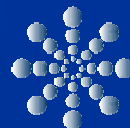


# Desenvolvimento de Software no “Silicon Valley North” (Ottawa/ON/Canada)

**Marcio Marchini**  
[marcio@bedarra.com](mailto:marcio@bedarra.com)

(palestra na UFSC, 2007/12)



**Bedarra Research Labs**

© 2007 Bedarra Research Labs – [www.bedarra.com](http://www.bedarra.com)

# Platéia Alvo



## Principal:

- Alunos UFSC disciplina INE5358

## Secundário:

- Curiosos, profissionais de informática, etc

# Silicon Valley North

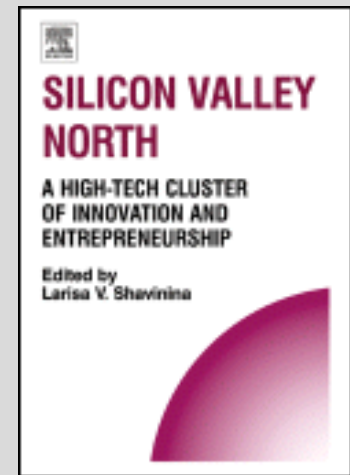


## Local:

- Ottawa, Ontario, Canadá
- Pop. 1,130,761
- [http://en.wikipedia.org/wiki/National\\_Capital\\_Region\\_\(Canada\)](http://en.wikipedia.org/wiki/National_Capital_Region_(Canada))

## High-Tech:

- Nortel (afundou bastante com o “Internet Bubble Burst”)
- Cognos (sendo comprada pela IBM - \$5 bilhões)
- MITEL (Terry Mathews e Mike Cowpland)
- Corel (Mike Cowpland)
- Newbridge (Terry Mathews , comprada pela Alcatel)
- ObjectTime (comprada pela Rational; influenciou UML)
- Rational (comprada pela IBM)
- QNX (comprada recentemente)
- Juniper Networks
- March Networks (Terry Mathews)
- JDS Uniphase
- RIM
- Bell, Rogers etc



# Background – Realidade de 1987



## Universidade (UFSC):

- Faz funcionar e entrega pro professor, tem festa na Lagoa!!!
- Hack & release
- Controle de Versão ?!?! Hã ?!?!
- Tem um maluco ensinando um tal de Smalltalk e uma tal de Orientação a Objetos no EDUGRAF... um tal de Melga

## Empresas:

- Folha de Pagamento, Sistema de Videolocadora, etc (dBase, Clipper, COBOL)
- Hack & Release
- Controle de Versão ?!?! Hã ?!?!

(Ainda realidade atual em várias empresas de software no Brasil)





## Waterfall:

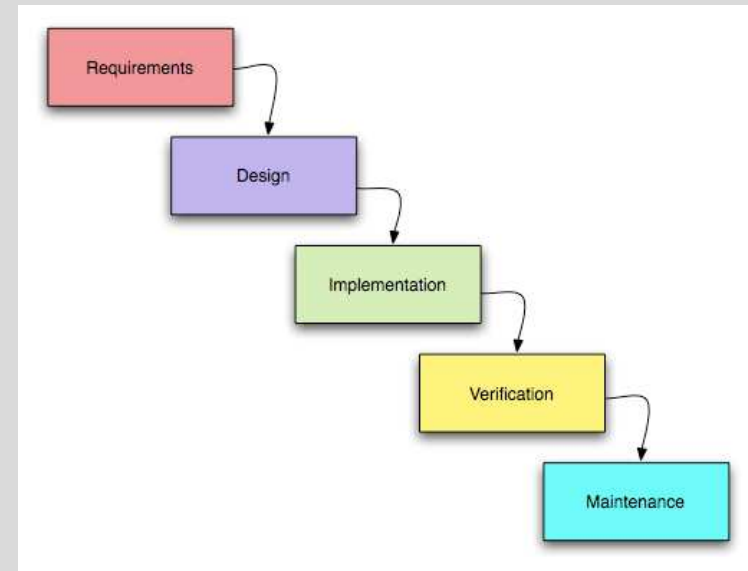
- Requirements
- Design
- Implementation
- Verification
- Maintenance

## Prototipação:

- Prototipação Rápida, Prototipação em Espiral, etc
- Hack & release
- Controle de Versão ?!?! Hã ?!?!

## Smalltalk:

- Uma tal de OTI no Canadá faz um Team Environment p/ Smalltalk (ENVY/Developer)
- Essa tal OTI roda Embedded Smalltalk em osciloscópios Tektronix!!!!??? Wow!!!
- OTI, é contigo mesmo!!! ...



# OTI – Brief History



From <http://www-03.ibm.com/software/ca/en/ottawalab/roots.html>

## Object Technology International (OTI): 1988-2002

A world leader in object-oriented technology founded in 1988, OTI is the company behind big-name products like VisualAge for Java, VisualAge Smalltalk, VisualAge Micro Edition, WebSphere® Studio Device Developer, WebSphere Micro Environment, WebSphere Custom Environment, WebSphere Studio Workbench, and Eclipse.

OTI is known for its lead role in the design and development of IBM's IDEs for Java, Smalltalk and embedded systems. More recently, OTI led in developing Eclipse, the open source, next-generation application development tools platform, and WebSphere Studio Workbench, the IBM supported version of Eclipse technology.

OTI was acquired by IBM in 1996, and operated as a wholly owned subsidiary for seven years. In 2003, OTI transitioned to become a full part of IBM with the formation of the new IBM Ottawa Software Lab.

Founder: Dave Thomas (see [http://en.wikipedia.org/wiki/David\\_A.\\_Thomas\\_%28software\\_developer%29](http://en.wikipedia.org/wiki/David_A._Thomas_%28software_developer%29) )



ENVY/Swapper (“serialization” do OTI Smalltalk/IBM Smalltalk)

Nossa “Mission Impossible”:

- Remover as limitações da implementação C (64KB max stack/object recursion)
- Fazer mais flexível que a implementação C (Class-based replacers, instance-based replacers, etc)
- Fazer em Smalltalk
- Mais Rápido do que em C

Solução:

- A velha regra do 80-20 (a.k.a. Pareto Principle – veja Wikipedia)





## O Mundo OTI:

- Vários Regression Tests de ENVY/Swapper. Hmm, na universidade não tinha disso.
- Swapper: Legacy que precisa evoluir. Hmm, na universidade não tinha disso.
- Time de Release Engineering, com builds. Hmm, na universidade não tinha disso.
- Milhares de clientes grandes pelo mundo, inclusive IBM, mainframes, etc. Tem que rodar bem e sem erro. Hmm, na universidade não tinha disso.
- Ninguém seguindo Waterfall ou usando OMT (precursor do UML). Mas também nada de prototipação. Hmm... O que é que esses caras usam??
- Componentes. APIs bem definidas nas “boundaries”. Times pequenos com Team Leaders. Sem Managers.
- “Eat Our Own Dog Food” (or “Drink Our Own Champagne”): Bug/Ticket System em Smalltalk. IDE em Smalltalk. Team tool em Smalltalk (ENVY/Developer). Mailer em Smalltalk!!!! Etc.
- CEO: Dave Thomas. Professor / Carleton University. ACM Distinguished Engineer.





