

Product Owner:

**Gestión de Productos con
Metodologías Ágiles**



Unidad 5:

Historias de Usuario a detalle

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Presentación:

En la presente unidad entendemos como un product owner realiza una gestión efectiva del backlog. Sabiendo cómo coordinar la entrega de valor con la adquisición de conocimientos de manera sustentable.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Objetivos:

Que los participantes:

- Entiendan qué son y por qué usamos las historias de usuario.
- cuáles son las buenas prácticas.
- Aprendan cómo dividir historias para maximizar la entrega de valor y feedback.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Bloques temáticos:

1. ¿Qué son las Historias de Usuario?
2. ¿Por qué usar Historias de Usuario?
3. Componentes de las Historias de Usuario.
4. Tipos de Historias de Usuario.
5. Técnicas para descubrir Historias de Usuario.
6. Cómo crear buenas Historias de Usuario: el criterio INVEST.
7. Recorte orgánico (Slicing horizontal) de Historias de Usuario.
8. Refinamiento del Backlog (Refinement).
9. Problemas comunes de las historias.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

1. Los foros proactivos asociados a cada una de las unidades.
2. La Web 2.0.
3. Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

** El MEC es el modelo de E-learning colaborativo de nuestro Centro.*



Tomen nota:

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



1. ¿Qué son las historias de usuario?

La construcción de un producto digital (y posiblemente de cualquier otro tipo) depende fuertemente del intercambio de información y opiniones entre individuos con habilidades y perfiles muy diferentes; sumado a los usuarios y clientes que lo utilizan. El lograr que este intercambio sea eficiente determina éxito o fracaso del objetivo. Es necesario lograr entre ellos una escucha efectiva, aprendizaje compartido y transparencia, para poder generar un producto de valor.

Los marcos ágiles apuntan a resolver este problema con la entrega de valor frecuente de manera colaborativa. No solo para dar entregas rápidas sino también para aprender de ellas y poder reducir incertidumbre y errores causados por mala comunicación o entendimiento. Aquí es donde las historias de usuario entran en uso.

El concepto de historia de usuario proviene de XP y simplificando se define como una manera de describir ítems de valor para el usuario o el dueño del producto.

“En lugar de que el backlog del producto se utilice para describir nuevas funcionalidades, temas a investigar, defectos que corregir y demás, el backlog del producto debe describir ítems de valor para el usuario o el dueño del producto”.

Mike Cohn, Libro: User Stories Applied: For Agile Software Development

Habiendo dicho eso, no existe un limitante teórico para solo usar historias de usuario (o incluso el formato que veremos más adelante) pero las prácticas han demostrado que en la mayoría de los casos es la manera más eficiente de mantener el backlog.

Diferencia con los Casos de Uso

Tal como Mike Cohn nos explica en su libro (User stories applied, 2005), los casos de uso son una herramienta con ya muchos años en la industria del software, que fueron introducidos inicialmente por Ivar Jacobsen (1992), y los templates que más popularizaron y se siguen utilizando en la actualidad son los propuestos por Alistair Cockburn en 2001.

Una de las principales diferencias que se pueden destacar es el alcance o “tamaño”. Ambos apuntan a la entrega de valor y se ven desde el punto de vista del usuario, pero por definición las historias suelen tener un tamaño más reducido, ya que se lo suele restringir activamente para que pueda construirse en una iteración. En cambio, los Casos de Uso, si bien nada impide que se



manejen esos tamaños, suelen ser bastante más grandes conteniendo muchas veces múltiples flujos y escenarios que en las historias suelen desagregarse.

Otra característica es **la duración o tiempo de vida de estos artefactos**, los casos de uso se suelen entender como artefactos con una duración de vida extensa, que existen mientras el producto está activo y a lo sumo se actualizarán de acuerdo a la evolución que tenga. En cambio, las historias de usuario no están pensadas para vivir más allá del periodo de construcción y entrega, luego del cual simplemente pueden ser archivadas.

Como tercera diferencia, podemos indicar que los casos de uso tienden a detallar mucho más las interfaces e implementación, mientras que las historias de usuario tienen a ser más abiertas a dar libertad en estos aspectos. En versiones más modernas esta práctica no se recomienda (Cockburn 2001; Adolph, Bramble, et al. 2002), pero sigue siendo una práctica común y por lo tanto otra diferencia.

Otra diferencia es el objetivo por el cual se escriben (Davies 2001). Los casos de usos son escritos en un formato que permita llegar a un acuerdo entre desarrolladores y clientes, mientras que las historias de usuario se escriben para facilitar la planificación de releases e iteración, y servir de guía para una conversación sobre los detalles de la necesidad.

Habiendo expresado las diferencias, podemos remarcar que no debemos pensar en los casos de uso y las historias de usuario como dos artefactos mutuamente excluyentes, sino como herramientas que pueden complementarse para dar valor y facilitar el entendimiento y mantenimiento del producto.



2. ¿Por qué usar Historias de Usuario?

Para responder esta pregunta, considero útil comenzar con una frase de Hannah Arendt que dice: *"Storytelling reveals meaning without committing the error of defining it."*(sic) / *"Narrar historias revela significado sin cometer el error de definirlo"*. Justamente las historias de usuario buscan eso, explicar una necesidad sin tomar atribuciones de la manera de implementarla.



Los requerimientos, como se los entiende de manera tradicional, indican cómo el producto DEBE actuar, a detalle... todo el detalle. Esto proviene principalmente del enfoque más tradicional (cascada) en el cual con toda esta definición se debe ser capaz de planificar completamente las fases y actividades requeridas para completar el sistema.

Siguiendo la línea de la IEEE 830 la especificación debe implicar el uso de la frase "El sistema debe..." que es el formato recomendado por la IEEE para escribir requerimientos funcionales¹.

Con eso en mente, una especificación debería ser del estilo:

- a) *El sistema debe permitir a la compañía pagar por una propuesta de empleo.*
- b) *El sistema de aceptar Visa, MasterCard y American Express*

¹ Jake Bartlett



- c) *El sistema debe permitir cambiar la tarjeta de crédito antes de que el puesto esté publicado.*
- d) *El sistema debe brindarle al usuario un número de confirmación único.*

Este estilo de documentación se vuelve bastante tedioso para hacer y leer, es tendiente a errores (particularmente de omisiones) y muy consumidor de tiempo para hacerlo, leerlo y entenderlo.

Las historias de usuario se las relaciona más comúnmente con el enfoque ágil, con fomento y mayor apertura a la discusión y colaboración que los requerimientos. Esta discusión y colaboración puede presentarse hasta el momento mismo de estar planificando (e incluso en cierto nivel construyendo) la historia. En cambio, con los documentos de requerimiento se espera un gran nivel de detalle ya definido con anticipación por el solicitante, no siendo esperable la modificación durante su implementación.

Veamos un ejemplo de cada enfoque con una necesidad para un sitio de e-commerce:

Una historia de usuario siguiendo el formato estándar podría ser:

COMO un cliente.

QUIERO acceder fácilmente los elementos en el carrito de compras.

PARA conocer y modificar mi selección de compra actual.

