

Nociones básicas de SCRUM



¿Qué es SCRUM?

Características

Roles

Reuniones

Documentos

Anexos:

¿Qué es SCRUM?

SCRUM no son siglas que signifiquen algo, más bien simboliza la idea de grupo o “piña” de un equipo de Rugby.



SCRUM es **una metodología para la gestión y desarrollo de software** que en contra del modelo en cascada está basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Los procesos ágiles de desarrollo de software intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

¿Qué es SCRUM?

En Ingeniería de software el **desarrollo en cascada**, también llamado **modelo en cascada**, es el enfoque metodológico que ordena rigurosamente las etapas del **ciclo de vida del software**, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Un ejemplo de una metodología de desarrollo en cascada es:

1. Análisis de requisitos
2. Diseño del Sistema
3. Diseño del Programa
4. Codificación
5. Pruebas
6. Implantación
7. Mantenimiento

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo. La palabra *cascada* sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.

Si bien ha sido ampliamente criticado desde el ámbito académico y la industria, sigue siendo el paradigma más seguido al día de hoy.

SCRUM es **una metodología para la gestión y desarrollo de software** que en contra del *modelo en cascada* está basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el [desarrollo ágil de software](#).

Los [procesos ágiles](#) de desarrollo de software intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

¿Qué es SCRUM? ejemplo



En un proyecto de desarrollo (*gestionado en cascada*) un cliente pide a un equipo una serie de **requisitos** básicos para que le fabriquen el deportivo amarillo de sus sueños:

- Vehículo biplaza con 4 ruedas
- Más de 300 cv de potencia
- Longitud mayor de 4 metros
- Color amarillo brillante con detalles en negro
- Llantas de al menos 17"
- Radio-CD con MP3
- ...

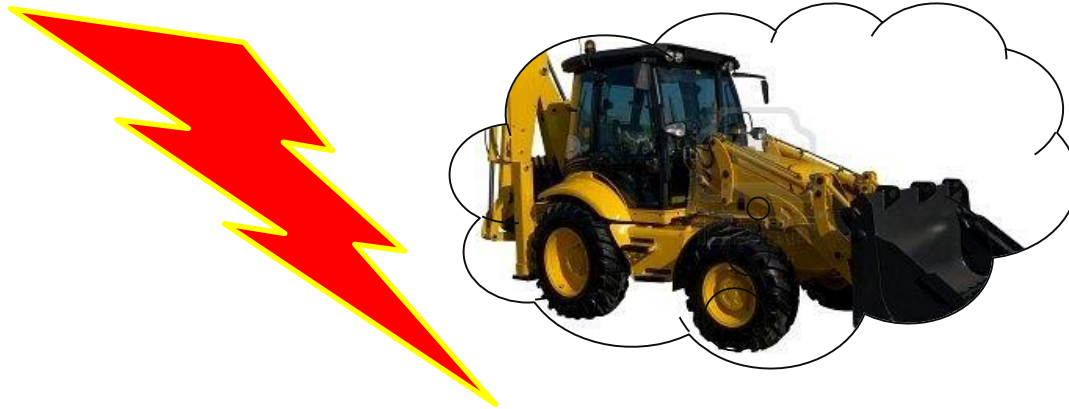
Acuerdan plazo de entrega y formas de pago



Tras unos cuantos meses/años de arduo trabajo, el equipo presenta el vehículo terminado y listo para circular al cliente esperando recibir la parte final del pago por su esfuerzo ...

¿Qué es SCRUM? ejemplo

Tras unos cuantos meses/años de arduo trabajo, el equipo presenta el vehículo terminado y listo para circular al cliente esperando recibir la parte final del pago por su esfuerzo ...



El resultado presentado no es el esperado por el cliente con la lógica pérdida de tiempo/dinero.

Posiblemente con la metodología SCRUM no hubiéramos llegado tan lejos de forma errónea.

