
Metodologías ágiles de desarrollo de software: una introducción a Scrum

CIT, San José, Costa Rica

25 Enero 2007

Alan Cymment
acyment@artinsoft.com

¿Quién soy?

- Lic. en Ciencias de la Computación (Universidad de Buenos Aires – Argentina)
- Experiencia como desarrollador y consultor metodológico tradicional (i.e. RUP)
- Organizador de primer curso abierto de certificación Scrum en América Latina
- ScrumMaster en el desarrollo del nuevo producto de ArtinSoft



Introducción

(i.e. para qué Scrum)



Creatividad

- Trabajadores del conocimiento → más en común con el arte que con la ingeniería¹
- Dejar el paradigma mecanicista → podemos explorar soluciones *creativas*



¹. e.g. "Artful Making" - Rob Austin & Lee Devin (2004)



Equipos ágiles

- Nuevos valores
 - Comunicación
 - Colaboración
 - Trabajo en equipo
- El equipo debe sentirse confiado, potenciado (*empowered*) y seguro
 - Debe poder tomar riesgos – y fallar
- Equipos → Individuos
 - Aprender a estar “en el límite”
 - Requiere práctica y un ambiente que lo apoye



Historia y Contexto

(i.e. cómo llegamos hasta acá)



Enfoques Definidos/Por fase

- El paradigma de facto desde los '70
- Modelo de producción de Taylor¹
 - Sistematizado
 - Autoritario y jerárquico



¹. "The Principles of Scientific Management" - Frederick W. Taylor (1911)

Enfoques Definidos/Por fase

- Mitigación de riesgos *upfront*
- La documentación transmite la información esencial
 - Traspaso entre grupos funcionales → Hitos del proyecto
- *Command & control*
- Resistencia al cambio