La misma necesidad planteado como requerimiento podría verse como:

- El acceso al carrito debe mostrarse en la esquina superior derecha de la pantalla
- Mostrar una lista con una fila por cada ítem del carrito
- Cada fila debe contener la cantidad, número de artículo, nombre e imagen del producto
- El usuario debe poder remover cualquiera de los elementos
- El usuario debe poder modificar la cantidad de elementos

En el primer caso se habilita al equipo de diseño, desarrollo y QA a plantear dudas y propuestas, tales como aprovechar previsualizaciones, como deseamos tratar el caso de un descuento, comportamientos que deberíamos tener en cuenta para probar, si necesitáramos reducir el tamaño podríamos negociar funcionalidades que excluir, etc

Un ejemplo de el “desperdicio” que podemos encontrar al usar los clásicos requerimientos es cuando el área funcional entrega al área de elaboración la especificación, la cual es revisada y

reescrita por ellos en otro documento con su interpretación. Esto es teóricamente lo mismo visto desde otra perspectiva pero que al usar otros términos y/o definiciones puede ser algo distinto o al menos más abierto a interpretaciones que lo escrito originalmente, por lo cual quien especifica el último documento es quien tiene el mayor poder ante posibles diferencias al terminar, ya sea por comportamiento distinto u omisiones (negociando contrato sobre colaborando con el usuario).

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



3. Componentes de las Historias de Usuario

Las historias de usuario normalmente se las compone por lo que Ron Jeffries llama “las tres C's de las Historias de Usuario” (por sus siglas en inglés): *Card, Conversation, Confirmation*

Tarjeta (card)

Una "tarjeta" (en ocasiones puede ser un Post-It), es una representación física de algo que, sino sería una simple abstracción, una conversación que tuvo lugar en un momento del proyecto entre varias personas (con distintos roles) sobre una función que el sistema. Claramente no contiene toda la información que hacen al requerimiento, solo lo suficiente para identificarlo y recordarlo. Esta conversación es mayoritariamente verbal, pero usualmente se complementa/acompaña con documentación (gráficos, diagramas, definiciones, escenarios, etc).

Un formato estándar que se le suele dar es:

COMO [un rol o usuario particular],
QUIERO [acción que se quiere realizar],
PARA [un objetivo],

Es importante entender que este estándar (template) de historia de usuario, al igual que todo template, debe entender como una guía de la información y cómo expresarla, el template en sí mismo puede no ayudar si esto no se entiende e incluso en equipos maduros podemos flexibilizar de acuerdo a la dinámica que se tenga

Conversación

La historia propiamente dicha es comunicada por parte del solicitante al equipo de elaboración mediante una conversación, brindando un intercambio de pensamientos opiniones y sensaciones. Esta conversación suele realizarse al momento de estimar o refinar la historia, pero no solo debemos limitarnos a utilizar esos espacios. Un Product Owner discute/conversa con el equipo sobre las próximas historias a lo largo de todo el ciclo de vida del proyecto para obtener feedback, ideas y orientar en la visión. Pudiendo iterar sobre la misma para mejorar la solución, clarificar inconsistencias que se detecten, particiones, etc. La conversación es mayoritariamente verbal,



pero puede (y suele) complementarse con documentación, gráficos, pantallas, diagramas y/o todo lo que al equipo le aporte valor para la correcta implementación.

Confirmación

Un hecho difícil de admitir y aceptar es que sin importar cuanta discusión o documentación produzcamos no podemos estar nunca completamente seguros sobre lo que se realizará al tomar la historia de usuario. Para mitigar esto existe la prueba de aceptación, a través de la cual el cliente informa al equipo de elaboración como comprobará que el resultado cumple con lo requerido. Con esta definición de las pruebas de aceptación el equipo de elaboración podrá demostrar que implementa correctamente la historia de usuario.

Equipos menos maduros con la dinámica es posible quieran utilizar casos de uso, documentos de especificación, etc. Si bien pueden resultar útiles en algunas ocasiones no suelen tener una relación costo/beneficio satisfactoria y tienden a atentar en contra de los principios ágiles de colaboración sobre documentación exhaustiva, por eso lo ideal es tener ejemplos claros de los resultados esperados que hasta puedan automatizarse para lograr un enfoque más fluido y eficiente. Un derivado de estos criterios de aceptación es un mejor entendimiento y estimación del tema pudiendo también facilitar el corte de historias (tema que detallaremos más adelante).

Con la definición de cómo confirmar la historia, el equipo de elaboración al final de la iteración puede mostrarle al usuario el resultado de la misma y con esto obtener feedback que permita seguir desarrollando la aplicación.

Otros componentes que pueden formar parte de las historias de usuario

Criterios de Aceptación

Los criterios de aceptación son una parte fundamental en establecer alcance esperado de una historia. Cuando mencionamos las 3C's de una historia mencionamos justamente la "Confirmación" la cual es clave para saber que alcanzamos el objetivo de lo "conversado" por la "tarjeta". En estos criterios buscamos clarificar como debe y como NO debe comportarse el producto. Podemos también indicar formatos y visualización.

La intención de esto que cualquiera pueda validar el cumplimiento o no de la historia al terminarla. Si bien los primeros en pedirla van a ser posiblemente los homologadores, claramente a nivel construcción es importante para entender cómo hacer la construcción y para el PO expresar lo esperado.



Pueden ser de tipo:

- Funcionales (los más usuales), indican comportamiento del sistema
- No- funcionales, en los cuales indicamos características de diseño, técnicos, etc.
- Performance Criterio: En estos criterios podemos indicar tiempos de respuesta o de consumo de recursos.

Ejemplos:

Como usuario QUIERO poder cancelar una reservación realizada por mi

Criterio de aceptación

- Verificar que solo pueda cancelar reservas hechas por el mismo usuario
- De acuerdo al tipo de rol puede tener un recargo
- Se debe enviar un mail de confirmación.
- Se debe confirmar la recepción por parte del proveedor que la reserva está cancelada. Caso contrario indicar al usuario que intente más tarde o contacte al servicio de posventa.

Escenarios

Los escenarios son una sección clave en los casos de uso que fueron incorporados en las historias de usuario de una manera más reducida. La intención es representar una secuencia de pasos que realiza el usuario en cuestión en donde cada paso se escribe como una oración sobre una meta que se cumple. Un template estándar que se suele utilizar es:

DADO QUE [sucede algo]
CUANDO [en que contexto]
ENTONCES [comportamiento esperado del sistema]

Esto no es mandatorio para todas y puede ser demasiado detalle para una historia en sprint. Se suelen encontrar de utilidad al generar las épicas, ya que permiten dar contexto, facilitar el slicing de historias y por sobre todo suelen fijar el alcance para el armado de casos de prueba y el equipo debe solicitar al PO todo caso en donde necesita su existencia.



Definición de Listo (Ready)

El llamado Definition of Ready en los textos originales, indica el criterio de inicio para tomar una historia, podemos estar hablando de un impedimento técnico (la instalación de un servidor o disponibilizar un servicio) así como también de una especificación. Esto suele acordarse con el equipo de elaboración en los refinamientos para saber qué impedimentos tendríamos si quisiéramos incluirlas en el próximo sprint y desbloquearlo con anticipación.

Definición de Hecho (Done)

La definición de hecho (DoD - Definition of Done) es un acuerdo realizado por el equipo para validar cuando una historia puede considerarse completada, evitando cuestionamientos posteriores y pudiendo planificar claramente sobre qué estará “hecho” y cuáles son las actividades y entregables abarcados en la historia.

