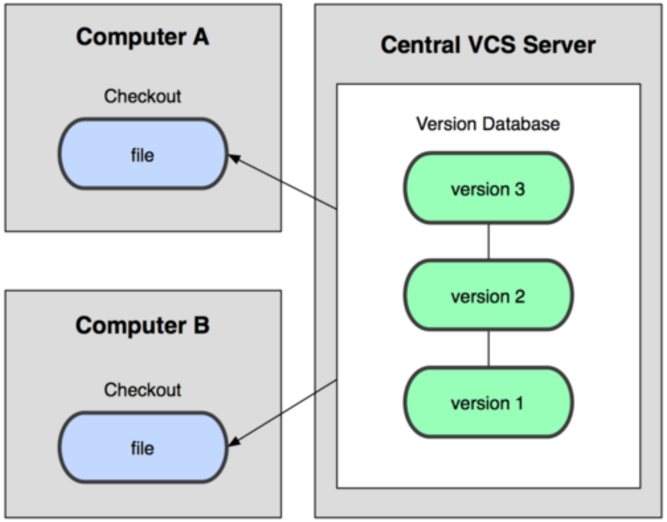
How to Git CHARA software development under control

Brian Kloppenborg
Georgia State University
bkloppenborg@gsu.edu

Status quo

CVS and email

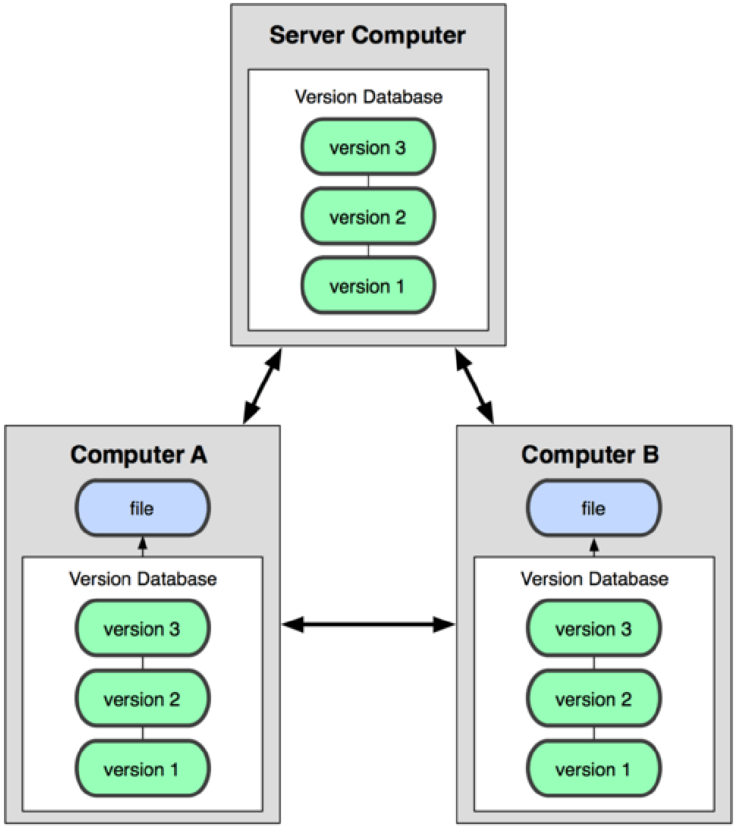
- A single CVS repository
- Bugs/features via. email
- Documentation in man pages



Problems:

- Network access required
- Repository not redundant
- Single branch development
- High level of interdependency
- No bug/issue/feature tracker
    - Design decisions hidden in email
    - No clear development plan or time line
- Documentation difficult to update

