



**“En un Scrum, hay que actuar como un unidad, no como 8 individuos. Todos tienen un rol. Nunca debemos olvidar que cuando trabajamos juntos como una unidad, el todo es más que la suma de las partes.”**

---

---

---

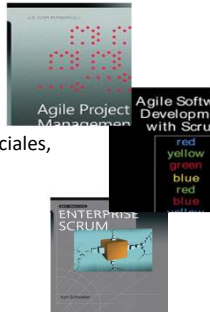
---

---

---

## Autores Scrum

- Ken Schwaber: desarrolló y formalizó Scrum para el desarrollo de sistemas
- Jeff Sutherland: pensamientos y prácticas iniciales, previo a su formalización con Ken.
- Mike Beedle: Utilizó y mejoró Scrum, integrándolo a XP.




---

---

---

---

---

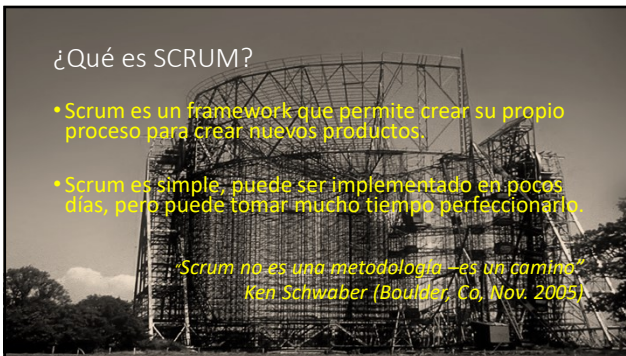
---

---

---

## ¿Qué es SCRUM?

- Scrum es un framework que permite crear su propio proceso para crear nuevos productos.
  - Scrum es simple, puede ser implementado en pocos días, pero puede tomar mucho tiempo perfeccionarlo.
- "Scrum no es una metodología – es un camino"*  
Ken Schwaber (Boulder, Co, Nov. 2005)




---

---

---

---

---

---

---

---

## Scrum es empírico

- Las metodologías rigurosas se basan en métodos definidos, con la idea de línea de ensamble
- El control en Scrum se alcanza con inspecciones frecuentes y correspondientes ajustes

---

---

---

---

---

---

---

---




---

---

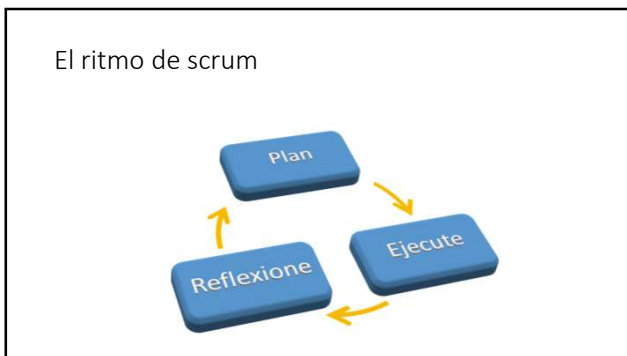
---

---

---

---

---




---

---

---

---

---

---

---

- Cimientos
- Empirismo
  - Auto-organización
  - Colaboración
  - Priorización
  - Time Boxing

---

---

---

---

---

---

---

## Cimientos

### • Empirismo

El **empirismo** es una teoría filosófica que enfatiza el papel de la experiencia, ligada a la percepción sensorial, en la formación del conocimiento. Para el empirismo más extremo, la experiencia es la base de todo conocimiento, no sólo en cuanto a su origen sino también en cuanto a su contenido. Se parte del mundo sensible para formar los conceptos y éstos encuentran en lo sensible su justificación y su limitación.

Procesos definidos y planificación detallada en las primeras fases son reemplazadas por ciclos de inspección y revisión just in time y ciclos adaptativos.




---

---

---

---

---

---

---

---

## Cimientos

### • Auto organización

La **auto-organización** es un proceso en el que la organización interna de un sistema, generalmente abierto, aumenta de complejidad sin ser guiado por ningún agente externo. Normalmente, los sistemas auto-organizados exhiben propiedades emergentes.

La auto-organización es objeto de estudio interdisciplinar, pues es una propiedad característica de los sistemas complejos, ya sean éstos matemáticos, físicos, químicos, biológicos, sociales o económicos.