# OTI JIT – O Segredo



## Just-In-Time Software Development

- Inspirado em Lean/Toyota
- Secreto/interno. “Competitive Advantage”

Nome Moderno: Agile

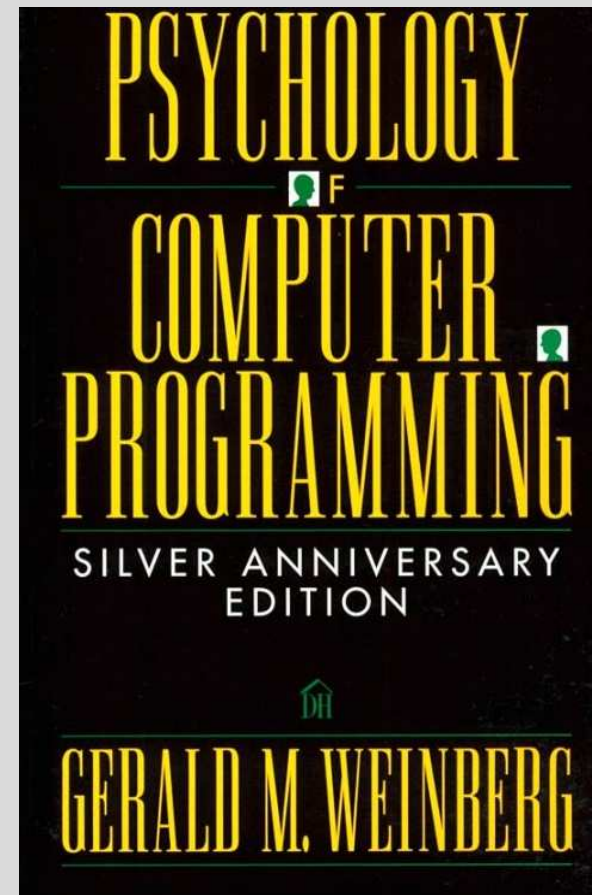
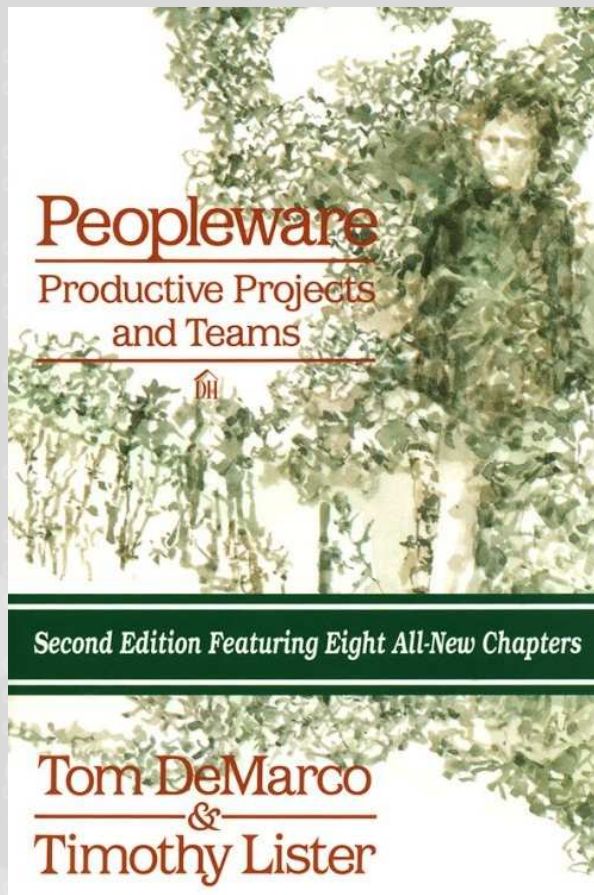


# OTI JIT – O Efeito Colateral



Smart People Attract Smart People!!! (experimente um Google dessa frase)

- Peopleware!!!!



# OTI JIT – Mais Efeito Colateral



## Assimilação Borg Reversa

- IBM Compra OTI e assimila o know-how. (ex. Erich Gamma, OTI Zurich)
- VMs Smalltalk, VMs Java. (John Duimovich & VM team)
- IDEs. (Greg Adams & teams)
- Método JIT de Software. (Dave Thomas & Brian Barry)
- Spin off: Eclipse Foundation. (Mike Millinkovitch: former OTI'er)
- IBM se tornando Ágil ?!?! (início: Websphere)

# 2007 - IBM Jazz – Sounds Familiar?



## IBM Jazz

- “People, not organizations, build great software”. <http://jazz.net/pub/index.jsp>
- “Jazz is a joint project between IBM Rational and IBM Research to build a scalable, extensible team collaboration platform for seamlessly integrating tasks across the software lifecycle.”
- Sounds Familiar???



# 2007 - IBM Project Zero – Sounds Familiar?



## Project Zero

- “Zero complexity. Zero overhead. Zero obstacles”. <http://www.projectzero.org/wiki/bin/view/>
- “Project Zero introduces a simple environment for creating, assembling and executing applications based on popular Web technologies. The Project Zero environment includes a scripting runtime for Groovy and PHP with application programming interfaces optimized for producing REST-style services, integration mash-ups and rich Web interfaces. Project Zero is an incubator project started within IBM that is focused on the agile development of the next generation of dynamic Web applications.”
- Sounds Familiar???

## Trivia

- Codename at OTI: UVM (Universal Virtual Machine)
- VisualAge for Java: Smalltalk VM rodando Smalltalk (IDE) e Java (runtime/app).

# Software em Empresas Grandes: 1994...2003



(não é todo mundo que nem a OTI ???)

<confidencial> (Telco)

- PBX: Pascal “envenenado” p/ “baixaria”, compilador in-house.
- Código massaroca
- UML? HAHAHAHAHA!!!! Se quiser, gera a partir do código!!!
- Unit Tests? HAHAHAHAHA!
- Mas... Continuous (Version Control, Bug Report, etc)

<confidencial> (Case Vendor)

- Propria ferramenta: Todo analisado/projetado up-front em UML, certo? HAHAHAHAHA!!!

<confidencial> (Grande empresa de hardware etc)

- UML: Usado entre releases, pros times “descansarem”. Brincar de desenhar diagramas. O gerente quer.
- Na hora do aperto: Hack and Release
- Mas... JIRA, Perforce. Continuous Builds com nossa ajuda.

<confidencial>

- Business Intelligence: 6M LOC em C, C++, APL, VB, Java, etc.
- UML? HAHAHAHAHA!!!!
- Unit Tests? HAHAHAHAHA!
- Todos os testes feitos “through the UI” com ferramentas tipo Mercury. Ouch!!!!
- Mas... Bug Tracking (in-house), Build System (in-house), Perforce, Smart Bear (code review)

# Empresas Grandes 2001...



## ISO 900x, 6-Sigma, CMM

- Teoria: Implante um Processo Rígido, a Qualidade Vai Vir
- “In theory, theory and practice are the same. In practice, they are different”.

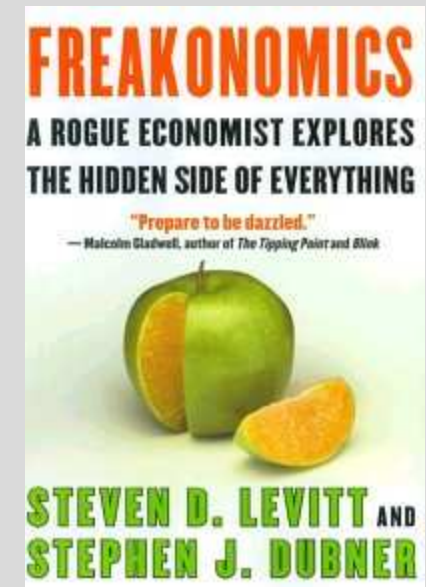
## ISO 900x, CMM etc

- A Máfia da Certificação. Scuba Divers, anybody?
- Consultores querendo vender seus livros
- “Pra Inglês Ver”: No papel a gente faz assim/assado, pro auditor (ISO|CMM|<insira sigla aqui>) ver. Na prática, bom, ....

“It’s possible to ship 0 lines of working software, but a pile of documentation, and be CMM-compliant. Not so with Agile.”

- <identidade protegida/removida>, Eclipse Foundation (ex-OTI, ex-prof. Carleton University), em uma conversa privada.

