

Ingeniería de Software II

Segundo Cuatrimestre de 2009

Clase 10: Introducción a los métodos ágiles y Scrum

Scrum: ¿Qué es?

¿Qué es un scrum?

*"Un scrum es un **agrupamiento** (formación fija) en Rugby. 8 integrantes de cada equipo, llamados "delanteros", se enfrentan agrupados para tratar de obtener la pelota, que es introducida por uno de los equipos en el centro de la formación."*



Pero... ¿qué tiene que ver con la Ingeniería de Software?

Scrum en el contexto de los métodos ágiles

- ▶ Los métodos ágiles son un subconjunto de los métodos de desarrollo de software iterativos e incrementales
- ▶ Están basados en el “Agile Manifesto”:
- ▶ Estamos *descubriendo* (“uncovering”) mejores formas de desarrollar software, haciéndolo y ayudando a otros a hacerlo. A través de este trabajo hemos llegado a valorar:
 - ▶ Individuos y sus interacciones sobre las herramientas y los procesos
 - ▶ Construir software sobre documentación “completa”
 - ▶ La colaboración con el cliente sobre la negociación de contratos
 - ▶ Responder ante el cambio sobre el seguimiento del plan

Los tres principios básicos

- ▶ Visibilidad: Los aspectos del proceso que pueden afectar el resultado deben ser visibles. El objetivo es evitar las sorpresas
- ▶ Inspección: Las variaciones de los procesos deben ser inspeccionados con la suficiente frecuencia tal que las variaciones inaceptables puedan ser detectadas
- ▶ Adaptación: Si el resultado de la inspección informa que hay aspectos del proceso que exceden los límites aceptables, el inspector debe ajustar el proceso o aquello que está siendo procesado. El ajuste debe hacerse lo más rápido posible para minimizar los desvíos

Más características de los métodos ágiles

- ▶ Empíricas (confían en la observación y la experiencia)
- ▶ “Livianas”
- ▶ “Adaptativas”
- ▶ Emergentes (cuidado con esto...)
- ▶ Rápidas, pero nunca apuradas
- ▶ Exponen pérdidas (wastefulness)
- ▶ Centradas en el cliente
- ▶ Empujan las decisiones hacia los niveles más bajos
- ▶ Favorecen la confianza, la honestidad y el coraje
- ▶ Favorecen la auto-organización
- ▶ Time Boxed

