

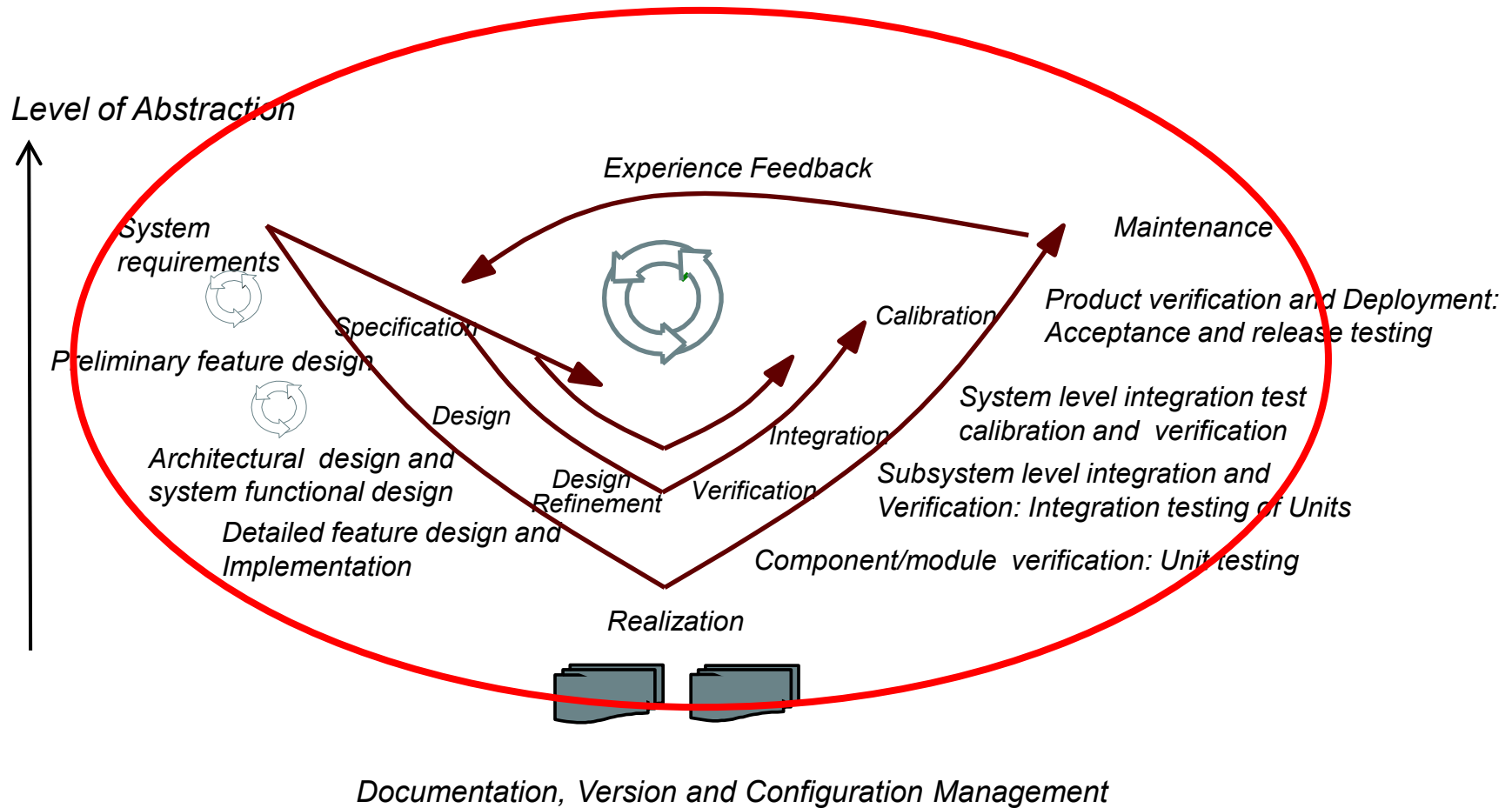
Software Engineering Theory

Software configuration management

Lena Buffoni/ Kristian Sandahl

Department of Computer and Information Science

2024-09-17



Project Management, Software Quality Assurance (SQA), Supporting Tools, Education