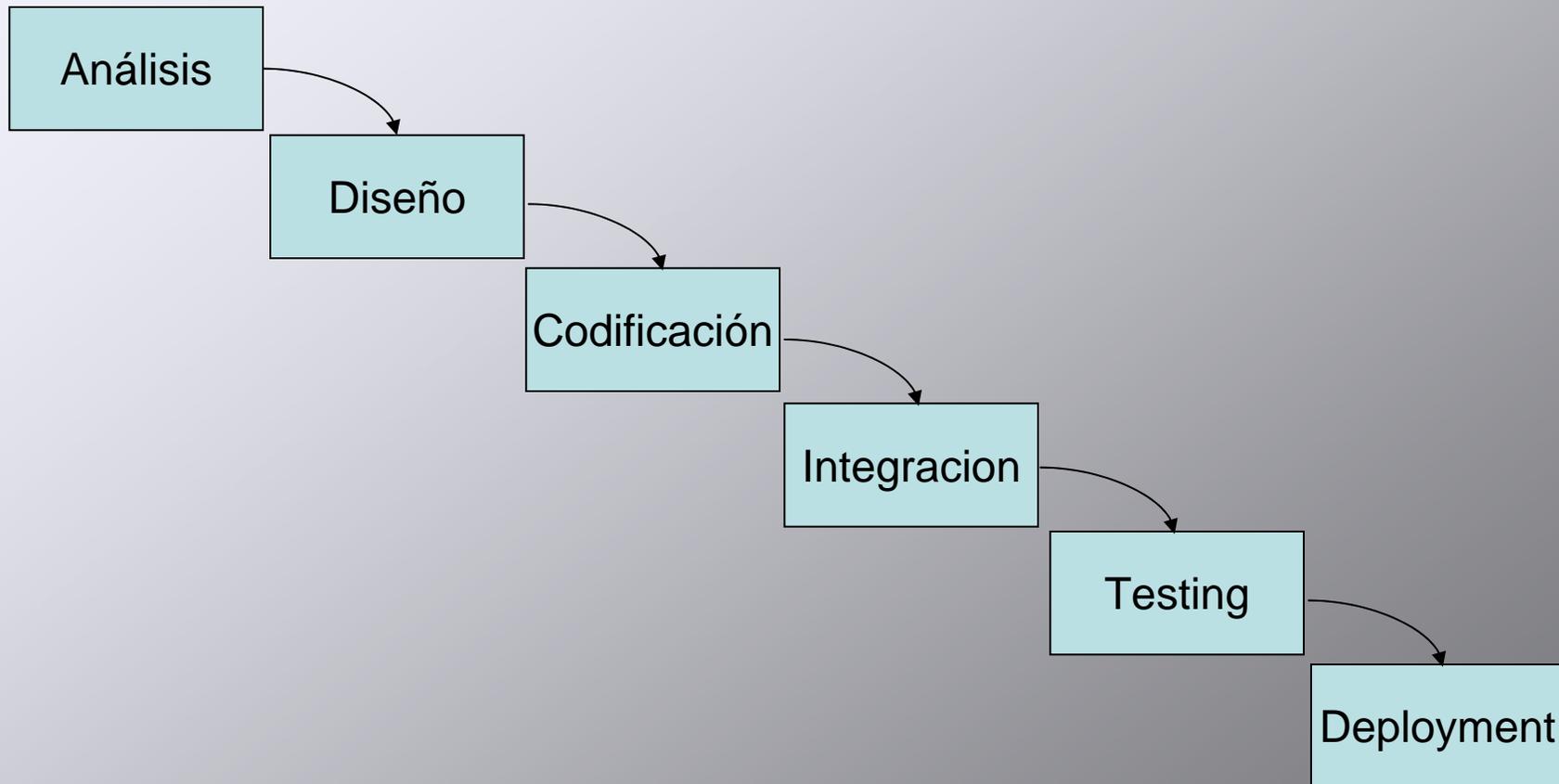


La cascada

- El más conocido – el más usado
- Basado en un paper escrito por Royce (1970)
- Royce describió el uso de la cascada como el modelo *ideal* de desarrollo
- Pero luego dijo que ese ideal es casi imposible de cumplir... en el mismo paper!



El enfoque por fase (i.e. La cascada)



El manifiesto ágil (2001)

“Lo mejor que se pueda” – ¡Un enfoque realista!

Estamos descubriendo mejores formas de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de esta experiencia hemos aprendido a valorar:

- **Los individuos y las interacciones** por sobre los procesos y las herramientas
- **El software ejecutable** por sobre la documentación completa
- **La colaboración con el cliente** por sobre la negociación de contratos
- **Responder al cambio** por sobre seguir un plan

Esto es, aunque los elementos de la derecha tienen valor, valoramos por encima de ellos los que están a la izquierda.

<http://agilemanifesto.org>



Características de un proceso ágil

- Empírico (basado en la observación y la experimentación)
- Simple
- Adaptable
- Emergente
- Rápido – pero nunca apurado
- Expone las tareas inútiles
- Centrado en el cliente
- Lleva las decisiones a los niveles operativos

Introducción a Scrum

(¡Por fin!)

¿Qué es Scrum?

- Un framework sencillo que puede ser entendido e implementado en unos días
- Una técnica para atacar problemas complejos
- Un ambiente que fomenta la auto-organización y la creatividad
- Un esfuerzo colaborativo que incluye tanto a desarrolladores como a clientes
- “*Scrum no es una metodología – es un camino*”
-- Ken Schwaber (Boulder, Co, Nov. 2005)

