



## Características de Scrum

**Scrum** es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

**Los roles principales en Scrum son:**

**El *ScrumMaster***, que mantiene los procesos y trabaja de forma similar al director de proyecto

**El *Product Owner***, que representa a los *stakeholders* (interesados externos o internos), y el *Team* que incluye a los desarrolladores.

Durante cada *sprint*, un periodo entre 15 y 30 días (la magnitud es definida por el equipo), el equipo crea un incremento de software *potencialmente entregable* (utilizable).

El conjunto de características que forma parte de cada sprint viene del ***Product Backlog***, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar.

Los elementos del ***Product Backlog*** que forman parte del sprint se determinan durante la reunión de *Sprint Planning*.

Durante esta reunión, el ***Product Owner*** identifica los elementos del ***Product Backlog*** que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.

Durante el **sprint**, nadie puede cambiar el ***Sprint Backlog***, lo que significa que los requisitos están congelados durante el sprint.

**Scrum** permite la creación de equipos autoorganizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

**Un principio clave de Scrum** es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado *requirements churn*), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada.

Por lo tanto, **Scrum** adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Existen varias implementaciones de sistemas para gestionar el proceso de **Scrum**, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.



**DISEÑO DE APLICACIONES WEB**  
**UNIVERSIDAD NACIONAL DE LA MATANZA**  
**Metodologías Ágiles**  
**Lic. Sergio D. Conde**

## **Roles en Scrum**

En Scrum se definen varios roles, estos están divididos en dos grupos: cerdos y gallinas. El nombre de los grupos están inspirados en la siguiente frase

En un plato de huevos con tocino el cerdo está comprometido, la gallina sólo está involucrada.

De esta forma, los 'cerdos' están comprometidos a desarrollar el software de forma regular y frecuente, mientras que todos los demás son 'gallinas' que sólo interesados en el proyecto, y si este falla, ellos no son los cerdos, es decir, ellos no fueron los que se comprometieron a hacerlo.

Las necesidades, deseos, ideas e influencias de los roles 'gallina' se tienen en cuenta, pero no de forma que pueda afectar, distorsionar o entorpecer el proyecto Scrum.

### **Roles**

#### **Product Owner**

El *Product Owner* representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

#### **ScrumMaster (o Facilitador)**

El *Scrum* es facilitado por un *ScrumMaster*, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El *ScrumMaster* no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

#### **ScrumTeam o Equipo**

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).



**DISEÑO DE APLICACIONES WEB**  
**UNIVERSIDAD NACIONAL DE LA MATANZA**  
**Metodologías Ágiles**  
**Lic. Sergio D. Conde**

### **Roles "Gallina"**

Los roles gallina en realidad no son parte del proceso Scrum, pero deben tenerse en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (stakeholders). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

### **Análisis de la frase "Rol gallina":**

La gallina alimenta al proyecto "poniendo huevos", no se ve comprometida como el cerdo que va al matadero.

#### **Usuarios**

Es el destinatario final del producto. Como bien lo dice la paradoja, El árbol cae en el bosque cuando no hay nadie ¿Hace ruido? Aquí la definición sería *Si el software no es usado ¿fue alguna vez escrito?*

#### **Stakeholders (Clientes, Proveedores, Inversores)**

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del sprint.

#### **Managers**

Es la gente que establece el ambiente para el desarrollo del producto.

## **Reuniones en Scrum**

### **Daily Scrum**

Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama "daily standup". El scrum tiene unas guías específicas:

La reunión comienza puntualmente a su hora. A menudo hay castigos - acordados por el equipo- para quien llegue tarde (por ejemplo: dinero, flexiones, llevar colgando una gallina de plástico del cuello, etc)

Todos son bienvenidos, pero sólo los "cerdos" pueden hablar.

La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.



**DISEÑO DE APLICACIONES WEB  
UNIVERSIDAD NACIONAL DE LA MATANZA**

**Metodologías Ágiles**

**Lic. Sergio D. Conde**

**Todos los asistentes deben mantenerse de pie (esto ayuda a mantener la reunión corta)**

**La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.**

**Durante la reunión, cada miembro del equipo contesta a tres preguntas:**

**¿Qué has hecho desde ayer?**

**¿Qué es lo que estás planeando hacer hoy?**

**¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del ScrumMaster recordar estos impedimentos).**

**Scrum de Scrum**

**Cada día normalmente después del “Daily Scrum”**

**Estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración.**

**Asiste una persona asignada por cada equipo.**

**La agenda será la misma que la del Daily Scrum, añadiendo además las siguientes cuatro preguntas:**

**¿Qué ha hecho tu equipo desde nuestra última reunión?**

**¿Qué hará tu equipo antes que nos volvamos a reunir?**

**¿Hay algo que demora o estorba a tu equipo?**

**¿Estás a punto de poner algo en el camino del otro equipo?**

**Reunión de Planificación del Sprint (Sprint Planning Meeting)**

**Al inicio del ciclo Sprint (cada 15 o 30 días), una “Reunión de Planificación del Sprint” se lleva a cabo.**

**Seleccionar qué trabajo se hará**

**Preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo.**

**Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint**

**Ocho horas como límite**

**Al final del ciclo Sprint, dos reuniones se llevaran a cabo: la “Reunión de Revisión del Sprint” y la “Retrospectiva del Sprint”**

**Reunión de Revisión del Sprint (Sprint Review Meeting)**

- **Revisar el trabajo que fue completado y no completado**
- **Presentar el trabajo completado a los interesados (alias “demo”)**
- **El trabajo incompleto no puede ser demostrado**



- Cuatro horas como límite

### Retrospectiva del Sprint (Sprint Retrospective)

Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

## Sprint

El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo en base a su propia experiencia. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo en base al ritmo del equipo, aunque sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y deberían entregar un producto con características de *utilizable por el cliente*. Asimismo se recomienda no cambiar los objetivos del sprint o *sprint backlog* a menos que la falta de estos cambios amenacen al éxito del proyecto. La constancia hace a la concentración y la mejor productividad del equipo de trabajo.

## Documentos

### Product backlog

El *product backlog* es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. priorizadas según su retorno sobre la inversión (ROI). Es el *qué* va a ser construido. Es abierto y cualquiera puede modificarlo. Contiene estimaciones *grosso modo*, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Esta estimación ayuda al *product owner* a ajustar la línea temporal y, de manera limitada, la prioridad de las diferentes tareas. Por ejemplo, si dos características tienen el mismo valor de negocio la que requiera menos tiempo de desarrollo tendrá probablemente más prioridad, debido a que su ROI será más alto.



**DISEÑO DE APLICACIONES WEB**  
**UNIVERSIDAD NACIONAL DE LA MATANZA**  
**Metodologías Ágiles**  
**Lic. Sergio D. Conde**

### **Sprint backlog**

El sprint backlog es un documento detallado donde se describe el *cómo* el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en *horas* con ninguna tarea de duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida en otras menores. Las tareas en el *sprint backlog* nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

### **Burn down**

La *burn down chart* es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.