It’s all about incentives:



# Empresas Grandes 2003...



## Agile

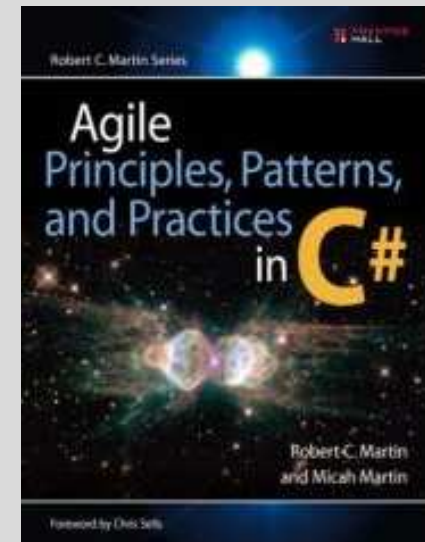
- SCRUM
- XP subset



# Agile



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



# Agile Manifesto



## Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck  
Mike Beedle  
Arie van Bennekum  
Alistair Cockburn  
Ward Cunningham  
Martin Fowler  
James Grenning  
Jim Highsmith  
Andrew Hunt

Ron Jeffries  
Jon Kern  
Brian Marick  
Robert C. Martin  
Steve Mellor  
Ken Schwaber  
Jeff Sutherland  
Dave Thomas

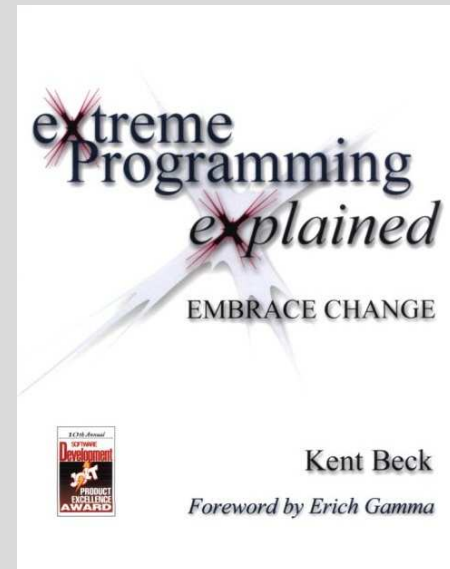


### XP Practices

- Whole Team - a.k.a. "customer present"
- Short Cycles (short iterations)
- User Stories
- Acceptance Tests
- Test Driven Development
- Pair Programming (a.k.a. Continuous Code Review)
- Collective Ownership
- Continuous Integration (detect breaking code early)
- Sustainable Pace (40-h week)
- Open Workspace
- The Planning Game (Business people decide on priorities, Developers decide on estimates)
- Simple Design (Do the simplest thing that could possibly work)
- Refactoring
- Metaphor (big picture is shared by all)

### Problemas

- Imagem: Extreme é coisa pra jovem. Arriscado.
- Religião: Pair Programming??!!
- Empresas grandes são geralmente conservadoras



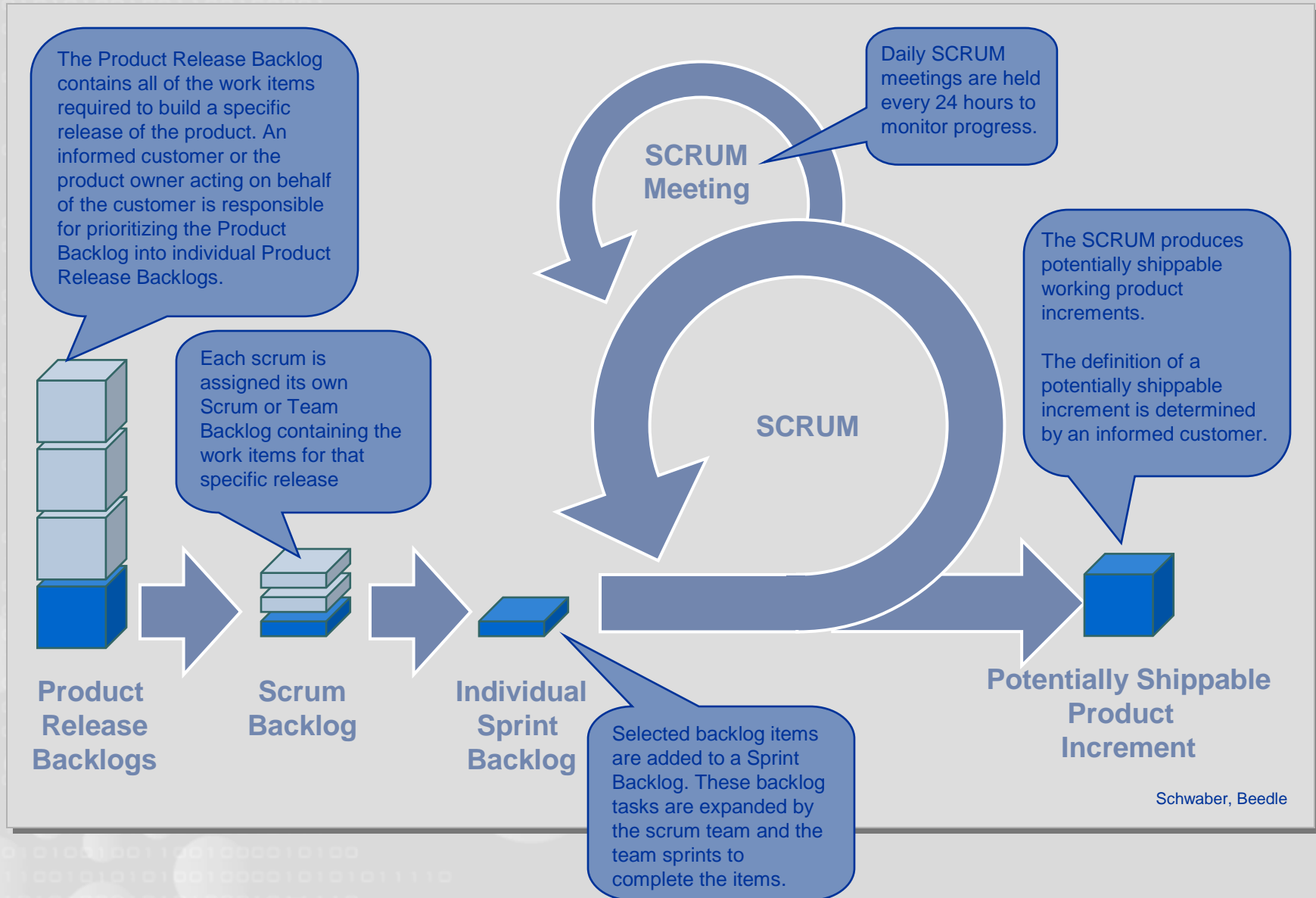
# Non-XP Workarounds/Subsets



- Code Review como substituto de Pair Programming (Smart Bear, etc)
- Unit Tests apenas pra novo código
- ...



# SCRUM





## SCRUM Practices

- Product Backlog
- Sprint (a.k.a. Short Iterations; Time-boxed)
- Sprint Planning Meeting
- Sprint Backlog
- Daily Scrum Meeting
- Sprint Review Meeting
- Roles

Product Owner

Scrum Master

Scrum Team

Repare: Nada quanto a “Engineering Practices”. Contraste com XP.

