



# Historias de Usuario

Ingeniería de requisitos ágil

v. 1.0



# Historias de Usuario

Ingeniería de requisitos ágil

Versión. 1.0 – Noviembre 2014

Imagen y diseño de cubierta: Scrum Manager.

© 2014 Alexander Menzinsky y Juan Palacio

© De la edición: Scrum Manager®

Información de derechos y licencia de uso:



<http://www.safecreative.org/work/1411262602694>

Scrum Manager® es marca registrada, propiedad de Iubaris Info 4 Media S.L.



# Contenido

<i>Contenido</i>	5
<i>Formación Scrum Manager</i>	6
Servicios de formación y asesoría Scrum Manager	6
<i>Mejora continua y control de calidad Scrum Manager</i>	7
<i>Ingeniería de requisitos ágil</i>	9
Historias de Usuario	9
Epics, temas y tareas	10
<i>Información en una historia de usuario</i>	11
<i>Calidad en las historias de usuario</i>	14
<i>Priorización de historias de usuario</i>	16
<i>Comparativa con otras formas de toma de requerimientos</i>	17
Historias de usuario versus casos de uso	17
Historias de usuario versus requisitos funcionales	18
<i>Bibliografía</i>	19
<i>Tabla de ilustraciones</i>	21
<i>Índice</i>	22

# Formación Scrum Manager

Este libro amplía el conocimiento troncal de formación Scrum Manager®

Formación  
Scrum Manager

Nivel I: Scrum técnico

Para aprender las reglas de scrum

Nivel II: Scrum pragmático

Para aprender a romper las reglas de scrum

Los contenidos de formación se mantienen regularmente actualizados. Puede descargar la última versión, o consultarla en línea en la dirección: <http://www.scrummanager.net/bok>

Este documento es un **recurso educativo abierto (OER)** y puede emplearse gratuitamente para consulta y autoformación.

No está permitido su uso para actividades comerciales.



**Los recursos educativos abiertos de Scrum Manager son posibles gracias al soporte de los:**

## Servicios de formación y asesoría Scrum Manager

- **Cursos** en convocatorias presenciales, o a medida para empresas o grupos de estudiantes.
  - Puede consultar el calendario de convocatorias en diferentes ciudades en la página de cursos de <http://www.scrummanager.net>:
  - Para información de cursos a grupos o in-company : [admin@scrummanager.net](mailto:admin@scrummanager.net)
- **Exámenes presenciales de certificación.** con supervisión de la ejecución de la prueba, e identificación del alumno
  - Incluidos en los cursos presenciales.
  - Disponibles en centros examinadores.

Más información:

<http://www.scrummanager.net> – (preguntas frecuentes)

<http://www.scrummanager.net/oks>

[admin@scrummanager.net](mailto:admin@scrummanager.net)

## Mejora continua y control de calidad Scrum Manager

Gracias por elegir los servicios de formación de Scrum Manager

***Su valoración es el criterio del control de calidad de Scrum Manager***, y decide la validez o no de los servicios de formación, y en su caso la continuidad de los cursos, centros y profesores.

Si ha participado en una actividad de formación auditada por Scrum Manager, le rogamos y agradecemos que valore la calidad del material, profesor, temario, etc. así como tus comentarios y sugerencias.

Scrum Manager anonimiza la información recibida, de forma que comparte con los profesores, y centros autorizados las valoraciones y aspectos de mejora, pero en ningún caso los nombres de los alumnos que las han realizado.

Puede realizar la valoración en la página accesible a miembros de Scrum Manager:

<http://scrummanager.net/qa>.





# Ingeniería de requisitos ágil

## Historias de Usuario

Las **historias de usuario** son utilizadas en las metodologías ágiles para la especificación de requisitos, **son una descripción breve de una funcionalidad software tal y como la percibe el usuario** (Mike Cohn, 2004).