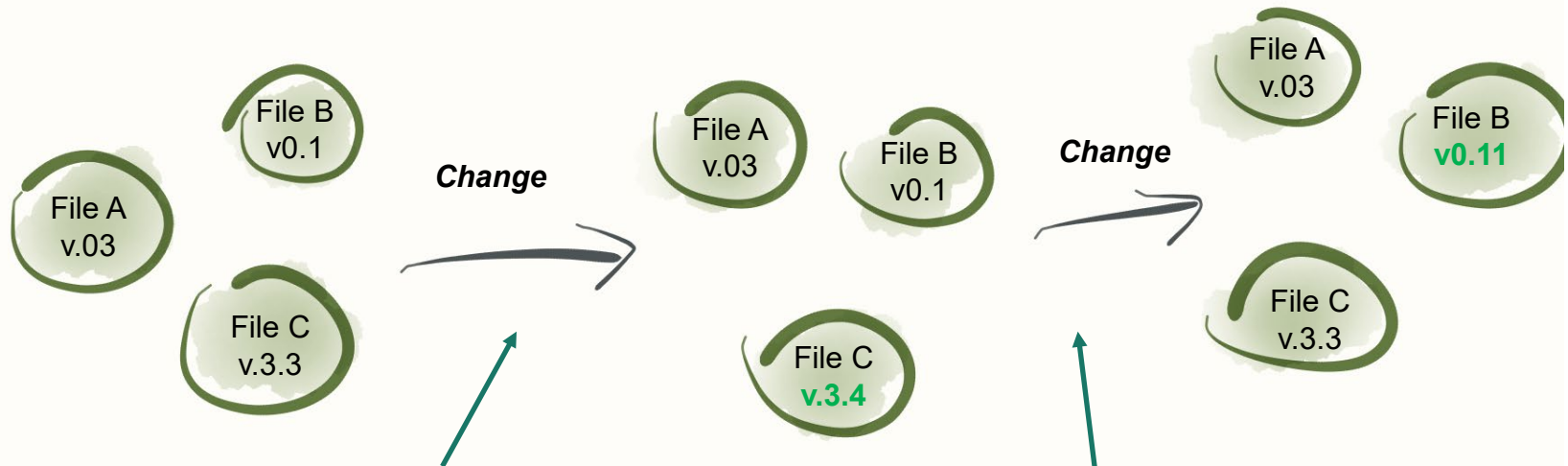
What is configuration management?

Configuration Item Identification

- Source code modules
- Test scripts
- Design documents
- Build systems



What is configuration management?



Baseline - a "snap-shot" of configuration items. Known to work together. Normally reviewed in some way. For example a release.

How do we control changes?

What is configuration management?

Configuration Item Identification

- Source code modules
- Test scripts
- Design documents
- Build systems

Configuration control

- Only authorized people may make changes
- Larger changes - change requests



Change Control Board (CCB)

- Make change decisions.
- Only large changes (e.g. new major requirements)