# How About FDD, Crystal, etc?



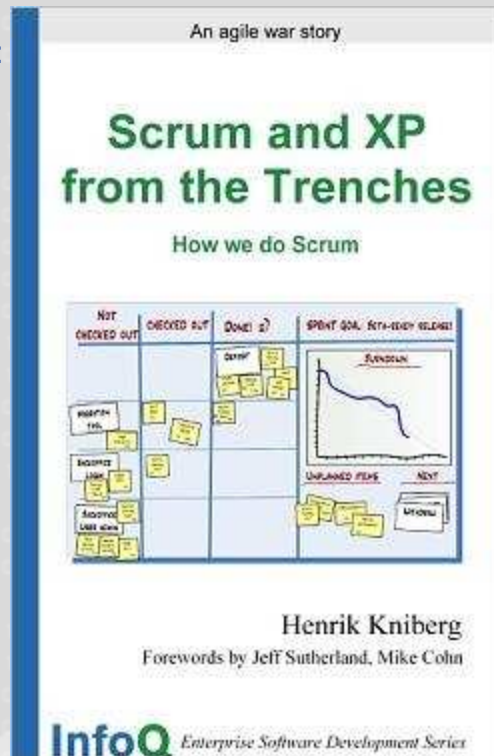
- Rare Animals – We haven't seen them in the wild yet



# Diga Mais Sobre XP + SCRUM



Baixe e leia o PDF:



Procure em video.google.com: SCRUM



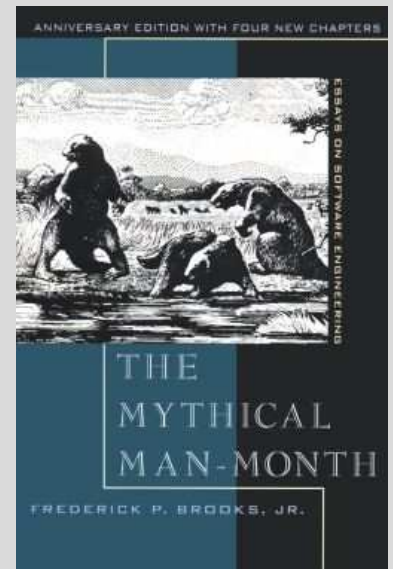
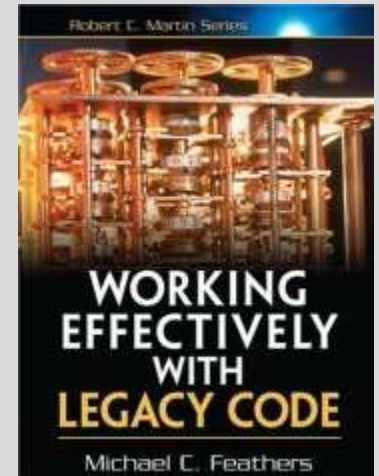
# O Lado Negro de Agile



- A Máfia da Certificação. Scuba Divers, anybody?  
(sim, sim, eu também fiz o curso de Scrum Master)
- A Máfia da Consultoria. Todo mundo agora é expert. Lembram-se de OO?  
(sim, sim, eu também dou consultoria em Agile)
- Micromanagement (O que você fez ontem? O que vai fazer hoje? Etc.)  
Algumas pessoas simplesmente não têm Peopeware skills.
- Visão simplória de alguns – falta de conciliação com empresas grandes com Legacy Code

Acima de tudo, não esqueça: No Silver Bullet (Fred Brooks)

[http://en.wikipedia.org/wiki/No\\_Silver\\_Bullet](http://en.wikipedia.org/wiki/No_Silver_Bullet)

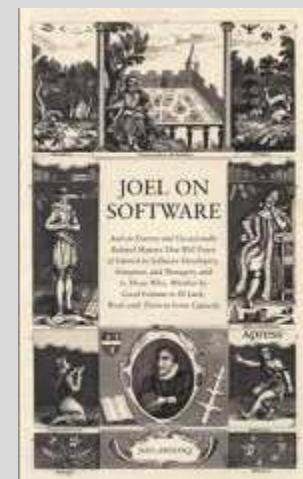


# Quão Bem Está Seu Grupo? Joel On Software - Test



- The Joel Test: 12 Steps to Better Code, By Joel Spolsky
- Wednesday, August 09, 2000
- <http://www.joelonsoftware.com/articles/fog0000000043.html>

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?



# Bedarra (Shameless Plug)



- Dave Thomas , Brian Barry (sounds familiar?)
- R&D Outsourcing
- Mentoring/Consulting
- CTIP: Continuous Test and Integration Platform (como usar alguns componentes Open Source de forma eficaz na construção ágil)
- Parceria com Object Mentor (Robert Martin) e Online-Learning.com. Blended Courses: online + on-site visit. Blended TDD: Test-Driven Development, 6 módulos. SCORM-compliant.
- Lean And Agile in the large. Agile 2007. <http://www.agile2007.org/index.php?page=sub/&id=790>



- Escolhemos, testamos & pré-configuramos:
  - Uma ferramenta *a la* make (e.g. Ant, Maven, etc)
  - Uma ferramenta de static analysis (e.g. Lint)
  - Uma ferramenta de Code Critics
  - Uma ferramenta de Code Coverage
  - Ferramenta de Unit Test e Acceptance Test
  - Um sistema de Continuous Build
  - ...
  - 2 exemplos de projetos funcionando & com reports
  - Disponível em 2 versões atualmente: Java e C++
  - Disponível em forma “InstallShield”, versões Developer, Server, Full (útil p/ demo/laptops)
  
- Idéia Principal: “Hit the ground running”
  - Copie os arquivos de configuração dos projetos exemplo
  - Adapte pro seu Projeto/Realidade
  - Best Practices
  - Instale facilmente nas desktops dos varios desenvolvedores (InstallShield) e servidor.



- Static Analyzer:

Bug Summary	Analysis Informations	List bugs by bug category	List bugs by package
-------------	-----------------------	---------------------------	----------------------

■ P1 
 ■ P2 
 ■ P3 
 ■ Exp.

**Malicious code vulnerability (1: 1/0/0/0)**

**MS: Mutable static field (1: 1/0/0/0)**

**Field isn't final but should be (1: 1/0/0/0)**

■ MS: In class `com.bedarra.continuousbuild.example.HelloWorldServlet`  
 In class `com.bedarra.continuousbuild.example.HelloWorldServlet`  
 Field `com.bedarra.continuousbuild.example.HelloWorldServlet.HELLO`

MS: `com.bedarra.continuousbuild.example.HelloWorldServlet.HELLO` isn't final but should be  
 A mutable static field could be changed by malicious code or by accident from another package.  
 The field could be made final to avoid this vulnerability.

- Continuous Builds:

Projects	Builds	Logs	Browser
----------	--------	------	---------

Project	Schedule	When to trigger	Latest build	Latest build date
<div style="border: 1px solid gray; padding: 2px;">  HelloWorld                 </div>				
HelloWorld	<span style="color: green;">●</span> Default	simple	<span style="color: green;">●</span> 1.1	2006.07.17 12:15
<div style="border: 1px solid gray; padding: 2px;">  Wumpus                 </div>				
Wumpus	<span style="color: green;">●</span> Default	simple	<span style="color: green;">●</span> 1.1	2006.07.17 12:19

- X-Ref código & User Stories:

User Stories Report							
Implemented Stories							
Story	Status	Developer(s)	Start	Last Changed	Implementation	Unit Tests	Acceptance Tests
<a href="#">StoryFallInPit</a>	Done	brianB;marciom	25-Apr-2006	25-Apr-2006	[type] <a href="#">HuntTheWumpus.Game</a> [field] pits [field] fellInPit [method] checkForPit() [method] putPitInCavern(int) [method] fellInPit()	[type] <a href="#">HuntTheWumpus.GameTest</a> [method] testFallInPit()	<a href="#">AtFallInPit</a>
<a href="#">StorySmellWumpus</a>	In-Progress	mikeT;daveT	25-Apr-2006	25-Apr-2006	[type] <a href="#">HuntTheWumpus.Game</a> [method] canSmellWumpus()	[type] <a href="#">HuntTheWumpus.GameTest</a> [method] testSmellWumpus() [method] testCantSmellWumpus()	<a href="#">AtSmellWumpus</a>
<a href="#">StoryHearBats</a>	Done	victorL;dougT	25-Apr-2006	25-Apr-2006	[type] <a href="#">HuntTheWumpus.Game</a> [method] canHearBats()	[type] <a href="#">HuntTheWumpus.GameTest</a> [method] testHearBats()	<a href="#">AtHearBats</a>
<a href="#">StoryWumpusMovesRandomly</a>	Done	daveT;mikeT	25-Apr-2006	25-Apr-2006	[type] <a href="#">HuntTheWumpus.Game</a> [field] wumpusCavern [field] wumpusFrozen [method] rest() [method] putWumpusInCavern(int) [method] getWumpusCavern() [method] moveWumpus() [method] freezeWumpus()	[type] <a href="#">HuntTheWumpus.GameTest</a> [method] testRandomWumpusMovement()	[type] <a href="#">HuntTheWumpus.fixtures.ChAtWumpusMovesRandomly</a>
StoryArrows			26-Apr-2006	26-Apr-2006	[type] <a href="#">HuntTheWumpus.GamePresenter</a> [method] shootArrow(String) [type] <a href="#">HuntTheWumpus.Game</a> [field] quiver	X	X



