

Integração Contínua

...usando software livre

CHRISTIANO MILFONT - <http://www.milfont.org>

cmilfont@gmail.com 20/10/2006

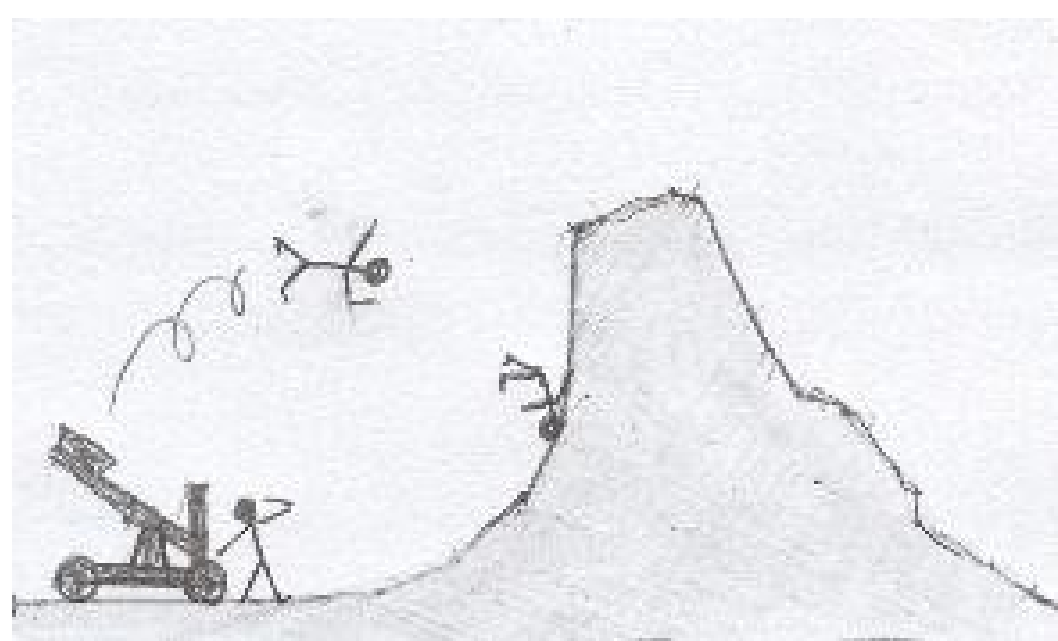
ROTEIRO



- 1. HISTÓRICO.**
- 2. GERÊNCIA DE CONFIGURAÇÃO.**
- 3. FERRAMENTAS.**
- 4. CONTROLE DE VERSÃO.**
- 5. CONTROLE DE MUDANÇAS.**
- 6. INTEGRAÇÃO CONTÍNUA.**
- 7. PRÁTICAS**
- 8. CONCLUSÃO.**
- 9. REFERÊNCIAS**