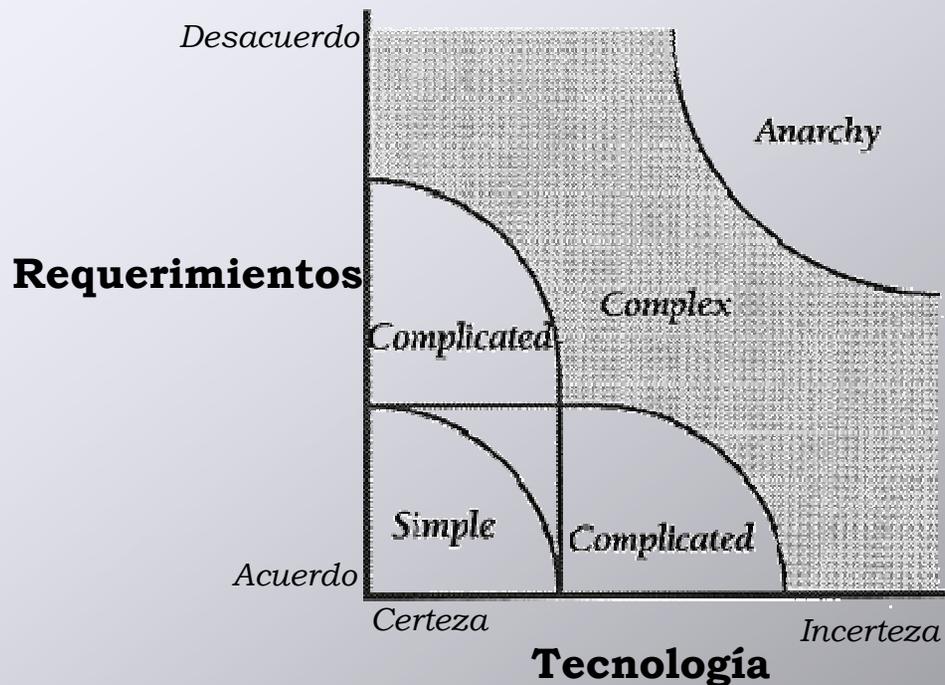
¿De dónde proviene?

- “The New New Product Development Game” - *Harvard Business Review*, 1986.
 - “The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”

¿Quién lo utiliza?

- Microsoft
- Yahoo!
- Hewlett-Packard
- Siemens
- Boeing
- Sun
- IBM
- etc...

¿Cuándo es útil?



- Scrum funciona mejor para el *espacio complejo*
- El desarrollo de nuevos productos y el trabajo de conocimiento (*knowledge work*) suelen caer en este espacio



Ejercicio

- Spaghetti
 - ¿Qué pasó?
 - ¿Cómo se sintieron?
 - ¿Qué conclusiones pueden sacar?