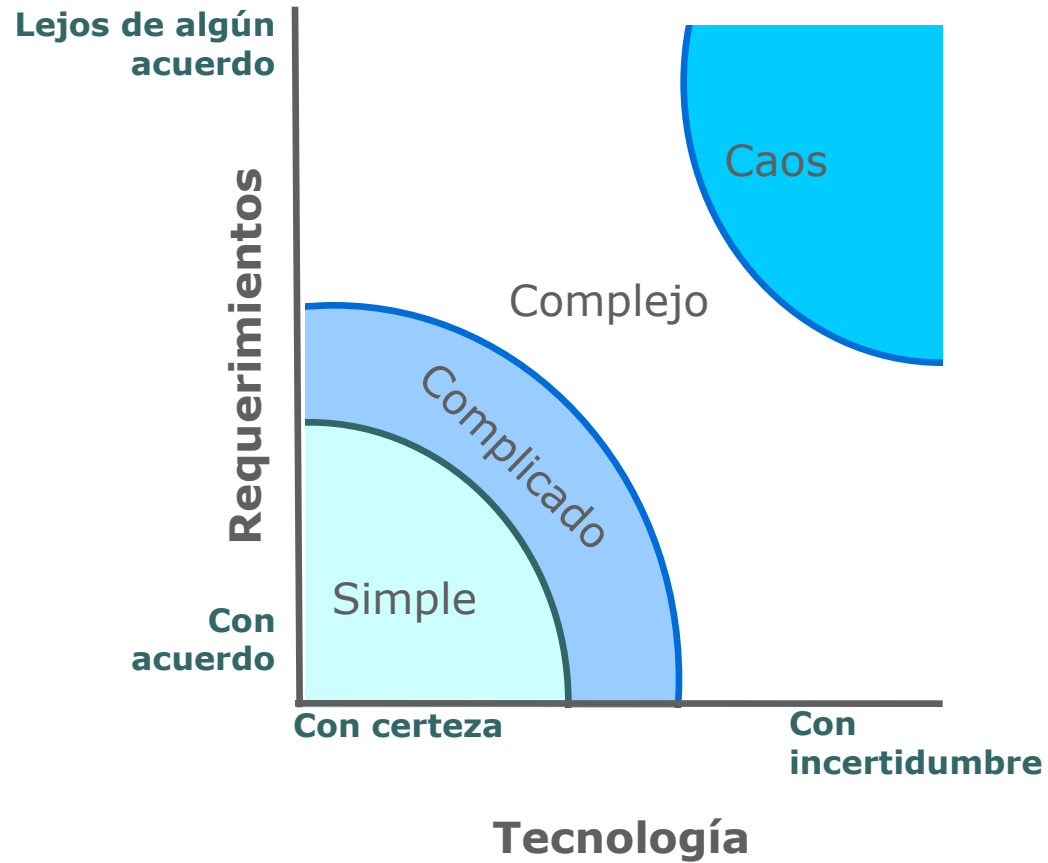
Scrum vs. Metodología Tradicional

- ▶ Scrum asume:
 - ▶ Requerimientos cambiarán durante el proyecto.
 - ▶ Scrum toma el cambio como una forma de entregar al final del desarrollo algo más cercano a la verdadera necesidad del Cliente (evitar el *Tunnel Effect*)
 - ▶ Sin embargo, Scrum tiene mecanismos de control y estructura.

Scrum según sus autores

- ▶ Un “framework” simple que puede ser entendido e implementado en pocos días
- ▶ Una forma de manejar problemas complejos
- ▶ Un entorno para soportar la auto-organización y la creatividad
- ▶ Un esfuerzo colaborativo que involucra a desarrolladores y clientes en un diálogo continuo
- ▶ “Scrum is not a methodology – it is a pathway”
 - ▶ -- Ken Schwaber