Describen lo que el cliente o el usuario quiere que se implemente y se escriben con una o dos frases utilizando el **lenguaje común del usuario**. Cada historia de usuario debe ser limitada, esta debería poderse memorizar fácilmente y escribir sobre una tarjeta o post-it. Poco antes de ser implementadas, estas van acompañadas de conversaciones con los usuarios y la definición de las pruebas de validación asociadas. Como los cambios son bienvenidos en agilidad, no vale la pena profundizar antes, ya que en el momento de la implementación estas pueden haber cambiado desde que fueron escritas. Las pruebas permiten al equipo, y luego al cliente o usuario, verificar que la historia ha sido completada.

Las historias de usuario forman parte de la fórmula de captura de funcionalidades definida en 2001 por Ron Jeffries de las **tres C's**:

### Card - Conversation - Confirmation

Estas son una forma ágil de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

Ventajas que aportan las historias de usuario:

- Al ser muy cortas, estas representan requisitos del modelo de negocio que pueden implementarse rápidamente (días o semanas).
- Necesitan poco mantenimiento.
- Mantienen una relación cercana con el cliente.
- Permiten dividir los proyectos en pequeñas entregas.
- Permiten estimar fácilmente el esfuerzo de desarrollo.
- Son ideales para proyectos con requisitos volátiles o no muy claros.

El origen de las historias de usuario viene de **XP “eXtremeProgramming”** o programación extrema, donde las historias de usuario deben ser escritas por los clientes. Esta metodología fue creada por Kent Beck y descrita por primera vez en 1999 en su libro *eXtreme Programming Explained*.

Las **historias de usuario** se aplican en la mayoría de las metodologías ágiles, siendo así una herramienta muy importante también en **Scrum**.

---

## Epics, temas y tareas

Se denomina **epic** a una **superhistoria de usuario que se distingue por su gran tamaño**, a diferencia de las historias de usuario, que tienen baja granularidad, las epics tienen una alta granularidad. Es una etiqueta que aplicamos a una historia grande, cuyo esfuerzo es demasiado grande para completarla de una sola vez o en un solo sprint. A medida que aumenta su prioridad y se acerca al momento de su implementación, el equipo la descompone en historias de usuario con un tamaño más adecuado para ser gestionada con los principios y técnicas ágiles: estimación y seguimiento cercano (normalmente diario).

Por encima de los epics se encuentran los **temas** que representan a **una colección de epics relacionados** para describir un sistema o subsistema en su totalidad. Por ejemplo en un sistema de software para gestión contable, el conjunto de epics: “Altas, bajas y mantenimiento de clientes”, “Facturaciones puntuales y recurrentes”, “Consultas de navegación y acciones de fidelización”, “Pedidos”, “Devoluciones” se podrían denominar como el tema de la gestión de clientes.

Por debajo de las historias de usuario se encuentran las **tareas**. Estas son resultado de la **descomposición por parte del equipo de las historias de usuario en unidades de trabajo** adecuadas para gestionar y seguir el avance de su ejecución.

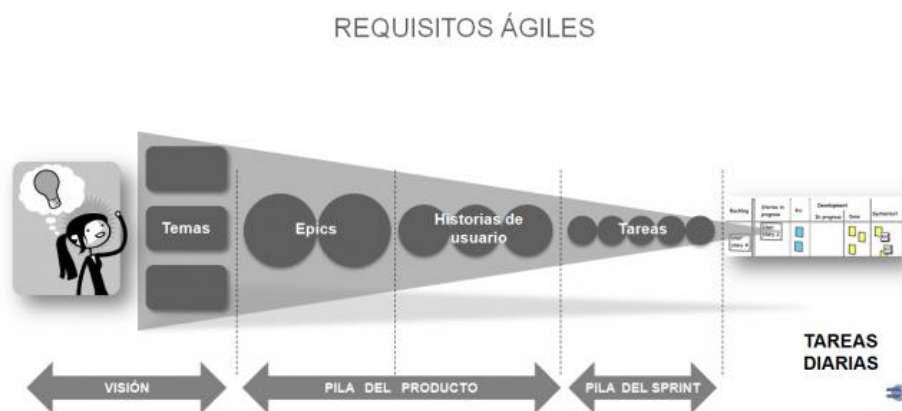


Ilustración 1: Gráfico con los cuatro niveles de tamaño con que trata los requisitos la gestión ágil