HISTÓRICO

- ANOS 80

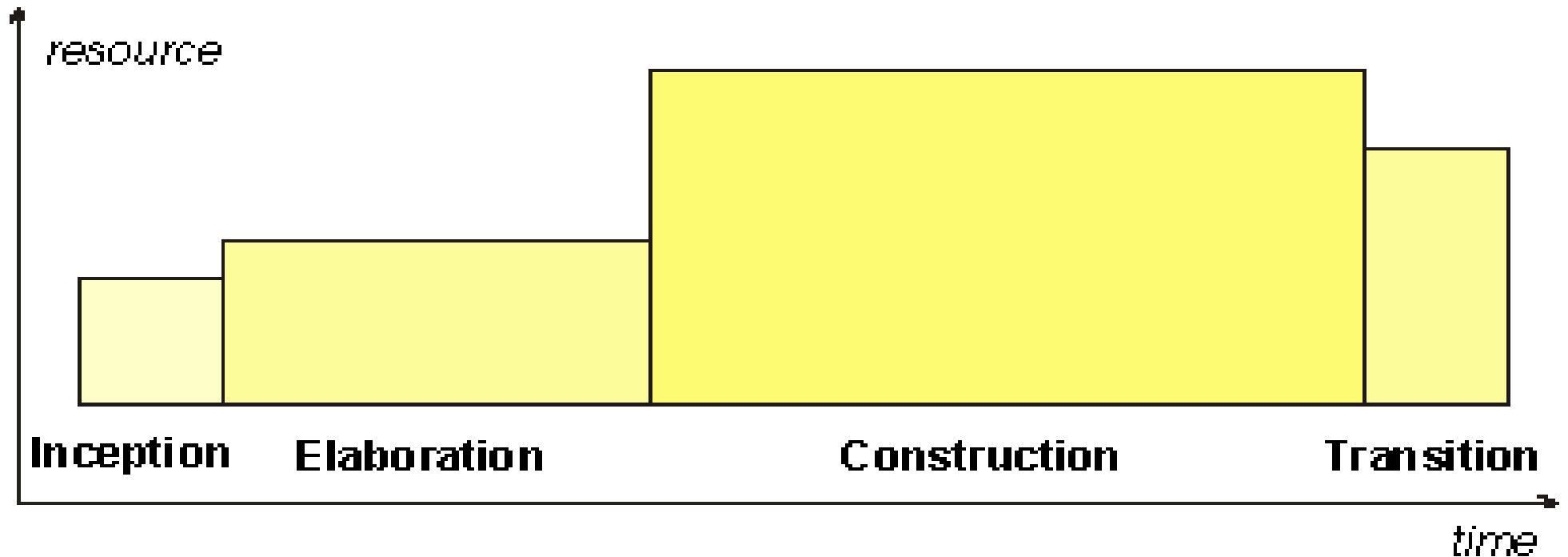


- **Anos 80 - década Code-and-Fix.**
- **Ausência de metodologias de desenvolvimento.**
- **Programação procedural e estruturada.**
- **Evolução da programação linear.**
- **Programas são: sequência, decisão e iteração.**
- **Dificuldade de simular relações entre entidades em processos de negócios.**

HISTÓRICO

- ANOS 90

- Linguagem UML.
- Processos unificados (UP).
- Metodologias Orientadas a Objetos.
- Fases bem definidas e controladas.
- Analogia com a Engenharia Civil.
- **Concepção, Elaboração, Construção e Transição.**



HISTÓRICO

- PROCESSOS UNIFICADOS

- Código complexo.
- Manutenção difícil.
- Baixa produtividade.
- Cronograma sempre atrasado.
- Insatisfação de todos.
- Design degradado.
- Documentação defasada, excessiva e ilegível.
- Fracasso nos projeto.



HISTÓRICO - MÉTODOS ÁGEIS

- **Lendário Projeto C3 - Comprehensive Compensation project – Payroll system (Chrysler).**
- **Melhores práticas (Design patterns).**
- **Análise Orientada a Objetos.**
- **Manifesto ágil.**
- **Agile Alliance (<http://www.agilealliance.org/>).**