**Pequeños grupos de trabajo que manejan su propia carga de tareas y se organizan entre ellos alrededor de un objetivo claro y tomando en cuenta las restricciones.**




---

---

---

---

---

---

---

---

## Cimientos

### • Colaboración

La **colaboración** se refiere abstractamente a todo proceso donde se involucre el trabajo de varias personas en conjunto. También cuando ayuda a una persona a hacer algo que se le dificulte, o en caso de que no pueda hacerlo.

**Líderes de Scrum, diseñadores de productos y clientes colaboran con los desarrolladores – ellos no los gerencian o dirigen.**




---

---

---

---

---

---

---

---

## Cimientos

- Priorización

Dar prioridad a alguna cosa

**Trabajar en lo más importante – no perder el tiempo haciendo foco en el trabajo que no tiene y/o agrega valor.**




---

---

---

---

---

---

---

## Cimientos

- Time Boxing

**Time boxing** es una técnica de planificación en proyectos, típicamente de software, donde el schedule (programación) es dividido en un número separado de períodos de tiempo (time box), normalmente entre 2 y 6 semanas, los cuales tienen sus propios entregables, fechas y costos.

**Timeboxing crea el ritmo que guía el desarrollo.**




---

---

---

---

---

---

---

## Entonces...¿Cómo trabaja Scrum?

- Equipos pequeños (< 10 personas, ideal 7+2)
- Una serie de Sprints (30 días)
- Incrementos visibles y usables
- Timeboxing
- Auto organización
- Colaboración
- Priorización

---

---

---

---

---

---

---

### Scrum se siente diferente...

- Menos tiempo planeando y definiendo tareas
- Menos tiempo creando y leyendo reportes
- Se pasa más tiempo con el equipo investigando la situación
- Scrum
  - Supone dominios impredecibles
  - No supone un proceso repetible

---

---

---

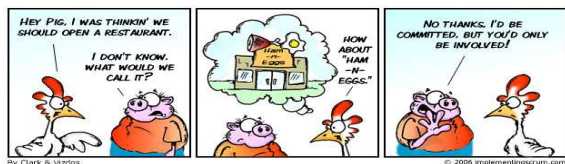
---

---

---

---

### Compromiso vs involucramiento




---

---

---

---

---

---

---

### Roles en Scrum

Scrum Master  
Product Owner  
Equipo de trabajo – Scrum Team




---

---

---

---

---

---

---

## Product Owner

- Controla y gestiona el Backlog.
- Una persona, no un Comité
- Nadie puede decirle al equipo una prioridad diferente
- Acciones visibles




---

---

---

---

---

---

---

## Scrum Master

- Es responsable de que las prácticas, valores y reglas se realicen
- Nexo entre la gerencia y el equipo
- Dirige los Scrum diarios, comparando el progreso planeado con lo real.
- Se asegura que se resuelven los impedimentos y se toman decisiones rápidas
- Trabaja con la gerencia y el cliente para identificar el dueño del producto
- El Scrum Master, el dueño del producto y el equipo producen un Backlog de producto




---

---

---

---

---

---

---

## Scrum Master - Actividades

- Toma decisiones en reuniones de Scrum
- Registra y resuelve problemas
- Mantiene el equipo enfocado
- Realiza el seguimiento de avance




---

---

---

---

---

---

---

## Responsabilidades del Scrum Master

- Mantener las reuniones diarias cortas mediante la aplicación de las reglas:
  - Hora y lugar fijos en lo posible
  - Los gerentes pueden asistir pero sólo los desarrolladores participan
  - Una reunión de Scrum no es una reunión de diseño

---

---

---

---

---

---

---

## El equipo

- 7+2 personas. Si hay más de 8 personas, hacer varios equipos trabajando sobre el mismo Backlog.
- Autónomos y auto-organizados
- Compromiso de entregar un conjunto de ítems del Backlog al final de un Sprint
- Libertad de acción
- Limitado por estándares y convenciones organizacionales
- No hay roles
- Todos codifican




---

---

---

---

---

---

---

## El equipo - Dinámica

- Scrum pretende ser una sombrilla bajo la cual el equipo puede dar lo mejor
- Scrum es empírico, y a veces se alcanza el objetivo reduciendo funcionalidad

---

---

---

---

---

---

---



## Ambiente de trabajo

- Abierto
  - El silencio absoluto es un mal signo
- Pizarrones
- El equipo define sus horarios




---

---

---

---

---

---

---

## Compromiso vs Involucramiento

- ¿Qué roles son Cerditos?
- Qué roles son Gallinas?
- Usuarios???
- Stakeholders???
- Dueño del producto???
- Gerencias/Gerentes??
- Scrum Master??

