AARHUS UNIVERSITET

Software Engineering and Architecture

Software Configuration Management (SCM)

Motivation

AARHUS UNIVERSITET

Motivation: Collaboration

- SCM helps to solve two *very nasty problems* in dev
  - **Team collaboration** on large code bases
  - **Releasing** large code bases **to customers**
- War stories from my dark past in the mid 1990'ies
  - SAWOS 'collaboration' on 150.000 lines of C++ code
    1. Each developer had a copy ☹☹☹
    2. All source files on central file server ☹☹☹
       1. *Inverted race problem – save **last,** for god's sake!*
    3. Manual locking ☹☹☹
       1. Thumbtacks in colors on each filename on whiteboard
    4. Daily Split-Merge cycle ☹
       1. Split in morning – merge manually in afternoon

AARHUS UNIVERSITET

Motivation: Release

- SAWOS release management
  - We made a full copy as ZIP file, copied that to a diskette, marked with revision number and date, and put into the company safe.

And used ‘son-father-grandfather’ backup strategy

  - (This actually worked quite well!)

AARHUS UNIVERSITET

Motivation

- So – we were idiots or what?

- Not quite
  - I worked in 1992 in Horsens
  - CVS was published in 1991 – first commonly used SCM tool
    - And our 20-person company in Horsens did not read papers
  - No internet to search!
    - Question? Send fax to MicroSoft in Stockholm and never get answer
  - No SCM teaching at university at all!

AARHUS UNIVERSITET

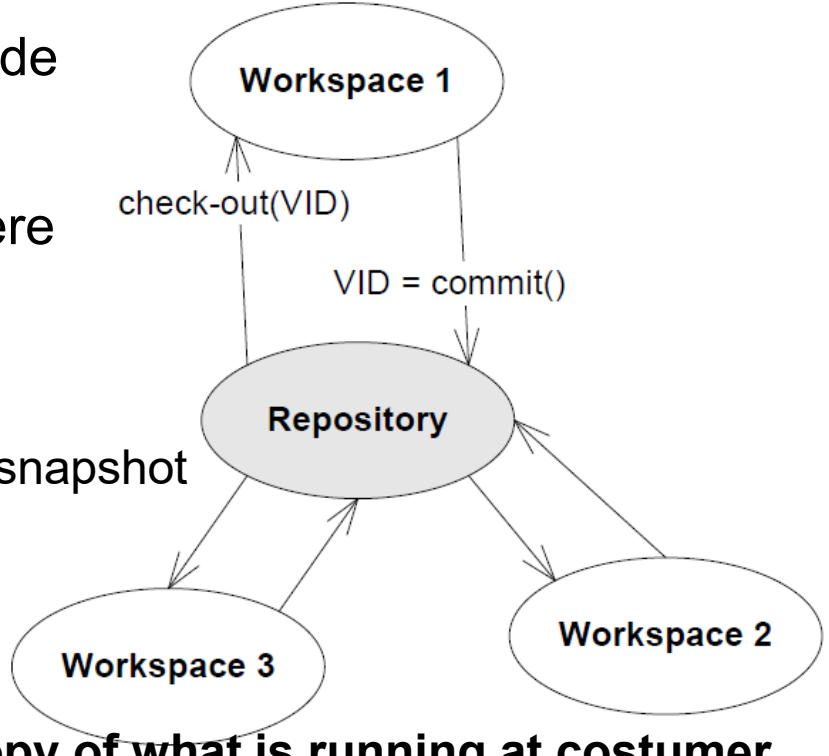
SCM in 2½ Slides

Read the book
or
Watch the Screencasts

AARHUS UNIVERSITET

SCM Versioning

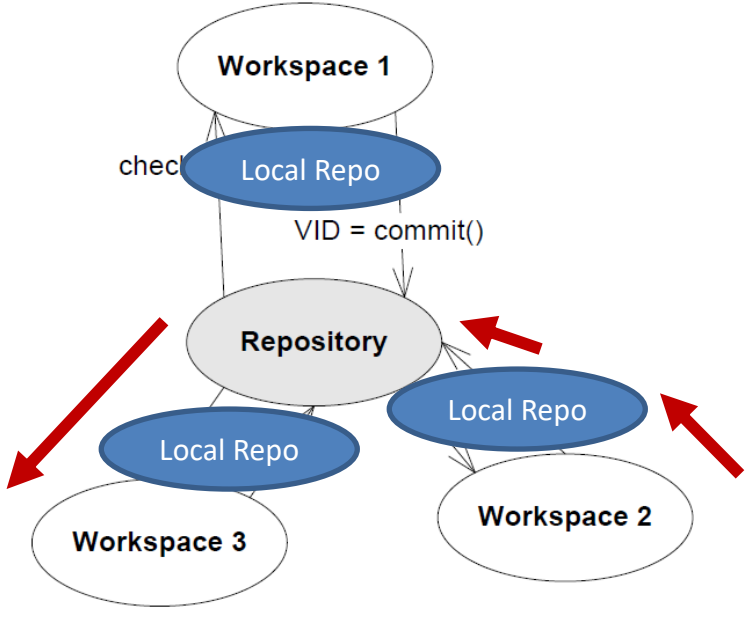
- Repository = Database
  - Stores all **versions** of all your code
- Workspace = Folder structure
  - Each developer do daily TDD there
- **Operations**
  - Commit (check-in)
    - Store new version of everything/snapshot
  - Check-out
    - Retrieve specific version