Ward Cunningham



Ron Jeffries



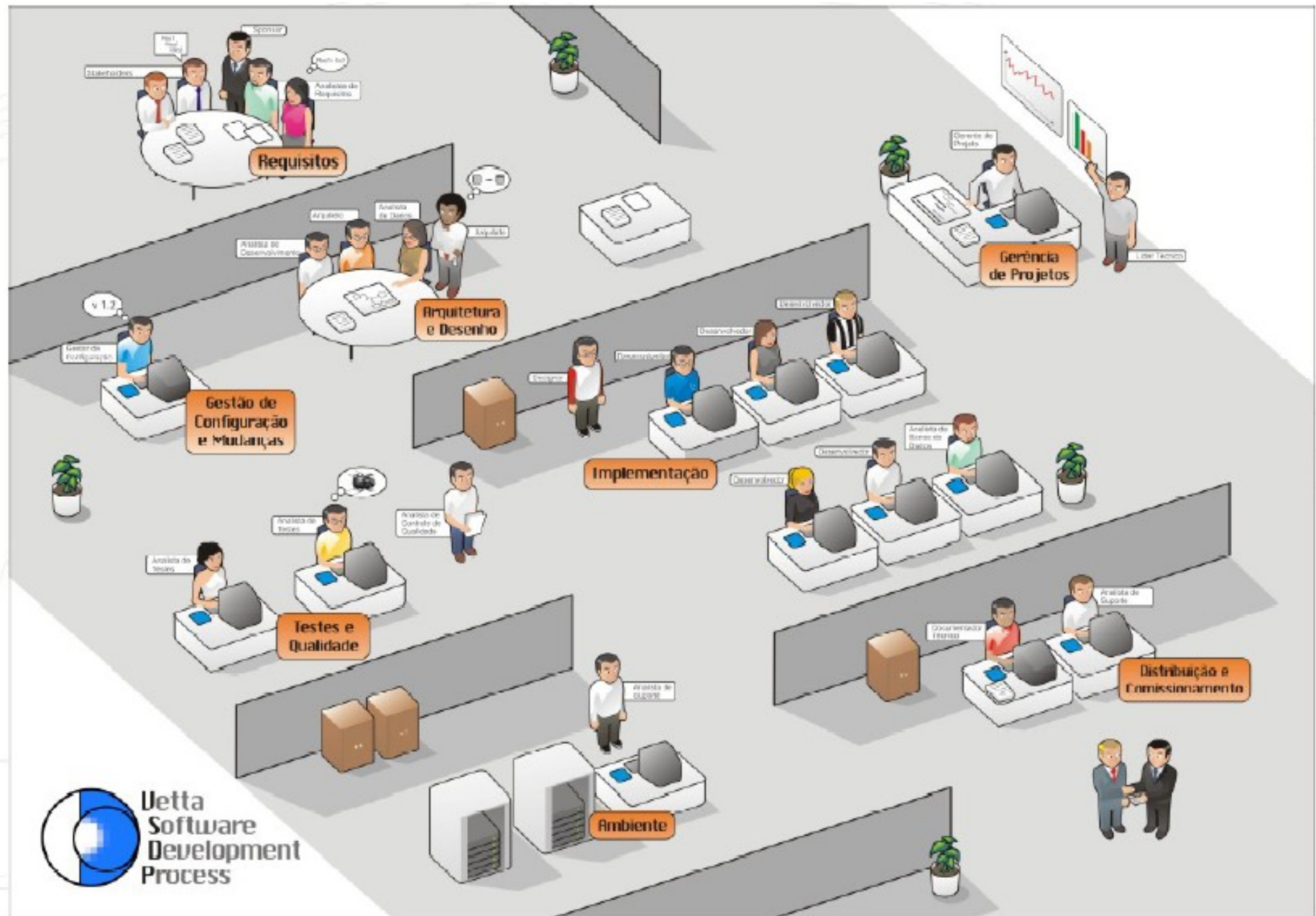
Kent Beck

EIS QUE SURGE...

GERÊNCIA DE CONFIGURAÇÃO

“Gerência de Configuração (GC) é um conjunto de atividades de apoio ao desenvolvimento que permite que as mudanças inerentes ao desenvolvimento sejam absorvidas pelo projeto de maneira controlada, mantendo a estabilidade na evolução do software.”

CONTROLAR AMBIENTES COMPLEXOS



O QUE É:

- **Gerencia as ferramentas que compõe o desenvolvimento de sistemas.**
- **Identifica as configurações de diferentes pontos no decorrer do tempo.**
- **Controla sistematicamente as mudanças realizadas.**
- **Fornece estatísticas da evolução dos sistemas.**
- **Faz parte do modelo de maturidade dos processos de desenvolvimento tais como CMMi, MPS-Br e o SPICE.**

O QUE NÃO É:

- **Comunica-se com gestão de redes, dados e suporte mas não gerencia essas áreas.**
- **Não controla o versionamento dos sistemas.**
- **Não determina as atividades a serem executadas.**

FERRAMENTAS DE APOIO

Gerência de Configuração

Controle de Versão

- **Armazenar.**
- **Versionar.**
- **Gerenciar.**
- **Ramificar.**
- **Recuperar.**

Controle de Mudanças

- **Identificar.**
- **Rastrear.**
- **Analisar.**
- **Controlar.**
- **Priorizar.**

Integração Contínua

- **Constuir.**
- **Testar**
- **Disponibilizar.**
- **Publicar.**
- **Notificar.**