---

---

---

---

---

---

---

## Entregables

Product backlog  
Sprint Backlog  
Producto de software potencialmente listo para producción




---

---

---

---

---

---

---

### El backlog de producto (product backlog)

- *“Es una cola priorizada de funcionalidades técnicas y de negocio, que necesitan ser desarrolladas”*

Ken Schwaber

- El Backlog de producto contiene la lista de requerimientos
- Se listan características, funciones, tecnologías, mejoras, bugs, etc., que serán aplicadas al producto
- El Backlog está incompleto inicialmente, aunque solo se necesita lo suficiente para realizar el primer Sprint de 30 días

---

---

---

---

---

---

---

### El backlog se origina en

- Marketing
- Ventas
- Desarrollo
- Soporte al cliente
- Características del Backlog
  - priorizado
  - también contiene problemas a ser solucionados, y que son dependencias de otros ítems del Backlog

---

---

---

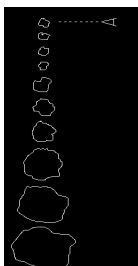
---

---

---

---

### Asteroides.. como refino un product backlog?



- Trabajar con un backlog de producto es como jugar a Asteroides
- Grandes rocas (epics) se rompen en pedazos de rocas más pequeñas (stories) hasta que son lo suficientemente pequeñas para ser eliminadas (desarrolladas y entregadas).

---

---

---

---

---

---

---

Estimaciones del Backlog

- Las estimaciones son un esfuerzo colaborativo entre las partes
- Las estimaciones son iterativas
- Si un ítem no puede ser debidamente estimado, se debe dividir en el Backlog.
- Los estimados sirven de base para la funcionalidad en el Sprint.



---

---

---

---

---

---

---

Product Backlog



- Los requisitos
- Una lista de todos los trabajos deseados en el proyecto
- Idealmente cada tema tiene valor para el usuarios o el cliente
- Priorizada por el Product Owner
- Repriorizada al comienzo de cada Sprint

---

---

---

---

---

---

---

Ejemplo de Product Backlog

Backlog item	Estimación
Permitir que un invitado a hacer una reserva.	3
Como invitado, quiero cancelar una reserva.	5
Como invitado, quiero cambiar las fechas de una reserva.	3
Como un empleado de hotel, puedo ejecutar informes de los ingresos por habitación disponible	8
Mejorar el manejo de excepciones	8
...	30
...	50

---

---

---

---

---

---

---

### Backlog del Sprint

- El equipo determina que debe hacerse para cumplir el objetivo del Sprint
- El dueño del producto suele asistir
- Se realiza una lista de tareas que tomen de 4 a 16 horas para completarse
- El Backlog no se modifica durante el desarrollo
- Si el equipo descubre que el Backlog no puede realizarse, el Scrum Master y el dueño del producto determinan si:
  - Algún ítem puede ser removido del backlog
  - Alguna funcionalidad puede eliminarse

---

---

---

---

---

---

---

### Ejemplo de Sprint Backlog

Tareas	L	M	M	J	V
Codificar UI	8	4	8		
Codificar negocio	16	12	10	4	
Testear negocio	8	16	16	11	8
Escribir ayuda online	12				
Escribir la clase foo	8	8	8	8	8
Agregar error logging			8	4	

---

---

---

---

---

---

---

### Producto de software potencialmente listo para producción

Parte del producto desarrollado en un sprint en condiciones de ser usado en el ambiente correspondiente.