En **Scrum**, y en las metodologías ágiles en general, una **pila de producto** puede contener tanto **historias de usuario** como **epics**. La pila de producto debe de estar ordenada por prioridad y el nivel de detalle de cada elemento debe de ir en relación a la posición del mismo dentro de la pila. En una lista larga, algo muy poco prioritario no tiene sentido tenerlo al detalle, porque probablemente cambiará a lo largo del proyecto, e incluso puede que ni se desarrolle. De la mitad hacia el final de la lista, donde está lo menos prioritario, es el lugar de los epics. A medida que nos acercamos a los elementos más prioritarios, el detalle debe aumentar, por tanto las historias de usuario son los elementos que deben de encabezar la lista.

# Información en una historia de usuario

Para decidir qué **información incluir en una historia de usuario** es preferible no adoptar formatos rígidos. Los resultados de **Scrum** y agilidad no dependen de las formas, sino de la institucionalización de sus principios y la implementación adecuada a las características de la empresa y del proyecto. Por tanto, aparte de **4 campos que se consideran necesarios**, se puede incluir cualquier campo que proporcione información útil para el proyecto.

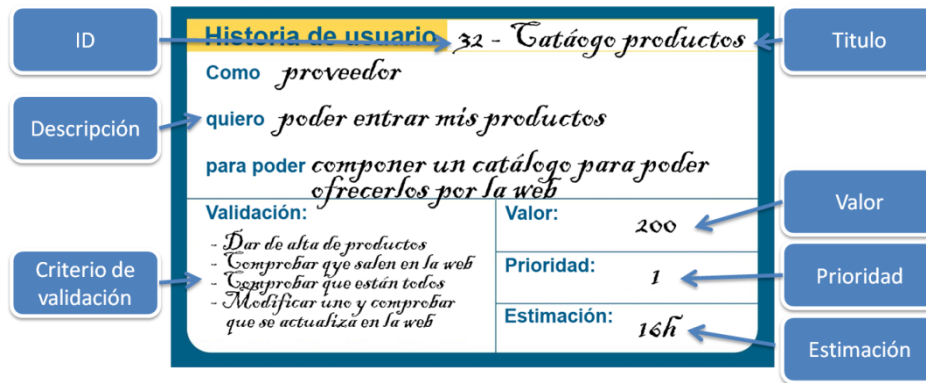


Ilustración 2: Ejemplo de una tarjeta de historia de usuario

Los campos que se consideran **más necesarios** para describir de una manera adecuada las historias de usuario son:

- **ID:** identificador de la historia de usuario, **único** para la funcionalidad o trabajo.
- **Descripción:** descripción sintetizada de la historia de usuario. El estilo puede ser libre, según mejor nos funcione, debe responder a tres preguntas: ¿Quién se beneficia? ¿Qué se quiere? y ¿Cuál es el beneficio? Mike Cohn recomienda seguir el siguiente patrón que garantiza que la funcionalidad esta descrita a un alto nivel y de una manera no demasiado extensa:

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

- **Estimación:** estimación del **esfuerzo necesario** en tiempo ideal de implementación de la historia de usuario. Según convenga al equipo también se puede utilizar unidades de desarrollo, conocidas como **puntos de historia** (estas unidades representan el tiempo teórico de desarrollo/persona que se estipule al comienzo del proyecto).
- **Prioridad:** **sistema de priorización** que nos permite determinar el orden en el que las historias de usuario deben de ser implementadas.

Dependiendo del tipo de proyecto, el funcionamiento del equipo y la organización, pueden ser aconsejables otros campos como:

- **Título:** título descriptivo de la historia de usuario.
- **Criterio de validación:** **pruebas de aceptación** consensuadas con el cliente o usuario. Estas son las pruebas que el código debe superar para dar como finalizada la implementación de la historia de usuario.