CHANGE REQUEST

Project: _____

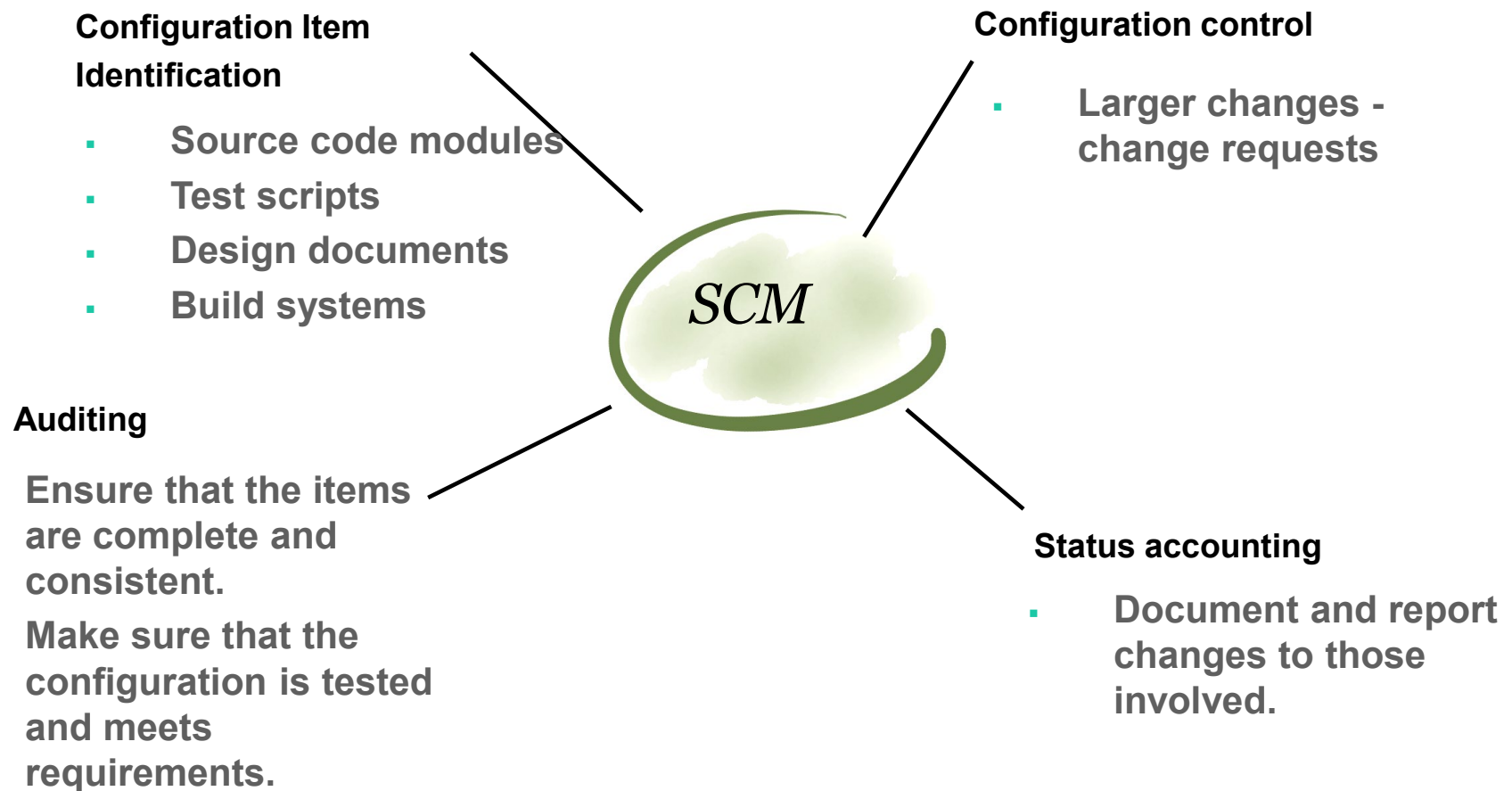
Classification: _____

Priority: _____

Date: _____

Change Description:

What is configuration management?



SCM tools

Workflow systems

■ Define processes for change requests

Change report system

- Bug-tracking
- New features
- Change request

(e.g free alternatives: Bugzilla, Trac)



*Change
manage
ment*

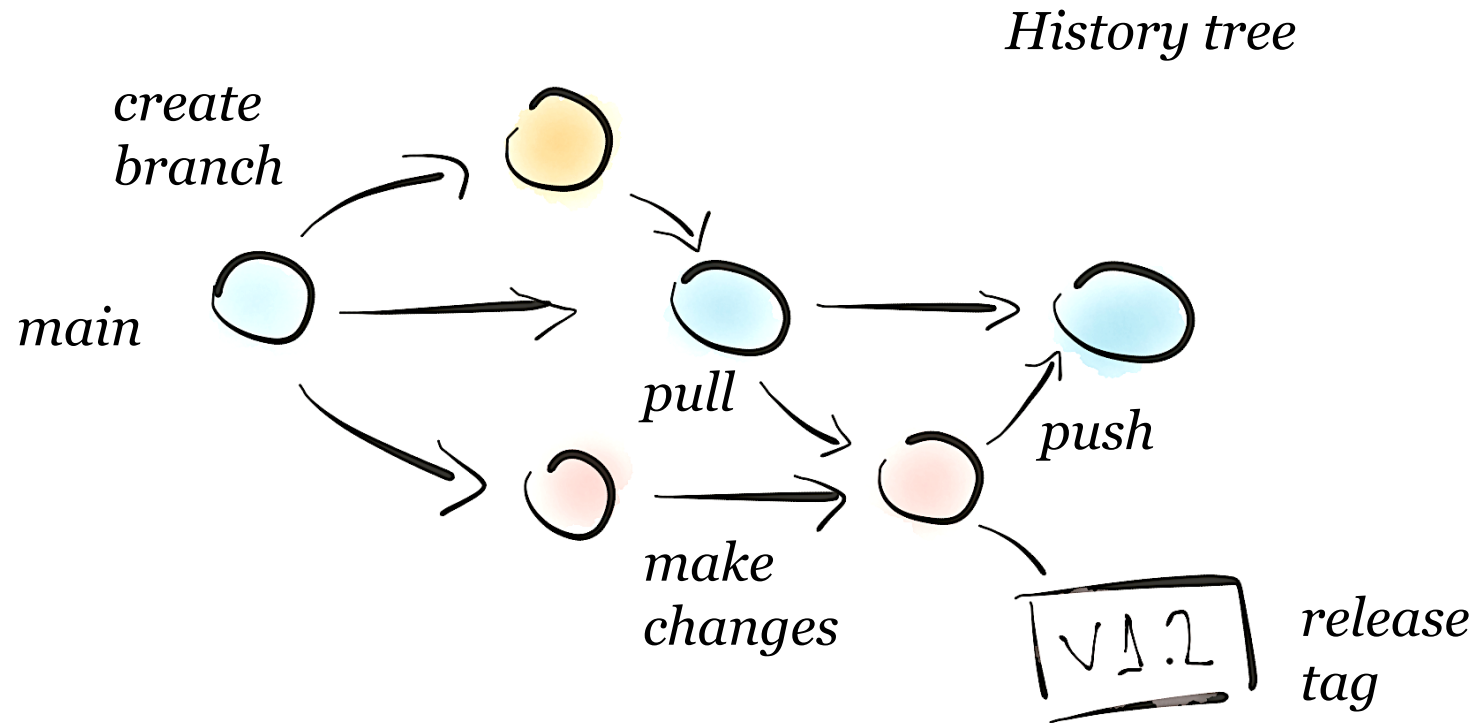
Tool examples:

- SVN, Git
- GitHub, GitLab
- Beanstalk



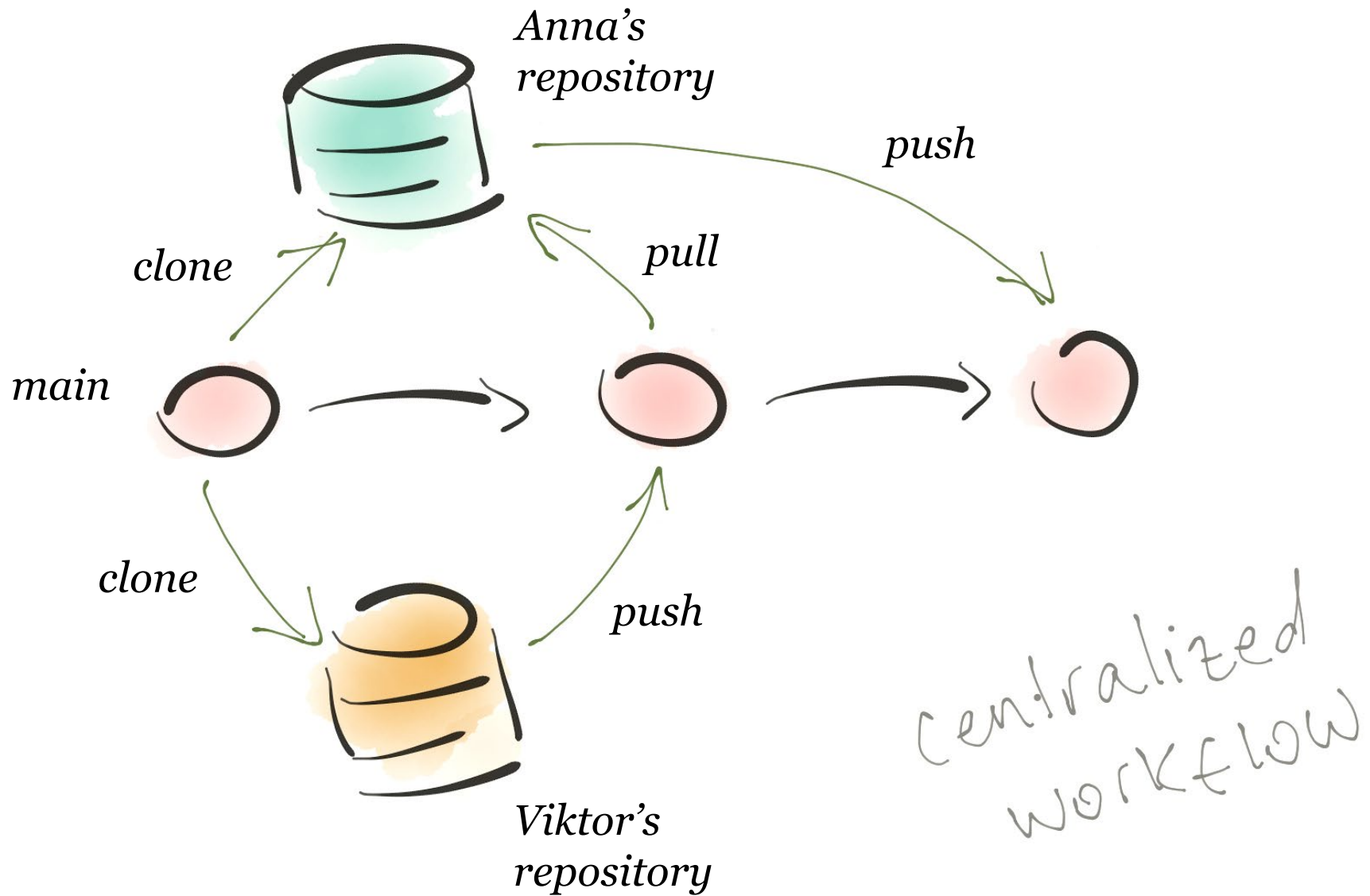
*Revision
control
systems*

Revision control: Some terminology



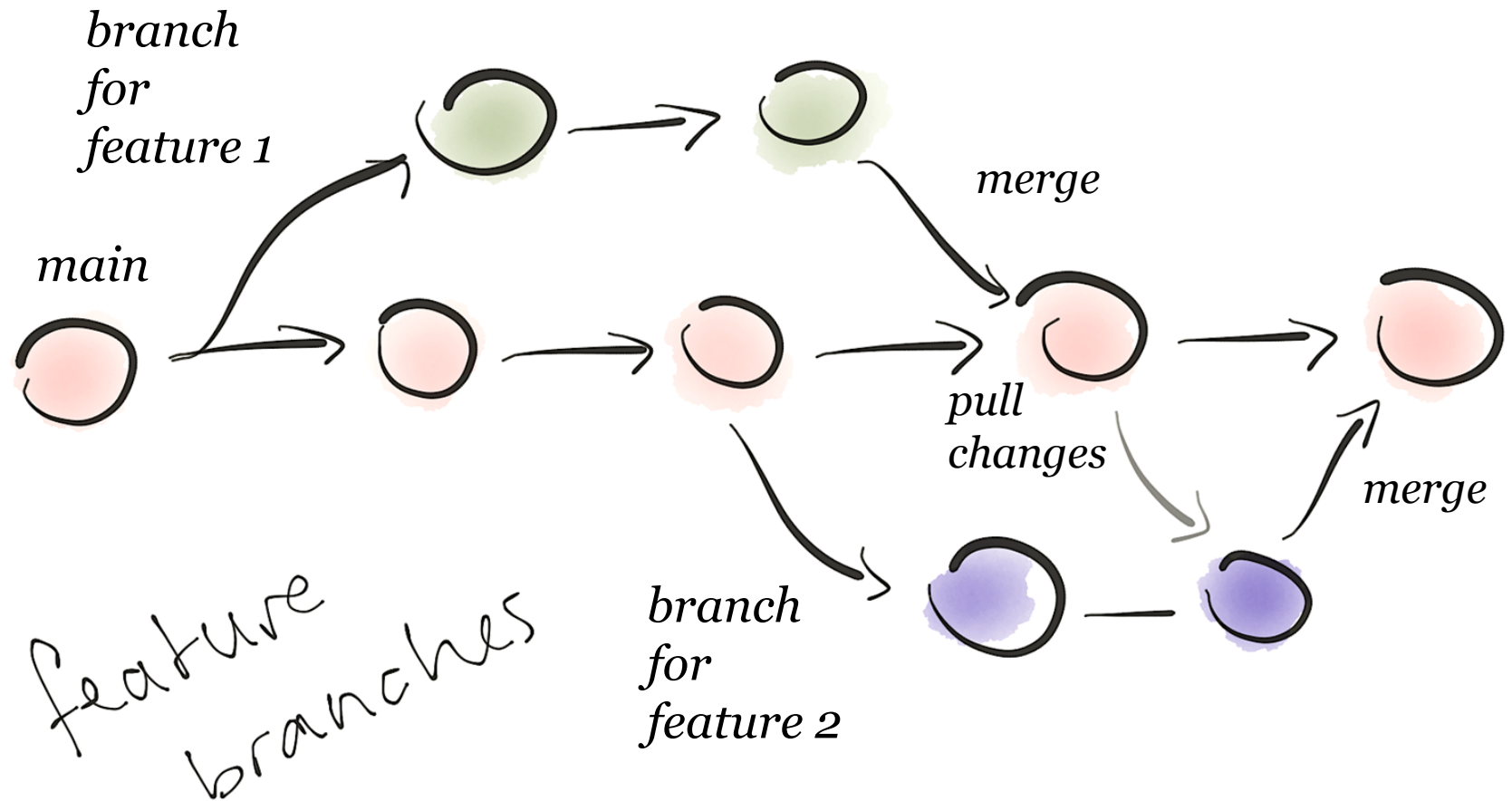
Defining a common workflow

- A structured way of collaborating
- Different people/teams working on different parts of the system



Pros/cons of centralized workflow

- + little setup
- + will work for small projects
- limited flexibility
- hard to maintain a continuously working product