---

---

---

---


---

---

---

# Reuniones

- Sprint Planning
- Daily
- Sprint Demo/review
- Restropectivas
- Story time/Grooming (opcional)




---

---

---

---


---

---

---

# Reuniones de planeamiento de Sprint

- Dos reuniones consecutivas
  - El equipo se reúne con el dueño del producto, la gerencia y los usuarios para decidir la funcionalidad a implementar
  - El equipo decide COMO llevarlo a cabo
- Inputs
  - Backlog, último incremento, métricas




---

---

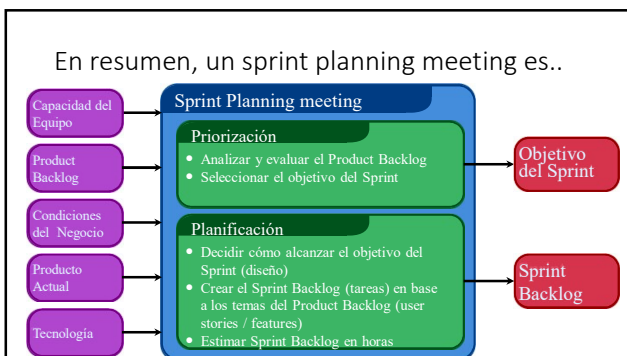
---

---

---

---

---




---

---

---

---

---

---

---

### Reuniones de Scrum diarias

- Cortas (15-30 min). Se analiza el estado. Facilitador: Scrum Master
- Asiste todo el equipo.
- No para soluciones de problemas
- El Scrum Master pregunta 3 cosas a los asistentes:
  - ¿Qué se completó desde la última reunión?
  - ¿Qué obstáculos se presentaron?
  - ¿Qué va a hacer hasta la próxima reunión?
- No es dar un status report al Scrum Master
- Se trata de compromisos delante de pares

---

---

---

---

---

---

---

---

### Sprint Demo/review

- Reunión informativa de unas 4 horas
- El equipo presenta el incremento desarrollado a gerentes, clientes, usuarios y el dueño del producto
- Informal
  - Regla de 2 hs preparación
  - No usar diapositivas
- Todo el equipo participa
- Se invita a todo el mundo
- Se reportan los problemas encontrados
- CUALQUIER ítem puede ser agregado, eliminado, re-priorizado
- Se estima el nuevo Sprint y se asignan tareas

---

---

---

---

---

---

---

---

### Sprint retrospective

- Periódicamente, se echa un vistazo a lo que funciona y lo que no
- Normalmente 1hr a 4hs
- Se realiza luego de cada sprint
- Todo el equipo participa
  - Scrum Master
  - Product Owner
  - Equipo
  - Posiblemente clientes y otros

---

---

---

---

---

---

---

---

Story time

- El equipo se reúne con el PO para discutir ítems del backlog de alta prioridad, determinar su criterio de aceptación y asignar la priorización a cada nuevo ítem.
- Esto ocurre inicialmente antes del desarrollo y después iterativamente en cada sprint.

---

---

---

---

---

---

---

Consideraciones generales

---

---

---

---

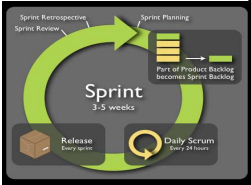
---

---

---

Sprint

- Período fijo de tiempo, sugerido 30 días
- Dentro del Sprint
  - No se puede cambiar el alcance
  - No se puede agregar funcionalidad
  - No se pueden modificar las reglas del equipo



---

---

---

---

---

---

---

### Objetivo del sprint

- Una declaración
- Un objetivo claro a alcanzar

---

---

---

---

---

---

---

### Reglas de un Sprint

- Foco en la tarea
- SIN interrupciones
- SIN cambios
- El mismo equipo puede descubrir más trabajo a ser hecho

---

---

---

---

---

---

---

### Elementos de un Sprint

- Reuniones de Scrum
- Se produce un incremento usable y visible
- Los incrementos se basan en un producto anterior
- Compromiso de los miembros a la asignación

---

---

---

---

---

---

---



### Mecánica de los Sprints

- En un Sprint se congelan 3 de las 4 variables de un proyecto:
  - Tiempo: 30 días
  - Costo: salarios + ambiente
  - Calidad: generalmente un estándar organizacional
- El Equipo puede cambiar funcionalidad siempre y cuando cumpla con el objetivo del Sprint.