- **Valor:** valor (normalmente numérico) que aporta la historia de usuario al cliente o usuario. El objetivo del equipo es **maximizar el valor y la satisfacción percibida por el cliente** en cada iteración. Este campo servirá junto con la estimación para determinar la prioridad con el que las historias de usuario deben de ser implementadas.
- **Dependencias:** una historia de usuario no debería ser dependiente de otra historia, pero en ocasiones es necesario mantener la relación. En este campo se indicarían los identificadores de otras historias de las que depende.
- **Persona asignada:** en casos en que queramos sugerir la persona que pueda implementar la historia de usuario. Recordar que en **Scrum** es en último término el equipo autogestionado quién distribuye y por tanto asigna las tareas.
- **Criterio de finalización:** la definición de **finalizada** incluye los criterios o actividades necesarias para dar por terminada una historia de usuario (desarrollada, probada, documentada...), que son las convenidas por el equipo y el propietario del producto.
- **Sprint:** puede ser útil para organización del propietario del producto incluir el **número de sprint** en el que previsiblemente se vaya a realizar la historia.
- **Riesgo:** riesgo **técnico o funcional** asociado a la implementación de la historia de usuario.
- **Módulo:** módulo del **sistema o producto** al que pertenece.
- **Observaciones:** para **enriquecer o aclarar** la información o cualquier uso que pueda ser útil.

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre historia: Cambiar dirección de envío	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Pérez	
<b>Descripción:</b> Quiero cambiar la dirección de envío de un pedido.	
<b>Validación:</b> El cliente puede cambiar la dirección de entrega de cualquiera de los pedidos que tiene pendientes de envío.	

Ilustración 3: Ejemplo una historia de usuario



Ilustración 4: Ejemplo una historia de usuario

Mike Cohn comenta que **si bien las historias de usuario son lo suficientemente flexibles como para describir la funcionalidad de la mayoría de los sistemas, no son apropiadas para todo. Si por cualquier razón, se necesita expresar alguna necesidad de una manera diferente a una historia de usuario, recomienda que se haga.** Por ejemplo, la maquetación de pantallas se suele describir con pantallazos, por tanto esta es la mejor manera de transmitir el diseño que queramos darle a una aplicación.

# Calidad en las historias de usuario

En 2003 Bill Wake desarrolló un **método llamado INVEST** para asegurar la calidad en la escritura de historias de usuarios. El método sirve para comprobar la calidad de una historia de usuario revisando que cumpla una serie de características:

- **I** - Independent (independiente)
- **N** - Negotiable (negociable)
- **V** - Valuable (valiosa)
- **E** - Estimable (estimable)
- **S** - Small (pequeña)
- **T** - Testable (comprobable)

## **Independent (independiente)**

Es ventajoso que cada historia de usuario pueda ser planificada e implementada en cualquier orden. Para ello las historias deberían de ser totalmente independientes (lo cual facilita el trabajo posterior del equipo). Resaltar que las dependencias entre historias de usuario pueden reducirse combinándolas en una o dividiéndolas de manera diferente.

## **Negotiable (negociable)**

Una historia de usuario es una descripción corta de una necesidad que no incluye detalles. Las historias deben ser negociables ya que sus detalles serán acordados con el cliente o el usuario durante la fase de conversación. Por tanto, se debe evitar historias de usuario con demasiados detalles porque limitaría la conversación acerca de las mismas.

## **Valuable (valiosa)**

Una historia de usuario tiene que ser valiosa para el cliente o el usuario. Una manera de hacer una historia valiosa es que la escriba el mismo.

## **Estimable (estimable)**

Una buena historia de usuario debe de poder ser estimada con la precisión suficiente para ayudar al cliente, usuario o propietario del producto a priorizar y planificar su implementación. La estimación generalmente la realizará el equipo de trabajo y está directamente relacionada con el tamaño de la historia de usuario (una historia de usuario de gran tamaño es más difícil de estimar) y con el conocimiento del equipo de la necesidad expresada (en el caso de falta de conocimiento, serán necesarias mas fases de conversación acerca de la misma).

## **Small (pequeña)**

Las historias de usuario deberían englobar como mucho unas pocas semanas/persona de trabajo. Incluso hay equipos que las restringen a días/persona. Una descripción corta ayuda a disminuir el tamaño de una historia de usuario facilitando así su estimación.

