

Universidad Tecnológica Nacional
Facultad Regional Buenos Aires
Diseño de Sistemas



Introducción a las Metodologías Ágiles

Nicolás Brailovsky
March 7, 2009

- 1 ¿Qué es una metodología?
- 2 Metodologías Ágiles
- 3 Scrum
- 4 XP
 - Test Driven Development
 - Integración Continua



¿Qué es una metodología ...

- en general?
- aplicada al desarrollo de software?
- ágil?





¿Para qué?

En las metodologías "pesadas" el desarrollo de software se ve

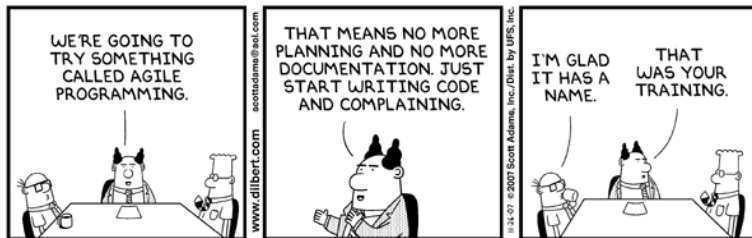
- como un proceso de manufactura
- predecible y estable
- estructurado en forma rígida

Resultados

- Productos de poca calidad
- Proyectos inestables
- Problemas de motivación en el equipo



¿Qué es una metodología ágil?



© Scott Adams, Inc./Dist. by UFS, Inc.

// TODO: Pedirle a Scott que nos deje usar esta imagen



Manifiesto Ágil

- Individuos e interacciones > procesos y herramientas
- Software en funcionamiento > documentación comprensiva
- Colaboración del cliente > la negociación de un contrato
- Responder al cambio > seguir un plan

agilemanifesto.org



Manifiesto Ágil

El "software en funcionamiento" es solo parte de los entregables

- ¿Qué otros entregables existen?
- Un manual de usuario, o la documentación de una API,
 - ¿son entregables?
 - ¿son software en funcionamiento?

Se puede clasificar la documentación como histórica y para soporte del proceso

- ¿A cuál se refiere el manifiesto?
- ¿Sin documentación comprehensiva == No documentar?



Distintos enfoques

- SCRUM
- XP, Extreme Programming
- Lean Software Development
- Crystal

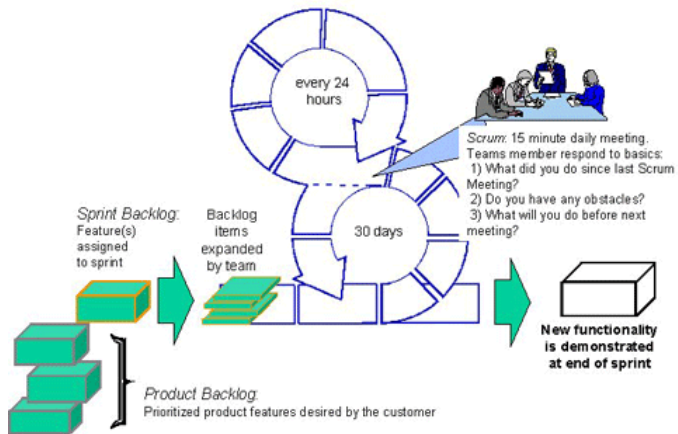


Introducción

- Metodología iterativa y adaptativa al proyecto
- Centrado en la gestión del proyecto
- Gestiona requerimientos y tiempos de entrega



Guía del proyecto



Sprint

Un proceso guiado por Scrum se basa en una serie de sprints

- Ciclo corto (2 o 3 semanas)
- Involucra a stakeholders, el equipo de desarrollo y un Scrum master
- Los stakeholders, con el grupo, crean y priorizan una lista de requerimientos
- La lista priorizada de requerimientos para el proyecto se llama product backlog
- La misma lista para el sprint es el sprint backlog
- El equipo se compromete a cumplir una determinada cantidad de RQs por sprint



Sprint

- La cantidad de historias por sprint determina la velocidad del equipo
- La velocidad del equipo sirve para estimar el siguiente sprint backlog
- La velocidad se basa en datos históricos: no es una estimación
- La velocidad no se mide, necesariamente, en unidades de tiempo
- Todos cometemos errores al estimar pero, estadísticamente, el error es consistente
- La estimación se puede realizar con un juego de estimación
- Al final del sprint debe existir un entregable con valor para los stakeholders



Introducción

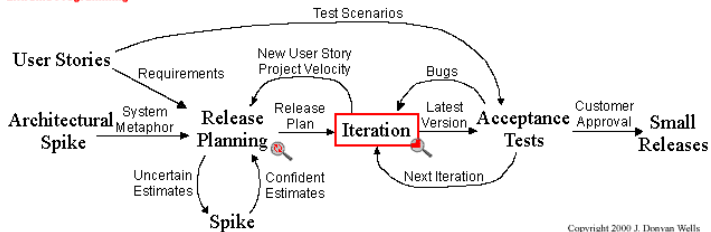
- Mejores prácticas
- Orientado al día a día
- Énfasis en el testing
- Nombre marketinero (?)