FERRAMENTAS DE APOIO

Gerência de Configuração

Controle de Versão

- **Subversion.**
- **CVS.**

Controle de Mudanças

- **Elips.**
- **Bugzilla.**
- **Mantis.**
- **Trac.**
- **Jira.**

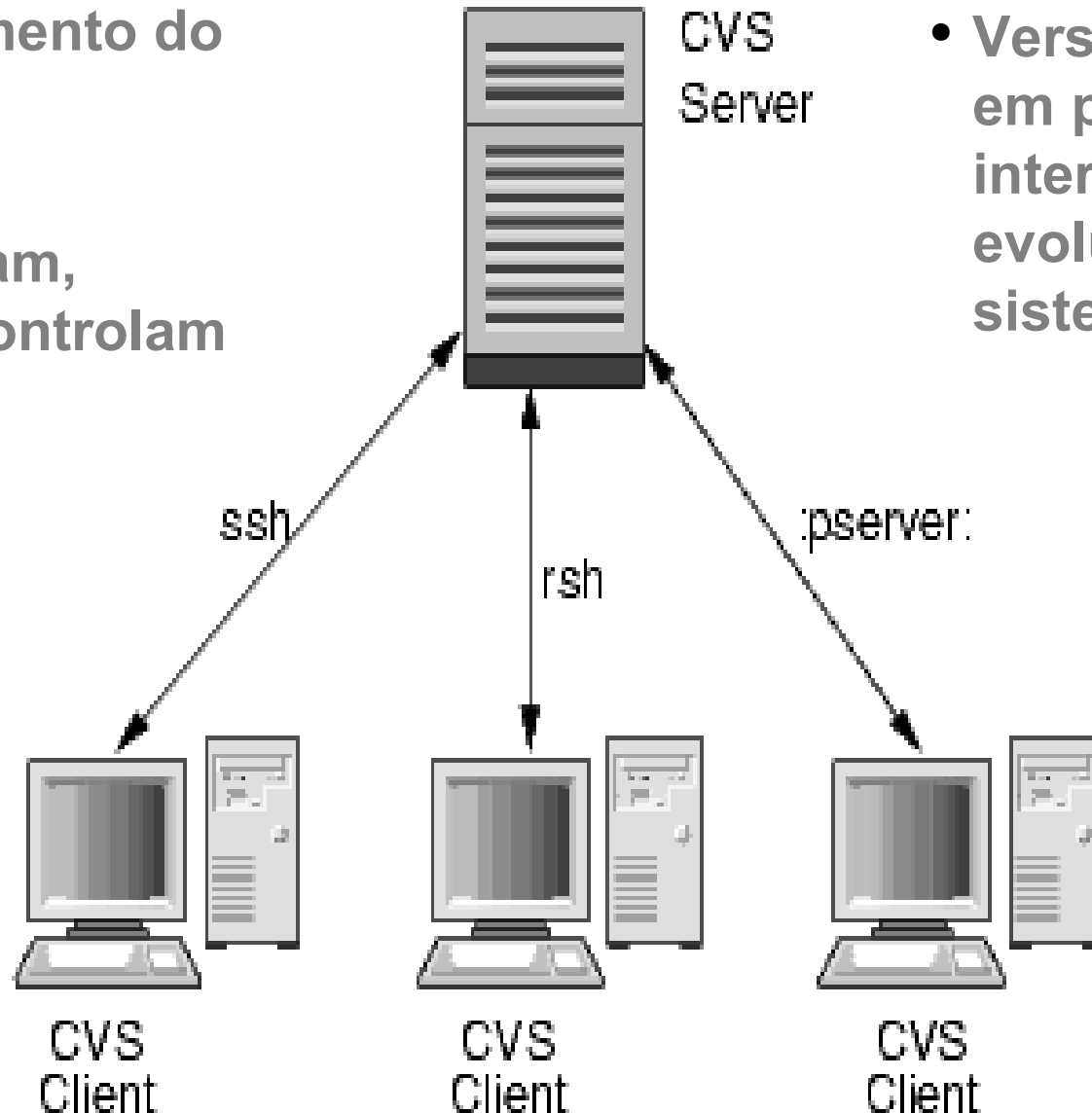
Integração Contínua

- **Ant.**
- **Ruby Rake**
- **Maven.**
- **Gump.**
- **CruiseControl.**
- **AntHill.**

CONTROLE DE VERSÃO

- Compartilhamento do código.
- SCM versionam, arquivam e controlam o projeto.

- Versões continuam em paralelo sem interromper a evolução do sistema.



CONTROLE DE MUDANÇAS

ISSUE TRACKING

- Fornece estatísticas e relatórios da consistência do projeto para facilitar a comunicação.

Requirements
Gathering
&
Management



Project Plan
&
Ad-hoc Tasks
Management

- Prioriza as funcionalidades a serem desenvolvidas ou consertadas.

- Mantém histórico de falhas e consertos no projeto.



Issue
Change Request
&
Defect
Tracking



- Cataloga feedback dos usuários.

CONTROLE DE MUDANÇAS

ISSUE TRACKING

Mantis

Logged in as: *prescience*

02-04-01 07:08 PST

mantis

[Main](#) | [View Bugs](#) | [Report Bug](#) | [Summary](#) | [Account](#) | [Manage](#) | [Edit News](#) | [Docs](#) | [Logout](#)

[[All Bugs](#)] [[Reported Bugs](#)] [[Assigned Bugs](#)]

any any any Show: Changed(hrs): Hide Resolved:

Viewing Bugs

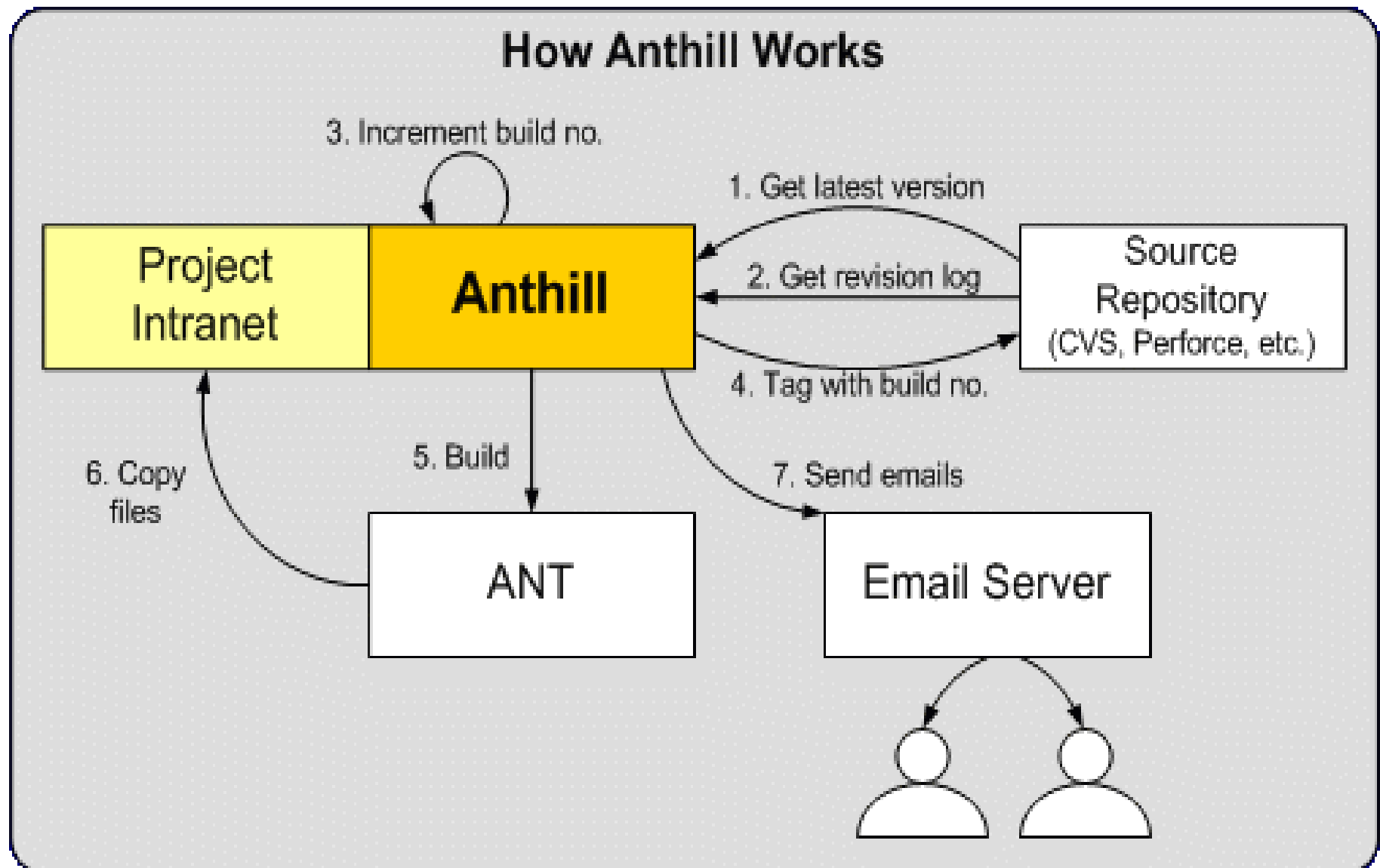
ID	#	Category	Severity	Status	Updated	Summary
0000132		security	tweak	new	02-03	Something that lets you do something you're not suppose to do.
0000130		bugtracker	block	new	02-02	Coucou
0000133	1	bugtracker	major	new	02-03	testing
0000129	2	test	major	feedback	02-02	Catalyst crash Switch 1
0000131		bugtracker	crash	acknowledged	02-03	testing summary
0000064	3	bugtracker	feature	assigned	01-02	user defined fields / projects
0000066	3	bugtracker	feature	assigned	12-27	change to filter select lists on viewview bug pages
0000081	2	bugtracker	feature	assigned	01-22	Remember Previous Entry

ATÉ QUE ENFIM...