Es importante destacar que esto no hace referencia solo a la funcionalidad entregada en si misma sino también a la calidad e ítems asociados requeridos para terminarlo; puede verse como una lista de actividades que completamos por defecto al hacer una historia (Codificación, Prueba unitaria, Generación de Casos de prueba, actualización de documentos, etc.). Estas actividades deben entenderse como de valor para los entregables (descartando toda actividad que no lo hace) y ser conocidas por el Product Owner y equipo de elaboración.

Esta definición no debe ser tomada a la ligera por el equipo, ya que es la manera en que el equipo considera puede garantizar calidad y sustentabilidad de sus entregas. Idealmente en Scrum una historia “hecha” implica potencialmente implementable en producción, pero se suele adaptar a la realidad que el equipo puede alcanzar normalmente en un sprint, la cual a medida que el equipo madura puede modificarse y evolucionar.

La definición de done también puede utilizarse a niveles mayores como Sprint o Release, a través del cual indicamos con anticipación como consideramos su cumplimiento.

¿Es la única manera de "Hacer"?

En la visión industrialista de construir hacemos foco en la producción o salida (output) del proceso de creación. Traducido a productos digitales es entregar código en producción, mayor cantidad de funciones o historias implementadas. En una gestión de producto moderna esto nos debe hacer ruido, ya que **nuestro foco debe ser lograr el mayor impacto posible al cliente con el menor esfuerzo**. El terminar ítems genera una sensación de avance y satisfacción personal, pero si no estamos beneficiando al usuario o negocio estamos destinando erróneamente recursos.

Es clave responder al feedback hasta que el resultado deseado es conseguido. Para esto debemos poder medir y tomar decisiones rápidas en ciclos cortos, esto se puede (y debería) llevarse a cada historia, pudiendo tener iteraciones en ella (sprints dentro de sprints) para ir validando y decidiendo constantemente cuantos recursos invertir.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



4. Tipos de Historias de Usuario

Necesidades de usuario

Son aquellas que hacen referencia a una necesidad de un rol usuario del sistema. Se podrían considerar las historias de usuario “per se” y deberían representar la mayoría de los ítems del backlog en general, ya que son los entregables de valor directo.

Historias “técnicas”

Este tipo de ítem suele generar algunos debates sobre si deben o no incluirse en el Product Backlog, algunos autores (especialmente décadas atrás) apuntaban a que el backlog solo contuviera necesidades del usuario, por lo cual todo ítem que no generará un entregable pedido por el quedaba excluido.

Esto claramente deja sin lugar a muchos temas que, si bien no son pedidos por el usuario, son necesidades de la aplicación o de perfil técnico que, si no realizadas se acumulan como deuda técnica, problemas a futuro o incluso incumplimientos normativos o de soporte. Los ejemplos más clásicos son la instalación de un nuevo parche en la base de datos, modificación de un framework de desarrollo, cambio de servidor, etc. que va a requerir del equipo (múltiples roles) y consumir tiempos del sprint, razón por la cual deben ser negociados por el PO para evaluar su prioridad con relación a los demás ítems y definir cuándo pueden ser tomados. Cabe mencionar que debe cumplir con los parámetros de los demás tipos de historias, caso contrario deberá darse un trato diferencial.

Historias para adquisición de conocimiento (spikes)

Este tipo de historias de usuario que no generan un entregable concreto o implementable para el usuario es un concepto que también viene de XP (eXtreme Programming) en analogía a la práctica realizada al escalar montañas. Cuando se escala se suele detener para introducir una estaca en la roca, esto no es escalar en sí mismo (no nos acerca a la cima) pero nos sirve de fundamento para los próximos pasos de escalar. De manera similar en el software, un Spike no produce código, pero permite la generación de nuevas historias de usuario en el backlog y nos sirve para entender módulos que nunca fueron tocados por el equipo, usos nuevos a herramientas



tecnológicas, respuesta del entorno y con esta información trabajar con mayor certidumbre sobre demás las historias de usuario.

Historias de limitaciones o restricciones (constraint)

Este tipo de “historias” no son entregas de valor concreto, sino que son necesidades no funcionales que tienen los usuarios. La mayoría de los autores expertos en el tema, indican que se manejar fácilmente con el mismo formato las llamadas “Historias de restricciones”. Como por ejemplo

COMO comprador del sitio

QUIERO recibir en menos de 2 segundos la respuesta de confirmación de stock.

Criterio de aceptación:

- Tiempo medido desde el clic en la opción para “validar stock”
- Tener 100 usuarios activos en el sistema y al menos 10 consultas en paralelo sobre el stock.

Niveles o Jerarquías en las historias de usuario

Las historias de usuario son una herramienta que ha probado su eficacia para la entrega de valor frecuente al usuario en el flujo de Scrum. Como vimos en las unidades anteriores el "tren" de construcción tiene distintos niveles de planeamiento, el más mencionado siempre es el nivel sprint que dura entre 1 y 4 semanas y luego tenemos planificaciones más grandes tales como Release (que en cierto sentido podemos verlo también como una iteración en sí misma) que puede durar varios meses. De acuerdo al periodo en que estamos tratando es el nivel de detalle o tamaño de las historias que deberíamos estar visualizando. Imaginémonos que si tenemos historias con un tamaño que permiten incluir 9/10 en un sprint, si tuviéramos 6 sprints en un release ya estaríamos hablando de 54/60 historias, lo cual puede resultar demasiado numeroso para negociar y priorizar, en especial porque también implicaría tener un alto nivel de detalle de los temas a construir en sprints que sucederán meses más adelante, lo cual (como también vimos en unidades anteriores) desperdicia los beneficios del enfoque iterativo adaptativo de Scrum. Es por este motivo que suelen utilizar historias menos detalladas y más abstractas que faciliten la planificación de más largo plazo y luego refinamos para generar las historias que se planificaran a nivel sprint.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



- a) Historias: tal como mencionamos antes, una historia de usuario es un pequeño deseo que el usuario tiene para el sistema y que puede ser tomado por el equipo de elaboración para entregarlo en el tiempo de un sprint (ver más detalles en la definición de historia de usuario). Debe poder descomponerse en tareas/actividades accionables por el equipo durante el sprint.
- b) Épicas: Podemos definir la definición de una épica como una historia de usuario muy grande que sabemos/suponemos será desagregada en partes más pequeñas (o historias contenidas) al momento de implementarlo, justamente su nombre se debe a esa razón en la cual podemos pensar en la saga del Señor de los Anillos como una Épica compuesta de historias más chicas que se van relatando. Este tipo de historias suelen tener un mayor nivel de abstracción y menor cantidad de detalle, lo cual implica un mayor sesgo para estimarlas, pero se acepta como válido dado que son utilizadas para la planificación de mayor plazo. Estas deberán bajarse a detalle y decantar en historias que sí puedan ser tomadas en una iteración o sprint de pocas semanas para su construcción.
- c) Temas: Los temas podemos verlos como una manera de agrupar historias de usuario que están relacionadas. Si bien uno podría decir que todas las historias están relacionadas ya que son sobre el mismo producto, cuando hablamos de una relación para agrupar podemos decir que son todas las que afectan a un determinado tipo de usuario, o impactan en una necesidad técnica particular.
 - Facilitan validar la consistencia y completitud
 - Permite priorizar por temas y luego historias
 - Suele facilitar refinamiento y priorización

5. Técnicas para descubrir Historias de Usuario

No necesitamos irnos a buscar bibliografía muy antigua para encontrar el uso de las palabras elicitación y captura de requerimientos. Podemos encontrarlas en libros que son cabecera para identificar requerimientos escritos por autores reconocidos tales como (Kovitz 1999; Lauesen 2002; Wiegers 1999 and Jacobson, Booch and Rumbaugh 1999).

