

Software Configuration Management

Web Site:

www.soebes.com

Blog:

blog.soebes.com

Email:

info@soebes.com

Dipl.Ing.(FH) Karl Heinz Marbaise

Agenda

1. Definitions
2. Role of Version Control
3. Definitions of Build-, Release-, Deployment- and Test Management
4. Version Control Definitions
5. Continuous Integration
6. Demo

1. SCM Definition

- In software engineering, **software configuration management (SCM)** is the task of **tracking** and **controlling changes** in the software. Configuration management practices include revision control and the establishment of baselines.

wikipedia: Software Configuration Management

1. SCM Definition

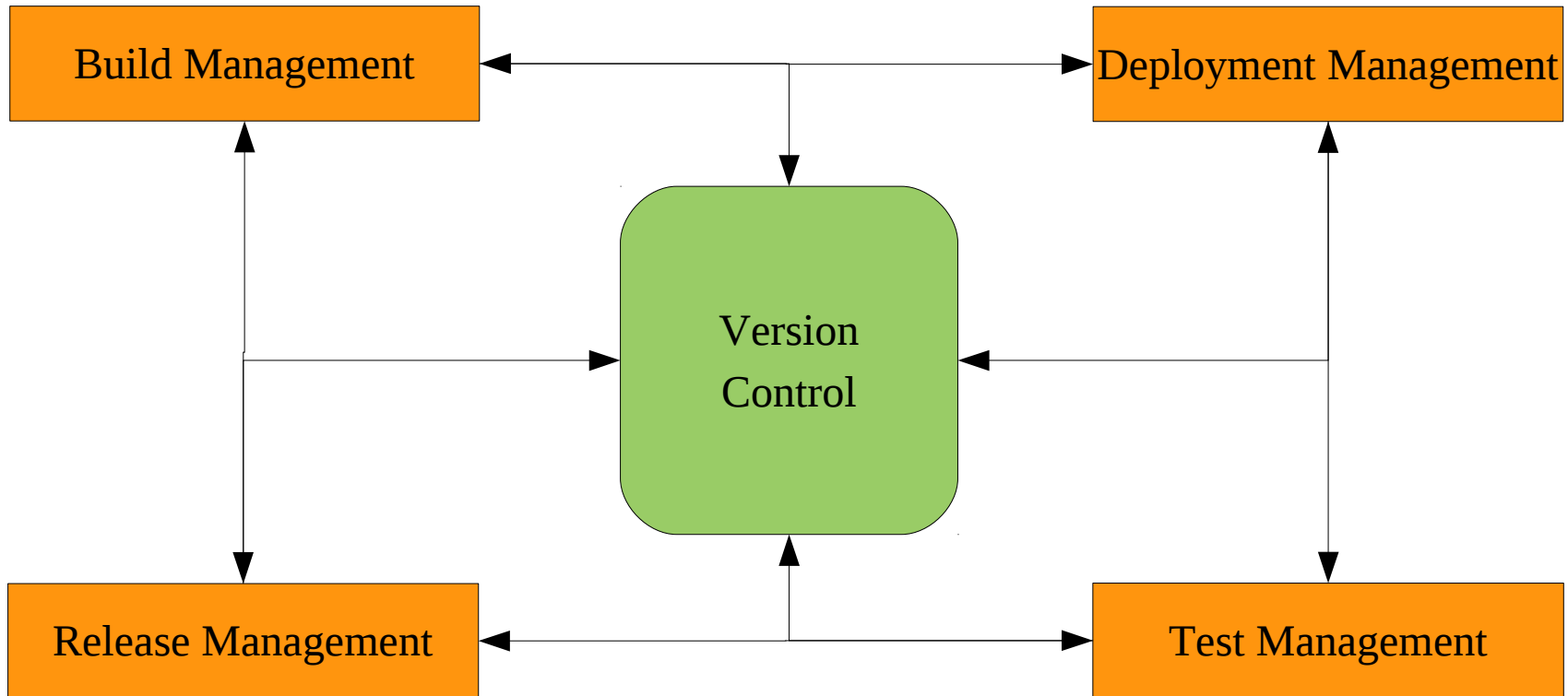
- The essence of the definition is:

REPRODUCIBILITY

TRACEABILITY

The result is usually summarized into a document called „Software Configuration Management Plan“.