Git: Distributed version control

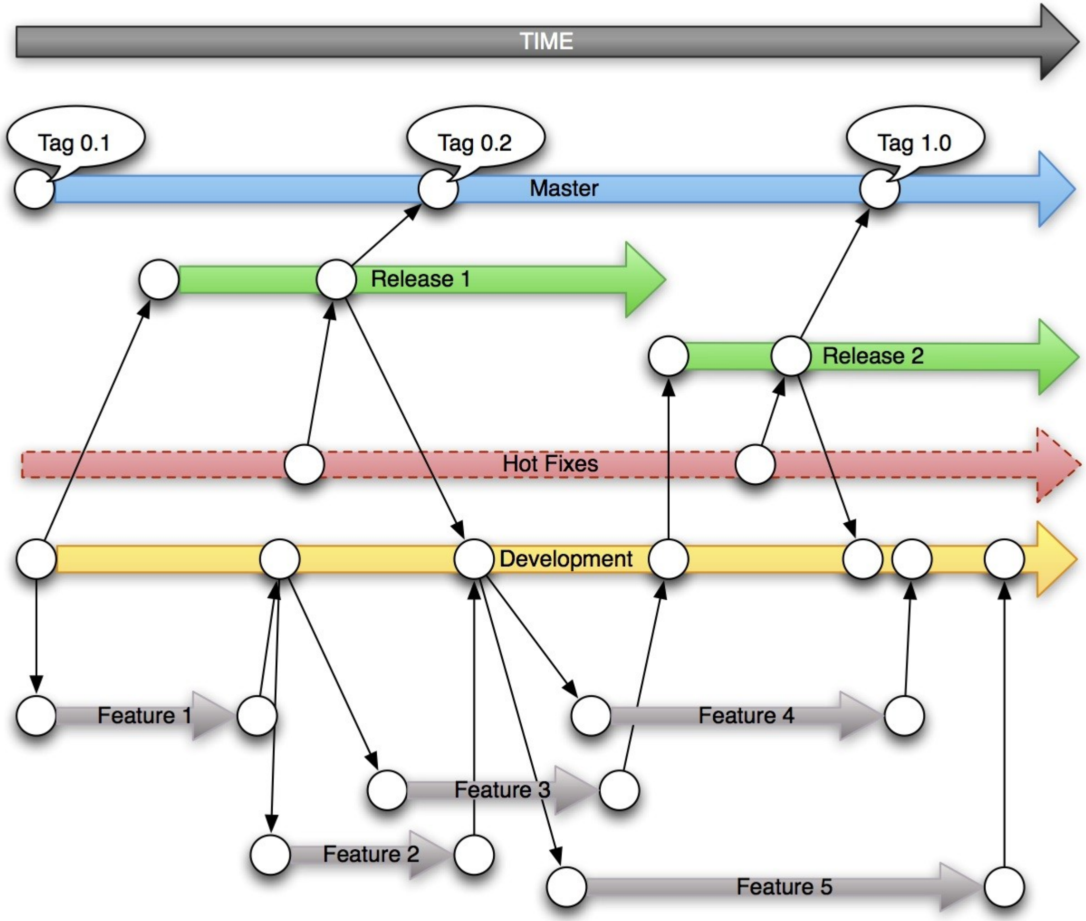


About git: http://git-scm.com/
Intro to git: http://git-scm.com/book

Benefits

- Works offline
- Intelligent merging
- Branches: fast and local
- Find regressions via. bisection
- Easy to share, send patches
- More than one “central” repository
- Name commits via. tags
- Submodules

Branching



Graphic from http://erickryski.com/2012/06/01/my-git-branching-strategy/

git – typical daily usage

- Checkout the repository:

  ```
  git clone user@yourserver.com:thing.git
  ```

- Switch to a different branch (this is really fast)

  ```
  git checkout [-b] branch_name
  ```

- Make changes (preferably in small, manageable portions)
- Add file(s) to staging area

  ```
  git add test1.cpp test1.h test2.* ...
  ```

- Commit to the local repoitory

  ```
  git commit -m "Short commit message, typically < 50 characters"
  ```

- Continue making changes.
- Once done, push your changes somewhere else

  ```
  git push destination branch_name
  ```

- If you mess up:
  - Remove file from staging area:

    ```
    git reset HEAD filename
    ```

  - Restore file to state in last commit:

    ```
    git checkout filename
    ```

  - Restore file to previous commit state:

    ```
    git checkout COMMIT_HASH_ID filename
    ```

Gitlab: Collaborative coding

3 minute video showing

1. Repository creation
2. Committing, pushing
3. Issue creation, resolution
4. Branching
5. Milestones

Migration time line

- Verify CHARA machines run git – done
- Trial CHARA CVS to git conversion – done
- Get gitlab server operational – done