- Code Coverage:

[all classes]

**OVERALL COVERAGE SUMMARY**

name	class, %	method, %	block, %	line, %
all classes	80% (4/5)	81% (13/16)	79% (96/122)	73% (27.8/38)

**OVERALL STATS SUMMARY**

total packages: 5  
 total executable files: 5  
 total classes: 5  
 total methods: 16  
 total executable lines: 38

**COVERAGE BREAKDOWN BY PACKAGE**

name	class, %	method, %	block, %	line, %
coa_bedarra_continuousbuild_example_tests_fitnessse	0% (0/1)	0% (0/2)	0% (0/9)	0% (0/2)
coa_bedarra_continuousbuild_example	100% (1/1)	75% (3/4)	55% (16/29)	42% (5/12)
coa_bedarra_continuousbuild_example_tests_httpunit_cactus	100% (1/1)	100% (4/4)	92% (44/48)	90% (10.8/12)
coa_bedarra_continuousbuild_example_tests_cactus	100% (1/1)	100% (4/4)	100% (29/29)	100% (9/9)
coa_bedarra_continuousbuild_example_tests_junit	100% (1/1)	100% (2/2)	100% (7/7)	100% (3/3)

- Acceptance Tests:

HuntTheWumpus.

## AllRequirements

COMMAND LINE TEST RESULTS

Test Pages: 72 right, 0 wrong, 0 ignored, 0 exceptions    Assertions: 1528 right, 0 wrong, 0 ignored, 0 exceptions

14 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.Arrows.PickUpArrow
42 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.Arrows.ShootArrow.TestAbbreviationsOfShootCommand
15 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.Arrows.ShootArrow.TestCanShootWhenThereAreArrows
4 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.Arrows.ShootArrow.TestCannotShootWhenThereAreNoArrows
6 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.Arrows.ShootArrow.TestPlayerDiesWhenArrowBouncesOffWallInSameCavern
14 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtArrows.AtPickUpArrow
42 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtArrows.AtShootArrow.TestAbbreviationsOfShootCommand
15 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtArrows.AtShootArrow.TestCanShootWhenThereAreArrows
4 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtArrows.AtShootArrow.TestCannotShootWhenThereAreNoArrows
6 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtArrows.AtShootArrow.TestPlayerDiesWhenArrowBouncesOffWallInSameCavern
0 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtBatTransportRandomly
8 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtFallInPit
0 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtHearBats
26 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtHearPit
40 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtKillWumpus
36 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtNavigatePlayer.TestCanMoveThroughMap
47 right, 0 wrong, 0 ignored, 0 exceptions	IterationOne.AtNavigatePlayer.TestCanTest

- Metrics:

**Summary**    [\[summary\]](#) [\[packages\]](#) [\[cycles\]](#) [\[explanations\]](#)

Package	Total Classes	Abstract Classes	Concrete Classes	Afferent Couplings	Efferent Couplings	Abstractness	Instability	Distance
HuntTheWumpus	10	1	9	1	6	0.1	0.86	0.04
HuntTheWumpus.fixtures	6	0	6	1	6	0	0.86	0.14
com.vladium.emma.rt	No stats available: package referenced, but not analyzed.							
fit	No stats available: package referenced, but not analyzed.							
java.io	No stats available: package referenced, but not analyzed.							
java.lang	No stats available: package referenced, but not analyzed.							
java.util	No stats available: package referenced, but not analyzed.							
junit.framework	No stats available: package referenced, but not analyzed.							

[\[summary\]](#) [\[packages\]](#) [\[cycles\]](#) [\[explanations\]](#)

**HuntTheWumpus**

Afferent Couplings: 1    Efferent Couplings: 6    Abstractness: 0.1    Instability: 0.86    Distance: 0.04

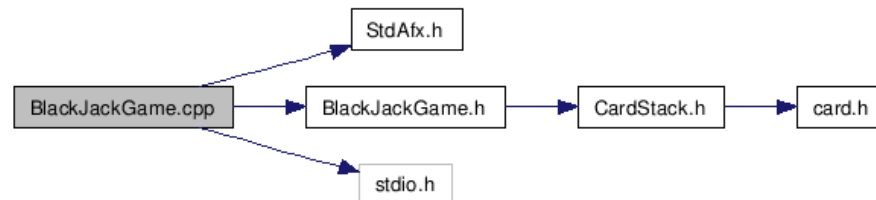


- Dependencies:

## BlackJackGame.cpp File Reference

```
#include "StdAfx.h"
#include "BlackJackGame.h"
#include <stdio.h>
```

Include dependency graph for BlackJackGame.cpp:

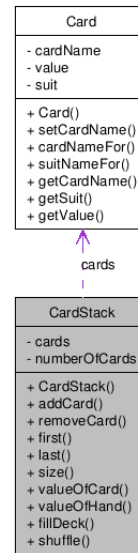


- Diagrams from the code:

## CardStack Class Reference

```
#include <CardStack.h>
```

Collaboration diagram for CardStack:



[List of all members.](#)

Collaboration graph

# Bedarra – Blended Courses



- O que o aluno vê.
  - Locker Page (home page própria no portal)
  - Módulos/cursos onde está registrado

**ONLINE-LEARNING.com**  
Home > Locker

Help  
Catalogue  
Change Password

## Welcome Back Marcio!

You are currently registered in the following courses:

---

**Module 1: Intro to Test Driven Development (OMTDD1-NOV07) [2007-11-22...2007-12-31]**

- ➔ Enter the course online
- ➔ Email your Instructor

---

No new messages from your Instructor

---

**Module 2: Principles, Patterns and Practices (OMTDD2-NOV07) [2007-11-22...2007-12-31]**

- ➔ Enter the course online
- ➔ Email your Instructor

---

No new messages from your Instructor

---

**Module 3: Refactoring (OMTDD3-NOV07) [2007-11-22...2007-12-31]**

- ➔ Enter the course online
- ➔ Email your Instructor

---

No new messages from your Instructor

---

**Module 4: Unit Test Tools (OMTDD4-NOV07) [2007-11-22...2007-12-31]**

- ➔ Enter the course online

**Object Mentor Incorporated**

Module 1: The Three Laws of TDD

- Has This Ever Happened To You?
- Automated Tests Provide A Safety Net
- The Three Laws of TDD
- The Development Cycle
- Calling your shots: Strategy
- What Is Tested?
- TDD Forces You To Focus
- The Benefits of TDD
- The Ethics Behind TDD
- Test-FIRST: Principles Of Test-Driven Development
- Test

## The Three Laws of TDD

Page 4 of 81

Test-Driven Development is a disciplined approach to design and implementation. At first it may be counter-intuitive but as we will see, the benefits are profound. Test-Driven Development can be summarized in three simple laws:

### The Three Laws of Test-Driven Development

1. You cannot write production code until you have a failing unit test.
2. You cannot write more of a unit test than is sufficient to fail. And not compiling is failing.
3. You cannot write more production code than is sufficient to pass.

While the Laws of Test-Driven Development should be taken seriously, they are not unbreakable. For example, not all unit tests are written first. On occasion, some tests are written after. The real goal is to write the tests and production code concurrently, with the tests generally a few minutes ahead of the production code.

[Click Here To View The Three Laws Video](#)

© 2007 Object Mentor. All rights reserved. Produced by www.online-learning.com.