Características

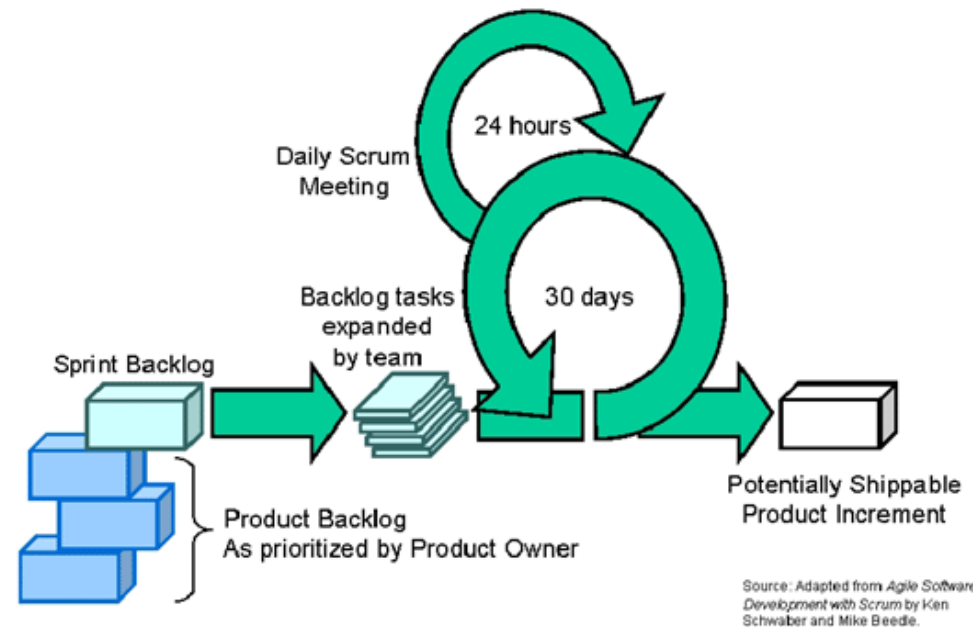
Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el **ScrumMaster**, que mantiene los procesos y trabaja de forma similar al director de proyecto, el **ProductOwner**, que representa a los Clientes 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. *(ejemplo del horno y las magdalenas)*



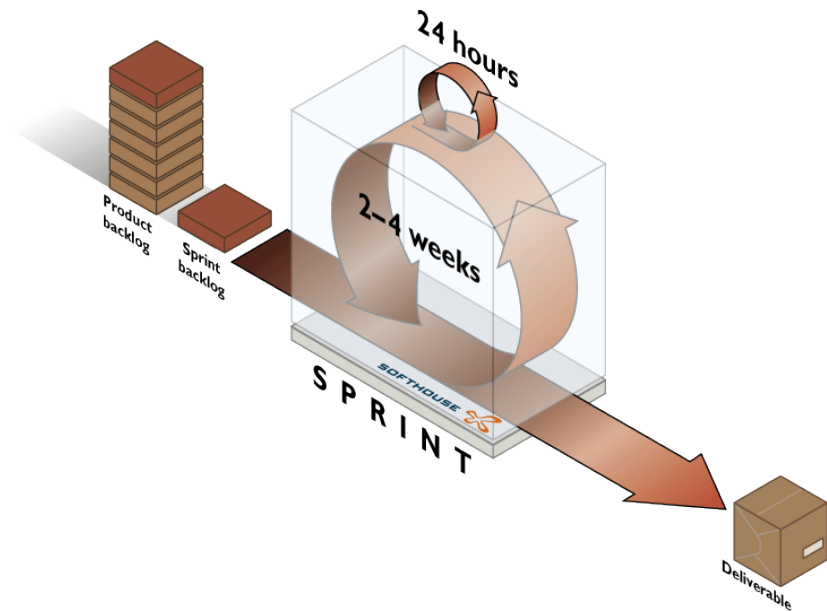
Características

Scrum permite la creación de equipos autoorganizados impulsando 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 (**REQUISITOS**), y que los **imprevistos** 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.



En Scrum se definen varios roles, estos están divididos en dos grupos: **cerdos y gallinas**.

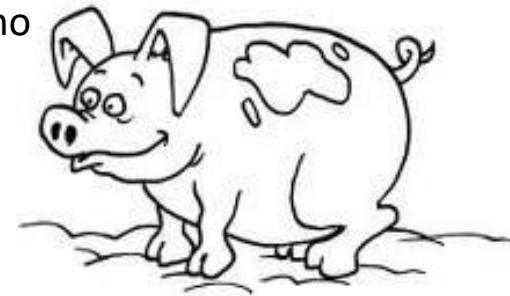
El nombre de los grupos están inspirados en el siguiente chiste sobre un cerdo y una gallina:

Un cerdo y una gallina se encuentran en la calle. La gallina mira al cerdo y dice: "Hey, ¿por qué no abrimos un restaurante?" El cerdo mira a la gallina y le dice: "Buena idea, ¿cómo se llamaría el restaurante?"