Guía de proyecto



Extreme Programming Project



Principios

XP basa sus prácticas en

- Comunicación, Feedback, Simplicidad, Valor, Respeto

Se adapta a proyectos con

- requerimientos inestables
- alto riesgo
- equipos con poca experiencia
- equipos chicos

Una de las innovaciones de XP es integrar al testing como parte del proceso y no como tarea de soporte



Prácticas

- Las historias de usuario sirven para crear test de aceptación y para estimar
- Entregables en ciclos cortos en incrementos pequeños
- Propiedad colectiva del código / Rotación interna del equipo
- Pair programming
- El cliente siempre está disponible
- Testing antes de codificar (TDD/BDD)
- Integración continua
- Refactor and consolidate
- KISS

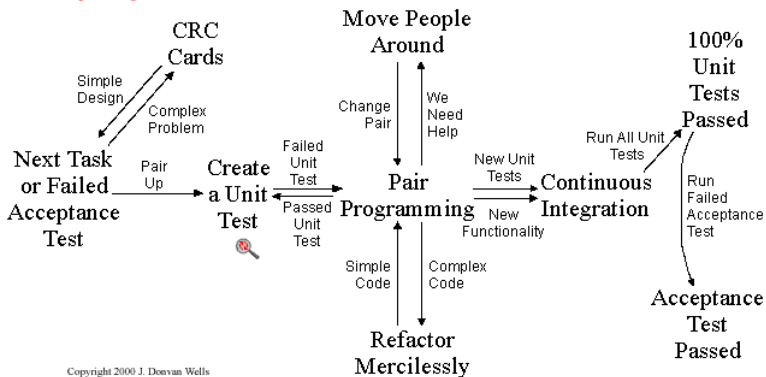


Prácticas



Collective Code Ownership

Zoom Out



Copyright 2000 J. Donovan Wells



Outline

- 1 ¿Qué es una metodología?
- 2 Metodologías Ágiles
- 3 Scrum
- 4 XP**
 - Test Driven Development
 - Integración Continua



Introducción

TDD: Una de las prácticas mas innovadoras de XP ¿Para qué escribir tests?

- Reutilización de código ⇒ bueno
- Tocar en un lado y que explote todo ⇒ malo

Los tests son una forma de asegurar que la interfaz del módulo no cambia. Además:

- Es una forma de documentar las historias de usuario
- No solo documenta, sirve como ejemplo!
- Incrementa la confianza en la fiabilidad del sistema
- Obliga al programador a pensar antes de codificar
- El trabajo termina una vez que todos los tests pasan



Metodología de trabajo

La metodología de trabajo en TDD se conoce como "Red - Green - Refactor"

- Escribir tests como especificación de un comportamiento
- Correr los tests; ver como fallan
- Implementar código
- Cuando los tests pasan (están en verde) el desarrollo está completo
- Ahora se puede refactorizar el código, sin miedo a romper nada



Behaviour Driven Development

Utilizando TDD

- se puede detectar una falla con el mismo cambio que la provoca
- se pretende definir comportamiento, no solo verificar la estructura del código

Esto se conoce como "BDD", Behaviour Driven Development: definir el comportamiento del sistema a partir de un test.

- Con TDD/BDD los tests deben correr en un tiempo razonable, es decir, se deben utilizar tests unitarios (no de integración!)



Outline

- 1 ¿Qué es una metodología?
- 2 Metodologías Ágiles
- 3 Scrum
- 4 XP**
 - Test Driven Development
 - Integración Continua**



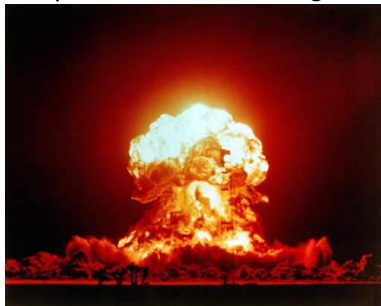
Introducción

En sistemas grandes el esfuerzo se divide en componentes, equipos, etc. ¿Qué pasa al intentar integrar todas las partes?



Introducción

En sistemas grandes el esfuerzo se divide en componentes, equipos, etc. ¿Qué pasa al intentar integrar todas las partes?



Prácticas

- La última versión del repositorio SIEMPRE compila ¹
- Los test de integración no deben tardar mas de 5 minutos
- Permite detectar errores de integración minutos (5) después del commit que lo causó

¹Práctica de XP: Quien rompa el build deberá traer facturas



Lectura Adicional I



Pressman, Roger

Ingeniería del Software: Un enfoque Práctico.
Mc Graw Hill.



Duvall, Paul et al.

Continuous Integration.
Addison Wesley, 2007



Agile Manifesto

<http://agilemanifesto.org>



Extreme Programming: A Gentle Introduction

<http://www.extremeprogramming.org/>



Martin Fowler

The New Methodology

<http://www.martinfowler.com/articles/newMethodology.html>



Lectura Adicional II



Martin Fowler

Continuous Integration

<http://www.martinfowler.com/articles/continuousIntegration.html>



Martin Fowler

Is Design Dead?

<http://www.martinfowler.com/articles/designDead.html>



Kent Beck

Extreme Programming

<http://c2.com/cgi/wiki?ExtremeProgramming>



James Shore

The Art of Agile

<http://jamesshore.com/>