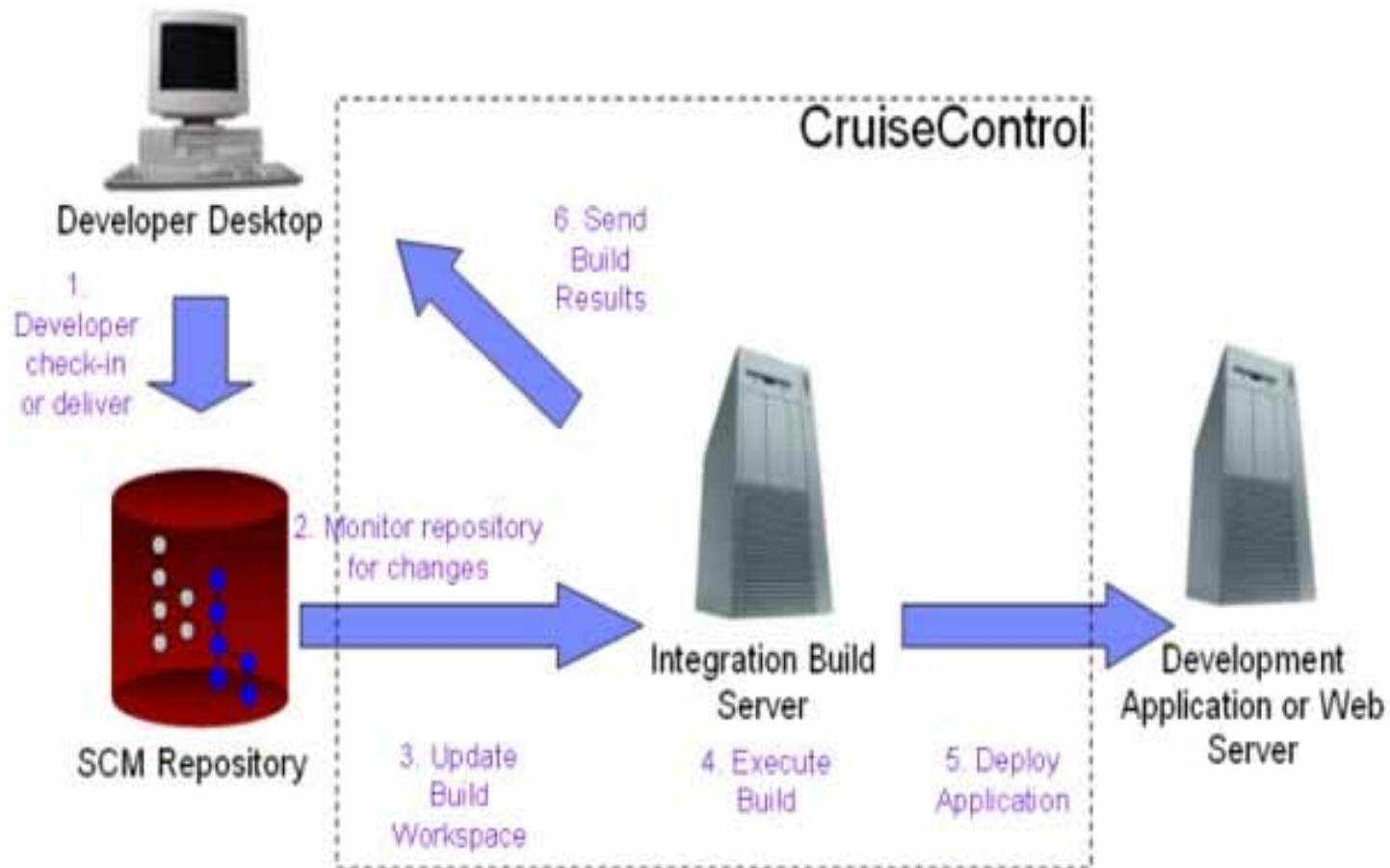
INTEGRAÇÃO CONTÍNUA

“O processo de integração contínua significa que todos os integrantes de um time atualizam suas modificações continuamente com testes que garantem que não quebram as funcionalidades já integradas.”

INTEGRAÇÃO CONTÍNUA



INTEGRAÇÃO CONTÍNUA



PROCESSO DE BUILD E DEPLOY

INTEGRAÇÃO CONTÍNUA - BUILDS



Current Build Started At:
02/28/2003 11:12:43

02/27/2003 16:50:21
(BuildNr. 24)

02/27/2003 15:49:13
(BuildNr. 23)

02/26/2003 22:54:20

02/26/2003 20:17:25

02/26/2003 18:15:28

02/26/2003 17:24:30

02/26/2003 17:05:26

02/26/2003 16:55:07

02/26/2003 16:53:04

02/26/2003 16:47:13

02/25/2003 20:43:18

02/25/2003 10:33:31

02/24/2003 20:12:02
(BuildNr. 22)

02/24/2003 10:42:46
(BuildNr. 21)

02/18/2003 13:54:08
(BuildNr. 20)

02/14/2003 18:41:59
(BuildNr. 19)

02/14/2003 17:40:01
(BuildNr. 18)

BUILD FAILED

Ant Error Message: file:D:/cruisecontrol/bugkilla/bugkilla/build.xml:469: No message

Date of build: 02/26/2003 17:24:30

Time to build: 59 seconds

Last changed: 02/26/2003 17:22:02

Last log entry: improved junit handling in cc-test all tests should run after failure of a single tes

Unit Tests: (106)

error testDeleteSequence

Unit Test Error Details: (1)

Test: testDeleteSequence

Type: java.lang.NullPointerException

Message:

```
java.lang.NullPointerException
    at de.oio.bugkilla.management.PersistenceManager.deleteTestSequence(PersistenceManager.java:100)
    at de.oio.bugkilla.management.TestPersistenceManager$DeleteTestSequence.run(TestPersistenceManager.java:100)
    at de.oio.bugkilla.management.TestPersistenceManager.testInPreparedEnvironment(TestPersistenceManager.java:100)
    at de.oio.bugkilla.management.TestPersistenceManager.testDeleteSequence(TestPersistenceManager.java:100)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:42)
```

Modifications since last build: (10)

modified dsohn /build.xml	improved junit handling in cc-test all tes
modified niete src/de/oio/bugkilla/recorder/RecorderState.java	add javadoc
modified niete src/de/oio/bugkilla/recorder/test/TestConnected.java	kicked out import of ValidSignal
modified niete src/de/oio/bugkilla/splitter/test/ServerSocketTestHelper.java	refactor BugkillaProperties.getInstance property files. and adding javadoc
modified niete src/de/oio/bugkilla/splitter/test/TestFilterManager.java	refactor BugkillaProperties.getInstance property files. and adding javadoc
modified niete src/de/oio/bugkilla/recorder/test/TestNotConnected.java	refactor BugkillaProperties.getInstance property files.

