L3 Mention Informatique
Parcours Informatique et MIAGE

Génie Logiciel Avancé

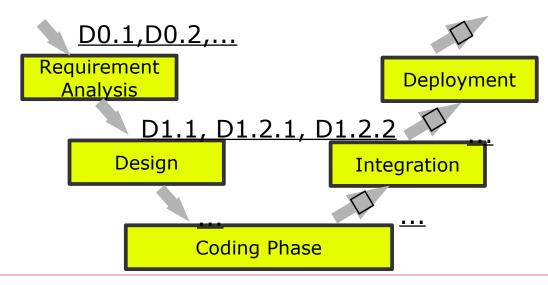## Part IV : Version and Configuration Management

Burkhart Wolff

wolff@lri.fr

Plan of the Chapter

- ❑ Motivation: Version and Configuration Management as „the" means for collaborative development

- ❑ Version management
  - ➢ Centralized Version Control
  - ➢ Distributed Version Control
  - ➢ Organizing Merges

- ❑ Beyond Versions: Configurations
  - ➢ Build Management
  - ➢ Advanced Configuration Management

Motivation

- Recall: SE Processes are based
  - on a large flow of documents and code
  - ... that have to be edited collaboratively
  - ... distributed consistently
  - ... while controlling access

D0.1,D0.2,...

| Requirement Analysis |

| Deployment |

D1.1, D1.2.1, D1.2.2 ...

| Design |

| Integration |

...

| Coding Phase |

...

Motivation

- Important TECHNICAL means to support the process
  - Explicit tools for Version Management
    (CVS, svn, git, mercurial, sourcedepot, ...)
  - Create and track revisions of files and file-trees
  - Create and track differences between files and file-trees
  - Acces control to the various parties of process
  - Locking of documents
  - Merging of revisions
  - Quality control
  - . . . actually, it is dead-useful for everything !!!

Concepts of Centralized Version Control

- Working copises
  (in user space)

- Repository
  (on the server-side)

- update:
  syncing with the
  repository

Centralized version control

Server

Repository

commit   update   commit   update   commit   update

Working copy   Working copy   Working copy

Workstation/PC #1   Workstation/PC #2   Workstation/PC #3

- commit: creating a new revision of a document
  (involves new registration, inclusion in documents,
   consistency checks)

- operations lock, checkout, import, ...
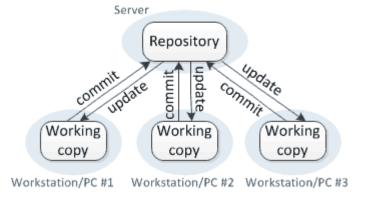
Concepts of Centralized Version Control

- ❑ First widely used system: CVS

- ❑ Nowadays in use : svn

- ❑ In connection with a gui-client: useable for end-users ... ... for everything ...

- ❑

![SUBVERSION®]

https://subversion.apache.org/

Centralized version control

Server

Repository

commit · update · commit · update · update · commit

Working copy · Working copy · Working copy

Workstation/PC #1 · Workstation/PC #2 · Workstation/PC #3

... for everything ...

□ my working
copy for
this course
material
...

□ Version -
management
is not just
for code
...

One step further:
Distributed Version Control

- Hierarchy of repositories:

Centralized version control

Distributed version control

- one more sync-level, but more precise history in practice... since everybody can check in locally
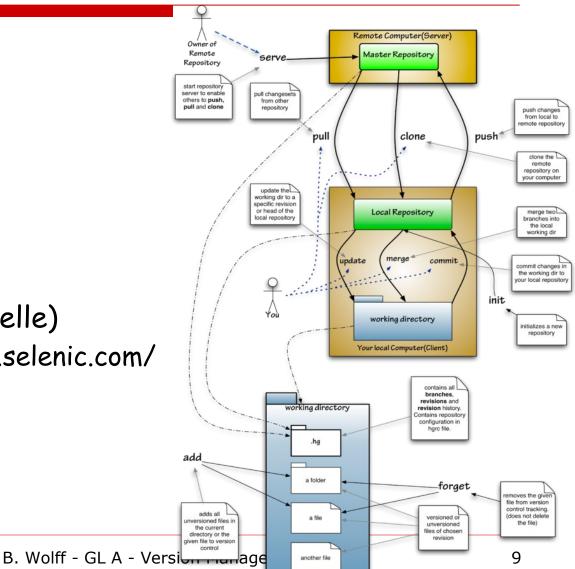
- hierarchy strictly speaking not necessary

Distributed Version Control Systems

- ❑ Opensource:
  - ➢ git (Linux)

