

División de Historias de Usuario

Subdirección de las Tecnologías de la Información y Comunicaciones

Área de Gobernanza y Calidad



Contenido

- [Histórico de cambios](#)
- [Ideas para dividir Historias de Usuario](#)

Resumen



- **Versión:** v01r01
- **Fecha publicación:** 29 de Octubre de 2020

Histórico de cambios

Los cambios en la documentación de apoyo vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución.

Versión	v01r01	Fecha publicación	29 de Octubre de 2020
Alcance			
• <u>Presentación de 15 técnicas para afrontar la división de Historias de Usuario</u>			

Introducción

El proceso de división de una Historia de Usuario comienza con el análisis del cumplimiento de dicha HU del principio INVEST (incluir enlace a página de Historias de Usuario). En el caso de que varios criterios de este principio no se cumplan, es recomendable reformular la HU (p.ej combinándola con otra HU). Si por el contrario la historia de usuario es Independiente, Negociable, de Valor, Estimable, Testeable pero tiene un tamaño mayor al 15% de la velocidad del equipo, se recomienda proceder a la división de la Historia de Usuario.

Ideas para dividir Historias de Usuario

La división de Historias de Usuario es en parte experiencia/técnica y en parte creatividad. De hecho, una misma Historia de Usuario puede dividirse siguiendo varios patrones de división.

Entonces, ¿qué patrón de división elijo? Quizás esta pregunta se pueda responder teniendo en cuenta tres premisas clave:

- **Obtén HUs de tamaño similar:** lo cual facilita la priorización por parte del PO.
- **Primero el valor, luego lo accesorio:** la regla de Pareto (o del 80/20) es igualmente aplicable a las HUs: el 80% del valor se encuentra en el 20% de las HUs. Por lo que siempre es positivo elegir aquel patrón de división que te permita dividir las HUs entre accesorias (de menor valor) y las críticas/core (alto valor).
- **Genera una estrategia de aprendizaje temprano y evolutiva:** Intenta elegir patrones de división que te permita testear de manera temprana tus asunciones más dudosas acerca del resultado esperado y/o descubrir los mayores riesgos técnicos y de desarrollo cuanto antes, de manera que , a través de "experimentos mínimos viables", generes una estrategia de aprendizaje temprano y evolutiva.

A continuación pueden encontrarse algunas de las estrategias de división de HUs más comunes:

#Idea	Tipo de Historias de Usuario	Estrategia División	Descripción
1	HUs que describe un workflow completo.	Divide los pasos de un workflow: comienza por los outputs.	En caso de que la HU original describa un workflow completo, se pueden estudiar dos alternativas: 1. Incluye una "rebanada" vertical simplificada de todo el flujo y añade iterativamente los detalles accesorios. 2. Incluye en una HU el primer y último paso del flujo, y desarrolla iterativamente el flujo intermedio.
2	HUs con mucha complejidad debido a la necesidad de cumplir con requisitos no funcionales (rendimiento, capacidad, etc).	Ponlo a funcionar y mejora el rendimiento /capacidad iterativamente.	Divide la HU original de manera que primero se pueda desarrollar la funcionalidad en sí (hacer que funcione) y luego, en HUs posteriores puedas cumplir con esos requisitos no funcionales. Por ejemplo: • Mejora incremental del rendimiento. • Aumento incremental de la capacidad del sistema: Tamaño de archivos, duración de la sesión, número total de usuarios con acceso al sistema, concurrencia de usuarios, número de items (archivos o recursos) disponibles a un sólo usuario,...

