los equipos trabajar de manera más coordinada y eficiente.

Recomendaciones sobre las épicas:

- Las épicas suelen **identificarse** en etapas tempranas del desarrollo, cuando el equipo se encuentra con grandes piezas de trabajo que son demasiado amplias para ser consideradas como una historia de usuario única.
- A medida que se acerca el momento de trabajar en una épica, el Product Owner en conjunto con el equipo **descompondrá la épica** en historias de usuario más pequeñas y manejables.
- Al igual que con las historias de usuario, **las épicas se priorizan.** Las épicas de alta prioridad se descompondrán antes que las épicas de baja prioridad.
- Dado que una épica se puede abarcar en múltiples sprints, se trabajará en las historias de usuario individuales que la componen a lo largo del tiempo.

> **Ejemplo de épica:** Renovar la experiencia de pago del sitio web para simplificar el proceso de compra y reducir el abandono del carrito.
>
> **Historias de Usuario derivadas de la Épica:**
>
> - *Como* cliente, *quiero* ver un resumen claro de mi pedido antes de confirmar el pago, *para* evitar errores.

> - *Como* cliente, *quiero* recibir un correo electrónico de confirmación inmediatamente después de hacer mi compra, *para* tener trazabilidad de la compra
> - *Como* cliente, *quiero* opciones de pago flexibles (tarjeta de crédito, PayPal, transferencia bancaria), *para* elegir dependiendo de mis necesidades y disponibilidad
>
> ... (y así sucesivamente, hasta que todas las facetas de la renovación de la experiencia de pago estén cubiertas).

Guía para identificar épicas sin morir en el intento ☝

- Las épicas suelen estar vinculadas a **objetivos estratégicos o de alto nivel.** Pregúntate: ¿Cuál es el gran objetivo o funcionalidad que queremos alcanzar?
- Si una historia de usuario parece **demasiado grande o ambigua,** es probable que sea una épica. Si al escribir una historia de usuario te encuentras agregando muchos **"y"** o **"o",** considera si puede ser desglosada en múltiples historias de usuario más pequeñas y, por lo tanto, convertirse en una épica.
- Si crees que una tarea o historia de usuario puede **extenderse más allá de un sprint**, es probable que estés frente a una épica.
- Discute las funcionalidades y objetivos con stakeholders y miembros del equipo. Ellos podrán proporcionar una perspectiva sobre la complejidad y el alcance, ayudando a identificar épicas.
- Si puedes desglosar un elemento en varias historias de usuario que son independientes entre sí pero que trabajan juntas para un objetivo común, entonces tienes una épica. (esto es que cada historia de usuario puede ser trabajada de forma independiente, pero que le apuntan al mismo objetivo).
- Utiliza herramientas como mapas de historias (story maps) para visualizar el flujo y las grandes áreas de funcionalidad. Estas áreas suelen ser buenas candidatas para épicas.
- Al identificar épicas, enfócate en el **"qué" (lo que se quiere lograr)** en lugar del **"cómo" (la solución técnica).** Las soluciones técnicas deben surgir al desglosar la épica en historias más pequeñas.
- A medida que el equipo avanza y se adquiere más conocimiento, es posible que algunas historias de usuario se expandan en alcance y se conviertan en épicas, o que algunas épicas se reduzcan y se conviertan en historias de usuario. Es esencial revisar y refinar regularmente.
- Si al intentar estimar una historia de usuario encuentras que el equipo tiene muchas incertidumbres o considera que llevará mucho tiempo, podría ser una señal de que estás tratando con una épica.
- Si un elemento tiene muchas dependencias o debe ser completado en un orden específico, podría ser una épica que necesita ser desglosada en historias de usuario que puedan ser trabajadas en paralelo o en un orden específico.

Revision #33
Created 24 February 2021 17:34:07 by CertMind
Updated 15 September 2023 18:58:16 by CertMind