La gallina piensa un poco y contesta: "¿Por qué no lo llamamos "Huevos con jamón?" "Lo siento pero no", dice el cerdo, "Yo estaría comprometido pero tú solamente estarías involucrada".

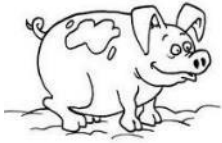
De esta forma, los *cerdos* están comprometidos a construir software de manera regular y frecuente, mientras que el resto son *gallinas*: interesados en el proyecto pero realmente irrelevantes porque, si éste falla, no son un *cerdo*, es decir, no son los que de manera comprometida ponen su propio pellejo (y carne) para sacar el proyecto adelante.

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.



Roles "Cerdo":

Los *Cerdos* son los que están **comprometidos** con el proyecto y el proceso Scrum; son los que "ponen el jamón en el plato".



- **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:** Su 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. Es el que hace que las reglas se cumplan.
- **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).

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. Es importante que participen y retroalimenten con respecto a la salida del proceso a fin de revisar y planear cada sprint.



- **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:** Son las personas encargadas de cada equipo de trabajo en general en la empresa.

Reuniones

➤ Daily Scrum:

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

- La reunión comienza puntualmente a su hora. A menudo hay castigos (acordados por el equipo) para quién llega tarde (por ejemplo: dinero, flexiones, llevar colgando una gallina de plástico del cuello, etc)
- Todos son bienvenidos, pero solo los "cerdos" pueden hablar.
- La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
- 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 como del Daily Scrum, además de 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:

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

- Seleccionar que 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
- Cuatro horas como límite

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

➤ Reunión de Revisión del Sprint (DEMO):

- 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:

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 no superable de cuatro horas.

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 valor para el negocio. Es el *qué* va a ser construido. Es abierto y cualquiera puede modificarlo. Contiene estimaciones a *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.

➤ 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 rota en mayor detalle. Las tareas en el *sprint backlog* nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.



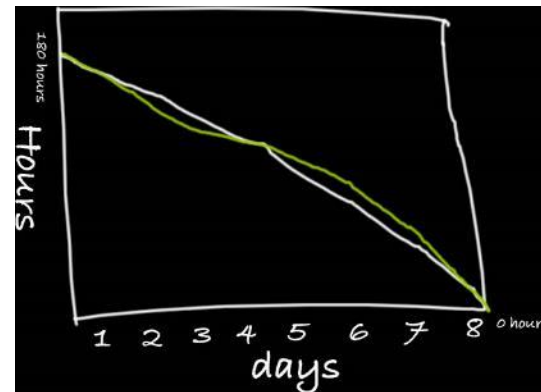
➤ Burn down:

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.

- En el eje horizontal se representa el **tiempo** del Sprint/Proyecto;
- En el eje vertical el número de **tareas a realizar** en ese tiempo determinado.

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.



Documentos



Anexos: chuleta de SCRUM

Roles



Scrum Team

- Team is cross-functional and consists of 5-9 people
- There are no set project roles within the team
- Team defines tasks and assignments
- Team is self-organizing and self-managing
- Maintains the Sprint Backlog
- Conducts the Sprint Review



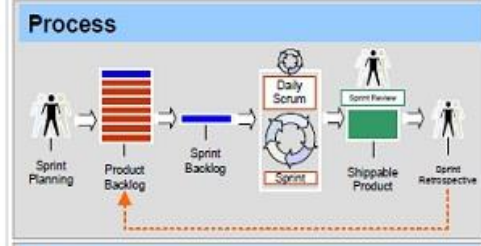
Product Owner (PO)

- Accountable for product success
- Defines all product features
- Responsible for prioritizing product features
- Maintains the Product Backlog
- Insures team working on highest valued features



Scrum Master (SM)

- Holds daily 15 minute team meeting (Daily Scrum)
- Removes obstacles
- Shields the team from external interference
- Maintains the Sprint Burndown Chart
- Conducts Sprint Retrospective at the end of a Sprint
- Is a facilitator not a manager