- **Release management = Get copy of what is running at costumer**

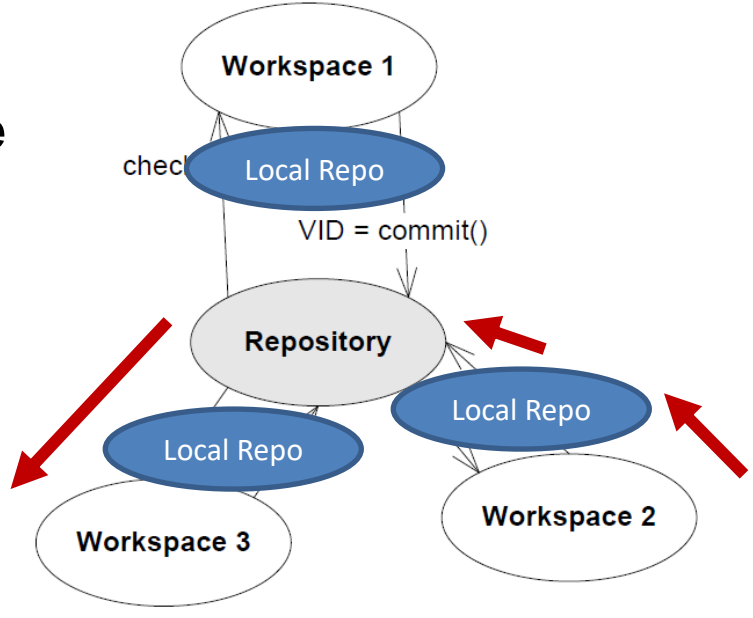
Git…

- Git's model is more complex ☹
  - Repositories form a *chain (workspace -> local -> Origin)*
- *Git commit –a –m "log msg"*
  - *Check-in into **local repository***
    - *Stored on **my** local harddisk*
- *Git push*
  - *Copy version to **origin repository***
    - In our case, AU GitLab
- *Git pull*
  - Copy version from origin to local *and* workspace



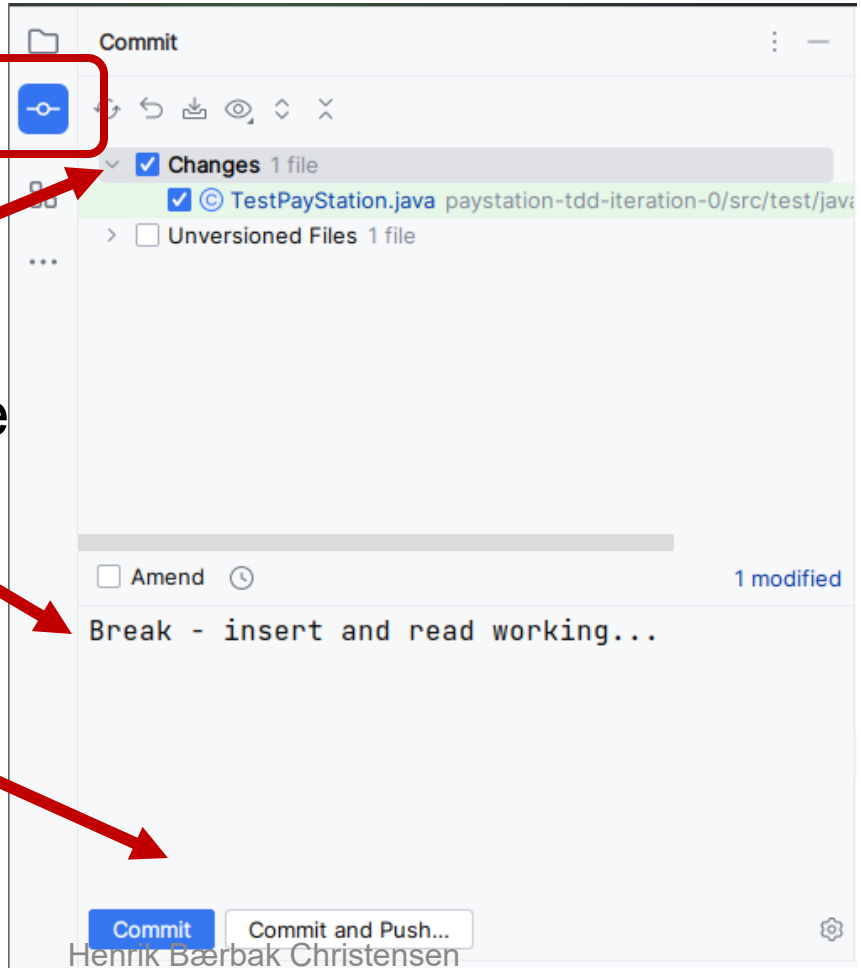
Git…

- So – to collaborate, Arne and Bente
- Arne does work on HotStone on his computer (workspace)
- Arne then **commits** and **pushes**
  - **The push is important! Otherwise the new version is only on Arne's computer**
- Bente then **pulls**
  - … to get Arne's work into her own workspace



Or… In IDE

- Click to set the ‘staging area’
- Enter log message
- Commit
  - And push!



SCM Collaboration