



***Técnico en***  
**< DESARROLLO DE SOFTWARE >**

***Metodología de Desarrollo de  
Software II***

(CC BY-NC-ND 4.0)  
International

Attribution-NonCommercial-NoDerivatives 4.0



## **Atribución**

Usted debe reconocer el crédito de una obra de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace.



## **No Comercial**

Usted no puede hacer uso del material con fines comerciales.



## **Sin obra derivada**

Si usted mezcla, transforma o crea un nuevo material a partir de esta obra, no puede distribuir el material modificado.

No hay restricciones adicionales - Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



# Metodología de Desarrollo de Software II

## Unidad II

### Metodología de Desarrollo Ágil SCRUM

#### 1. Conociendo Scrum

**Scrum** es una metodología ágil para la gestión de proyectos relacionados con la construcción (desarrollo) de Software. Pete Deemer, Gabrielle Benefield, Craig Larman y Bas Vodde, definen Scrum en el libro The Scrum Primer (2009), con los siguientes párrafos:

*“Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados Sprints. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra. Los Sprints son de duración fija – terminan en una fecha específica aunque no se haya terminado el trabajo, y nunca se alargan. Se limitan en tiempo. Al comienzo de cada Sprint, un equipo multi-funcional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos. [...]” (The ScrumPrimer, 2009, pág. 5)*

Un Sprint tiene dos características fundamentales:

- Una duración fija entre 1 a 4 semanas.
- Cada Sprint se ejecuta de forma consecutiva (sin tiempo muerto entre un sprint y otro).

El objetivo principal es transformar un conjunto de ítems requeridos por el cliente en un incremento de funcionalidad 100% operativa para el software.

*“Todos los días el equipo se reúne brevemente para informar del progreso, y actualizan unas gráficas sencillas que les orientan sobre el trabajo restante. Al final del Sprint, el equipo revisa el Sprint con los interesados en el proyecto, y les enseña lo que han construido. La gente obtiene comentarios y observaciones que se puede incorporar al siguiente Sprint. Scrum pone el énfasis en productos que funcionen al final del Sprint que realmente estén “hechos”; en el caso del software significa que el código esté integrado, completamente probado y potencialmente para entregar [...]” (The Scrum Primer, 2009, pág. 5)*

Según Jeff Sutherland<sup>5</sup>, uno de los creadores de Scrum y del método de control empírico de procesos en el cual se basa dicha metodología, éste se sostiene en la implementación de tres pilares: transparencia (visibilidad de los procesos), inspección (periódica del proceso) y adaptación (de los procesos inspeccionado).

## 2. El Marco de Trabajo de Scrum

En un sentido amplio, el marco de trabajo de Scrum se compone de una serie de reglas, que definen roles que integran los equipos, artefactos necesarios para los procesos, bloques de tiempo preestablecidos y ceremonias que deben respetarse.

Los equipos de Scrum definen tres roles bien diferenciados:

1. El Scrum Master, responsable de asegurar los procesos;
2. El Dueño de Producto, responsable de maximizar el valor del producto;
3. El Equipo, responsable de realizar el trabajo.

La definición de bloques de tiempo iterativos (de 2 a 4 semanas), está destinada a crear continuidad y regularidad en las cuáles, se basarán las seis ceremonias (reuniones) que aseguran el cumplimiento de objetivos:

1. Reunión de Planificación de la Entrega
2. Reunión de Planificación del Sprint
3. El Sprint – corazón de Scrum
4. Reunión diaria
5. Reunión de revisión
6. Reunión de Retrospectiva

Cómo herramientas aplicables a todos los procesos anteriores, Scrum emplea cuatro artefactos:

1. **Backlog de Producto** (una lista priorizada de todo lo que requiere el software).
2. **Backlog de Sprint** (una lista de tareas necesarias para convertir parte del Backlog de Producto, en incremento de funcionalidad del Software),
3. **Scrum taskboard** (un tablero físico que permite la transparencia de los ítems de trabajos pendientes, en curso y terminados de un mismo Sprint),
4. **Diagrama de Burndow** (un gráfico que permite medir visualmente, el progreso de los ítems del Backlog de Producto)

*“[...] Las reglas sirven de unión para los bloques de tiempo, los roles y los artefactos de Scrum [...]. Por ejemplo, una regla de Scrum es que sólo los miembros del equipo - la gente comprometida a convertir el Product Backlog en un incremento - pueden hablar durante un Scrum Diario [...]” (Scrum Guide, 2008, Pág. 5)*

### 3. Los Roles en Scrum

Como comentamos anteriormente, Scrum define tres roles bien diferenciados: el Scrum Master, el dueño de Producto y el Equipo. En este capítulo, veremos en detalle, las responsabilidades y características de cada uno de ellos.