3	HUs con un núcleo sencillo que aporta la mayor parte del valor y complementos accesorios que añaden complejidad.	Identifica si la complejidad no reside en el núcleo de valor.	Divide el núcleo sencillo (y de alto valor) y el complejo de la HU original, de manera que primero pueda desarrollarse el núcleo sencillo e iterativamente el complejo.
4	HU que gestiona varias interfaces distintas.	Divide por interfaces.	Divide la HU original por tipo de interfaz, de manera que con cada HU resultante se gestione un tipo de interfaz (p.ej tipo de buscador, versión o diferente hardware).
5	HU con interfaz compleja.	Evoluciona iterativamente la complejidad de la interfaz de usuario.	Intenta dividir la HU original de manera que se comience con una versión simplificada de la interfaz y se vaya añadiendo complejidad de manera iterativa (p. ej. inicialmente atacas una Interfaz mínima viable y en iteraciones posteriores se mejora el estilo).
6	HU que aplica la misma funcionalidad sobre distintos tipos de datos.	Segmenta el tipo de datos.	Divide la HU original de manera que primero se gestione un tipo de dato o un set de datos e iterativamente se vaya ampliando la cobertura a los demás datos. P.ej: • Primero ataca registros válidos y posteriormente datos invalidos. • Trabaja primero en los datos más frecuentes y posteriormente lo complementas con otros más infrecuentes.
7	HU con variedad en las reglas de negocio (p. ej. campos flexibles).	Implementa iterativamente las reglas de negocio.	Divide la HU original de manera que se pueda hacer primero un subconjunto de las reglas y ampliar iterativamente con otras HUs el número de reglas cubiertas.
8	HU que incluye múltiples operaciones (p. ej. sólo lectura, lectura/edición, gestión de contenido, etc).	Divide múltiples operaciones.	Divide la HU original de manera que cada operación se recoja en una HU.
9	HU conceptualmente compleja, con alta incertidumbre o que no sepa cómo puede dividirse.	Planifica tus conocidos-desconocidos.	Cuando el tamaño/complejidad de una HU resida en la falta de conocimiento de la misma o, incluso, seamos incapaces de dividir la misma, planifica un Spike (una reserva de tiempo para investigar) para ayudar a mejorar el entendimiento de la HU. Esto puede ayudar a estimar el esfuerzo restante y/o dilucidar caminos adicionales para la división de la misma.
10	HU para remodelar los "outputs" de una funcionalidad (p.ej. un reporte) de un sistema Legacy.	Comienza por los Outputs, no por los inputs.	En estos casos es recomendable comenzar la división de HUs por aquellas que modelen los Outputs finales del sistema en lugar de los inputs o procesos intermedios (p.ej. inicialmente centrarse en el reporte final de una funcionalidad en lugar de comenzar la división por los formularios de registro que conducen a dichos reportes).
11	HU que sirve a varios usuarios.	Segmenta tus usuarios.	Divide la HU original en función de los segmentos de usuarios a los que se pretende servir. La segmentación de clientes puede realizarse en base a múltiples dimensiones, las cuales a su vez dependerán del tipo de producto/funcionalidad que pretenda desarrollarse: • Productos de consumo: territorio, edad, competencias, localización (casa, trabajo o en itinerario), ocupación, necesidades,... • Aplicaciones IT corporativas internas: tipo de trabajo, departamento,...
12	HU que utiliza datos de referencia de otros sistemas (p.ej bbdd, servicios remotos o "cajas negras") y estos no suelen cambiar frecuentemente.	Empieza "Dummy", evoluciona hacia "Dynamic".	En estos casos, las ventanas de entradas de datos suelen ser foco de atención inicial debido a que es necesario tener los datos antes de que se procesen. Sin embargo la entrega de valor no reside en los "inputs" del producto, sino en los "outputs". Por ello en estos casos puede ser recomendable orientar la división de HUs de manera que en las primeras iteraciones los datos de referencia sean "hard coded" en lugar de hacer importaciones dinámicas desde las fuentes oficiales e ir iterativamente conectando los campos de entrada definidos con las fuentes de datos relevantes.

13	HUs que dan como resultado varios formatos de reporte.	Divide y simplifica los "outputs" inicialmente.	Divide la HU original de manera que se obtenga una HU por cada tipo de reporte (p.ej. PDF, Excel, otros formatos), siendo la primera HU la que procese los datos y genere el "output" más simple.
14	HUs en las que se necesita que un proceso de negocio se implemente completamente para ser funcional.	Simplifica al mínimo la interacción de los usuarios.	En lugar de proporcionar usabilidad a los usuarios, céntrate inicialmente en proporcionar utilidad básica (interacción limitada del usuario con la herramienta). La extracción de utilidad básica normalmente requiere de la ejecución de procesos semiautomáticos o de chequeos para verificar inconsistencia de datos. Ejemplos de esta estrategia: • Evitar utilizar formularios previamente rellenos y pedir al usuario introducir toda la info. cada vez que sea necesario. • En lugar de permitir la selección de varios items a la vez, restringir la selección a un item cada vez. Esta estrategia es arriesgada especialmente en aquellos productos en los que la experiencia de usuario es un punto crítico. Por ello, se recomienda utilizarla en dos situaciones: cuando quiere testearse la utilidad de trasfondo o cuando es necesario implementar un proceso urgentemente y los usuarios pueden sobrevivir con una interacción limitada.
15	HU compleja, que no se sepa cómo puede dividirse o que la única división que parece viable es la separación de tareas técnicas.	Método de la Hamburguesa.	En estos casos otra alternativa para mejorar la división y entendimiento de la HU es utilizar el método de la hamburguesa .