## **Testable (comprobable)**

La historia de usuario debería ser capaz de ser probada (fase confirmación de la historia de usuario). Si el cliente o usuario no sabe como probar la historia de usuario significa que no es del todo clara o que no es valiosa. Si el equipo no puede probar una historia de usuario nunca sabrá si la ha terminado o no.

Hay empresas que han diseñado sus propias tarjetas como estas de Braintrust que, aparte de dar un look profesional, son completas en información y coherentes en espacio para los campos.

The image shows two example cards from Braintrust. The first card is a 'USER STORY' card with a yellow header. It contains the text 'As a', 'I want', and 'So that' in a large font. Below this is a section with the word 'INVEST' in large letters, followed by six small yellow squares. To the right of this section are two fields: 'Size:' and 'Business Value:'. The second card is an 'ACCEPTANCE CRITERIA' card with a yellow header. It contains the text 'Meets team's definition of ready?' followed by a small yellow square. At the bottom right of the card is the Braintrust logo and website information.

Ilustración 5: Ejemplo de tarjetas diseñadas por Braintrust

### Consejos de buenas prácticas:

- Siempre escribir las historias con el “qué”, evitar el “cómo”.
- No escribir una descripción exhaustiva, solo lo justo.
- Escribir el criterio de validación y ser suficientemente explícito.
- Estimar todas las historias, no hacerlo puede crear falsas expectativas.
- No fiar toda la información en las tarjetas, a veces es buena idea una documentación externa como una wiki por ejemplo.
- Nunca dar una historia por finalizada cuando está “prácticamente hecha”.

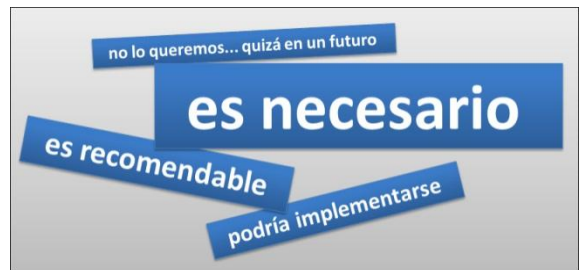
## Priorización de historias de usuario

Aunque todas las historias de usuario puedan ser importantes, para poder **focalizarnos en el trabajo de forma eficiente**, es necesario destacar aquellas que den mayor valor al sistema, por tanto, las historias de usuario deben de estar priorizadas. Estas deben de tener **asignadas un valor** que intervenga en el **sistema de priorización**, un valor asignado por el propietario del producto y se basará básicamente en las siguientes variables:

- Beneficios de implementar una funcionalidad.
- Pérdida o coste que demande posponer la implementación de una funcionalidad.
- Riesgos de implementarla.
- Coherencia con los intereses del negocio.
- Valor diferencial con respecto a productos de la competencia.

Uno de los aspectos a tener en cuenta es que la definición de "valor" puede variar para cada uno de nuestros clientes. Más allá de un sistema de clasificación de tipo prioridad alta, media o baja es muy recomendable utilizar algún tipo de escala cualitativa, una que tenga un **significado intrínseco**. Este es el caso de la **técnica MoSCoW**, en la que el usuario responsable de asignar la prioridad es consciente del efecto real que producirá su elección. Esta técnica fue definida por primera vez en el año 2004 por Dai Clegg de Oracle UK Consulting en el libro *Case Method Fast-Track: A RAD Approach*. Su finalidad es obtener un entendimiento común entre cliente y el equipo del proyecto, en concreto sobre la importancia de cada historia de usuario. La clasificación es la siguiente:

- **M - MUST HAVE (es necesario)**: Se debe tener la funcionalidad implementada en la solución, sino esta fallará o la solución no puede ser considerada un éxito.
- **S - SHOULD HAVE (es recomendable)**: Se debería tener la funcionalidad implementada en la solución ya que es una funcionalidad de alta prioridad. La solución es prescindible, no fallará si no existe pero debería de haber causas justificadas para no implementarla.
- **C - COULD HAVE (podría implementarse)**: Es deseable, por tanto sería conveniente tener esta funcionalidad implementada en la solución, dependerá de las posibilidades de los tiempos y el presupuesto del proyecto.
- **W - WON'T HAVE (no lo queremos... quizá en un futuro)**: Se trata de una funcionalidad de muy baja prioridad o descartada en ese momento, pero que en futuro pueda ser relevante. Posteriormente, cuando cobre importancia, puede pasar a alguno de los estados anteriores.