#### El Dueño de Producto (Product Owner) en Scrum

El Dueño de Producto (Product Owner) en Scrum El Dueño de Producto es la única persona autorizada para decidir sobre cuáles funcionalidades y características

funcionales tendrá el producto. Es quien representa al cliente, usuarios del software y todas aquellas partes interesadas en el producto.

*“El propietario del producto puede ser un miembro del equipo, que también participe en las tareas de desarrollo. Esta responsabilidad adicional puede disminuir la capacidad del Propietario del Producto de trabajar con los interesados del proyecto o producto. Sin embargo, el propietario del producto no puede ser nunca el ScrumMaster. ”*

*(Scrum Guide, 2008, Pág. 7)*

## Funciones y responsabilidades

- Canalizar las necesidades del negocio, sabiendo "escuchar" a las partes interesadas en el producto y transmitir las en "objetivos de valor para el producto", al scrum team.
- Maximizar el valor para el negocio con respecto al Retorno de Inversión (ROI), abogando por los intereses del negocio.
- Revisar el producto e ir adaptándole sus funcionalidades, analizando las mejoras que éstas puedan otorgar un mayor valor para el negocio.

## Aptitudes que debe tener un Dueño de Producto

- Excelente facilidad de comunicación en las relaciones interpersonales.
- Excelente conocimiento del negocio.
- Facilidad para análisis de relaciones costo/beneficio.
- Visión de negocios.

## El Scrum Master

El Scrum Master es el alma mater de Scrum. Un error frecuente es llamarlo "líder", puesto que el Scrum Master no es un líder típico, sino que es un auténtico Servidor neutral, que será el encargado de fomentar e instruir sobre los principios ágiles de Scrum.

*Un buen Scrum Master es aquel capaz de trabajar a la par del equipo, sintiéndose parte de él y siendo capaz de colocarse al servicio de éste, y no, al mando del mismo.*

## Funciones y responsabilidades

- Garantizar la correcta aplicación de Scrum. Esto incluye, desde la correcta transmisión de sus principios a las altas gerencias, hasta la prevención de la inversión roles (es decir, guardar especial cuidado en que el dueño de producto no actúe en nombre del Scrum Team y viceversa, o que la audiencia se inmiscuya en tareas que no le son propicias)
- Resolver los conflictos que entorpezcan el progreso del proyecto.
- Incentivar y motivar al Scrum Team, creando un clima de trabajo colaborativo, fomentar la auto-gestión del equipo e impedir la intervención de terceros en la gestión del equipo.

## Aptitudes que debe tener un Scrum Master

- Excelentes conocimientos de Scrum: Debe contar con amplios conocimientos de los valores y principios del agilísimo, así como de las pautas organizativas y prácticas de Scrum.
- Amplia vocación de servicio: Lisa llanamente, hablamos de tener vocación de servicio evitando la ambición de poder. El Scrum Master DEBE ser servicial al Scrum Team. Si



solo se pretende “llevar el mando”, liderar y sobresalir, no será una buena idea convertirse en Scrum Master.

- Tendencia altruista.
- Amplia capacidad para la resolución de problemas: La facilidad para resolver los impedimentos detectados será una característica clave de la cual, si se carece de ella, no habrá éxito posible.
- Analítico y observador: El poder de observación será fundamental a la hora de detectar los impedimentos que el Scrum Team deba enfrentar.
- Saber incentivar y motivar
- Capacidad docente e instructiva: ser didáctico y contar con una alta capacidad para transmitir los conocimientos de Scrum a toda la organización y principalmente, a los miembros del Scrum Team y Dueño de Producto
- Buen carisma para las negociaciones: Ser carismático y contar con una marcada tendencia negociadora, abrirá las puertas a que los cambios organizacionales de lo tradicional al agilísimo, se conviertan en un hecho que no provoque “roces”.