Tools

Task Board

- White Board containing teams Sprint goals, backlog items, tasks, tasks in progress, "DONE" items and the daily Sprint Burndown chart.
- Scrum meeting best held around task board
- Visible to everyone

Artifacts

Product Backlog - (PB)

- List of all desired product features
- List can contain bugs, and non-functional items
- Product Owner responsible for prioritizing
- Items can be added by anyone at anytime
- Each item should have a business value assigned
- Maintained by the Product Owner

Sprint Backlog - (SB)

- To-do list (also known as Backlog item) for the Sprint
- Created by the Scrum Team
- Product Owner has defined as highest priority

Burndown Chart - (BC)

- Chart showing how much work remaining in a Sprint
- Calculated in hours remaining
- Maintained by the Scrum Master daily

Release Backlog - (RB)

- Same as the Product Backlog. May involve one or more sprints dependent on determined Release date

"DONE" = Potentially Shippable!

FAQ

- **Who decides when a Release happens?** At the end of any given Sprint the PO can initiate a Release.
- **Who is responsible for managing the teams?** The teams are responsible for managing themselves.
- **What is the length of a task?** Tasks should take no longer than 16 hours. If longer then the task should be broken down further.
- **Who manages obstacles?** Primary responsibility is on the Scrum Master. However, teams must learn to resolve their own issues. If not able then escalated to SM.
- **What are two of the biggest challenges in Scrum?** Teams not self-managing, Scrum Master managing not leading.

Meetings

Sprint Planning - Day 1 / First Half

- Product backlog prepared prior to meeting
- First half - Team selects items committing to complete
- Additional discussion of PB occurs during actual Sprint

Sprint Planning - Day 1 / Second Half

- Occurs after first half done - PO available for questions
- Team solely responsible for deciding how to build
- Tasks created / assigned - Sprint Backlog produced

Daily Scrum

- Held every day during a Sprint
- Lasts 15 minutes
- Team members report to each other not Scrum Master
- Asks 3 questions during meeting
- "What have you done since last daily scrum?"
- "What will you do before the next daily scrum?"
- "What obstacles are impeding your work?"
- Opportunity for team members to synchronize their work

Sprint Review

- Team presents "done" code to PO and stakeholders
- Functionality not "done" is not shown
- Feedback generated - PB maybe reprioritized
- Scrum Master sets next Sprint Review

Sprint Retrospective

- Attendees - SM and Team. PO is optional
- Questions - What went well and what can be improved?
- SM helps team in discovery - not provide answers

Visibility + Flexibility = Scrum

Glossary of Terms

- **Time Box** - A period of time to finish a task. The end date is set and can not be changed
- **Chickens** - People that are not committed to the project and are not accountable for deliverables
- **Pigs** - People who are accountable for the project's success
- **Single Wringable Neck** - This is the Product Owner!

SCRUM CHEAT SHEET

Estimating

User Stories

- A very high level definition of what the customer wants the system to do.
- Each story is captured as a separate item on the Product Backlog
- User stories are NOT dependent on other stories
- **Story Template:**
- "As a <User> I want <function> So that <desired result>"
- **Story Example:**
- As a user, I want to print a recipe so that I can cook it.

Story Points

- A simple way to initially estimate level of effort expected to develop
- Story points are a relative measure of feature difficulty
- Usually scored on a scale of 1-10, 1=very easy through 10=very difficult
- **Example:**
- "Send to a Friend" Story Points = 2
- "Shopping Cart" Story Points = 9

Business Value

- Each User Story in the Product Backlog should have a corresponding business value assigned.
- Typically assign (L,M,H) Low, Medium, High
- PO prioritizes Backlog items by highest value