Es importante distinguir entre prioridad y valor para el cliente. Puede ser que una historia de usuario no tenga ningún valor para el cliente o usuario, pero que esta sea absolutamente necesaria, por tanto de alta prioridad. Por ejemplo la infraestructura necesaria para la implementación de un software, no aporta valor al cliente en sí, pero sin ella no se puede desarrollar ni ejecutar la solución desarrollada.



# Comparativa con otras formas de toma de requerimientos

## Historias de usuario versus casos de uso

Siempre que se menciona **casos de uso** e **historias de usuario** se produce algo de confusión. Hay quien ha logrado incluir casos de uso en su proceso ágil pero eso no quiere decir que las historias de usuario sean equivalentes a los casos de uso.

Un caso de uso se basa en **UML**, un lenguaje descriptivo pensado inicialmente para sencillez en la comunicación pero que no es cercana a la comunicación humana. En cambio una historia de usuario está escrita en un lenguaje coloquial, que funciona a modo de recordatorio de la conversación con el cliente.

Como comenta Alistair Cockburn en 2001 en su libro *Agile software development*, realmente las historias de usuario están más cerca de la captura de requisitos, la fase que sirve para extraer las necesidades del usuario, que la especificación de requisitos, como son los casos de uso.

Podríamos decir que una **historia de usuario dice el “qué”** quiere el cliente o usuario, un **caso de uso es un “cómo”** lo quiere.

En el criterio de validación también hay diferencias de concepto, a diferencia de los casos de uso que requieren matrices de seguimiento de requisitos con porcentajes de terminado, las historias de usuario que incluyen el criterio de validación incluyen el terminado de forma binaria, o vale o no vale.

La propia agilidad se hace patente en el hecho de que las historias de usuario son vivas. El análisis funcional y técnico se hace poco antes del desarrollo, en la reunión de inicio de sprint en caso de **Scrum**, por tanto el desglose en tareas lo hace el equipo, con lo que nivel de detalle y previsión supera en mucho al que pueda hacer un único arquitecto o analista funcional en los casos de uso.

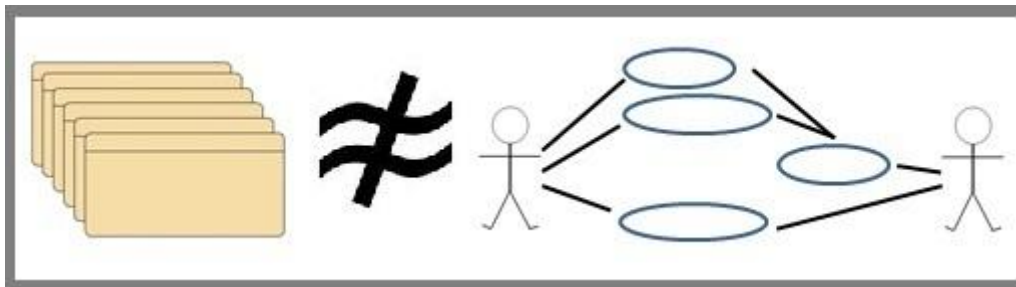


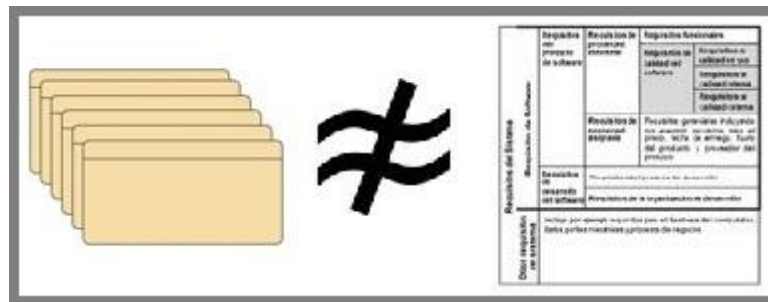
Ilustración 6: Historias de usuario no son parecidas a casos de uso

Cuando un proyecto comienza a seguir una metodología ágil se deberían de olvidar completamente los casos de uso y el equipo debería de centrarse solo en la realización de historias de usuario.