*"El ScrumMaster puede ser un miembro del Equipo; por ejemplo, un desarrollador realizando tareas del Sprint. Sin embargo, esto conduce frecuentemente a conflictos cuando el ScrumMaster tiene que elegir entre eliminar obstáculos o realizar las tareas. El ScrumMaster nunca debe ser el Propietario del Producto."*

*(Scrum Guide, 2008, pág.7)*

## Actitudes que un buen Scrum Master debe evitar indefectiblemente

- Entrometerse en la gestión del equipo sin dejarlos definir su propio proceso de trabajo: recordemos que el Scrum Team debe ser un equipo auto-gestionado. El Scrum Master NO PUEDE involucrarse en las decisiones organizativas del trabajo del equipo, pues no le corresponde.
- Asignar tareas a uno o más miembros del equipo.
- Colocarse en actitud de autoridad frente al equipo: el Scrum Master no tiene “superpoderes”. Debe ser servil y no intentar “dominar”. Debe ser “parte del equipo”. Una actitud autoritaria, podría provocar el quiebre del equipo, obteniendo resultados poco deseables.
- Negociar con el Dueño de Producto la calidad del producto a desarrollar: el Scrum Master debe tener muy en claro, que solo el cliente tiene autoridad para decidir que otorga valor al negocio y que no.
- Delegar la resolución de un conflicto a un miembro del equipo o a un tercero: el Scrum Master tiene la obligación de tomar en sus manos la resolución de un problema. Cuando ésta, no esté a su alcance, deberá “golpear las puertas” que sean necesarias, y ser él, quien realice un seguimiento a fin de asegurarse que el problema ha sido resuelto.

**El peor error que un Scrum Master puede cometer**, es delegar la resolución de un conflicto a un miembro del equipo, o intentar colocarse en posición de autoridad frente al Scrum Team.

## El Equipo de Desarrollo (Scrum Team o Equipo Scrum)

El Scrum Team (o simplemente "equipo"), es el equipo de desarrolladores multidisciplinario, integrado por programadores, diseñadores, arquitectos, testers y demás, que en forma auto-organizada, será los encargados de desarrollar el producto.

### Funciones y responsabilidades

- Convertir el Backlog de Producto, en software potencialmente entregable.

### Aptitudes que deben tener los integrantes de un Scrum Team

- Ser profesionales expertos o avanzados en su disciplina.
- Tener "vocación" (la buena predisposición no alcanza) para trabajar en equipo.
- Capacidad de auto-gestión.

### Un buen Scrum Team, actúa como un verdadero equipo

Las actitudes personales con las cuales, los miembros del Scrum Team deben contar y aquellas, que deben ser evitadas, se describen a continuación:

- **Ser solidario y colaborativo:** si quieres formar parte de un verdadero equipo, debes ser consciente de que tus compañeros, pueden necesitar tu ayuda profesional. Cuando un compañero de equipo no sepa algo que tú sí sepas, debes ser solidario con tu conocimiento y colaborar con las tareas de tu compañero.
- **Ser motivador,** en vez de pretender sobresalir: No es lo mismo decir “yo puedo resolver eso” que “me gustaría que intentáramos resolverlo juntos”. Que un compañero de trabajo

sienta que lo apoyas, es motivarlo. Pero si te pones en el papel de “sabelotodo”, solo estarás adoptando una postura odiosa. En un equipo, no debes querer sobresalir: debes sentirte a gusto, logrando objetivos entre todos.

- **Evitar la competencia:** ¿cuál es el objetivo de querer ser “mejor” que tu compañero? Intentando ser “mejor que”, malgastas tus energías en competencias sin sentido. Puedes aprovechar toda esa energía, en aprender de tus compañeros aquello que ignoras, y enseñarles aquello en lo que tú eres bueno. Notarás la gran diferencia, al finalizar cada sprint, y ver que el resultado de lo entregado, es lo que verdaderamente sobresale.

*Aprende a disfrutar de los logros de tu equipo y verás como el orgullo es mayor, que cuando solo intentas sobresalir en algo.*

---

### **Descargo de responsabilidad**

La información contenida en este documento descargable en formato PDF o PPT es un reflejo del material virtual presentado en la versión online del curso. Por lo tanto, su contenido, gráficos, links de consulta, acotaciones y comentarios son responsabilidad exclusiva de su(s) respectivo(s) autor(es) por lo que su contenido no compromete al área de e-Learning del Departamento GES o al programa académico al que pertenece.

El área de e-Learning no asume ninguna responsabilidad por la actualidad, exactitud, obligaciones de derechos de autor, integridad o calidad de los contenidos proporcionados y se aclara que la utilización de este descargable se encuentra limitada de manera expresa para los propósitos educativos del curso.