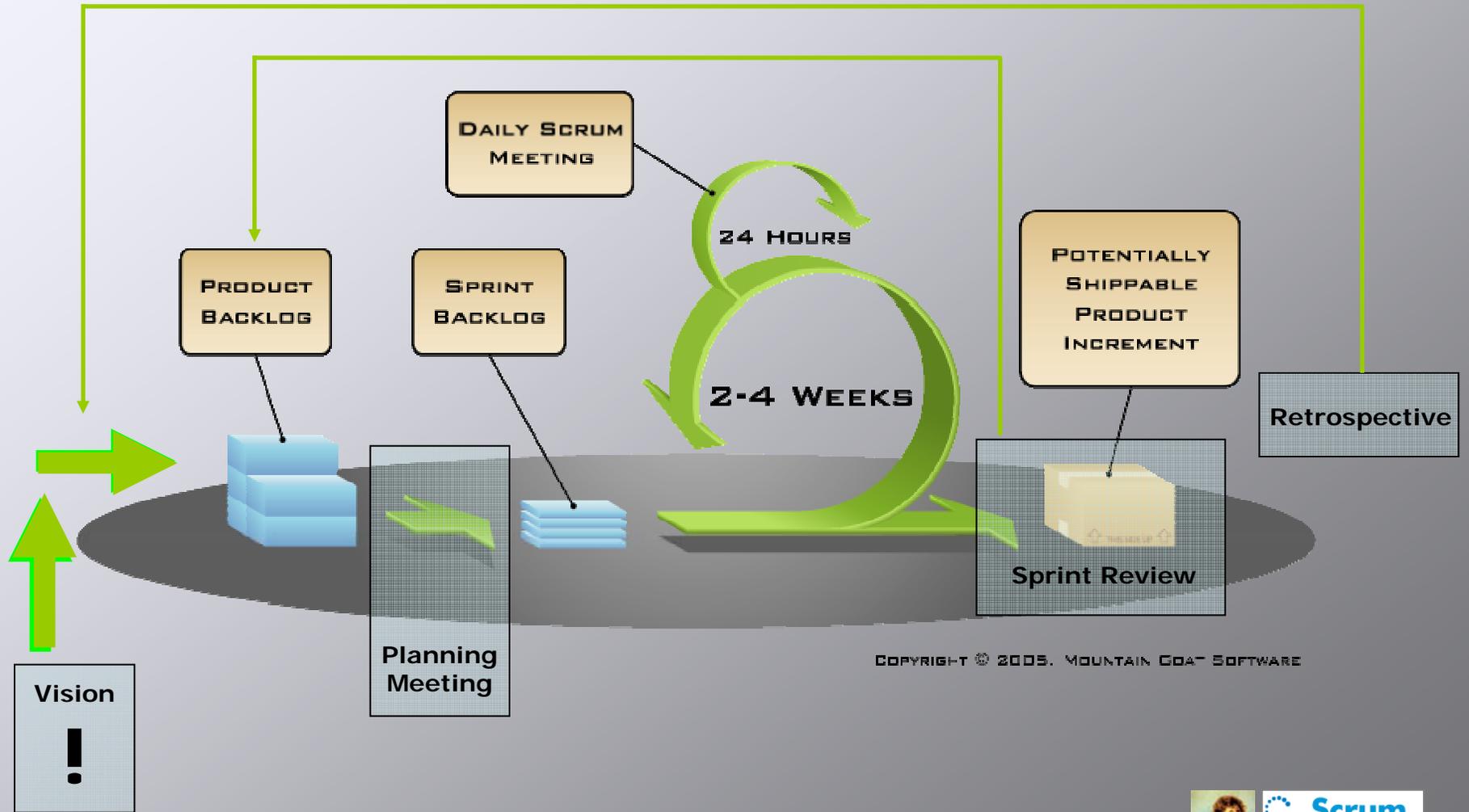
Los principios de Scrum

- Proceso empírico
 - La planificación detallada-anticipada y los procesos definidos se reemplazan por ciclos de inspección + adaptación

¡CRUCIAL!

- Auto-organización
 - El equipo se auto-administra y organiza alrededor de ciertas metas, a partir de límites bien claros

EI Framework



Roles

- **Product Owner**
 - Administra la Visión y los Requerimientos
- **ScrumMaster**
 - Administra el proceso
- **Equipo**
 - ¡Se administran a si mismos!

Product Owner

- Define los features del producto
- Decide la fecha de release
- Es responsable de la rentabilidad del producto
- Prioriza los features según su valor de mercado
- Ajusta los features y su prioridad en cada iteración
- Acepta o rechaza el trabajo realizado por el Equipo

ScrumMaster

- Responsable de poner en práctica los valores y prácticas de Scrum
- Elimina impedimentos
- Hace lo necesario para que el equipo trabaje con máxima productividad
- Promueve el trabajo en equipo entre todos los roles y las funciones
- Protege al equipo de interferencias externas

Equipo

- Usualmente 5-9 personas
- Multidisciplinario
 - Testers, programadores, diseñadores de GUI, etc
- Los miembros deben trabajar tiempo completo
 - Algunas excepciones (ej: DBA)
- Se auto-organiza
 - Idealmente no hay títulos
- La composición sólo puede variar entre sprints

Reuniones

- **Planning Meeting**
 - Ocurre al comienzo de cada sprint
 - El Equipo y el Product Owner negocian el sprint
- **Daily Scrum**
 - Cada día
 - 15 minutos como máximo
 - Los integrantes del equipo comunican unos a otros su progreso, intención a futuro e impedimentos
 - No es para resolver problemas
- **Review**
 - Inspección y adaptación del *producto*
 - El Equipo muestra el software ejecutable construido durante el sprint
 - Generalmente en forma de demo
- **Retrospectiva**
 - Inspección y adaptación del *proceso*
 - El Equipo analiza qué anduvo bien y qué puede mejorarse