Scrum: Tipos de Proyectos

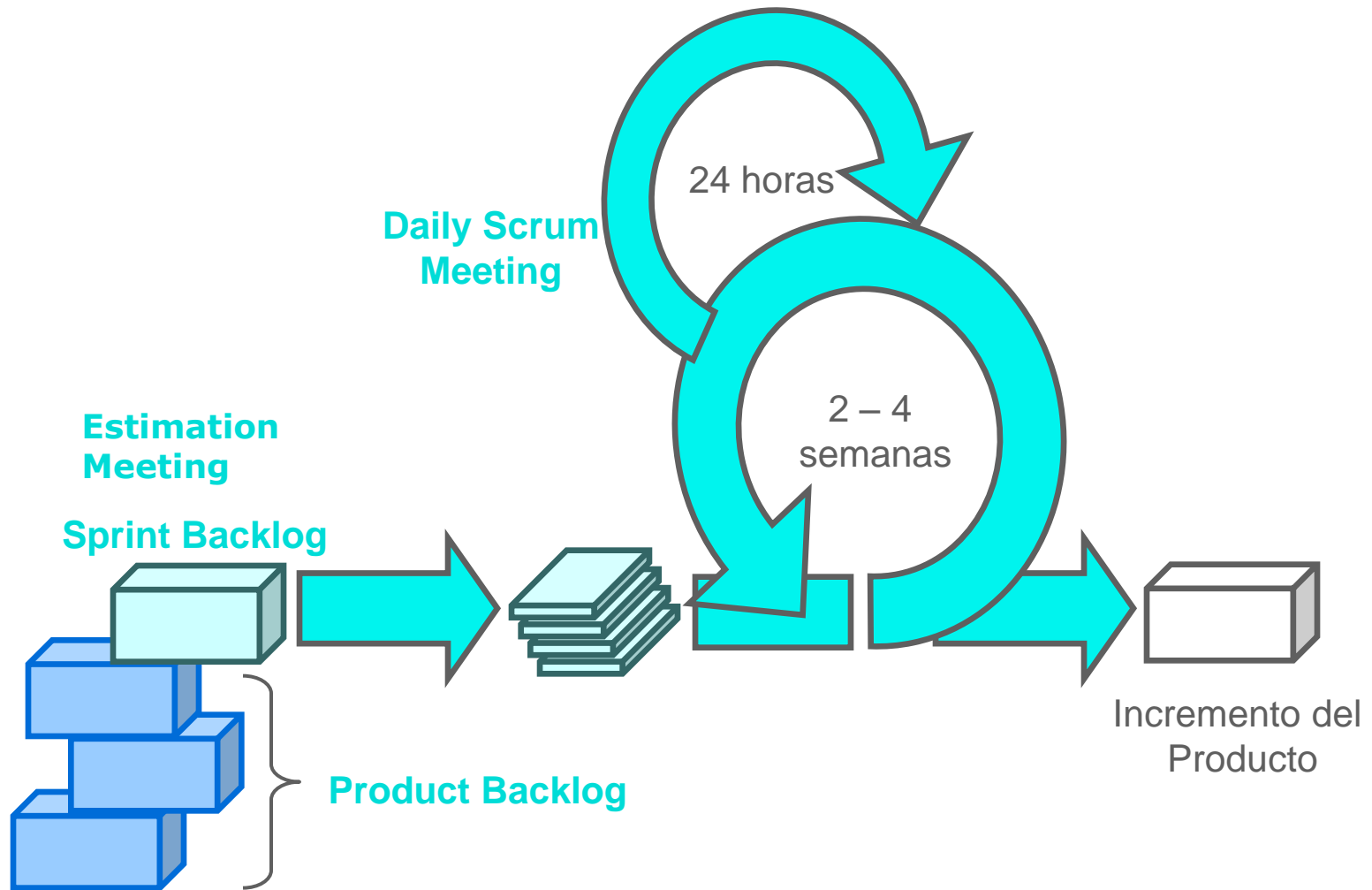


Scrum: Historia

HISTORIA

- **Jeff Sutherland**
 - Scrums en “Easel Corp” en **1993**
 - No solo para proyectos triviales
- **Ken Schwaber**
 - Definición Inicial de Scrum en **OOPSLA 96** con Sutherland
- Actualmente utilizado en diversas empresas de **todo el mundo**, principalmente en Estados Unidos

Esquema Scrum para el Desarrollo



Scrum - Roles

- ScrumMaster
 - Facilitador del equipo. Remueve impedimentos
 - Líder, pero no demandante
 - Responsable de implementar y administrar el proceso Scrum
- Product Owner
 - Representa los intereses del cliente. Es el responsable de poner los requerimientos en términos claros.
 - Es el responsable de maximizar el Retorno de Inversión – Prioriza los requerimientos con mayor valor para el negocio.
 - Resume el input de los usuarios, stakeholders y otras partes interesadas en una vista única de los requerimientos del sistema, con prioridades definidas
 - Puede cambiar funcionalidades y prioridades cada 30 días
 - Acepta o rechaza el trabajo hecho por el equipo
- Team
 - Pilares fundamentales: Comunicación y Cooperación
 - 7 +/- 2 miembros – Máximo 11 personas
 - Compuesto por Desarrolladores, Funcionales y Testers (skills, no roles)
 - Auto-organizado y auto-administrado
 - Estima y define los alcances y límites de cada iteración

Backlog

- ▶ Product Backlog: todo lo que se sabe que quiere incluirse en el producto final
- ▶ Sprint Backlog: subconjunto del Product Backlog seleccionado para hacerse en un Sprint
 - ▶ Se descomponen en tareas
 - ▶ Las tareas se estiman en horas 1-16
 - ▶ Los miembros se postulan a las tareas
 - ▶ La estimación se actualiza día a día.