Estas palabras dan la sensación de que todos los requerimientos del proyecto están disponibles y son conocidos en detalle y solo requieren de buen entrenamiento y técnica para su especificación. En cambio, Robertson and Robertson (1999) eligen relacionar el proceso con el término “trawling” (pesca con red) lo cual nos da claramente la idea de que los requerimientos se pueden ir “hallando” mientras se avanza con el bote, incluso pudiendo usar redes de distinto tamaño para elegir el tamaño de los requerimientos que queremos “atrapar”, pudiendo priorizar los “peces grandes” y luego hacer otras pasadas con el fin de capturar los medianos dejando los más chicos para después. Esta metáfora aplica de manera bastante directa con tamaño de valor para el negocio y/o decisiones técnicas. Una segunda ventaja de esta manera de pensar los requerimientos (tal como menciona Mike Cohn, 2009) es que no los identifica como entidades estáticas sino como organismos vivos que nacen, maduran y mueren. Por lo tanto, los requerimientos de hoy pueden no ser necesarios mañana, así como también peces chicos de hoy pueden haber crecido y ser los peces grandes de mañana (máxima prioridad).

Si bien el enfoque ágil reconoce valora la adaptación y el maximizar el trabajo no realizado, evitando un relevamiento minucioso en estadías tempranas del proyecto dado la dificultad y la posibilidad de que cambien durante el curso del mismo, se debe hacer un intento de visualizar a alto nivel el alcance esperado del producto. Aquí yace una de las ventajas de las historias de usuario sobre los requerimientos, ya que por naturaleza permiten diversos grados de granularidad y detalle pudiendo quedar abierta para posteriores refinamientos.

Habiendo expresado eso, debo remarcar que no es buscada una clásica fase de análisis de meses en la cual estamos escribiendo historias de usuario, sino utilizar técnicas colaborativas para plantear un roadmap a largo plazo, que luego puede descomponerse en releases de mediano plazo los cuales serán refinadas para poder tomarse en sprints a medida que el periodo de tiempo se acorta.

Técnicas más comunes para descubrir Historias de Usuario

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



No existe una técnica que aplique perfecto para todo contexto, ni una manera de obtenerlas todas juntas. En lugar de eso, es un proceso iterativo y de múltiples acercamientos para cubrir el detalle. Si bien con las metodologías ágiles surgieron nuevos estilos de ceremonias y prácticas que pueden aplicar, nada limita a utilizar técnicas ya probadas en el pasado, solo teniendo en cuenta de que sean sencillas y no intrusivas para que puedan aplicarse de manera continua.

- Entrevistas con el usuario, Las entrevistas con usuarios es la técnica más tradicional para comenzar el relevamiento y obtención de historias. Un factor clave para obtener resultados satisfactorios es seleccionar los correctos entrevistados (keyusers), los mismos deben estar involucrados en el proyecto y deben cubrir diversos roles y áreas involucradas. ¿El proceso de entrevistar para conseguir las historias nunca es tan simple como preguntar “Que necesita?”, esto requiere un esfuerzo, técnicas y prácticas que faciliten una conversación fluida haciendo las preguntas que lleven a eso.
- Cuestionarios, Este tipo de herramientas es particularmente útil para obtener información de un número muy grande de gente, así como también poder priorizar historias, pero limitado para obtener la esencia del producto y bajarla a historias, por lo tanto se recomienda usarla como complemento en casos que se presenten con esas características.
- Observaciones, Esto consiste en compartir las actividades cotidianas y usos del sistema. En ocasiones la posibilidad de aplicar esta técnica es limitada pero el poder presenciar cómo interactúan con el sistema y como este se comporta tiene un enorme potencial para poder conocer la experiencia usuario y tener un feedback valioso para la evolución del producto. Esto es justamente un motivador para realizar sprint reviews y también tener releases cortos que nos permitan tener feedback y validaciones.
- Workshops, este tipo de reuniones es lo que considero el estilo más clásico para obtener historias de usuario de manera ágil. Estos eventos en general involucran desarrolladores, usuarios, analistas funcionales, gente de procesos y demás roles que puedan contribuir a las historias. Un formato muy conocido es el de Story Mapping (detallado en otra unidad) por el cual con rapidez pueden setearse expectativas, marcar prioridades, definirse un MVP y próximos releases, con un alto nivel de abstracción y menor detalle cuanto más lejano y/o prioritario sea un tema.



6. Cómo crear buenas historias de Usuario: el criterio INVEST

El criterio INVEST ("Invertir" en inglés) es un nemónico muy popular creado por Bill Wake, reconocido autor y profesor de temas tales como eXtreme Programming o design patterns, está aceptado como criterio de facto para validar si tenemos o no una buena historia de usuario, caso negativo el equipo puede reformular o incluso reescribirla para cumplir con ellos.

Detallaremos que representa cada letra:

Independiente

Para facilitar su estimación, priorización y planificación no debería haber un acoplamiento (al menos fuerte) entre historias de usuario. Se entiende que puede existir (estamos trabajando sobre un mismo sistema) pero debemos tratar de minimizarlas lo más posible.

Negociable

Los detalles de una historia deben ser negociables, este punto nos recuerda que no son una definición de requerimientos realizados en una etapa de análisis que debe concretarse cual contrato, sino que es un elemento que se debate en equipo y sus detalles negociados para acordar una implementación.

Esta negociación y/o debate busca que se capture la esencia deseada por el negocio de manera colaborativa para tener una solución implementable en una iteración o dividida en múltiples entregables más chicos que así lo permitan evitando debates posteriores entre las partes por no haber entendido el objetivo y/o conceptos principales y el porqué de ellos (ejemplo cuestiones normativas, limitaciones contractuales, etc).

Valioso

Las historias deben aportar valor al usuario o cliente (idealmente para ambos), los clientes eligen pagar por ellas y los usuarios las utilizan, por lo tanto, si una historia no es valiosa para ellos no tiene lugar el backlog del producto. Es difícil imaginarnos que al construir una casa construyamos una habitación que no tiene uso, lo mismo con nuestro producto, por lo tanto, si una historia no tiene valor debe descartarse (siguiendo el principio ágil de maximizar el trabajo no realizado).



En este punto nos encontramos con el dilema de aquellos temas que son de valor técnico pero el usuario puede no valorarlos haciendo difícil su inclusión, por lo cual debe negociarse entre el product owner y el equipo para entender por qué invertir recursos en ellas y no en historias de usuario y así priorizarse en el backlog.

Estimable

Las historias de usuario deben ser estimables por el equipo que diseña, construye y las prueba, esto permite poder compararlas en “tamaño” relativo y su correspondiente costo para terminar de definir la prioridad de cada una en el backlog y/o poder realizar un refinamiento o desagregación en historias más chicas que permitan una implementación más eficiente.

La estimación además presenta otro valor que es poder comprobar el entendimiento y/o certidumbre que tiene el equipo de la misma para poder estimarla. Si es muy grande se va a requerir trabajar con el PO para poder desagregar en partes más manejables ya sea por desconocimiento, riesgo o complejidad.