---

---

---

---

---

---

---

### Tareas obligatorias

- En un Sprint hay 2 tareas mandatorias
  - Reuniones Scrum diarias: participan TODOS los miembros del equipo
  - Backlog del Sprint: debe estar actualizado y con los últimos estimados de los desarrolladores

---

---

---

---

---

---

---

### Cancelación de un Sprint

- Se puede cancelar un Sprint si las circunstancias hacen que no sea necesario
  - El objetivo se vuelve obsoleto
  - Las condiciones técnicas o de mercado cambian
- Decidido por el equipo
  - Porque no se puede alcanzar el objetivo
  - Se encuentra un problema muy grande
- Cancelar un Sprint es un costo

---

---

---

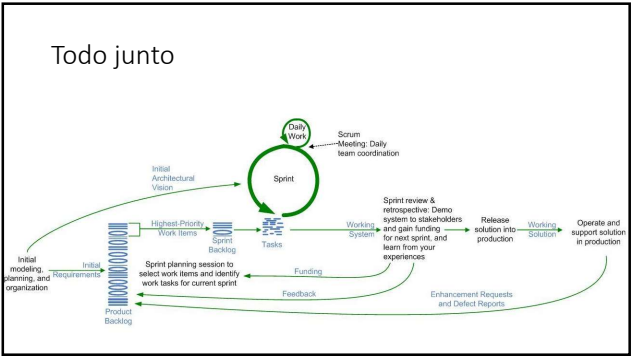
---

---

---

---

Todo junto



---

---

---

---

---

---

---

---

Herramientas de Scrum



---

---

---

---

---

---

---

---

TASKBOARD

**TARJETA**

- Código del ítem Backlog
- Número del Sprint
- Nombre de la Actividad
- Iniciales de la Persona
- Tiempo Estimado en Horas

---

---

---

---

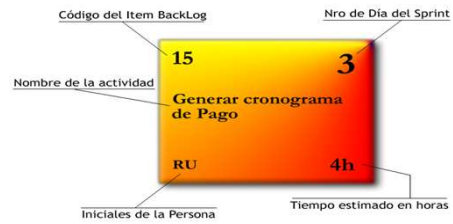
---

---

---

---

Taskboard



---

---

---

---

---

---

---

Taskboard

- TABLERO**
- Story
  - To Do
  - Work In Process
  - To Verify
  - Done

---

---

---

---

---

---

---

Taskboard

Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... 9 Test the... SC 8
	Code the... 2	Code the... 6	Test the... SC 8		Test the... SC 5 Test the... SC 6
	Test the... 8	Test the... 4			
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC 5 Test the... SC 6
	Code the... 4	Code the... 6			

---

---

---

---

---

---

---



---

---

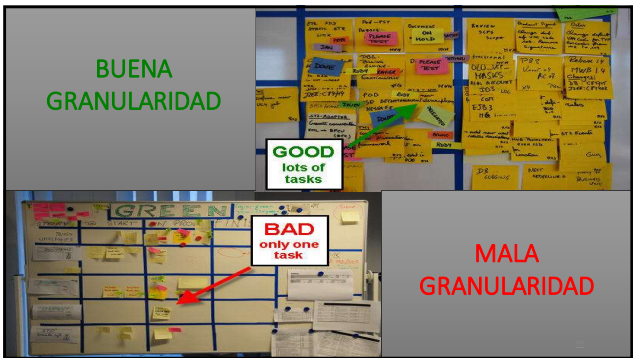
---

---

---

---

---



---

---

---

---

---

---

---

Gráficos de Backlog

- La gerencia necesita datos acerca de:
  - Progreso del Sprint
  - Progreso del Release
  - Progreso del producto
- El backlog de trabajo es la cantidad de trabajo que queda por ser realizado
- Tendencia del Backlog: trabajo que queda vs. tiempo

---

---

---

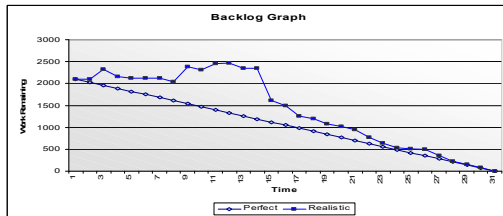
---

---

---

---

## Ejemplo




---

---

---

---

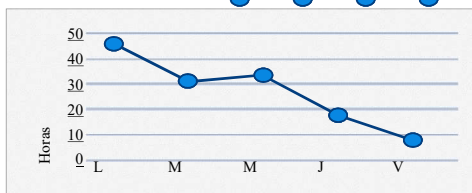
---

---

---

---

Tareas	L	M	M	J	V
Codificar UI	8	4	8		
Codificar Negocio	16	12	10	7	
Testear Negocio	8	16	16	11	8
Escribir ayuda online	12				