Artefactos

- **Product Backlog**
 - Lista priorizada de todos los requerimientos conocidos sobre un producto
 - Cada requerimiento se conoce como Product Backlog Item (PBI)
 - Representa el *qué* del proyecto
 - En constante movimiento
 - Re-priorización a cargo del Product Owner según necesidades del negocio
- **Committed Backlog (aka Sprint Backlog)**
 - El subconjunto del Product Backlog al que se ha comprometido el Equipo (durante un sprint)
 - Esta parte del backlog *no se toca*
 - Los PBIs prometidos se subdividen en tareas: el *cómo*
- **Lista de Impedimentos**
 - Lista priorizada de todas las dificultades e impedimentos encontrados
 - Dividida en dos secciones: *Organizacional* y *Equipo*
- **Incremento del Producto**
 - Al final de un sprint se entrega software “terminado”
 - Aceptado (o rechazado) por el Product Owner y los Stakeholders

¿Preguntas?

Additional Material...

The 4 Qualities of Artful Making

- **Release**
 - A method of control that accepts wide variation within known parameters. Release contrasts sharply with *restraint*.
- **Collaboration**
 - The letting go of vanity, fear, inhibition and preconceptions in order to treat the contributions of team mates as material to create with, not as positions to argue against.
- **Ensemble**
 - The emergent quality borne out of collaboration, where individuals generate a whole greater than the sum of its parts.
- **Play**
 - The real-time interactions within a team, and between the team and their customers. Play is always supportive, never blocking.

Adapted from “Artful Making” by Rob Austin and Lee Devin (2004)

X and Y Management

- **Theory X**

Theory X Management assumes that the average person:

- Dislikes work and attempts to avoid it.
- Has no ambition and wants no responsibility.
- Resists change.

- **Theory Y**

– **Theory Y Management** assumes that the average person:

- Enjoys work and wants to do a good job
- Is creative and seeks responsibility.
- Embraces change, and uses it as a motivating factor

It may be noticed that each style of management is, to some extent, self-fulfilling.

Ref: "The Human Side of Enterprise" by D. McGregor – <http://www.netmba.com/mgmt/ob/motivation/mcgregor/> »

The Edge of Chaos

1. Equilibrium is a precursor to death. When a living system is in a state of equilibrium, it is less responsive to changes occurring around it. This places it at maximum risk.
2. In the face of threat, or when galvanized by a compelling opportunity, living things move toward the edge of chaos. This condition evokes higher levels of mutation and experimentation, and fresh new solutions are more likely to be found.
3. When this excitation takes place, the components of living systems self-organize and new forms and repertoires emerge from the turmoil.
4. Living systems can not be directed along a linear path. Unforeseen consequences are inevitable. The challenge is to disturb them in a manner that approximates the desired outcome.

Ref: "Surfing the Edge of Chaos" by Richard Pascale, Mark Millemann and Linda Gioja

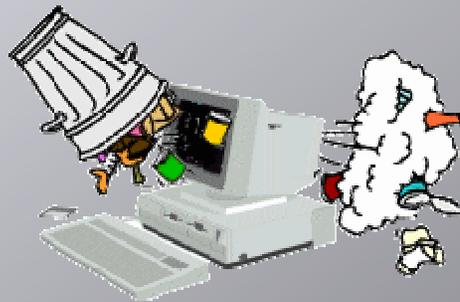
Diseño Emergente

- Entregar “rebanadas” funcionalmente completas a intervalos regulares
- El Diseño y la Arquitectura son tan importantes que deben ocurrir *todo el tiempo*



El Product Backlog

- Scrum comienza con una Visión del Producto (i.e. Product Backlog inicial)
- Backlog pobremente definido → Producto que no cumple con expectativas del cliente



“Garbage in – Garbage out”