Scrum: Product Backlog

	Item #	Description	Est	By
Very High				
PRIORIDAD	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
	Add licensing		-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
	5	File formats we support are out of date	160	TG
High	6	Round-trip Analyses	250	MC
	Enforce unique names		-	-
	7	In main application	24	KH
	8	In import	24	AM
	Admin Program		-	-
	9	Delete users	4	JM
	Analysis Manager		-	-
	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab		8	TG
	10	Other	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of numbers to allow easy numbers	16	T&A
	13	Horizontal scrolling	12	T&A
	Population Genetics		-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
	Pedigree Manager		-	-
	20	Validate Derived kindred	4	KH
Medium				
	Explorer		-	-
	Launch tab synchronization (only show queries/analyses for logged in users)		8	T&A
	21	Delete settings (?)	4	T&A

DESCRIPCIÓN

ESTIMACIÓN

ESTIMADO

POR...

Scrum: SPRINT Backlog

Days Left in Sprint		15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
Total Estimated Hours:		554	458	362	270	0
	User's Guide					
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
	Misc. Small Bugs					
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
	Environment					
TBD	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
	Database					
KH	Finish 2.206 database patch	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

RESPONSABLE

DESCRIPCIÓN

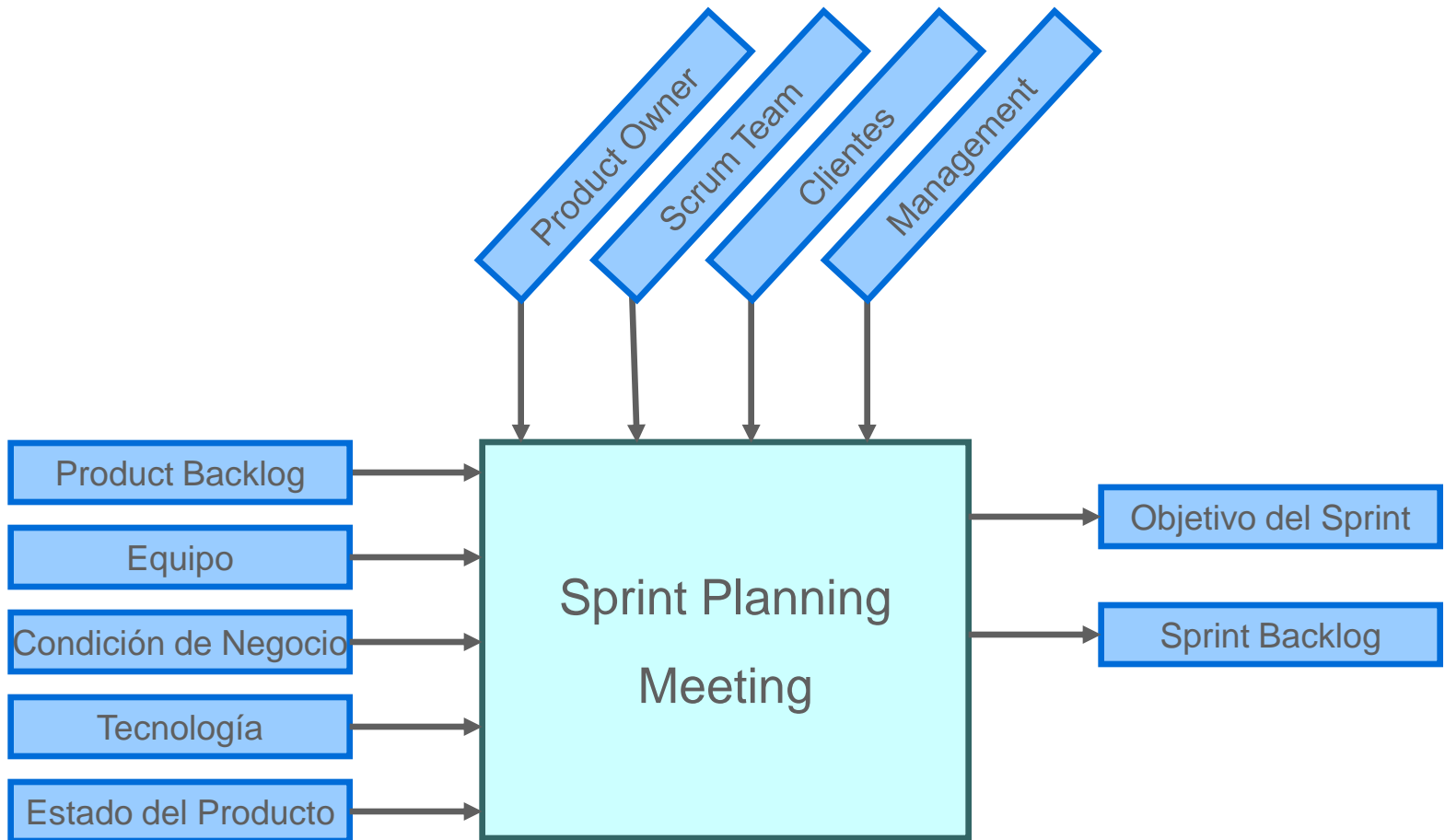
Reuniones de Scrum

- ▶ Se hacen cuatro tipos de reuniones:
 - ▶ Sprint planning
 - ▶ Sprint Review
 - ▶ Daily Standup
 - ▶ Sprint Retrospective

Reuniones: Sprint Planning Meeting

- Contexto y Características
 - Antes del Sprint
 - Product Owner define funcionalidades
 - El equipo decide qué puede realizar en el sprint
- Participantes:
 - Product Owner
 - Product Manager
 - Otros “Stakeholders”
 - Team
 - Management

Reuniones: Sprint Planning Meeting



Estimaciones y Formato del Plan

- ▶ Las estimaciones con Scrum se hacen de manera “relativa” (por ejemplo de 1 a 5), a través de una especie de “piedra, papel y tijera”
- ▶ Se busca consenso
- ▶ El plan generado en la reunión tiene sólo dos niveles: stories y tareas
- ▶ Se usa un “taskboard”, donde las tareas están divididas en “Planificadas, En curso, Terminadas”