Small / Size appropriately - Pequeña / De tamaño adecuado

Al momento que el equipo de elaboración toma una historia para su implementación, la misma debe poder realizarse dentro del intervalo de tiempo e idealmente en ese periodo de tiempo deseamos poder trabajar en varias historias, por lo tanto, el tamaño debe permitirlo. A su vez si hablamos de una historia que no estaremos tomando en el corto plazo puede la definamos en menor detalle y con mayor tamaño (concepto de “Épica”) lo cual es un adecuado para algo que solo pensamos a nivel Release y/o para sprint más lejanos que luego dedicaremos esfuerzo en refinar.

Testable

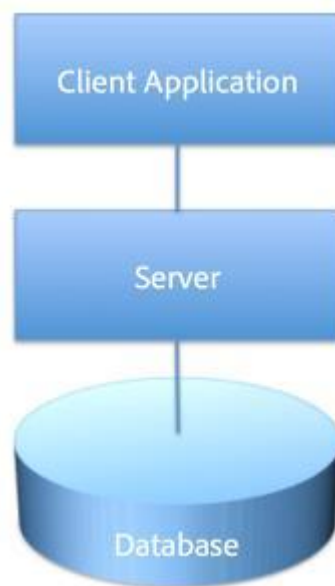
Una historia debe poder validarse claramente que cumple o no un criterio de aceptación o confirmación (como fue explicado antes en las 3 C's de una historia de usuario). Estos criterios de aceptación usualmente facilitan el entendimiento y estimación de las historias.

Es posible que en historias grandes (o épicas) no se puedan expresar casos de prueba o sean demasiados los incluidos, siendo una señal de que debemos desagregar antes de planificarla en el sprint y previo a eso no los incluyamos en detalle.



7. Recorte orgánico (Slicing horizontal) de Historias de Usuario

Cuando tratamos de llevar el paradigma de trabajo tradicional a la división de historias nos puede llevar a hacerlo utilizando los llamados **cortes horizontales** en los cuales hacemos los cortes guiados por las capas de arquitectura; por ejemplo, realizar las modificaciones en la base de datos, la capa lógica y por último la interfaz usuario, luego de lo cual se integra y se prueba el funcionamiento integral.



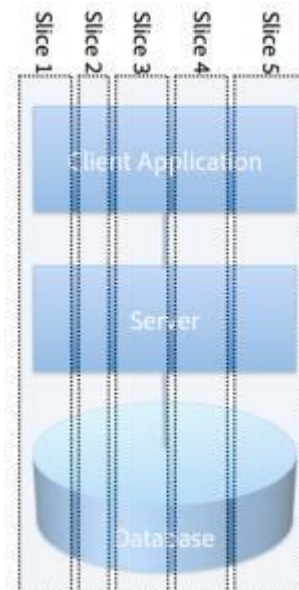
¿Por qué cortar las historias?

Esta organización permite una alta especialización de cada área busca en teoría asegurar organización, y si bien puede sonar intuitivo y lógico tiene como puntos negativos la demora en obtener feedback y el trabajo en “silos” que atenta contra mirada más holística del paradigma ágil. En cuando buscamos acelerar el feedback necesitamos buscar cortes que atraviesan diversas capas y nos permitan tener software operativo buscando que los diversos roles realicen sus partes de manera altamente integrada y colaborativa para su integración, testeo y rápida posibilidad de release

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Este tipo de recorte es el buscado cuando seguimos el paradigma ágil, en inicio puede verse como ineficiente ya que implica varias veces trabajar sobre una misma parte, pero esta supuesta “ineficiencia” se ve compensada por tener feedback frecuente del usuario para detectar diferencias y ganar aprendizaje sobre software construido.

Existen varias heurísticas para el llamado slicing de historias, detallaremos dos patrones de los autores más reconocidos sobre el tema.

Los patrones definidos por Richard Lawrence²:

Patrón 1: Pasos de workflows

COMO administrador de contenido
QUIERO publicar artículos en el sitio corporativo
PARA el acceso de nuestros clientes

Esta historia puede parecer bastante simple y concreta, hasta que empezamos a indagar en un refinamiento y vemos que de acuerdo al tipo de historia, rol o incluso en todas, este es un proceso

² agile for all; vertical slices and scale



de varios pasos que requiere revisiones, correcciones, etc. Las cuales si buscamos realizarlas desde el inicio hacen difícil que se termine la historia en una iteración o al menos que podamos incluir otras.

Una manera que nos reduce la complejidad sería particionar la historia por los distintos pasos que puede tener para construir la opción más simple y directa que podamos mostrar y validar en funcionamiento, sobre la cual luego incorporamos los distintos pasos en otras historias:

Dividiríamos inicialmente en:

COMO administrador de contenido

QUIERO publicar artículos directamente en el sitio corporativo sin validaciones

Y luego:

COMO administrador de contenido QUIERO publicar artículos con revisión del editor.

... QUIERO publicar artículos con revisión del sector legal.

... QUIERO publicar artículos en el sitio de homologación.

... QUIERO publicar artículos desde homologación en producción.

Esta división permite por lo pronto poder publicar directamente en el sitio, que suena lo de mayor valor, quitando las validaciones por parte de las otras áreas que tal vez solo suceden en la minoría de los casos o que podemos hacerlo manualmente mientras ya utilizamos el software producido. Justamente por esta mención es que hacer un corte por cada paso sería incorrecto, porque al construir historias de esa manera estaríamos haciendo algo similar a los cortes horizontales por arquitectura, ya que al finalizar cada historia no tendríamos algo operativo.

Patrón 2: Variaciones en las reglas de negocio

Este tipo de patrón puede darse cuando existen distintas reglas complejas para hacer la operación. Así por ejemplo en la siguiente historia:

COMO usuario

QUIERO buscar por mis operaciones bancarias con fechas flexibles

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Al refinar "Fechas flexibles" podemos encontrar diversas reglas:

COMO usuario QUIERO buscar por mis operaciones bancarias entre X e Y

... QUIERO buscar por mis operaciones bancarias en una semana de diciembre

... QUIERO buscar por mis operaciones bancarias desde X hasta el día de hoy.

... QUIERO buscar por mis operaciones bancarias N días desde X

Siendo cada una independiente esta división nos permite reducir la complejidad en partes.

Patrón 3 Mayor Esfuerzo

En ocasiones una historia puede contener el mayor esfuerzo en una parte que luego es aprovechada por otras, así por ejemplo puedo tener

COMO usuario

QUIERO pagar la compra con VISA, MasterCard y American Express

En esta historia la mayor complejidad podría estar en la infraestructura para el pago electrónico, luego de preparado para una, sería aprovechada para implementar las otras 2. Por lo cual podríamos escribirlo de la siguiente manera:

COMO usuario

QUIERO poder para con una tarjeta de crédito (de Visa, MasterCard o American Express)

... QUIERO poder pagar con los 3 tipos de tarjeta de crédito (de Visa, MasterCard o American Express).

El no expresar una tarjeta puntual de manera inicial nos da libertad para que el PO priorice a último momento cual es la tarjeta con la cual empezar y de acuerdo a eso estimar las restantes.

Patrón 4 Complejo/Simple

Es posible que en ciertas historias al ir pensando los distintos escenarios existen cada vez más opciones y complejidades

En ese sentido nos conviene priorizar un escenario que nos sirva como la implementación desde la cual agregaremos los siguientes.