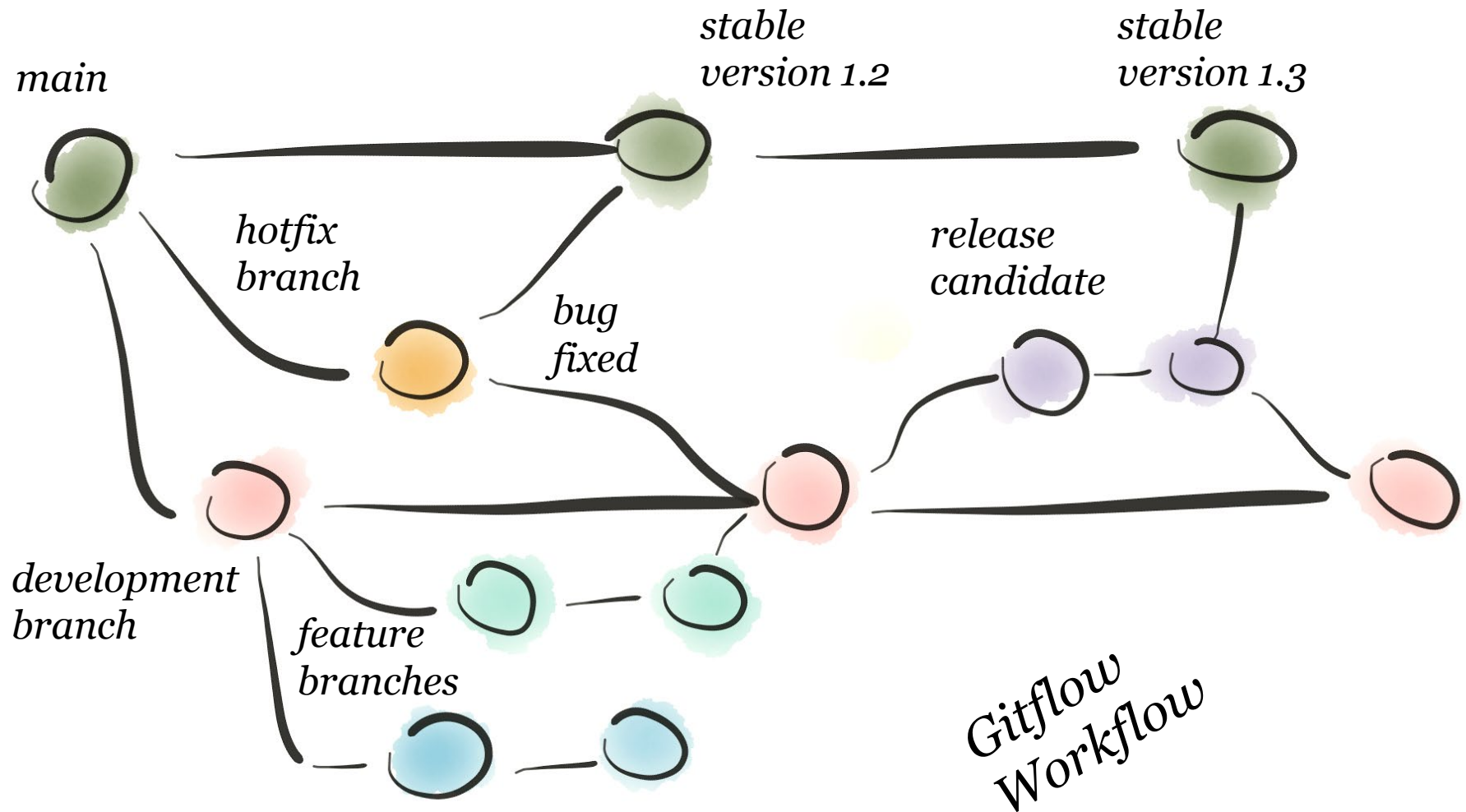


Pros/cons of feature branches

- + easier conflict resolution
- + better control of what features to include when
- + possibility of using pull requests
- additional steps
- long living branches might be hard to merge