2. Role of Version Control



3. Release Management Definition

- Release Management is the discipline to define a consistent state of your software while integration of bug fixes and/or features into a define release line.

http://en.wikipedia.org/wiki/Release_management

3. Build Management Definition

- Build means to create the runnable software from it's source code and it's components.

http://en.wikipedia.org/wiki/Build_management

3. Build Management Tools

- Open Source
 - Maven, Gradle, Ant + Ivy, Ant, SCons, CMake, Phing, Leiningen, BuildR etc.

3. Deployment Mgmt. Definition

- Deployment Management means to deploy a defined state of the software into dev, test, pre-live and production environment

Software Deployment
Application Lifecycle Management

3. Deployment Mgmt. Tools

- RPM's, packages
- Capistrano (RoR)
- Chef (RoR)

Ergänzungen:

Puppet

3. Test Management Definition

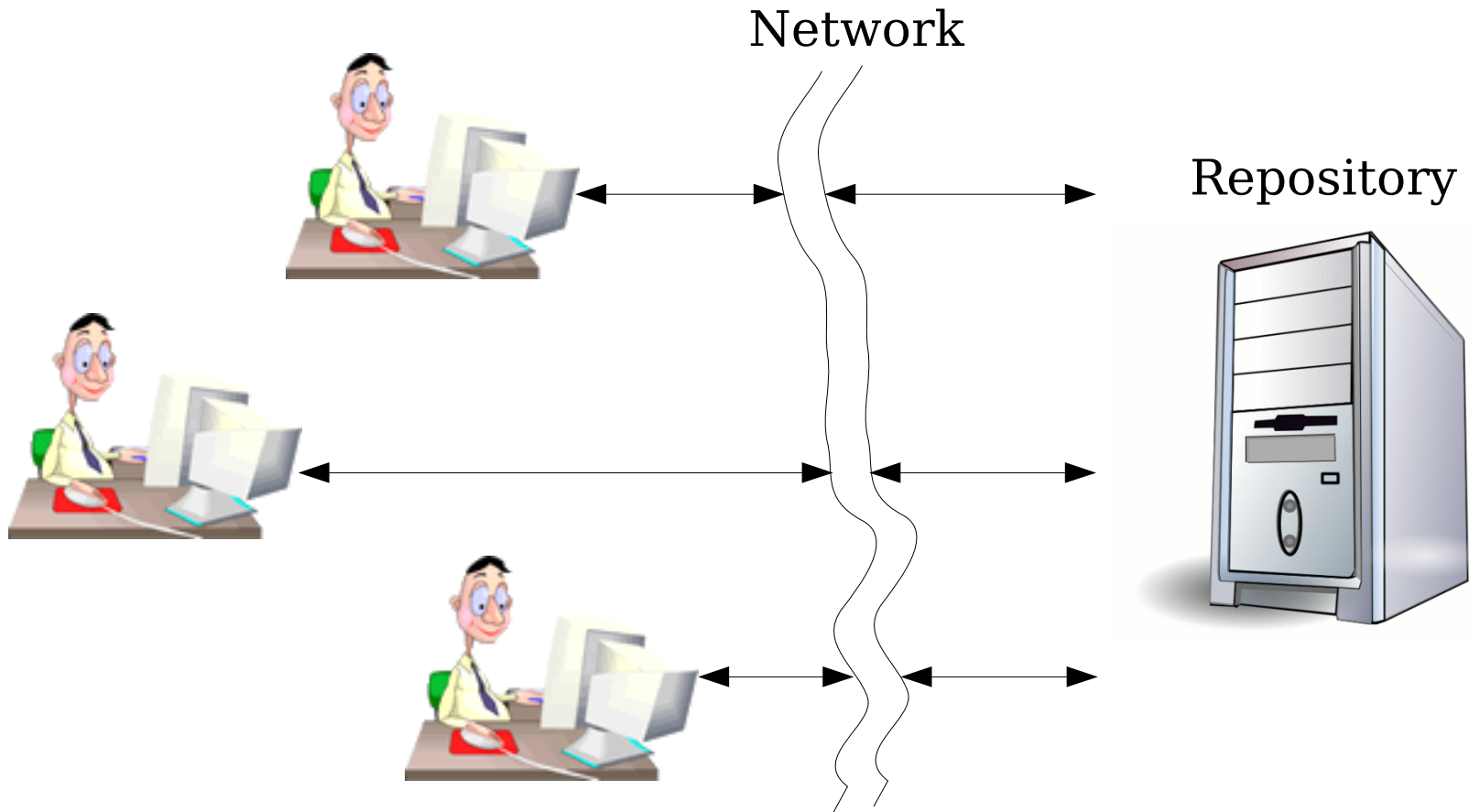
- “Test management is the activity of managing some tests. [...] Test management tools often include requirement and/or specification management modules [...]”