Así por ejemplo podemos tener:

COMO usuario

QUIERO poder realizar la búsqueda de los vuelos entre 2 destinos

Siendo que existen muchas opciones de búsqueda para esos parámetros podemos dividirlos de la siguiente manera:

COMO usuario QUIERO buscar vuelos directos entre 2 destinos

... QUIERO buscar vuelos con N paradas entre 2 destinos

... QUIERO buscar vuelos entre 2 destinos incluyendo aeropuertos cercanos.

Patrón 5 Variaciones en Datos

En ocasiones los tipos de datos que podemos tener son extremadamente variados lo cual impide cerrar un alcance claro. Esto puede verse por ejemplo en los sistemas de venta de pasajes, para los cuales las diversas empresas tienen sus propias firmas y reglas internas que permiten distintas opciones de acuerdo a como se lo requiera, en este caso nos conviene seleccionar la que consideramos de más valor (costo/beneficio) e ir agregando opciones a continuación.

COMO usuario QUIERO buscar pasajes entre 2 destinos

En este caso podemos iniciar por el proveedor de pasajes que utiliza el formato que queremos más práctico y creamos las siguientes historias:

COMO usuario

QUIERO buscar pasajes entre A y B con proveedores que permitan cargar origen y destino indicados como ciudades

... QUIERO buscar pasajes entre A y B con proveedores que devuelvan las ciudades más cercanas de cada uno



Patrón 6 Manera de entrada de datos

En ocasiones la complejidad puede estar dada por la interfaz usuario en lugar de por la funcionalidad en sí misma. En cuyo caso podemos dividir la historia en la implementación de la funcionalidad en sí misma y luego incorporar las mejoras de UX

COMO usuario

QUIERO poder realizar la búsqueda de operaciones entre 2 fechas (cargando manualmente en formato DD/MM/YYYY)

... QUIERO poder realizar la búsqueda de operaciones entre 2 fechas con un calendario gráfico.

Patrón 7 Diferir performance

En ocasiones el optimizar los tiempos de respuesta pueden requerir un esfuerzo importante para llegar a lo deseado, pero es algo que tal vez podemos diferir para un siguiente paso y poder avanzar en obtener feedback del resto de la funcionalidad (primero hacer que ande, luego hacerlo rápido)

COMO usuario

QUIERO poder buscar las operaciones entre 2 fechas (aunque tenga baja performance, indicar un icono animado de en proceso)

... QUIERO poder buscar en menos de 5 segundos,

Patrón 8 Operaciones

En muchos casos se utiliza la palabra "administrar" la cual puede tener aparejada múltiples operaciones que pueden implementarse por separado (y algunas diferirse con algún proceso manual momentáneamente). Por ejemplo:

COMO usuario QUIERO administrar mi cuenta

Puede dividirse en:

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



COMO usuario QUIERO poder crear una cuenta
... QUIERO editar los datos de mi cuenta
... QUIERO eliminar mi cuenta.

Patrón 9 Crear una Spike

Una historia puede ser costosa a causa de una alta incertidumbre, en cuyo caso lo aconsejable es hacer una spike con duración fija que nos permita obtener información sobre la cual decidir. Debemos ser claros en los objetivos a cubrir y los mismos no deberían ser ambiguos o demasiado extensos.

Por último, incluyo el diagrama realizado por el autor para guiarnos en esto. Puede resultarnos muy útil para recordarlo durante el refinamiento.





S.P.I.D.R de Mike Cohn

Este es el enfoque que Mike Cohn creó para desagregar historias de usuario rápidamente. Él lo denomina S.P.I.D.R. (práctica común en los estadounidenses de usar anagramas, en este caso similar a Spider- araña en inglés-). Esta heurística es un tanto más simple que el anterior, aunque tiene algunas prácticas compartidas. Los patrones son:

- **Spike**, En lugar de tener una historia grande con alto nivel de incertidumbre, mejor hacer una pequeña investigación desde la cual obtener los datos requeridos para que el equipo obtenga más información. Con esta información poder estimar correctamente y facilitar la desagregación de historias en tamaños manejables
- **Paths (Camino o secuencia)**; en estos casos podemos evaluar los flujos incluidos en la historia y dividir cada uno en una historia distinta. También se puede considerar que cada paso será una historia en sí misma.
- **Interfaces**, Una opción bastante usada que suele ser simple es dividir las historias por interfaces, e incluso si sigue siendo demasiado grande por tecnología (explorador o dispositivo). Es muy recomendable iniciar con una construcción mínima de la interface dejando su estilización para un paso posterior.
- **Data (Datos)**, En muchos casos podemos necesitar trabajar con múltiples formatos de datos. Considerar cuales son los más usados o críticos pueden ser una buena manera para dividir la historia, incluso utilizar una versión simplificada para probar alguna hipótesis también puede probar ser útil.
- **Rules (Reglas)**, En ocasiones la existencia de validaciones, tecnologías, estándares, etc pueden complejizar las historias. Reducir su aplicación en primeras versiones es una práctica útil para dividir las historias, incorporando las reglas en historias posteriores.

Prácticas usuales

A las prácticas recomendadas por estos reconocidos autores, agrego varias que considero útiles:

- Por tipos de usuario, de acuerdo a las necesidades de los diversos roles y “personas” evaluar sus prioridades y valor económico para saber por dónde empezar.
- Diferir performance y arquitectura en general, en este caso es particularmente útil al inicio de la vida del producto, donde queremos validar la hipótesis de negocio antes de invertir más recursos.



- Diferir la interfaz de usuario, al menos las funcionalidades no críticas o puramente estéticas. Este enfoque obviamente es más simple en backoffice que en frontends de productos de empresas .COM.
- Según los datos, podemos inicialmente no aceptar todos los diversos tipos o formatos de datos haciendo foco en el/los más comunes. Así por ejemplo solo trabajamos con medidas en metros/ kilómetros y luego incorporamos la posibilidad de trabajar en millas y pulgadas.
- Crear una Spike cuando tenemos mucha incertidumbre, Haciendo foco en aprender
- El camino "feliz" completo sacrificando toda validación e ir incorporando progresivamente nuevos caminos o validaciones.
- Por Reglas de negocio, este caso similar al anterior, uno puede demorar su implementación o realizarlas manualmente.
- "Mockear" las interacciones que incluyan a sistemas externos. Esto nos permite destrabar el avance y poder validar lo demás hasta tener la interfaz formal.
- "Paretizar" las funcionalidades, para quedarnos con lo más valioso hasta llegar a un tamaño "manejable".

Ejemplo de slicing

Imaginemos un nuevo equipo multidisciplinario (UX, gente de frontend, backend, analista funcional) que recibe la asignación de lanzar un sistema de venta online para comenzar a comercializar los productos de una empresa con múltiples sucursales, pero sin presencia web, lo cual se considera fundamental para hacer crecer su facturación.

1- Funcionalidades requeridas:

- Cargar Producto
- Seleccionar productos
- Realizar Compra (con tarjeta, MP, en efectivo, etc.)
- Envío a domicilio

2- Primer mapeo de Historias de Usuario

US1: COMO usuario del e-commerce.

QUIERO seleccionar múltiples productos.



PARA comprarlos y obtener la mercadería.