# Bedarra – Blended Courses



- O que o instrutor vê (e opcionalmente a empresa que paga pra treinar seus funcionários).
  - Quão bem o aluno foi nas “quizzes”
  - Quão bem o aluno foi nos “assignments”- os quais são corrigidos pelo time da Object Mentor (Bob Martin etc)
  - Quanto tempo o aluno gastou no curso, qual a nota final, etc

## OMTDD1-NOV07 : Quizzes, Assignments and Grades on November 30, 2007, 4:34 pm

- Click on a column heading to sort by that column.
- Hover on a quiz score to see when it was taken.
- Hover on any 'Last Visit' cell to see the URL for the last page visited.
- Please note that 'Browser Time' is an approximation. It can be higher than expected if the browser is left open on a page while not being used. It can also be lower than expected if the course is not closed properly.

Student	Quizzes	Submitted Text [ <a href="#">Hide All</a> ] [ <a href="#">Show All</a> ] [ <a href="#">Show Without Feedback</a> ]	Last Visit	Browser Time (HHH:MM:SS.SS)	Final Mark
Marchini, Marcio	[Q0]=28.33	<p>Submitted Text # 0 (SCO: content_index_html)</p> <pre>1 function htmlTextAsTableWithLineNumbers (\$text) { 2     \$lines = explode ("&lt;br&gt;", \$text); 3     \$result = "&lt;TABLE cellpadding=\"0\" cellspacing=\"0\" class=\"lineNumbersAndText\"&gt; 4     \$lineNumber = 0; 5     foreach (\$lines as \$lineText) { 6         \$lineNumber++; 7         \$result .= "&lt;TR&gt;&lt;TD class=\"lineNumber\"&gt;\$lineNumber&lt;/TD&gt;&lt;TD class=\"lineText\"&gt;\$lineText&lt;/TD&gt;&lt;/TR&gt;"; 8     } 9     \$result .= "&lt;/TABLE&gt;"; 10    return \$result; 11 }</pre> <p>Instructor Feedback (use HTML markup/notation):</p> <div style="border: 1px solid #ccc; padding: 5px;"><p>Font: [v] Font size: [v] Choose style: [v]</p><p>[B] [I] [U] [List-Group icons]</p></div> <p>[Save]</p>	2007-11-30 13:49:40	0006:55:18.27	<a href="#">21</a>

# Bedarra – Lean and Agile In The Large



- Designed for large and small software development projects
- Designed for projects that span multiple teams
- Designed for requiring a supplier-customer relationship
- Designed for any type of project such as software applications, components, internal, third-party, or consulting projects

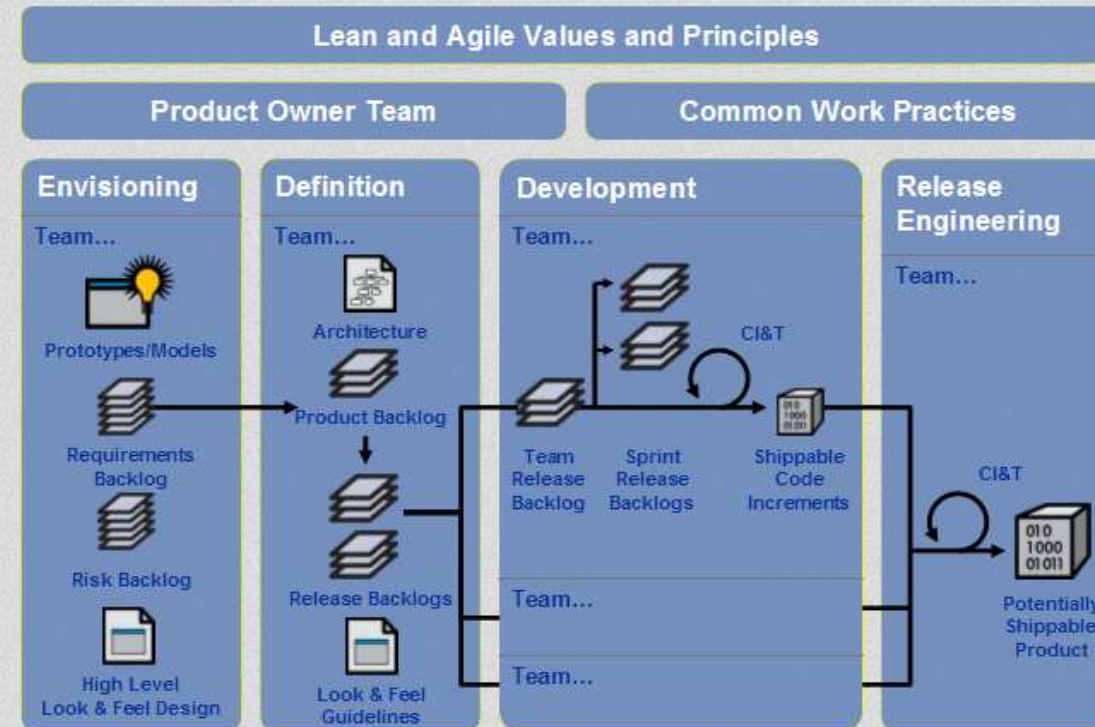


- Lean and Agile In The Large D
  - Process Overview
  - Envisioning Phase
  - Definition Phase
  - Development Phase
  - Release Engineering Phas
  - Agile Product Team
  - Agile Principles and Practic
- Quick Reference
  - List of Agile Roles
  - List of Agile Practices
  - List of Agile Artifacts

## Process Overview

© 2007 Bedarra Research Labs Page 2 of 12

The diagram below illustrates the Lean and Agile In The Large Development Process. The process consists of an integrated product team that spans all phases of the process, a set of principles and practices that are used to guide development, and a four-phase iterative design cycle that is used to deliver a product. The different parts of the process are described in subsequent sections of this document.



# 1-Slide Process Overview



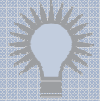
## Agile Principles

## Agile Product Team

### Envisioning Phase

Assess market risks  
Assess R & D risks  
Gather requirements  
Develop vision

#### Scrums...



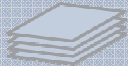
Vision Prototypes



Requirements Backlog



Risk Backlog



Look & Feel Guidelines

### Definition Phase

Convert requirements to features  
Design architecture (ADD)

#### Scrums...



Product Architecture



Product Backlog



Release Backlogs

### Development Phase

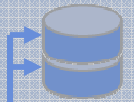
Adopt test-driven development model  
Implement & re-factor code using pair development  
Employ a Continuous Integration & Test Environment