---

---

---

---

---

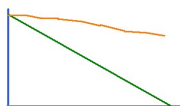
---

---

---

## BURNDOWN Chart

### Diagnóstico?



- No todo lo incluido en el sprint será entregado en la fecha propuesta. El backlog no fue modificado para alcanzar la fecha (remover requerimientos o modificar la fecha)
- Posibles causas:
  - Pobres técnicas de estimación
  - Mal manejo de riesgos
  - Rotación del equipo o renuncias
- Este gráfico muestra que el Scrum Master (SM) y el Product Owner (PO) no reaccionaron para evitar esta situación.
- Evite llegar a este estadio!

---

---

---

---

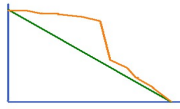
---

---

---

---

### Burndown charts



#### Diagnóstico?

- Este BC es mucho mejor que el anterior
- Al comienzo del sprint hubo problemas, pero en el medio el equipo de trabajo reaccionó y tomó medidas correctivas. Por la línea se nota que el PO quitó requerimientos para mantener la fecha de release.
- También pone en evidencia un mejor manejo de riesgos lo que facilitó aumentar la velocidad en la segunda parte del sprint.

---

---

---

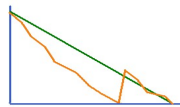
---

---

---

---

### Burndown charts



#### Diagnóstico?

- Este BC también tiene características negativas.
- Este no es un ejemplo de desarrolladores con las mejores habilidades técnicas. Demuestra pobres técnicas de estimación.
- El PO y el SM deberían haber agregado más requerimientos para ese release.

---

---

---

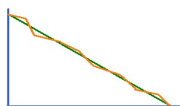
---

---

---

---

### Burndown charts



#### Diagnóstico?

- Ideal
- Buenas técnicas de estimación.
- Buena velocidad de desarrollo
- Buen manejo de impedimentos
- Todos los requerimientos prometidos son los entregados.
- Buen manejo de la asignación de tareas (tareas cortas, reporte diario)
- Quizás.....demasiado bueno para ser verdad ☺

---

---

---

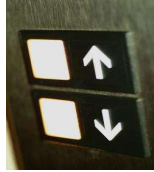
---

---

---

---

## BENEFICIOS DE SCRUM




---

---

---

---

---

---

---

### Beneficios de Scrum

- Se gestionan los cambios de requerimientos
- Se incorpora la visión de mercado
- Los clientes ven incrementos que refinan los requerimientos en un tiempo razonable
- Mejores relaciones con el cliente

---

---

---

---

---

---

---

Poniendo todo junto y repasando.....




---

---

---

---

---

---

---

## SCRUM en 100 palabras

### Algunos usuarios de SCRUM:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

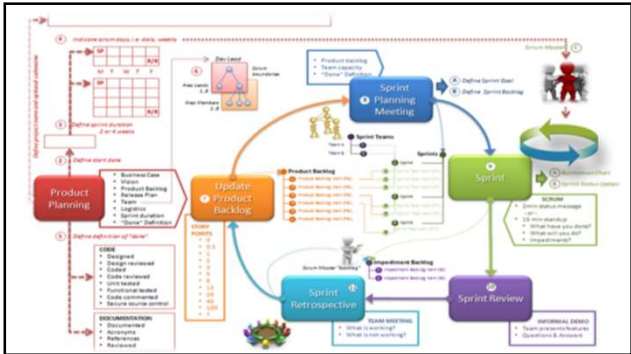
- Scrum es un proceso ágil que nos permite centrarnos en ofrecer el más alto valor de negocio en el menor tiempo.
- Nos permite rápidamente y en repetidas ocasiones inspeccionar software real de trabajo (cada dos semanas o un mes).
- El negocio fija las prioridades. Los equipos se auto-organizan a fin de determinar la mejor manera de entregar las funcionalidades de más alta prioridad.
- Cada dos semanas o un mes, cualquiera puede ver el software real funcionando y decidir si liberarlo o seguir mejorándolo en otro sprint.