source: http://en.wikipedia.org/wiki/Test_management

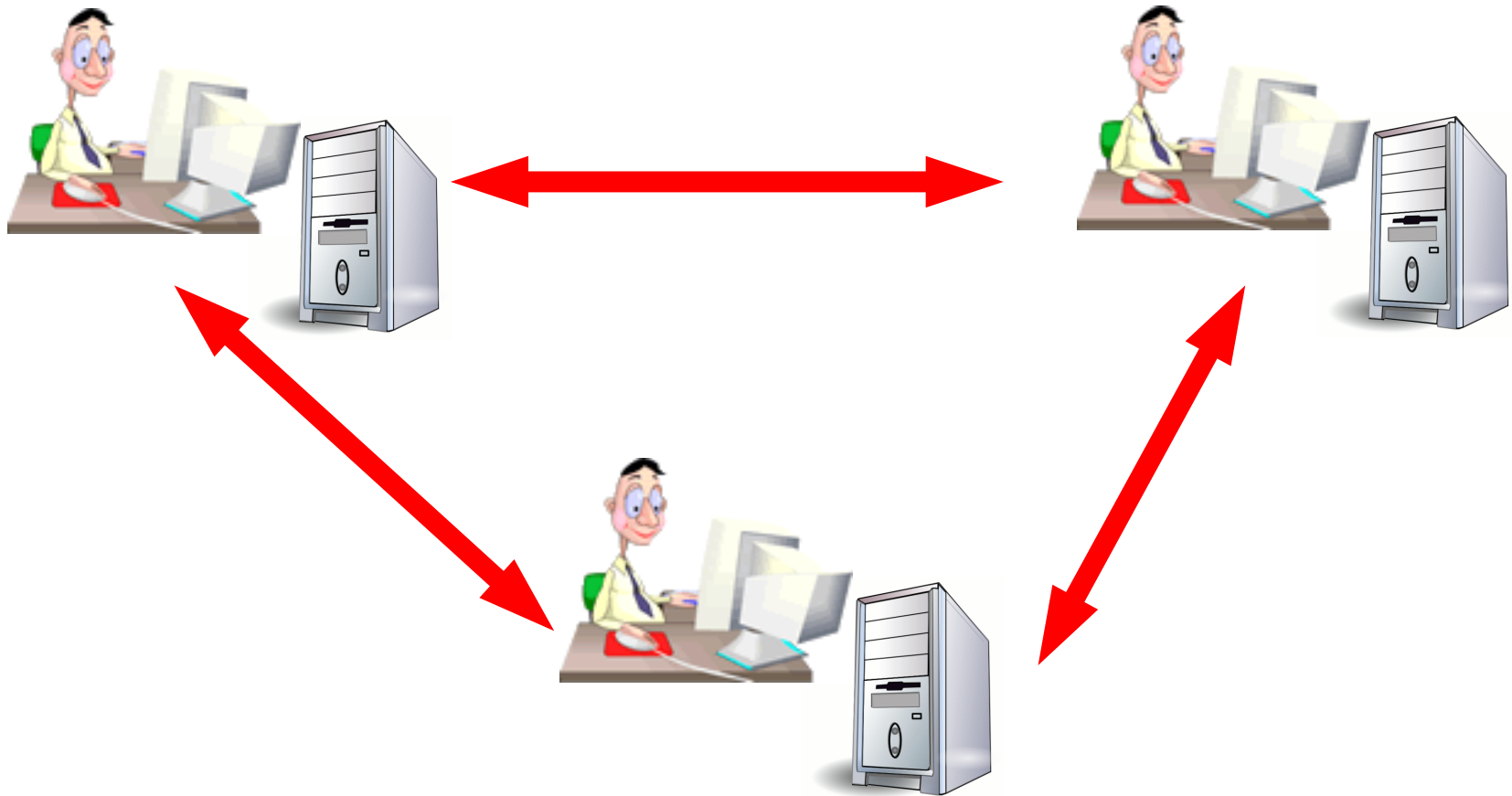
4. Version Control Definitions

- Centralized Version Control System (CVCS)
 - Sometimes called centralized systems.
- Distributed Version Control System (DVCS)
 - Sometimes called decentralized systems.