Reuniones: Daily Scrum o Daily Standup

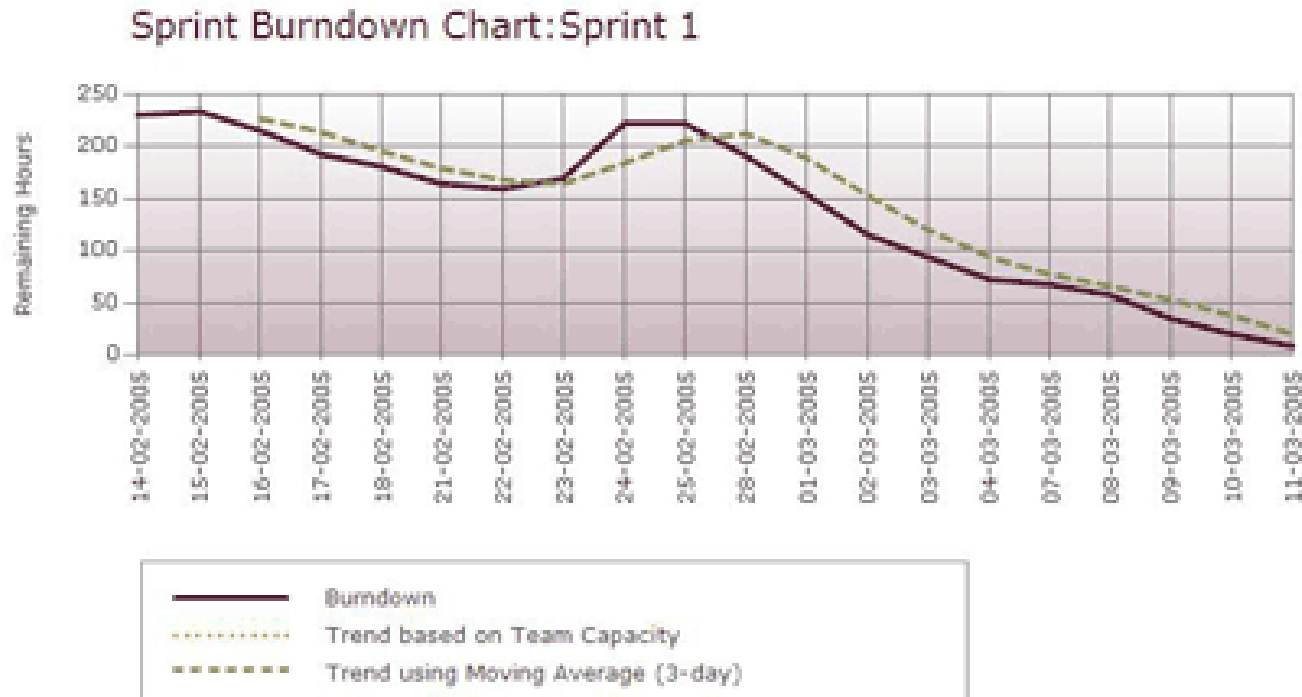
- ▶ Contexto y Características
 - ▶ Diariamente
 - ▶ 15-minutos
 - ▶ De pie (recomendación)
 - ▶ No se pretende resolver problemas
- ▶ Cada integrante del equipo contesta
 - ▶ ¿Qué hice ayer?
 - ▶ ¿Qué voy a hacer hoy?
 - ▶ ¿Qué obstáculos hay en mi camino?
- ▶ Participantes:
 - ▶ Pollitos y cerditos están invitados
 - ▶ Solo los cerditos pueden hablar!!!

Reuniones: Daily Scrum o Daily Standup

- ▶ ¿Por qué diariamente?
 - ▶ Recordemos... “¿Cómo puede un proyecto retrasarse un año?”
 - ▶ “Un día por vez”
 - ▶ Fred Brooks, “The Mythical Man-Month”.
- ▶ Se pueden reemplazar las reuniones de Scrum por reportes vía email?
 - ▶ NO!!!
 - ▶ Todo el equipo debe ver la “big picture” cada día.
 - ▶ Crear la responsabilidad de hacer lo que uno dice que va a hacer.

Reuniones: Daily Scrum o Daily Standup

Actualización Sprint Burndown Chart



Puede ser "Story burndown" o "Task burndown" eje Y

Reuniones: Sprint Review

- ▶ Contexto y Características
 - ▶ Cada 30 días aprox.
 - ▶ 2 hs. de preparación
 - ▶ No mas de 4 horas
 - ▶ Demo In-House
 - ▶ Demo resultado producido
- ▶ Participantes:
 - ▶ Clientes
 - ▶ Management
 - ▶ Product Owner
 - ▶ Otros involucrados

Reuniones: Sprint Retrospective

- Al final de cada sprint
 - Qué funcionó bien
 - Qué puede mejorarse
 - Acciones a tomar para mejorar
-
- Deben hacerse sistemáticamente