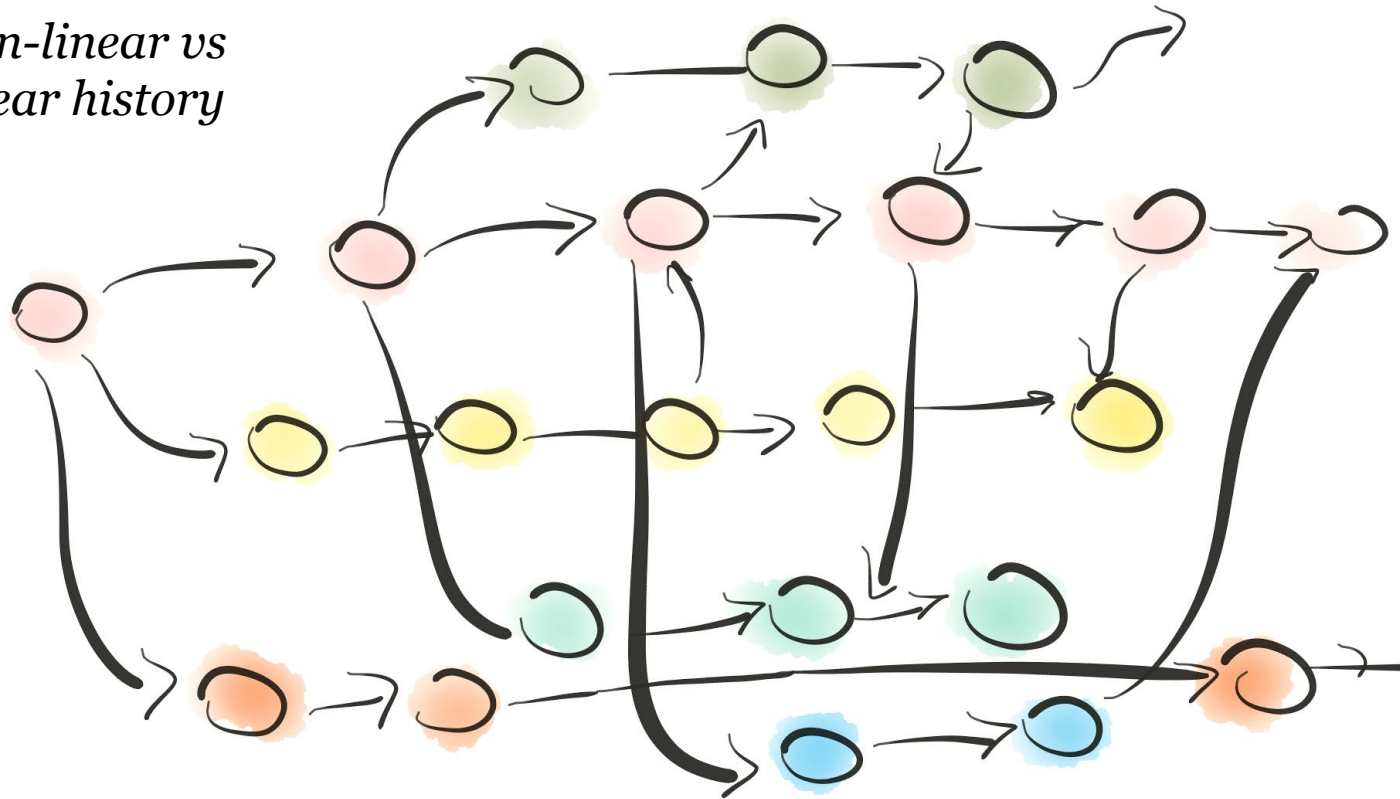
Push vs pull request

- Pull request: Submitting your code for consideration for inclusion in the main repository
 - + an opportunity to review code and make changes
 - + less risk of needing a rollback
 - puts extra work on the repository owner
 - can increase time before changes are integrated