#### SCRUM Team 1



SCRUM Release Backlog

Sprint Release Backlogs



Shippable Code Increments



#### SCRUM Team 2

#### SCRUM Team N

### Release Engineering Phase

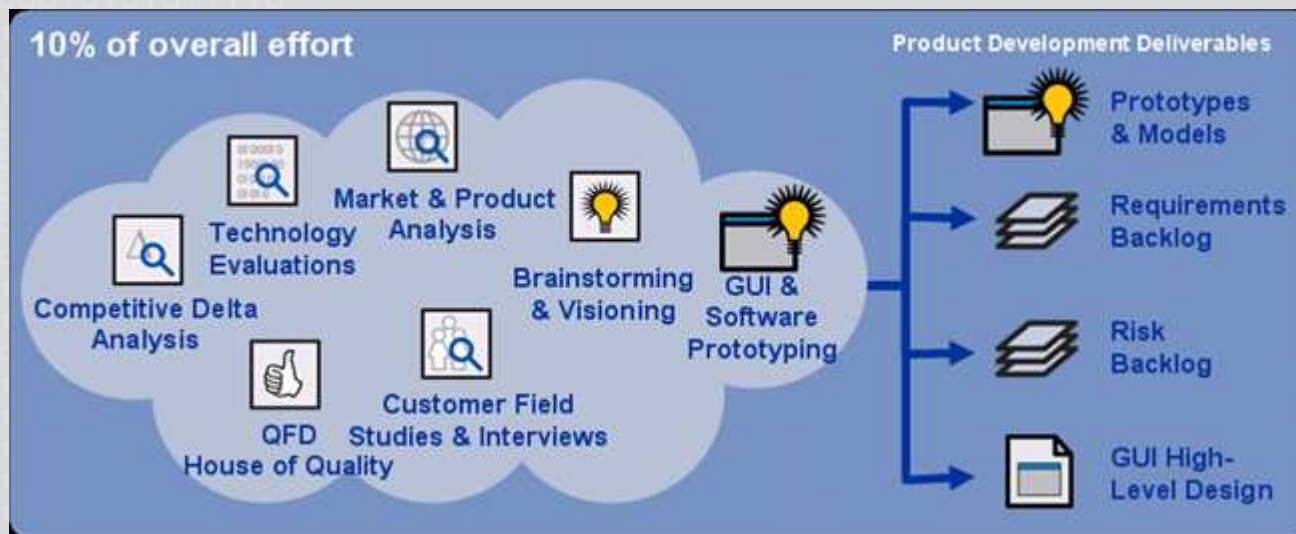
Continuous integration and test  
Manage dependencies  
Manage the "End Game"