4. Version Control CVCS



4. Version Control DVCS

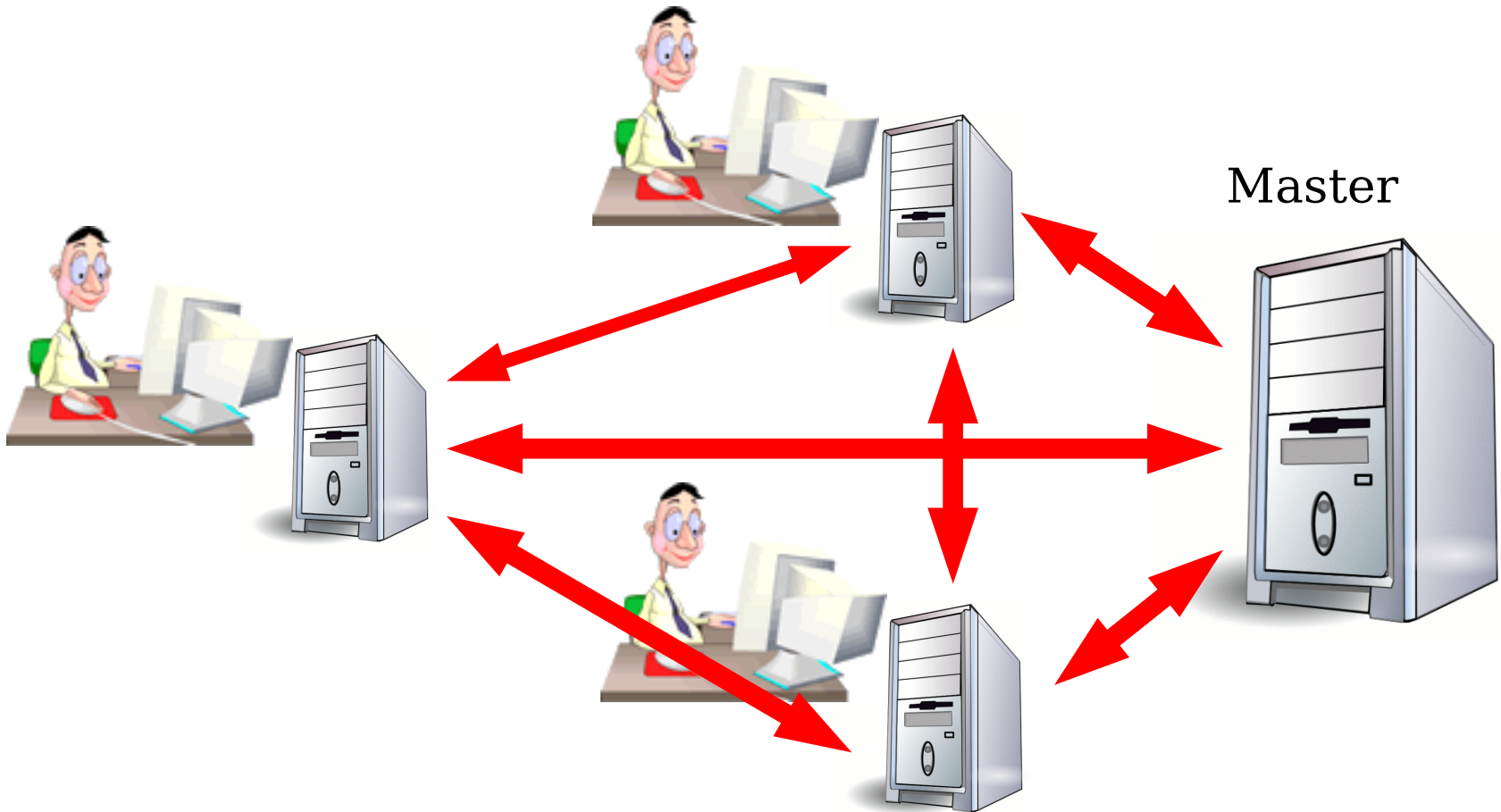


4. Version Control

DVCS

- Based on practical reasons we need a „centralized point“ in decentralized systems.

4. Version Control DVCS (hands-on)



4. Version Control