Criterio de aceptación:

- Múltiples productos
- Editar la selección (cantidad, agregar, eliminar, etc)
- Acceder a la descripción de los productos
- Tener envío a domicilio
- Retirar en sucursal

US2: COMO administrador

QUIERO cargar el catálogo de productos

PARA disponibilizarlos a los clientes

Criterio de aceptación

- Quedan accesibles para realizar compras
- Se realiza un control de stock, ocultado aquellos no disponibles.
- Se pueden cargar imágenes
- Características técnicas
- Descripción de los productos
- Listado de fabricantes

3- Primer corte: Selección de productos / efectuar compra

US3: COMO usuario del e-commerce

QUIERO poder seleccionar productos

PARA tenerlos disponibles en mi carrito de compra

Criterio de aceptación

- Múltiples productos.
- Editar la selección (cantidad, agregar, eliminar, etc).

US4: COMO usuario del e-commerce

QUIERO comprar la selección disponible en mi carrito

Criterio de aceptación

- Poder pagar con Tarjeta de Crédito.
- Poder pagar con MercadoPago.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



- Poder pagar contrareembolso.
- Poder pagar en sucursal.
- Tener envío a domicilio.
- Retirar en sucursal.

4- Obtención de un MVP

US5: COMO usuario del e-commerce

QUIERO poder seleccionar UN producto

PARA tenerlo disponible en mi carrito de compra

Criterio de aceptación

- Queda almacenado en un carrito de compra
- Puede visualizarse.

US6: COMO usuario del e-commerce

QUIERO comprar el producto seleccionado en mi carrito con pago y retiro en sucursal

Criterio de aceptación

- Poder pagar en sucursal
- Retirar en sucursal

Para la historia de US2, se acuerda con el usuario no realizar desarrollos. La carga se realizará manualmente en la base de datos con la intención de evaluar primero el comportamiento del usuario antes de invertir recursos en el backoffice.

5- Porciones de US4

US7: COMO usuario del e-commerce

QUIERO pagar el producto seleccionado en mi carrito con pago electrónico desde la web

Criterio de aceptación

- Poder pagar con Tarjeta de Crédito
- Poder pagar con MercadoPago
- Realizar retiro en sucursal

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



En esta historia el equipo decide aplicar slicing dada la independencia entre ambos tipos de pago y el esfuerzo involucrado en c/u

US8: COMO usuario del e-commerce

QUIERO pagar el producto seleccionado en mi carrito con MercadoPago

US9: COMO usuario del e-commerce

QUIERO pagar el producto seleccionado en mi carrito con Tarjeta de Crédito

US10: COMO usuario del e-commerce

QUIERO poder seleccionar múltiples productos

PARA tenerlos disponibles en mi carrito de compra

Criterio de aceptación

- Poder eliminar cualquiera de las selecciones

Ante desconocimiento técnico el equipo sugiere no implementar la edición desde el carrito, Comenzando con una versión en la cual se deba eliminar y luego agregar nuevamente en ese caso y luego implementar esa funcionalidad en otra historia.

US11: COMO usuario del e-commerce

QUIERO poder editar la cantidad seleccionada en cualquiera de los productos disponibles en mi carrito

PARA simplificar la operación si cambio de opinión o cometo un error.



8. Refinamiento del Backlog (Grooming o Refinement)

El refinamiento del backlog es una actividad fundamental para mantener un ritmo constante en la elaboración del producto, incluye dentro de este proceso mantenerlo actualizado descartando lo que ya no aplica y/o despriorizando según corresponda. Igualmente vamos a hacer foco en la preparación de las historias para su construcción.

Como hemos mencionado anteriormente el refinamiento de historias lo vamos realizando acorde se aproxima el momento de comenzar a construir, siendo el tamaño relativo de la historia acorde al de la iteración (release o sprint). La palabra más usada para generar historias de menor tamaño es dividir, y si bien también la vamos a utilizar aquí, en ocasiones puede no reflejar completamente lo esperado, por eso aconsejo pensarlo como un proceso de desagregación, porque cuando dice dividir imagina un corte en el cual la suma de todas las partes da el resultado original mientras que el verdadero proceso puede generar partes que se solapan y/o modifican con el feedback.

Cuál es el tamaño correcto de las historias del Sprint

La regla que autores como Lawrence o Cohn suelen recomendar es incluir aproximadamente entre 6-10 historias por iteración. El tamaño las mismas estará dado por la velocidad del equipo. El número es bastante razonable ya que asegura incluir una cantidad de historias sin abrir demasiado el espectro de temas. Igualmente, de acuerdo a la complejidad y tamaño del equipo este número lo debemos ir ajustando a nuestra comodidad.

Un consejo que me ha resultado útil es tratar de dividir en historias de tamaño similar para facilitar la priorización al PO por su valor al negocio, ya que, si una historia de 8 puntos la descomponemos en 2 de 5 y 3 estamos obligando a elegir un orden, en cambio si hacer 4 de 2 queda de su lado elegirlo según sus prioridades.

Responsabilidades de los participantes

Equipo de elaboración

- Trabajar junto con el PO que las mismas cumplan con el criterio INVEST en lugar de hacer foco en la especificación.
- Preguntar cuanto sea necesario para encontrar la raíz de la necesidad y tratar de encontrar la manera de dar valor minimizando esfuerzo.



- Colaborar con el PO para conversar sobre las historias y tener claro el valor que aportan al usuario antes que los demás de tecnología.
- Participar de refinamiento de las historias, incluso cuando un integrante puntual no participe de ella.
- Ser parte del proceso de identificación de los roles y personas involucradas en el uso del producto, entenderlos con claridad para poder diferenciarlos y mientras se construye comprender el uso que le podrán dar y experiencia que va a tener.
- Automatización de los casos de prueba (en caso de utilizar esta práctica)
- Creación de casos de prueba.
- Realizar testing unitario y de integración según corresponda.
- La estimación en equipo de las historias utilizando una unidad coherente que sirva para comparar tamaño entre sí (por ejemplo, dos historias de 3 puntos deberían ser de la misma magnitud).
- Colaborar con el PO para encontrar alternativas y optimizar la priorización.
- Indicar siempre que existan problemas estructurales o deuda técnica para una correcta priorización en relación con las entregas de valor.
- Evitar iniciar demasiadas historias en paralelo. El trabajo en curso (WIP) debería ser minimizado para concentrarnos en terminar cosas antes de empezar otras.

Product Owner

Es el responsable final (aunque debe recibir colaboración del equipo) en:

- Escribir las historias teniendo presente las 3 C's descritas por Jeffries que vimos anteriormente.
- Garantizar de que las historias cumplan el criterio INVEST.
- Buscar todos los posibles usuarios (y particularmente roles) que pueden usar el producto.
- Pensar el producto desde el punto de vista de los distintos roles, no haciendo foco indebidamente en un solo subgrupo.
- Las historias se deben escribir tan pronto como se puede para ubicarlas en la posición acorde en el backlog del producto para su posterior refinamiento cuando deba realizarse.
- Siendo el representante de los usuarios y stakeholders debe entender las opciones y costos de cada uno para comunicarlas y tomar decisiones informadas.
- Coordinar la realización de workshops para la generación de historias cuando le sea necesario.
- Escribir los criterios de aceptación y escenarios.