## Se ha utilizado SCRUM para:

- Software comercial
- Desarrollos internos
- Desarrollos bajo Contrato
- Proyectos Fixed-price
- Aplicaciones Financieras
- Aplicaciones certificadas ISO 9001
- Sistemas Embebidos
- Sistemas con requisitos 7x24 y 99.999% de disponibilidad
- Joint Strike Fighter
- Desarrollo de video juegos
- Sistemas críticos de soporte vital, aprobados por la FDA
- Software de control satelital
- Sitios Web
- Software para Handheld
- Teléfonos portátiles
- Aplicaciones de Network switching
- Aplicaciones de ISV
- Algunas de las más grandes aplicaciones en uso

## Mas que mil palabras





---

---

---

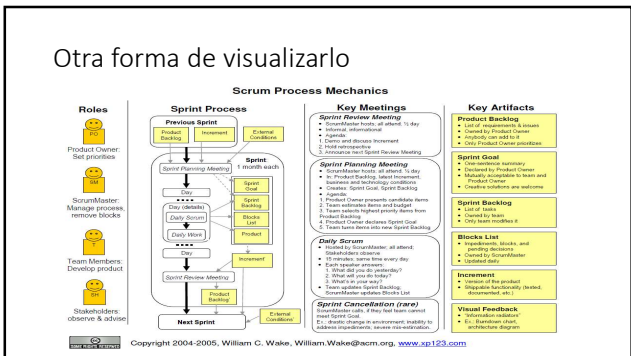
---

---

---

---

---



---

---

---

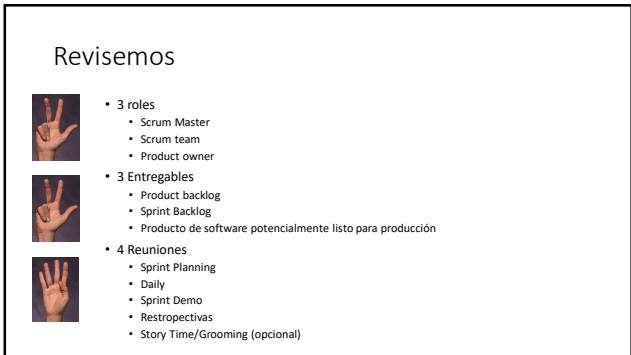
---

---

---

---

---



---

---

---

---

---

---

---

---

Cambios en un sprint??

**NO!!!!!!!!!!!!!!**

Y que se entienda, muchas implementaciones exitosas se basan en diseñar la duración de los sprints para contener los cambios fuera del sprint.

---

---

---

---

---

---

---

Pensamiento Final

*“Scrum **no es para todos**, sino para aquellos que tienen que trabajar con sistemas funcionando con la complejidad de tecnologías inestables y el surgimiento de requerimientos”*

Ken Schwaber

---

---

---

---

---

---

---

Para pensar....

- ¿Por qué la gente está tan “emocionada” con estas “nuevas” metodologías?
- ¿Cuándo scrum es apropiado?
- ¿Cuándo algo agile es no apropiado?
- Sino puede aplicar todo, que aplicaría?

---

---

---

---

---

---

---



---

---

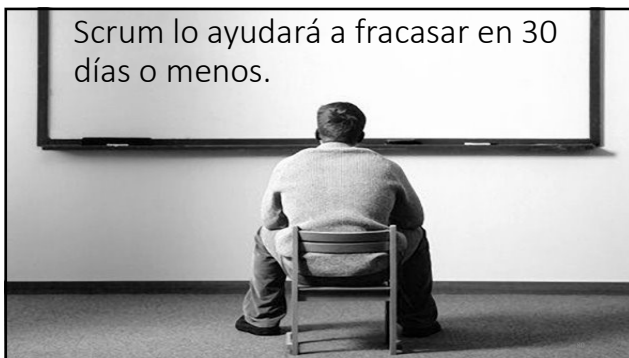
---

---

---

---

---



---

---

---

---

---

---

---



---

---

---

---

---

---

---

### Regla de Oro Ágil sobre Métricas

- La medición es una salida, no una actividad

Una filosofía minimalista sobre las

Medir la que sea necesario  
*y nada más.*

---

---

---

---

---

---

---

### Dos principios ágiles que guían la elección de las Métricas

"Nuestra mayor prioridad es satisfacer al cliente por medio de entregas tempranas y continuas de software valioso."

"El Software trabajando es la principal medida de progreso."

---

---

---

---

---

---

---

### Capacidad

- **Capacidad es...**

Una estimación de cuanto trabajo puede completarse en un período de tiempo dado.