INTEGRAÇÃO CONTÍNUA – COMO INTEGRAR

Standup Meeting @ 9AM

Pair Up

Test

Q&A

Code

Refactor

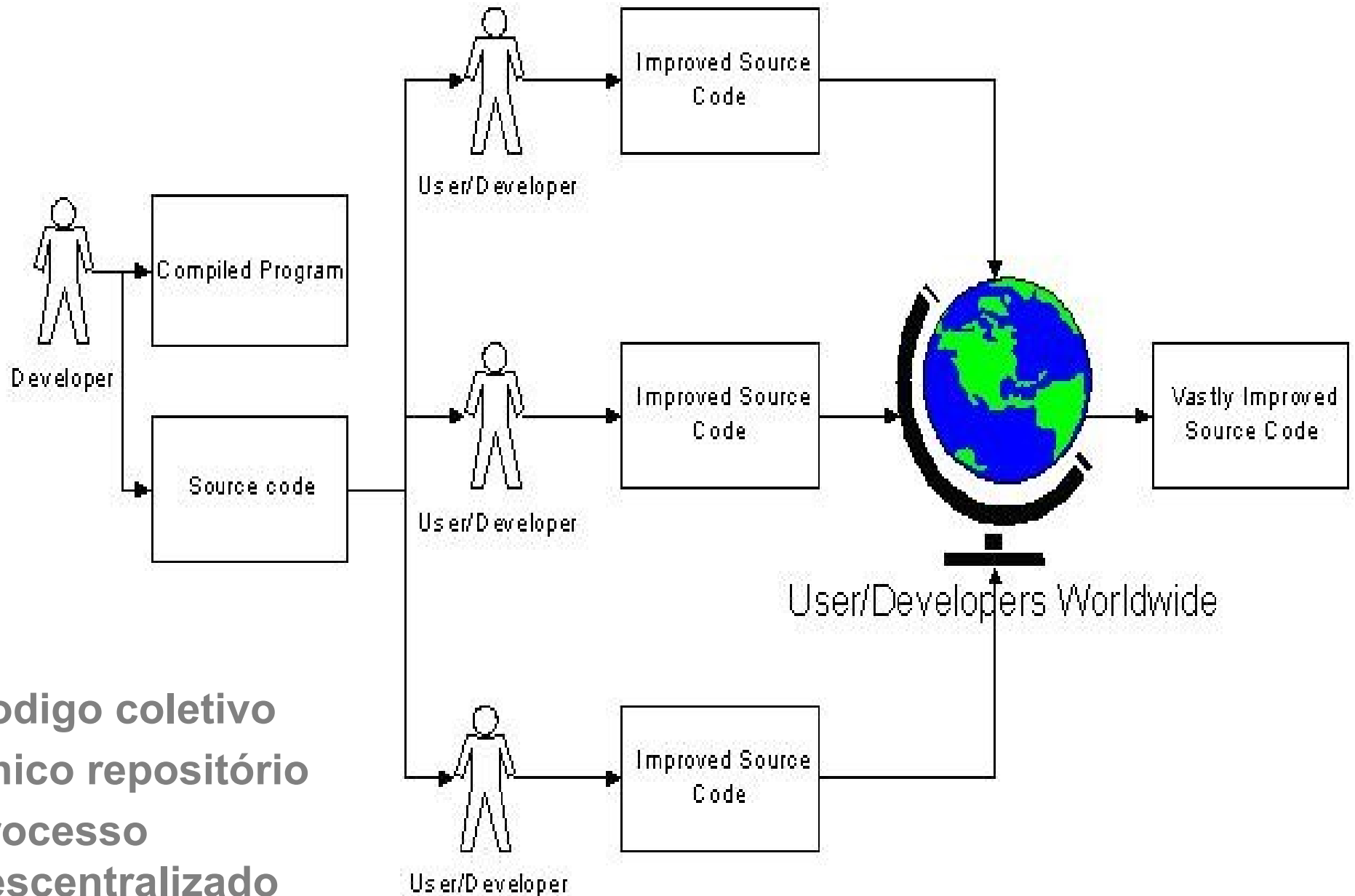
Integrate or Toss

Go Home @ 5PM

- Testar todos os componentes e módulos do sistema automaticamente.
- Fornecer relatórios de todas as atividades executadas.
- Construir e versionar.
- Integrar módulos e sistemas.
- Disponibilizar e homologar.
- Publicar em produção.
- Notificar participantes do projeto após publicação.