SCRUM - Productos

- ▶ Product Backlog
 - ▶ Lista priorizada de requerimientos del producto. Pueden ser requerimientos funcionales, no funcionales, o “issues”
 - ▶ Puede o no tener una estimación asociada
 - ▶ Tiene un “business value” asociado
- ▶ Sprint Backlog Items
 - ▶ Items: Lista de requerimientos extraídos del Product backlog que se implementarán durante el sprint.
- ▶ Sprint Backlog Tasks
 - ▶ Tasks: Lista de tareas. Define el trabajo del equipo para el Sprint. Las tareas en la lista son las que se desprenden de los requerimientos definidos para ese sprint. Son definidas por el equipo, no por el Scrum Master.
- ▶ Impediment List
 - ▶ Es una lista de tareas que usa el Scrum Master para hacer seguimiento de los impedimentos que deben ser resueltos.
 - ▶ Cualquier cosa alrededor de un proyecto de Scrum que afecta su productividad y su calidad.
 - ▶ Es responsabilidad del ScrumMaster hacer el seguimiento de las impedimentos presentes en esta lista

Especificación en Scrum

- Epics – Grupos de funcionalidad
 - Casos de Uso conceptuales de muy alto nivel
 - Son independientes del product / sprint backlog
- Sirven para dar una visión de la funcionalidad de la aplicación
- User Stories:
 - Breves descripciones de funcionalidad descriptas desde la perspectiva del usuario, que tienen valor para el usuario (business value)
 - Interpretan los Epics (Casos de uso bajados a detalle)
 - Tienen su respectivo Epic Asociado
 - Tienen la forma: Requerimiento + Casos de aceptación
 - Template: "As a User, I Want X, So That I can Y" + acceptance Cases
 - No deben requerir un esfuerzo mayor a 5 días
 - Forman parte del Product / Sprint Backlog

Especificación en Scrum – Acceptance Cases

- Las expectativas de los clientes son documentadas como criterios de aceptación.
- El criterio de aceptación debe ser más específico que la descripción de la user story propiamente dicha. Aportan información adicional a la story, la cual ayuda a los desarrolladores a saber cuando una user story está cumplida.
- Los criterios de aceptación deben ser escritos de la forma más simple posible, para que las asunciones y expectativas de los clientes sean tempranamente comunicadas a los integrantes del equipo.
- El criterio debe ser conceptual y no debe incluir temas relacionados con validaciones, restricciones, etc.
- Estructura estándar:
 - El <Usuario que utilizará la funcionalidad>
 - Puede <realizar una acción>
 - Y Obtener <resultado esperado>

Ejemplo de Story:

- User Story
 - Como un: Doctor a cargo de una emergencia
 - Quiero: poder ver un mapa con los lugares de derivación candidatos
 - De tal forma que: pueda saber a dónde llevar los afectados por la emergencia rápidamente, minimizando el tiempo de traslado de los accidentados
- Casos de Aceptación
 - Existe una opción de “ver mapa”
 - El mapa muestra la posición actual
 - El mapa muestra los centros de derivación con una indicación de su tipo:
 - Hospitales municipales: rojo
 - Clínicas: azul
 - El mapa muestra el camino teniendo en cuenta la dirección de las calles
 - Etc!

Scrums de Scrum

- ▶ SCRUM diario de SCRUMs
- ▶ Reuniones semanales:
 - ▶ ¿Qué han hecho los equipos la semana pasada?
 - ▶ ¿Qué harán esta semana?
 - ▶ ¿Cuáles son sus impedimentos?

¿Dónde dicen los de Scrum que no hay que usar Scrum?

- ▶ SCRUM NO es apropiado para:
 - ▶ Equipos muy grandes
 - ▶ Estructuras complejas de equipos
 - ▶ Equipos distribuidos geográficamente (depende)
 - ▶ Aplicaciones críticas (depende)
 - ▶ Gente con poca experiencia

Más información

- ▶ <http://www.mountangoatsoftware.com/scrum>
- ▶ <http://www.controlchaos.com>
- ▶ scrumdevelopment@yahoogroups.com
- ▶ Agile Software Development with Scrum
 - ▶ Ken Schwaber and Mike Beedle
- ▶ Agile Project Management with Scrum
 - ▶ Ken Schwaber and Mike Beedle
 - ▶ <http://www.agilealliance.com>
 - ▶ <http://www.controlchaos.com/certifiedscrum>