DVCS

- Open Source
 - git, mercurial, monotone, bazaar, darcs, SVK etc.
- Commercial
 - BitKeeper

4. Version Control CVCS

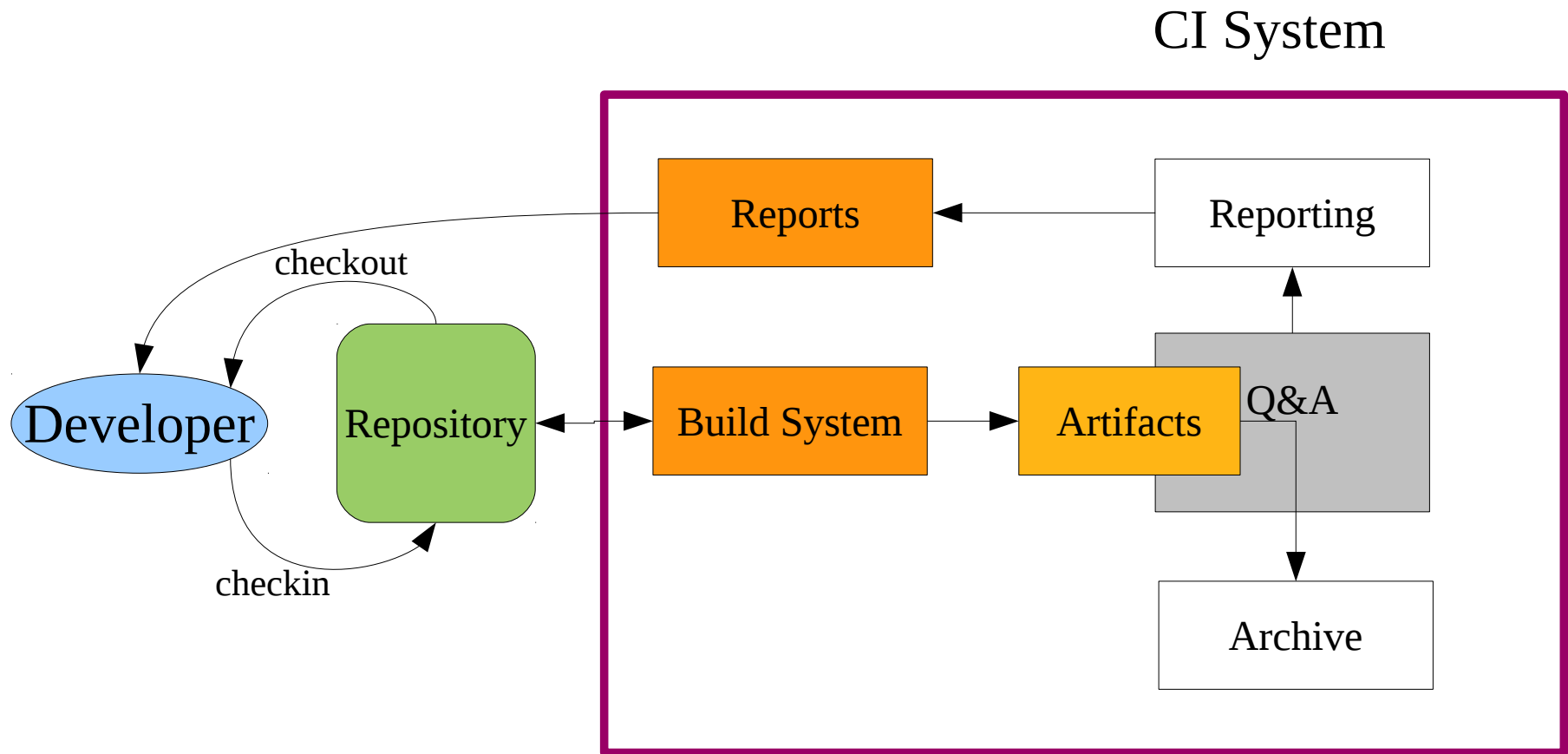
- Open Source
 - Subversion
 - CVS
- Commercial
 - Perforce
 - ClearCase
 - etc.