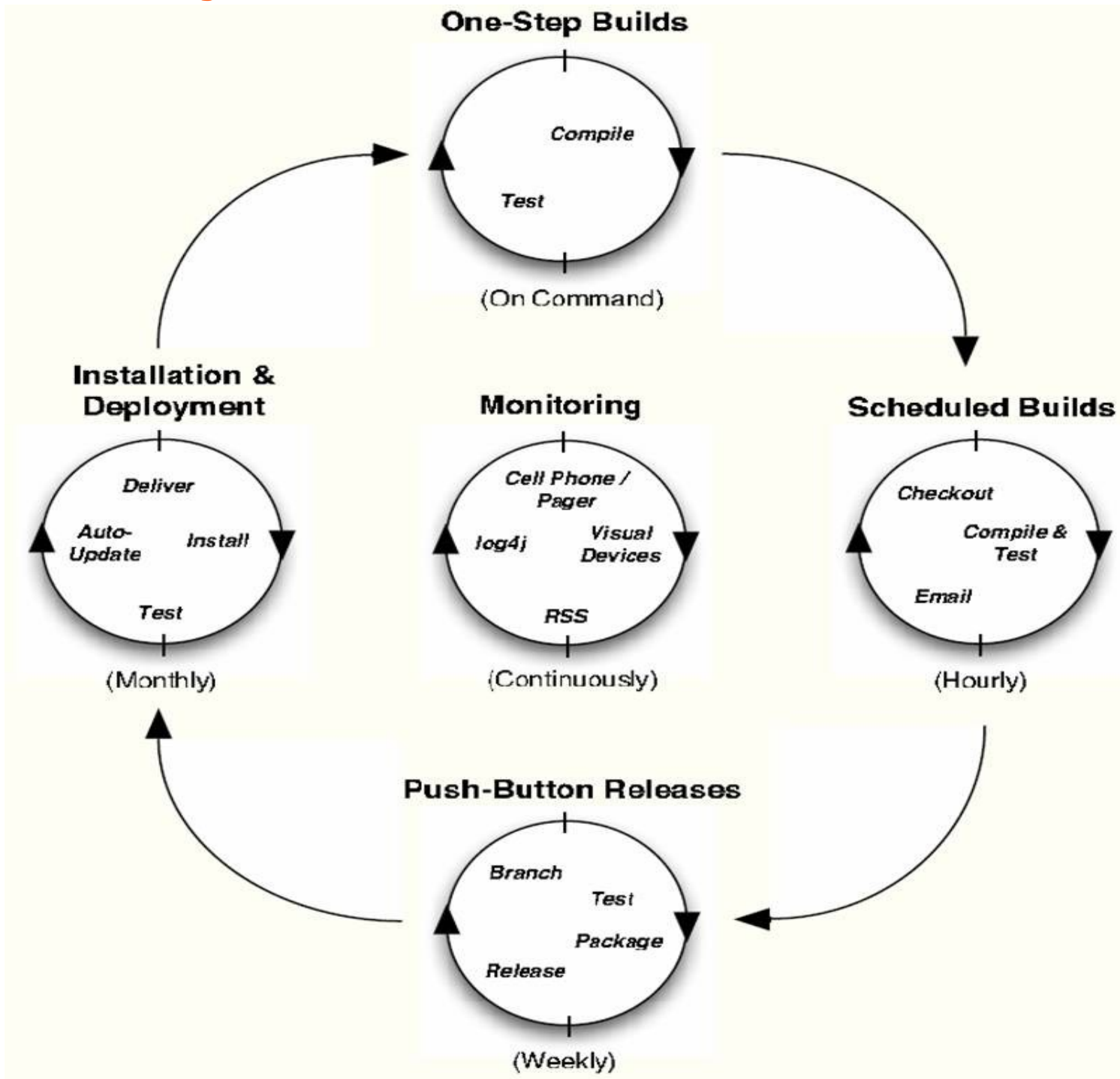
SÍNCRONA Vs ASSÍNCRONA

INTEGRAÇÃO CONTÍNUA ASSÍNCRONA



- Código coletivo
- Único repositório
- Processo descentralizado

INTEGRAÇÃO CONTÍNUA - ONE STEP BUILD



CONCLUSÃO

- **Problemas:**
 - **Custo em treinamentos específicos.**
 - **Esforço na adequação aos processos.**
 - **Sucesso do projeto fica atrelado a configuração correta das ferramentas.**
- **Soluções:**
 - **Agilidade nas mudanças.**
 - **Facilidade na comunicação do projeto.**
 - **Produtividade no ciclo de vida do produto.**
 - **Menor incidência de falhas com testes automáticos.**
 - **Riscos menores de descumprimento dos prazos.**
 - **Controle preciso no gerenciamento do projeto.**

REFERÊNCIAS

- Martin Fowler - <http://www.martinfowler.com/articles/continuousIntegration.html>
- Vinícius Manhães Teles - <http://www.improveit.com.br/xp/praticas/integracao>
- James Shore - <http://www.jamesshore.com/Blog/Continuous-Integration-on-a-Dollar-a-Day.html>
- Wikipedia - http://pt.wikipedia.org/wiki/Sistema_de_controle_de_vers%C3%A3o
- Jay Flowers - http://jayflowers.com/joomla/index.php?option=com_content&task=view&id=26
- IBM - <http://www-128.ibm.com/developerworks/rational/library/sep05/lee/index.html>