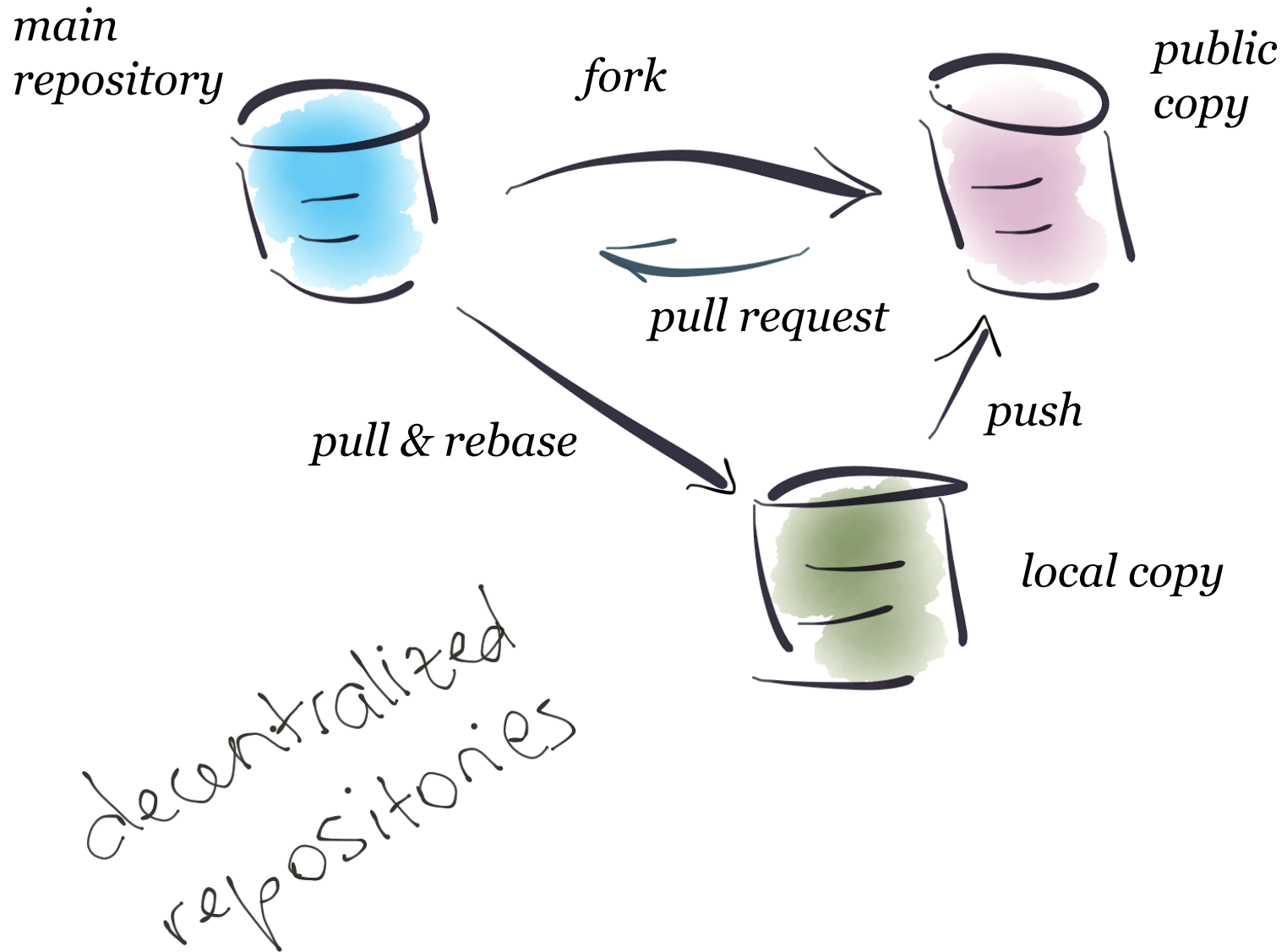
Things can get complicated

*Non-linear vs
linear history*



Pros/cons of gitflow workflow

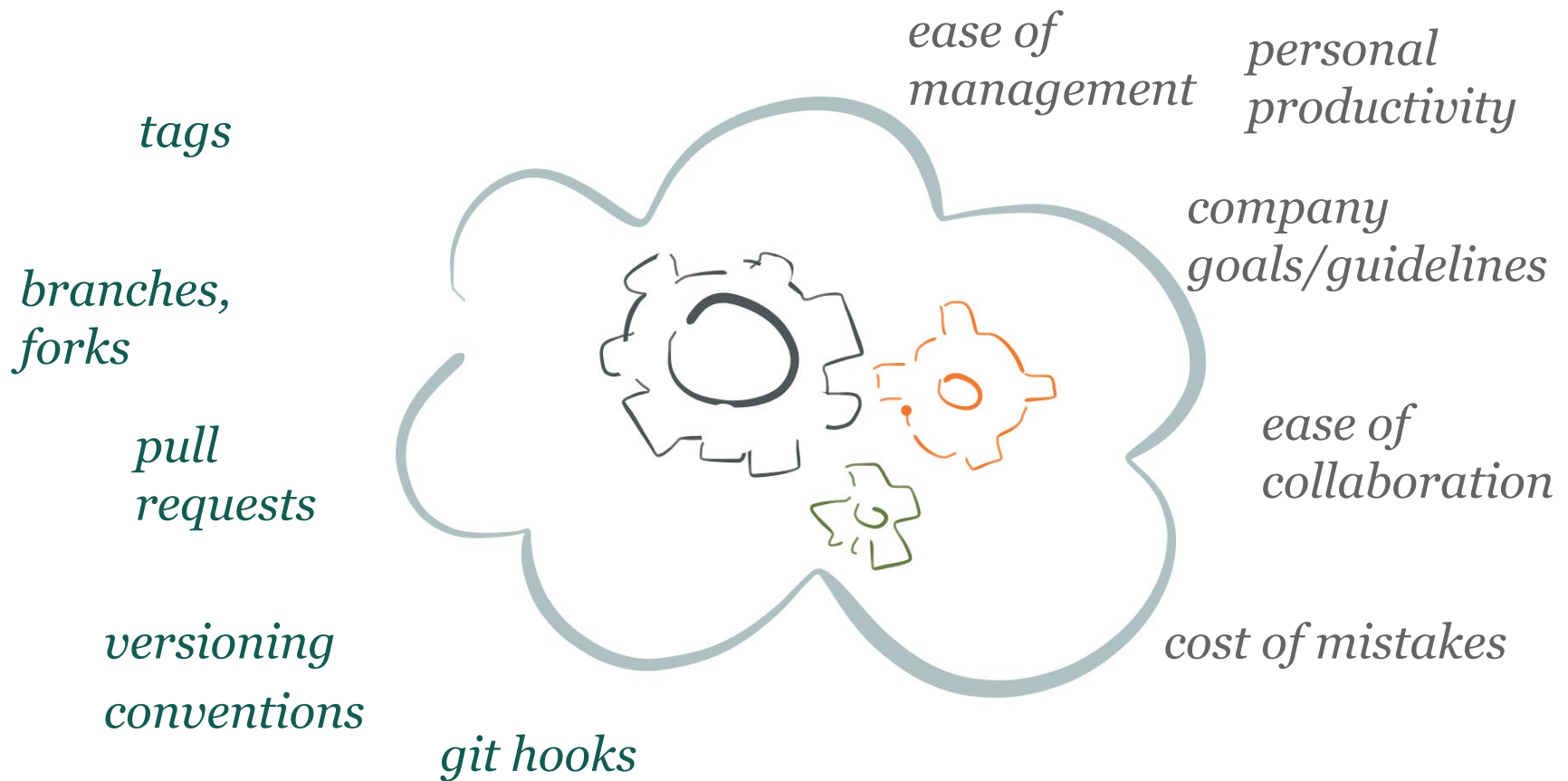
- + always stable code in the master
- + can work on release and future features simultaneously
- + fix issues without disrupting workflow
- not good for supporting multiple product versions
- history can get complicated
- many merging conflicts to resolve



Pros/cons decentralized workflow

- + no need to provide write access to main repository
- + good for large open-source projects
- more complicated to set up and maintain

Make your own work-flow



Summary

- Configuration management processes
- Development workflows

The end. Thank you! Questions?

www.liu.se