5. CI Definition

- Continuous integration is the process of assembling software everytime code changes. CI Helps to „Integrate Early, Integrate Often“ reducing the lifespan of a defect.
- Compiling, linking, running your unit tests, install the software and run integration tests.

<http://www.slideshare.net/thyagk/continuous-integration-system>

5. Setup for CI



5. Demo

- Java Project
- Maven as build tool
- Jenkins as CI
- SVN as Version Control Tool

On-line Sources

- DVCS
 - Git <http://git-scm.com>
 - Bazaar <http://bazaar-vcs.org/en>
 - Mercurial <http://mercurial.selenic.com/wiki>
 - Darcs <http://www.darcs.net>
 - SVK
<http://svk.bestpractical.com/view/HomePage>
 - Monotone <http://www.monotone.ca>
 - BitKeeper <http://www.bitkeeper.com/>
(commercial)

On-line Sources

- CVCS
 - CVS (Concurrent Versions System) (wikipedia)
 - Subversion
 - IBM ClearCase (commerical)
- Articles
 - <http://www.infoq.com/articles/dvcs-guide>

On-line Sources

- Maven
 - [Maven Build Tool](#)
 - [Maven](#)
 - [Jenkins CI Tool](#)
 - [Hudson CI Tool](#)
- Continuous Integration
 - [Continuous Integration](#)
 - [CI Best Practice](#)
 - [Sonar CI](#)

On-line Sources

- Build Tools
 - <http://www.gradle.org>
 - <http://buildr.apache.org>
 - <http://cmake.org/>
 - <http://www.scons.org/>
 - etc.

On-line Sources

- Deployment
 - <http://wiki.opscode.com/display/chef/Home>
 - <https://github.com/capistrano/capistrano/wiki>
- Continous Deployment
 - [Five Easy Steps](#)
 - [Continous Deployment \(german\)](#)
 - [Continous Deployment: Easier Said Than Done](#)

Questions?

jugf2011@soebes.com

Thank you for your attention.