Para los que estén interesados en complementar historias de usuario con casos de uso les invito a leer el libro de Cockburn, en él describe los problemas que pueden surgir y como les dio solución para sobrellevarlo.

Por lo general se asocian las **historias de usuario** a **requisitos funcionales**, unas como los artefactos de las metodologías ágiles y los otros los de las metodologías tradicionales. Sin embargo detrás de las historias de usuario hay aspectos que las diferencian, diferencias que desconocidas llevan a confusiones muy comunes.

Igual que en los casos de usos, las historias de usuario dicen el “qué” pero no el “cómo” se desarrollará la funcionalidad. Por tanto, de igual manera, las historias de usuario no deben tener el nivel de detalle que tiene la especificación de un requisito funcional.



### Ilustración 7: Historias de usuario no son parecidas a requisitos funcionales

# Bibliografía

Alistair Cockburn (2001). *Agile software development*. Boston: Addison-Wesley Professional.

Bill Wake (2003). *INVEST in good stories, and smart tasks*. <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

Dai Clegg (2004). *Case Method Fast-Track: A RAD Approach*. Boston: Addison-Wesley Longman.

Javier Garzás (2014). *Agilidad y Lean. Gestionando los proyectos y negocios del s. XXI*.  
<https://www.miriadax.net/>

Kent Beck (1999). *eXtreme Programming Explained*. Boston: Addison-Wesley Professional.

Mike Cohn (2004). *User Stories Applied for Agile Software Development*. Boston: Pearson Education, Inc.

Ron Jeffries, Ann Anderson, Chet Hendrickson (2001). *Extreme Programming Installed*. Boston: Addison-Wesley.

---



# Tabla de ilustraciones

Ilustración 1: Gráfico con los cuatro niveles de tamaño con que trata los requisitos la gestión ágil .....	10
Ilustración 2: Ejemplo de una tarjeta de historia de usuario.....	11
Ilustración 3: Ejemplo una historia de usuario.....	12
Ilustración 4: Ejemplo una historia de usuario.....	13
Ilustración 5: Ejemplo de tarjetas diseñadas por Braintrust .....	15
Ilustración 6: Historias de usuario no son parecidas a casos de uso .....	17
Ilustración 7: Historias de usuario no son parecidas a requisitos funcionales .....	18

# Índice

Asignación, 12  
Buenas prácticas, 15  
Campos, 11  
Caso de uso, 17, 18  
Comprobable, 14  
Criterio de finalización, 12  
Criterio de validación, 11, 15, 17  
Dependencia, 12  
Descripción, 11  
Epic, 10  
Equipo, 10, 11, 12, 14, 17, 18  
Esfuerzo, 11  
Estimable, 14  
Estimación, 11, 12, 14, 15  
Fórmula tres C's, 9, 14, 18  
Funcionalidad, 9, 11, 16, 18  
Granularidad, 10  
Historia de usuario, 9, 10, 11, 12, 14, 16, 17, 18  
ID, 11  
Independiente, 14  
Información, 11  
Método INVEST, 14

Módulo, 10, 12  
Negociable, 14  
Observación, 12  
Pequeña, 14  
Pila de producto, 10  
Prioridad, 10, 11, 12, 16  
Propietario del producto, 12, 14, 16  
Pruebas, 9, 11, 14  
Punto de historia, 11  
Requisito funcional, 18  
Riesgo, 12  
Scrum, 9, 10, 11, 12, 17  
Sprint, 12  
Tarea, 10  
Tarjeta, 11, 13, 15  
Técnica MoSCoW, 16  
Tema, 10  
Título, 11  
UML, 17  
Valiosa, 14  
Valor, 12, 14, 16  
XP, 9

---