Basado en la cantidad de tiempo ideal disponible del equipo.

- **...Se puede medir en...**

Esfuerzo (horas)

Puntos de Historia (Story Points)

---

---

---

---

---

---

---

## Capacidad

- **Horas de Trabajo Disponibles por día (WH) X Días Disponibles Iteración (DA) = Capacidad**

$$WH \times DA = \text{Capacidad}$$

### Ejemplo:

- Equipo de 8 miembros
- 4 miembros disponibles los 2 primeros sprints.
- 1 miembro se casa en sprints 5 y 6
- 6 horas de trabajo

---

---

---

---

---

---

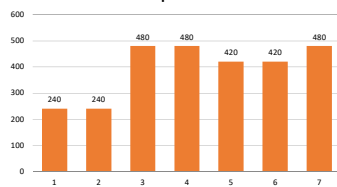
---

---

## Capacidad

Sprint	1	2	3	4	5	6	7	Total
Horas	240	240	480	480	420	420	480	2760
Puntos de Historia	30	30	45	60	58	52	60	335

Capacidad




---

---

---

---

---

---

---

---

## Capacidad: consideraciones

- Individuos calculan capacidad realista
- Aplicar estimaciones honestas a sus tareas
- Considerar una capacidad máxima de 50%-70%
- Comprender la capacidad a largo plazo con la velocidad y los puntos de historia-
- ¿Cuál es el promedio de finalización de un punto de historia para un equipo/individuo?

11

---

---

---

---

---

---

---

---

## Velocidad

*Velocidad es...*

- Una observación empírica de la capacidad del equipo para completar el trabajo por iteración.
- Comparable entre iteraciones para un equipo dado y un proyecto dado

*...y no es...*

- una estimación
- un objetivo a alcanzar
- comparable entre equipos
- comparable entre proyectos

---

---

---

---

---

---

---

---

## Unidad de medida de la Velocidad

Cómo planea el equipo	Unidad de Medida
Compromiso con las historias	Historias
Tamaño relativo (puntos)	Puntos de Historia
Estimación (horas ideales)	Horas Ideales

- ¿Qué cuenta para la velocidad?
- **Solo cuenta el trabajo completado para la velocidad**

---

---

---

---

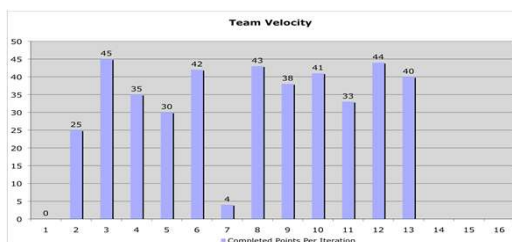
---

---

---

---

## Velocidad




---

---

---

---

---

---

---

---

Dudas, Preguntas? Donaciones  
pro-vacaciones!!!???




---

---

---

---

---

---

---

---

### Fuentes de Información

- Agile Software Development with Scrum. Ken Schwaber & Mike Beedle
- Agile Estimating and Planning, Mike Cohn,  
Pearson Education, ISBN: 9780137126347
- <http://www.planningpoker.com/detail.html>
- <http://www.controlchaos.com/> Ken Schwaber's site
- <http://www.Xbreed.org> Mike Beedle's site
- <http://www.agilealliance.com/articles> Agile Alliance's site
- <http://www.jeffsutherland.org/scrum/index.html>
- <http://www.lindarising.org> -- click on Articles
- <http://agilemanifesto.org/> Manifesto for Agile Software Development
- <http://www.infoq.com/>

---

---

---

---

---

---

---

---

### Videos

- <http://www.youtube.com/user/techexcel?v=kYajjGi5-M&feature=pyv&ad=7914006731&kw=scrum>
- <http://www.youtube.com/watch?v=Q5k7a9YEoU>

---

---

---

---

---

---

---

---

### Una lista de lecturas sobre Scrum

- *Agile and Iterative Development: A Manager's Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen
- *Agile Software Development Ecosystems* by Jim Highsmith
- *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle
- *Scrum and The Enterprise* by Ken Schwaber
- *User Stories Applied for Agile Software Development* by Mike Cohn
- Artículos semanales en [www.scrumalliance.org](http://www.scrumalliance.org)

---

---

---

---

---

---

---