  - ➢ mercurial (Isabelle)
    http://mercurial.selenic.com/

Consequences for a development process
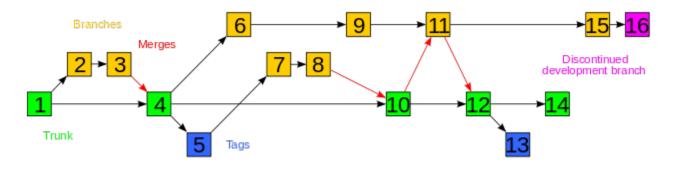
❑ The entire development becomes visible, trackable, reproducible, and can be an object in its own right (if formal): apply changeset XYZ ...

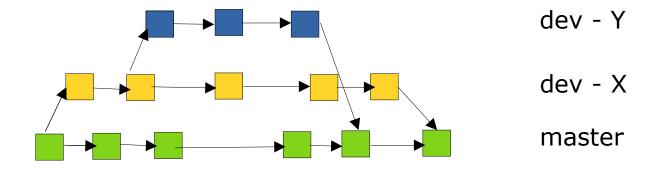# Problems of a distributed development

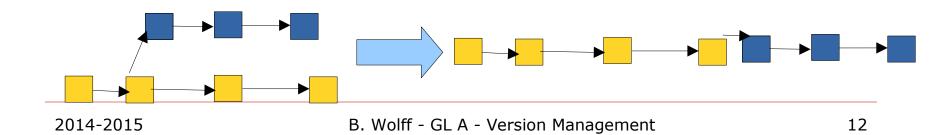- ❑ If not synced via explicit locks,
  a development looks like this:

- ❑ Merging: For binary formats, only special purpose merges work (as in MS Word, for example ...)
- ❑ For textual formats :

Problems of a distributed development

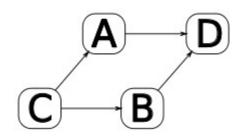- Pragmatics: Merging Strategy
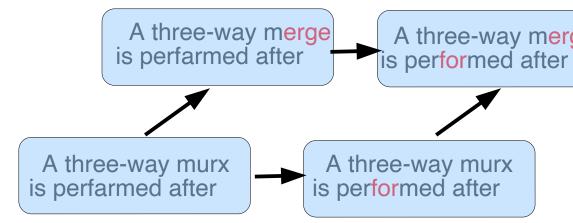
dev - Y

dev - X

master

- Rebase Strategy:

A partial solution ...

❑ ## For textual formats:
A three-way merge is performed after an automated difference analysis between a file 'A' and a file 'B' while also considering the origin, or common ancestor, of both files. It is a rough merging method, but widely applicable since it only requires one common ancestor to reconstruct the changes that are to be merged.
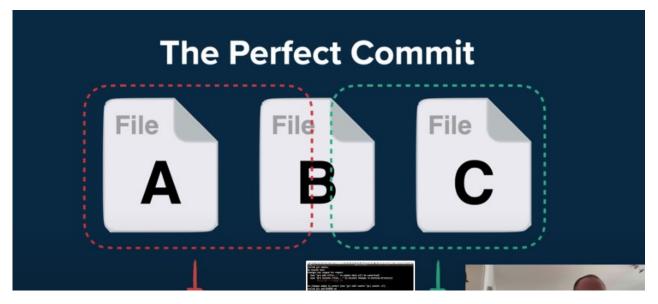

A three-way merge is perfarmed after

A three-way merg is performed after

A three-way murx is perfarmed after

A three-way murx is performed after

Tools: For example xmerge, which also offer conflict resolution by hand ...

More on Git

- More : partial adds , partial commits

- Youtube on git:

    https://www.youtube.com/watch?v=Uszj_k0DGsg

A better solution ...

- [ ] In a distributed development process, where a very large number of merges occurs routinely, the validation of the intermediate results becomes crucial.
  (This limits the use of informal documents.)

  - > validation on any check-in
    (for example: automated type-checking)

  - > validation for UML - documents
    (self-defined consistency checkers for UML models)

  - > automated static analysis and tests during systematic and periodic builds ...

  - > ... more advanced methods, using proof techiques.

Build Management: A Build-Server

Build Management: A Build-Server

Towards Configuration Management

- ❑ ... generalizes Version-Management
  - ➢ by configuration descriptions (including functionality environment, hardware,...)
  - ➢ attempts to GENERATE a revision on the basis

    of meta-data over dependencies and change-sets
  - ➢ works with heavy virtualization techniques nowadays (Google, Microsoft)