IRISA

Institut de recherche
en informatique et systèmes aléatoires

Using GIT with Kerrighed project

Kerrighed Summit '07

February 2nd 2007

Jean Parpaillon <jean.parpaillon@irisa.fr>

Table of contents

- Kerrighed SCM
- Subversion
- GIT
- GIT with Kerrighed
- References

Kerrighed SCM - Facts

- One official repository
  - http://scm.gforge.inria.fr/svn/kerrighed
- Kerrighed module, tools, libs
  - 5 development branches
- Kerrighed kernel patch
  - 5 development branches
  - distro kernels
  - w/o KDB
  - Linux vanilla

Kerrighed SCM - Issues

- Development branches hard to merge (scripts)
- Flavours
  - Hard to maintain
  - Increasing number (distro support)
- No Linux upstream merging

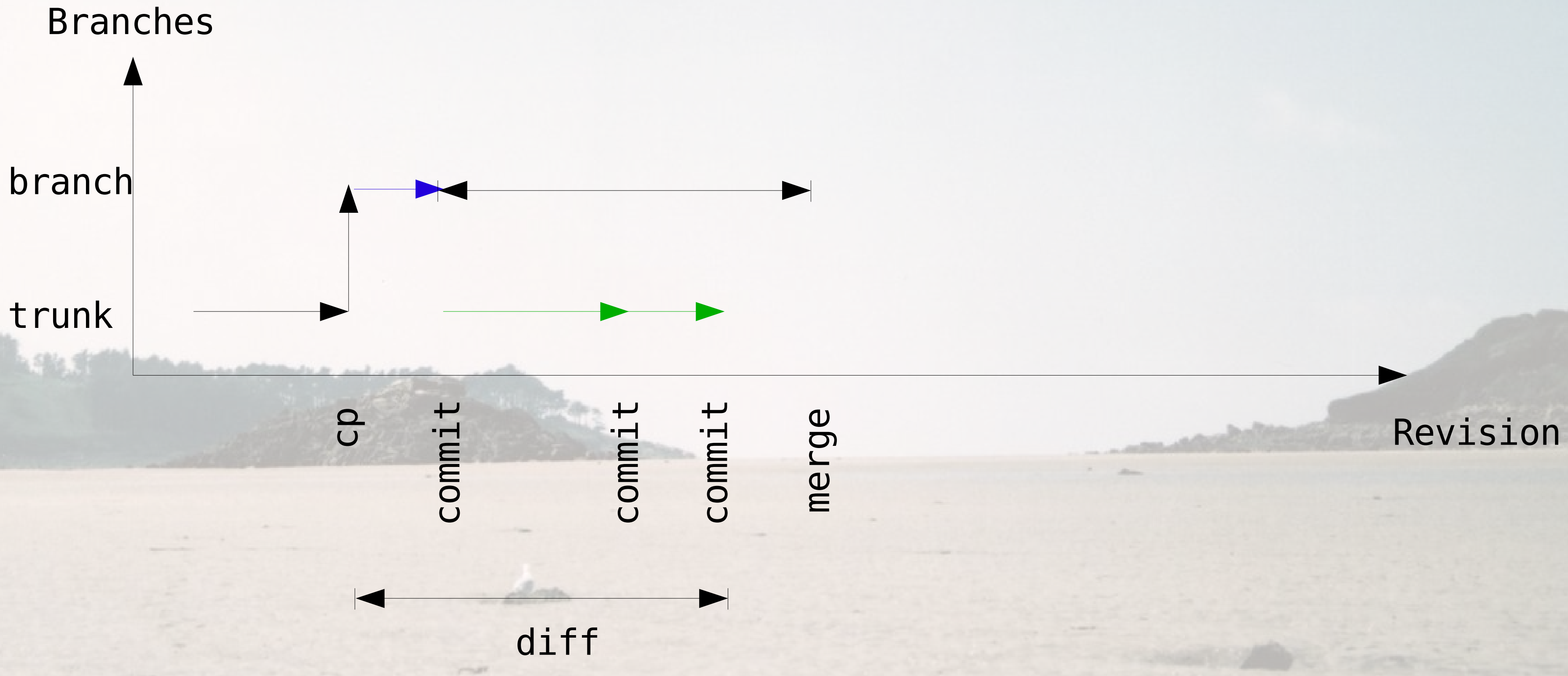
- Is SVN the best solution for the Kerrighed project ?