Estimate Team Capacity

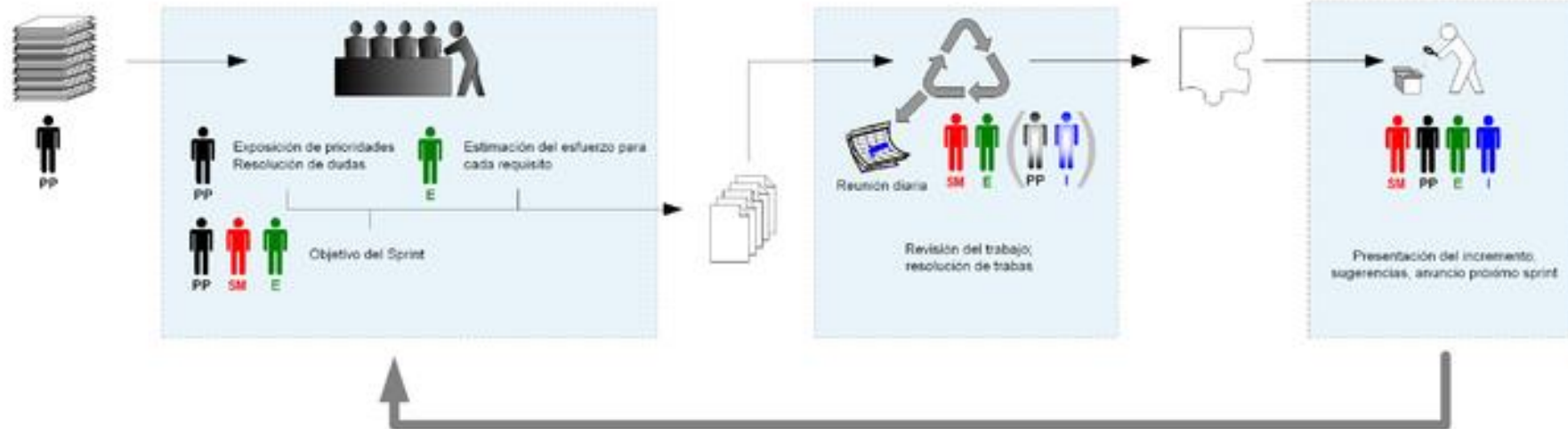
- Capacity = # Teammates (Productive Hrs x Sprint Days)
- Example - Team size is 4, Productive Hrs are 5, Sprint length is 30 days.
- Capacity = 4 (5 x 30) = 600 hours
- **NOTE:** Account for vacation time during the Sprint!

Velocity

- The rate at which team converts items to "DONE" in a single Sprint - Usually calculated in Story Points.

Anexos: proceso y módulos

PROCESO



ROLES

- PROPIETARIO DEL PRODUCTO (PP)**
Determina las prioridades. Una sola persona.
- SCRUM MANAGER (SM)**
Gestiona y facilita la ejecución del proceso.
- EQUIPO (E)**
Construye el producto.
- INTERESADOS (I)**
Asesoran y observan.

COMPONENTES

- PILA DEL PRODUCTO**
Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.
- PILA DEL SPRINT**
Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.
- INCREMENTO**
Parte del producto desarrollada en un sprint, en condiciones de ser usada (pruebas, codificación limpia y documentada).

REUNIONES

- PLANIFICACIÓN DEL SPRINT**
1 jornada de trabajo. El propietario del producto explica las prioridades y dudas del equipo. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El Scrum Manager define en una frase el objetivo del sprint.
- REUNIÓN DIARIA**
15 minutos de duración, dirigida por el Scrum Manager, sólo puede intervenir el equipo. ¿Qué hiciste ayer?, ¿Cuál es el trabajo para hoy?, ¿Qué necesitas?. Se actualiza la pila del sprint.
- REVISIÓN DEL SPRINT**
Informativa, aprox. 4 horas, moderada por el Scrum Manager, presentación del incremento, planteamiento de sugerencias y anuncio del próximo sprint.

SPRINT

- Ciclo de desarrollo básico de SCRUM, de duración máxima de 30 días en el que se desarrolla un incremento del producto.

VALORES

- Empowerment y compromiso de las personas
- Foco en desarrollar lo comprometido
- Transparencia y visibilidad del proyecto
- Respeto entre las personas
- Coraje y responsabilidad

¿Preguntas?

"There are no stupid questions. There are only stupid answers."

- T. Wilusz.



Alberto Jorge

Responsable de Proyectos



@albertoj



albertojorgej



<http://www.linkedin.com/in/albertoj>



https://www.xing.com/profile/Alberto_JorgeyJorge



<http://albertojorgej.blogspot.com>



Socio fundador <http://www.aercomunidad.org>



✓ Bibliografía recomendada: [SCRUM y XP desde las trincheras](#)



✓ Web de referencia recomendada: <http://www.scrummethodology.net>