#### Scrums...

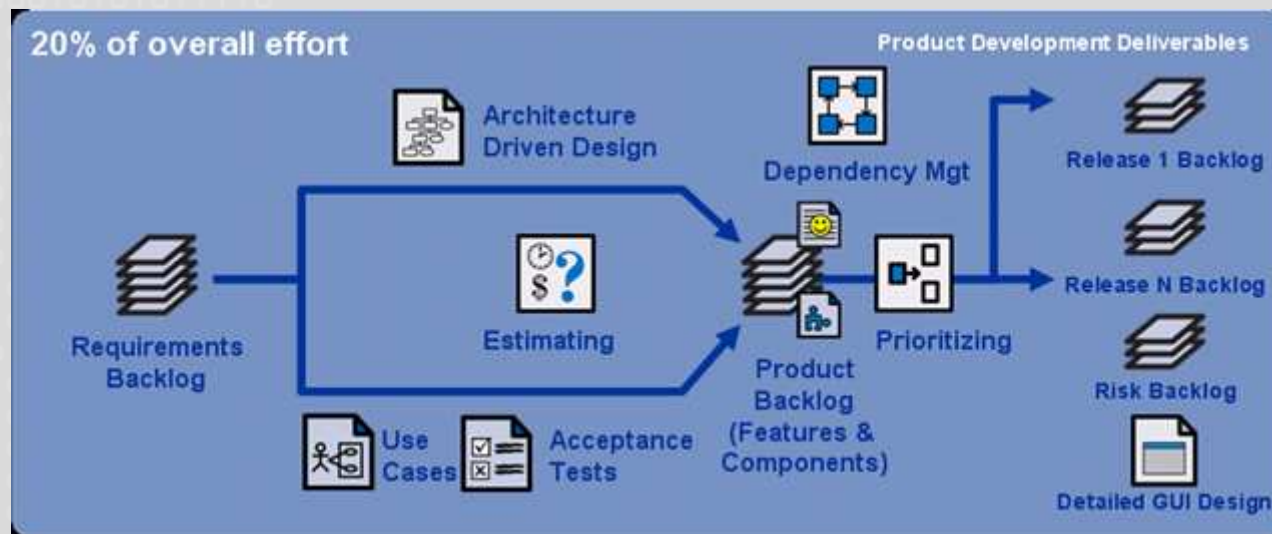


Potentially Shippable Product

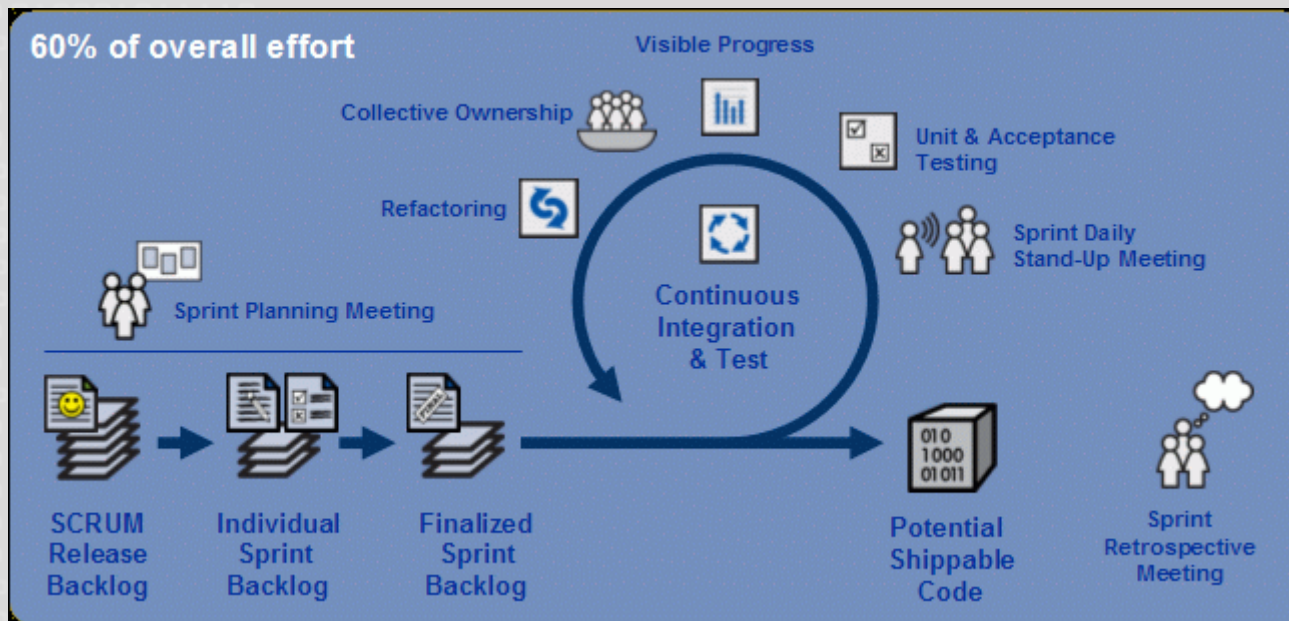
# AIL - Envisioning



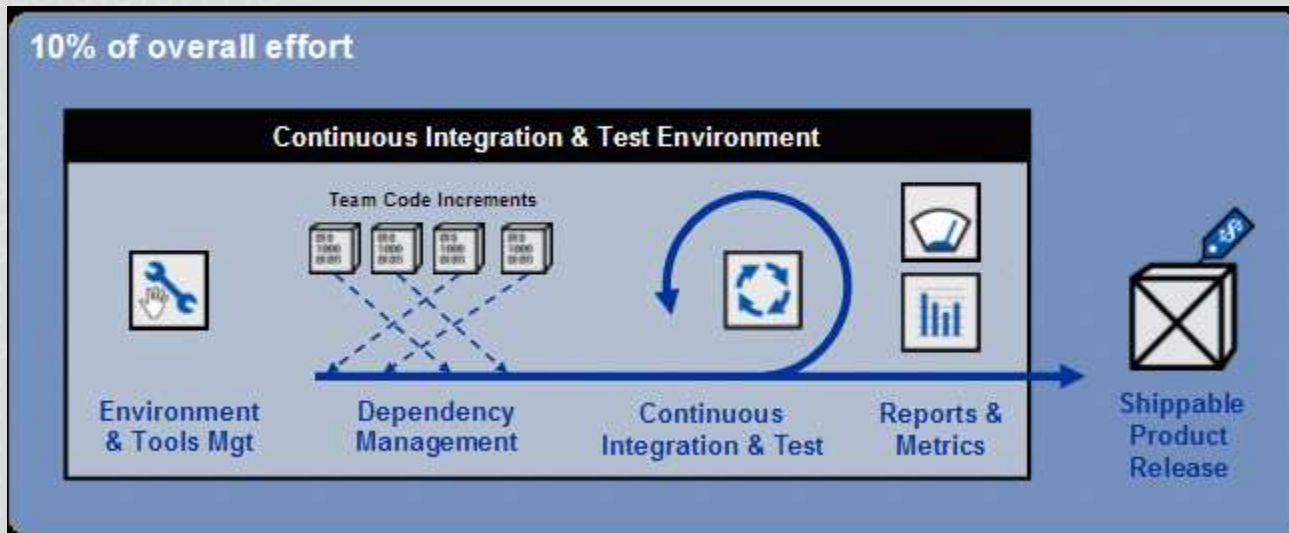
# AIL - Definition



# AIL - Development



# AIL – Release Engineering





# Bedarra em Floripa ?!



- Audaces

Fase 1: Levantamento da Situação

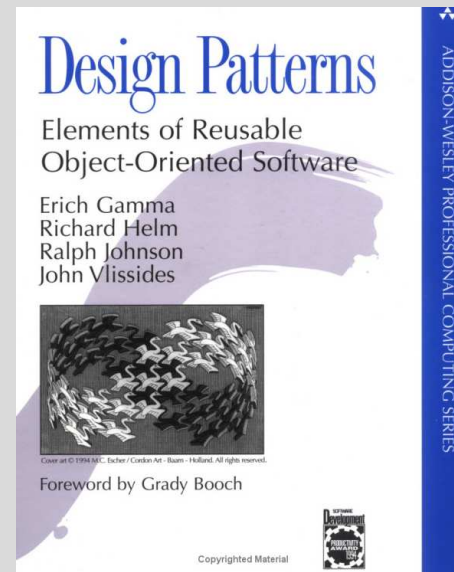
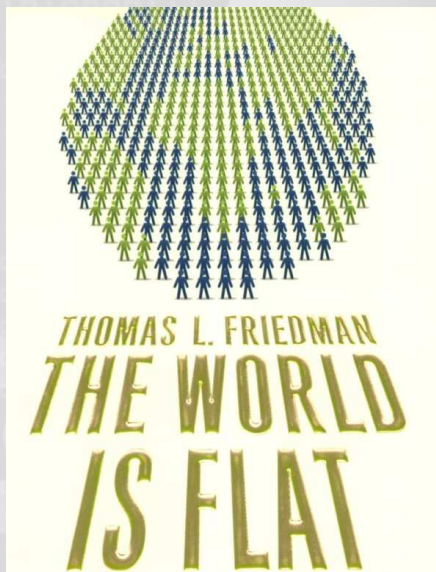
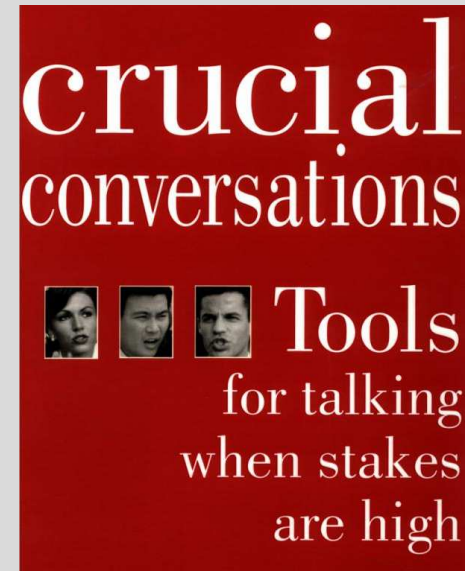
Fase 2: CTIP

Fase 3: Agile Mentoring (Blended TDD, AIL)

- Santo de Casa



# Conselhos a um Jovem Profissional



# Conselhos a um Jovem Profissional



- TortoiseCVS / TortoiseSVN
- Wiki
- TRAC (+ SCRUM Plugin etc)
- ...

# Conselhos a um Jovem Profissional



"Complexity kills. It sucks the life out of developers,  
it makes products difficult to plan, build and test.  
... Each of us should ...  
explore and embrace techniques to reduce complexity."

Ray Ozzie, CTO, Microsoft

KISS: Keep It Simple, Stupid!

[http://en.wikipedia.org/wiki/KISS\\_principle](http://en.wikipedia.org/wiki/KISS_principle)

# Conselhos a um Jovem Professor



- Modelo de Co-Op Canadense: estudantes gastam 3 ou 4 “terms” em empresas. Faz parte do currículo. R&D abundante. (contraste com ser estagiário num sistema de videlocadora)
- Trabalhos de programação poderiam ser passados com Acceptance Criteria *a la* FIT/Fitness. O resultado submetido deveria passar algo como CTIP:
  - Funciona – build com sucesso, fora do IDE.
  - Passa nos Unit Tests e Acceptance Tests
  - Code Coverage mostra que estás testando 90%+
  - Code Metrics mostra que não fizeste uma massaroca
  - Static Analyzer mostra que não tens bugs escabrosos escondidos
  - Histórico no sistema de controle de versão disponível (não basta submeter a versão final)
  - ...
- Pre-req de trabalhos. Exemplo: Compilador na cadeira de compiladores tem que funcionar na máquina virtual (seja ITISIAC ou JVM) que implementaste na cadeira anterior. Valor:
  - Legacy
  - Knowledge In The Head versus Knowledge in The World. (Don Norman)
  - Dirty kitchen (Ron Jeffries, Bob Martin)



Inspirar-se no modelo Canadense: IRAP, NSERC, etc - \$ p/ R&D, oriundo dos impostos das próprias empresas.

## **Silicon Valley North: The Formation of the Ottawa Innovation Cluster**

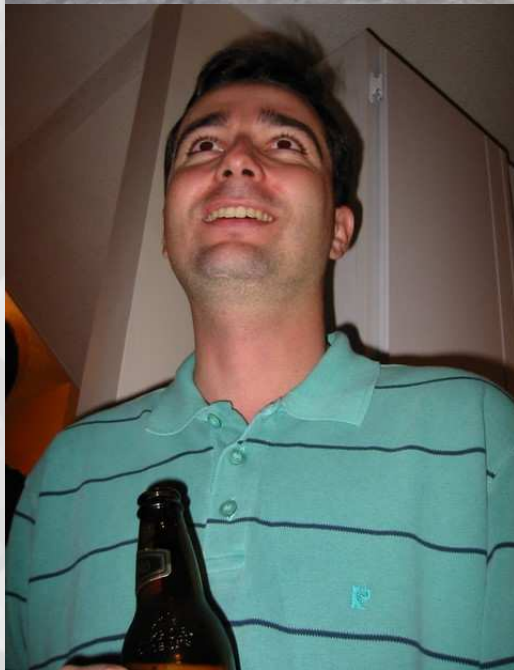
<http://www.innovationstrategy.gc.ca/gol/innovation/site.nsf/en/in02340.html>

“Ottawa's strong R&D emphasis is regarded as unusual among comparable clusters in North America. Measured on a per capita basis, Ottawa in the late 1990s was doing three times the R&D of other Canadian cities. The region's powerful research strengths have twin roots. The first lies in the growth after World War II of public scientific research labs, including the National Research Council (NRC), the Defence Research Establishment, the Communications Research Centre (CRC) and later, the Atomic Energy of Canada Laboratory (AECL). The second root lies in a decision taken in the late 1950s by Northern Electric (now Nortel) to choose Ottawa as the site for its small but rapidly growing R&D operation. “

# Conselhos a um Jovem Empresário



- Sua empresa vai conseguir atrair gente interessante ou operas apenas como uma Sweat Shop?
- Sua empresa vai conseguir competir num mundo Ágil & “Plano”?
- Seu ISO 9000 / CMM / 6-Sigma / etc é “pra inglês ver” ou produz com qualidade comparável às “Agile Best Practices”?







- Dave Thomas
- Luiz Fernando Bier Melgarejo (o “Melga”)

# Obrigado!



■ Perguntas?