Subversion – Concepts

- Began in 2000
- Another implementation of CVS model
  - Centralised repository
  - Release based
- Differences
  - Versioned meta-datas and directories
  - Atomic commits (by date with CVS)
  - `rename` and `mv`

Subversion – Branches

- Easy branching, hard merging



- Need to know branching revision
- Don't keep track of changelog
- No branch semantic
  - Committing in a branch increase revision number

Subversion – Access and tags

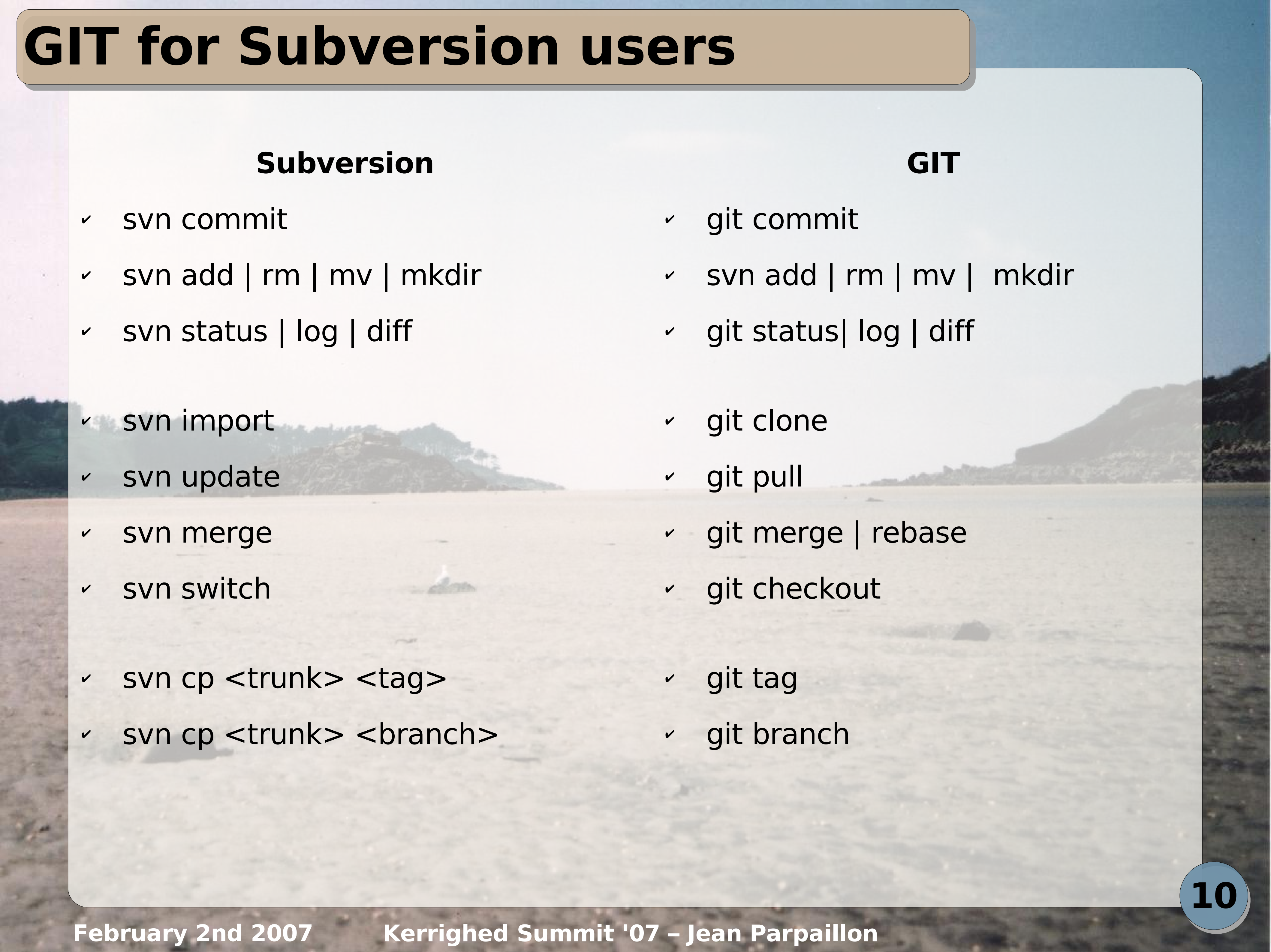
- One repository
- Basic access configuration (read/write)
  - No developper classes
- Very hard to follow upstream devels with read access
- No tag semantic
  - Tag can be different from a version to another

GIT - Concepts

- Decentralized
  - Each repository is a branch
- Semantic
  - a *tree* represents a directory
  - a *commit* represents a *tree*, the preceding *commit(s)* and a message
  - a *tag* is an alias for a commit
- A *commit* is designed by its checksum
- A *commit* can have several parents
  - Represents a merge of 2 development lines

GIT - History

- Originally a very basic tool
  - *cogito* written as a set of shell scripts above *git*
- Now *git* can be used alone
- Used for *Linux* project
  - Very decentralized



GIT for Subversion users

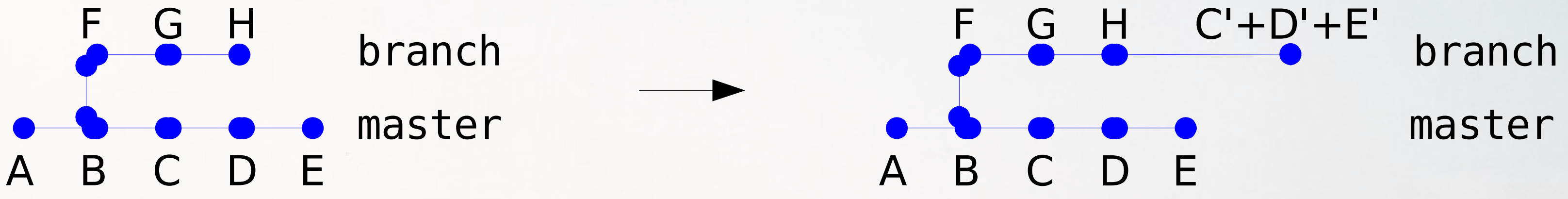
Subversion	GIT
svn commit	git commit
svn add | rm | mv | mkdir	svn add | rm | mv | mkdir
svn status | log | diff	git status| log | diff
svn import	git clone
svn update	git pull
svn merge	git merge | rebase
svn switch	git checkout
svn cp <trunk> <tag>	git tag
svn cp <trunk> <branch>	git branch

GIT – Branches (1/3)

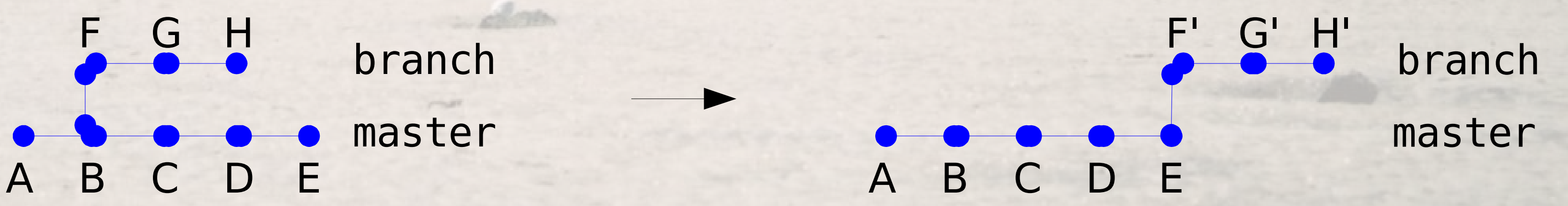
- trunk = master
- A local repository is a branch
- `git clone`
  - Clone a remote repository
  - Exception: remote *master* = local *origin*
  - ~ `svn checkout`
- `git pull`
  - Fetch updates from remote repository
  - ~ `svn update`
- `git push`
  - Upload objects to remote repository
  - ~ `svn commit`

GIT – Branches (2/3)

- Creating a local branch: `git branch <name>`
- SVN-like merge: `git merge`

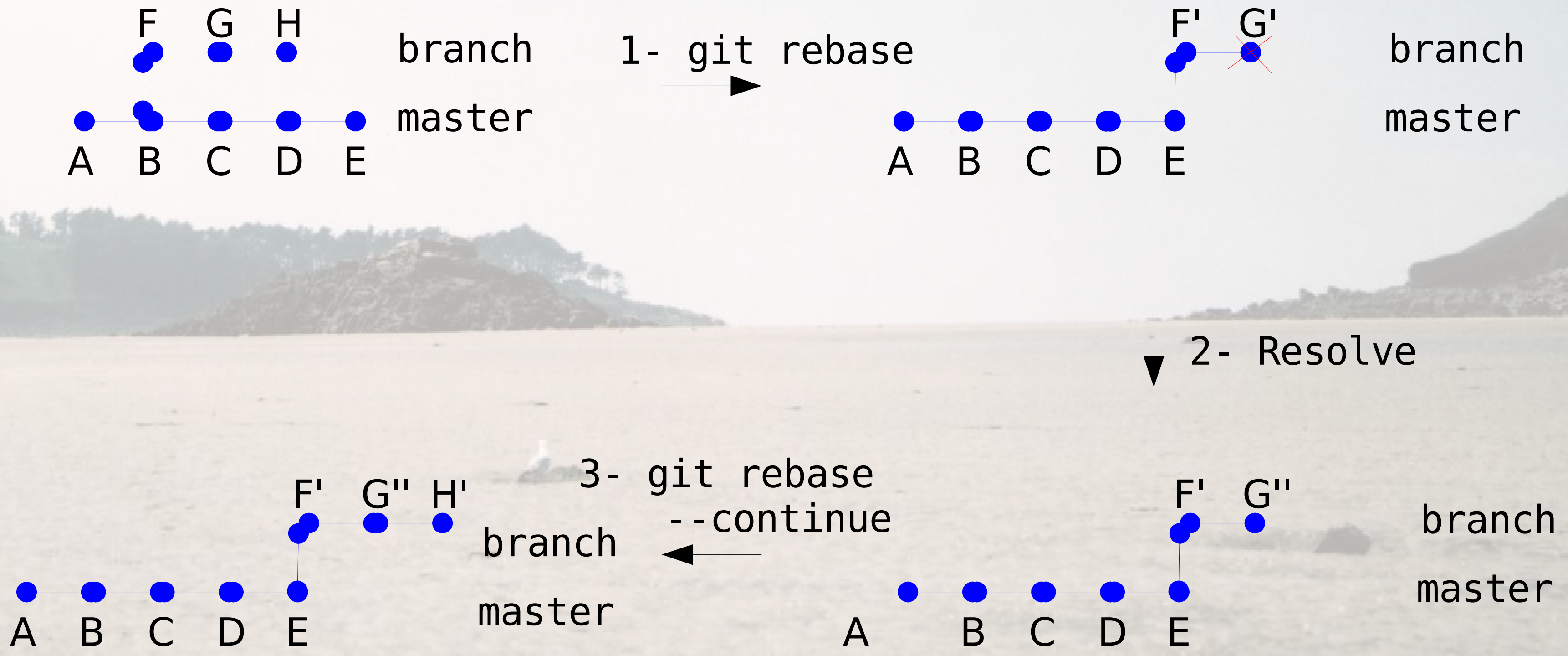


- Merge++: `git rebase`
  - Keep patches historic



GIT – Branches (3/3)

- Rebase conflict management



1- git rebase

2- Resolve

3- git rebase --continue

GIT – Repositories access (1/2)

- ✓ Decentralized SCM allows centralized development
- ✓ Centralized SCM hardly allows decentralized development
- ✓ GIT offers centralized accesses

GIT – Repositories access (2/2)

- git-daemon
  - git://
  - rsync://
- gitweb
  - http://
- Examples:
  - git-http-pull http://rsync.kernel.org/pub/scm/git/git.git/
  - git-clone git://www.kernel.org/linux/kernel/git/torvalds/linux-2.6.git

GIT – SVN import

- SVN import
  - Import *trunk branches* and *tags*
  - Keep synchronization with SVN

GIT – Bisect

- Find a breaking commit by binary search

```
(a)git bisect start
(b)git bisect bad
(c)git bisect good <good_commit_id>
  # Checkout the commit between these 2
(d)git bisect good
(e)git bisect good
(f)git bisect bad
  # etc.
```

GIT – Other features

- Allows hook scripts
- gitk: git GUI
- Create mails with patches/apply patches from mail
- git archive: create archive from tree

GIT with Kerrighed

- Possible main repository
- ...but local developments stay locals
- Many tools to:
  - apply patches
  - look for breaking patches
  - merging branches
  - syncing branches (rebase)
- Interfaces: web, tk
- SVN synchronisation
- Linux development done with git

References

- GIT online documentation
  - `http://www.kernel.org/pub/software/scm/git/docs/tutorial.html`
  - `/usr/share/doc/git-doc/index.html` (Debian)
  - git help <command>
- Everyday GIT With 20 Commands Or So
  - `http://www.kernel.org/pub/software/scm/git/docs/everyday.html`
  - `/usr/share/doc/git-doc/index.html` (Debian)
- Les nouveaux systèmes de gestion de version, *Stéphane Bortzmeyer*, http://2005.jres.org/paper/2.pdf