- Ejecutar las pruebas de aceptación una vez entregada la historia.
- Participar de las ceremonias de estimación para aclarar dudas y clarificar donde el equipo necesite. No debe caer en la tentación de querer estimar.
- Mantener el backlog priorizado y refinado de acuerdo a eso, agregando, modificando, cambiando prioridades e incluso eliminando según corresponda.
- Indicar los hitos que existan con anticipación de manera honesta para que se planifique en consecuencia.
- Mantener siempre visible el valor para el negocio de las historias en construcción
- Participar de las ceremonias de planificación y sincronización (daily meeting)
- Conocer la velocidad real del equipo y tener presente cuando las planificaciones iniciales deben ajustarse de acuerdo a ella.
- Cancelar la iteración si el objetivo no puede lograrse o si hubo un cambio que lo amerite.

Creación de historias por un equipo de Product Owner

Cuando existen múltiples personas que pueden escribir historias, lo importante en primera instancia a recordar (como lo mencionamos anteriormente) una persona debe ser la responsable de establecer prioridades con claridad para todo el backlog, evitando cualquier conflicto por este tema de cara al equipo de elaboración.

A su vez esta persona debe facilitar que exista una integración entre las historias realizadas por las distintas personas, manteniendo una línea y una visión integradora en los cambios sobre el producto.

Algunos consejos:

- a) Hacer foco en la necesidad desde la perspectiva del usuario usuario/clientes, esto ayuda a no perder el foco de porqué lo hacemos y mantener ese norte.
- b) Tener en cuenta los roles que se involucran para entender las distintas visiones del producto. Poder hablar con el lenguaje acorde a cada uno y facilitar las negociaciones.
- c) No buscar que se convierta en una especificación sino buscar lograr una captura de la esencia de la conversación sobre la funcionalidad.
- d) El descubrimiento de las historias debe ser un trabajo colaborativo de mucha creatividad, dar esos espacios y fomentar el trabajo colaborativo y de debate constructivo.



- e) Periódicamente realizar el refinamiento para detallar las próximas historias a implementar, así como también aprovechar el conocimiento obtenido en las iteraciones anteriores para mejorar las ya "terminadas".
- f) Usar lenguaje simple y desde el punto de vista de usuario. Evitar las ambigüedades y clarificar con términos claros para todas las partes, manteniendo no solo la documentación sino las conversaciones que se requieran.
- g) Enfocarse en la información importante descartando todo aquello no esencial (menos suele ser más).
- h) Reescribir las historias de usuario tantas veces como sea necesario para mejorarlas, el proceso en sí mismo también es iterativo.
- i) Los criterios de aceptación además de garantizar el correcto testeo aportan información y contexto a las historias, es conveniente comenzar a crearlos en paralelo con las historias, dado que a partir de los mismos puede facilitarse incluso la desagregación de historias en historias más chicas.
- j) Como heurística podemos decir que una historia de usuario de tamaño promedio debería tener entre tres y cinco criterios de aceptación para contar con un contexto válido sin necesidad de exceder el nivel de detalle.
- k) Ser concreto en las historias dándoles una definición cerrada, con límites claros. Por ejemplo, decir: COMO referente del centro de contactos QUIERO poder administrar productos. Esta no es historia con límites concretos, la administración de productos puede extenderse sin un fin claro y nunca terminaría la historia. Por lo tanto, sería mejor expresarla en una serie de historias que cierre el alcance tales como:
 - COMO referente del centro de contactos QUIERO poder cargar nuevos productos
 - COMO referente del centro de contactos QUIERO poder modificar la información de cada producto.
 - COMO referente del centro de contactos QUIERO poder borrar productos.
- l) Mantener la definición de la interfaz de usuario para lo más tarde posible, esto es para evitar dedicar esfuerzo a algo que posiblemente no estaremos trabajando en lo inmediato, así como también permitimos conservar libertad de maniobra para ajustarnos a cómo vamos avanzando en el sistema y las posibilidades/sugerencias del equipo técnico.
- J) El cliente o usuario las debería escribir, pero siempre contando con soporte del agile coach y equipo de elaboración.



9- Problemas comunes de las historias

Historias demasiado grandes

En este caso no podemos terminarla en el sprint y se cae total o restando las pruebas. Esto es un problema y no debe aceptarse y simplemente continuarla en el próximo sprint ya que impide tener software terminado al finalizar. El equipo debe trabajar en encontrar una división de las historias que permitan terminirlas por completo en el sprint con los patrones mencionados antes.

Historias demasiado pequeñas

En estos casos podemos habernos ido al otro extremo y cada entregable puede aportar un mínimo o nulo valor al usuario en sí mismo, dándonos más costo de operación que entrega en valor.

Historias dependientes entre sí

Si bien es imposible eliminar cierta dependencia en algunas historias, debemos reducirla para independizar la planificación del sprint de historias no priorizadas pero que debemos incluirlas por las dependencias existentes. En el caso de que esto no es posible tal vez debamos unir las historias para ya saber de antemano que la inclusión de un tema trae aparejado un costo mayor.

Incluir demasiados detalles en las historias

Este es un problema que podemos tener cuando priorizamos la documentación de la historia por sobre la conversación, en este caso incluimos un alto nivel de detalle en la misma pensando que eso puede evitarnos diferencias, pero esto es costoso y muchas veces puede resolverse con una simple conversación o reduciendo el tamaño de la historia para hacerla más simple.

Adelantarnos demasiado en el refinamiento

Este es un clásico caso desperdicio, ya que debemos mantener el tamaño de la historia acorde al momento en que avanzaremos en la construcción. Esto en parte puede venir del pensamiento más tradicional de analizar con alto nivel de detalle al principio del proyecto, pero debemos



priorizar la concreción de producto terminado para aprovechar el feedback y refinar acorde a esta información lo próximo a construir.

No tener criterios concretos de aceptación.

Seguir pensando en ciclo cascada implementado en sprints.

Este es un caso muy típico de equipos en estado inicial de la implementación de Scrum, en los cuales para poder cumplir con terminar la historia en un sprint se procede a dividirla en las fases clásicas del desarrollo de software, así por ejemplo en un sprint se hace el análisis, los 2 siguientes son historias de desarrollo y el cuarto pruebas. Puede que las historias se hayan completado, pero ninguna era un entregable concreto que permite dar valor y obtener feedback.

No priorizar o no saber cómo hacerlo

Esto puede darse por varias razones, las más comunes son; no tenemos al PO informado o capacitado para su tarea, por lo cual debemos solucionar este inconveniente si deseamos tener éxito en el proyecto, o por tener historias que no tienen un claro valor para el negocio, en cuyo caso debe rehacerse para definir esto y poder priorizar acorde.



Bibliografía utilizada y sugerida

Libros y otros manuscritos

Carlucci Daniela, Schiuma Gianni. The knowledge value chain: how intellectual capital impacts on business performance.2013.

Cohn Mike. Agile Estimating and Planning, Prentice Hall, 2005,

Humble, Jez, Molesky, Joanne y O'Reilly, Barry. Lean Enterprise: How High Performance Organizations Innovate at Scale, O'Really, 2015,

Patton Jeff, User Story Mapping, O'Really, 2015,

Pitchler, Roman. Strategize: Product Strategy and Product Roadmap Practices for the Digital Age; 2016.

v7

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Lo que vimos:

Es esta unidad hicimos foco en el tipo de componente del product backlog más utilizado, como son las historias de usuario, comprendiendo sus componentes y cómo crearlos con efectividad.



Lo que viene:

En la próxima unidad podremos conocer el estadió en el mercado del rol de Product Owner, las especializaciones y certificaciones existentes.



Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning