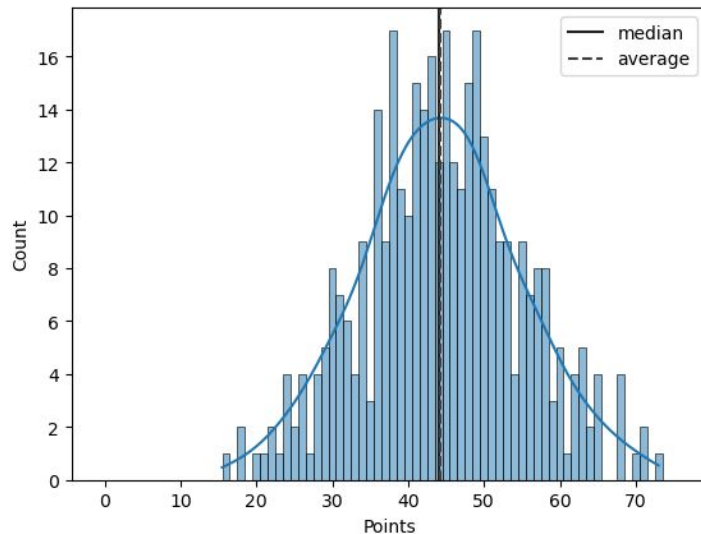


# Human aspects of software development

CS-214 - 26 Nov 2025

Clément Pit-Claudé

# Quick announcements



## Debrief for week 10

is up, with webapp tips

## Webapp status

Planning and first prototype!

## Webapp showcase

<https://cs214-apps.epfl.ch>

## Exam grades

Out tomorrow morning, 9AM

## Final exam

Dry run

# TODO

TDD and collaboration for webapp

Last week:

# Distributed version control

Learning objectives:

- 1. Handle divergence between codebases and resolve conflicts**
2. Describe and apply distributed collaboration schemes

- Single-user git recap
- Distributed Git basics
  - Remotes
  - Fetching
  - Branches
- Managing short-term divergence
  - Patches
  - Cherry-picks
  - Rebases
- Managing long-term divergence
  - Branching
- Handling conflicts
  - 3-way diffs
  - Conflict resolution
- Social aspects of distributed dev

# Distributed git demo

# At-home topics

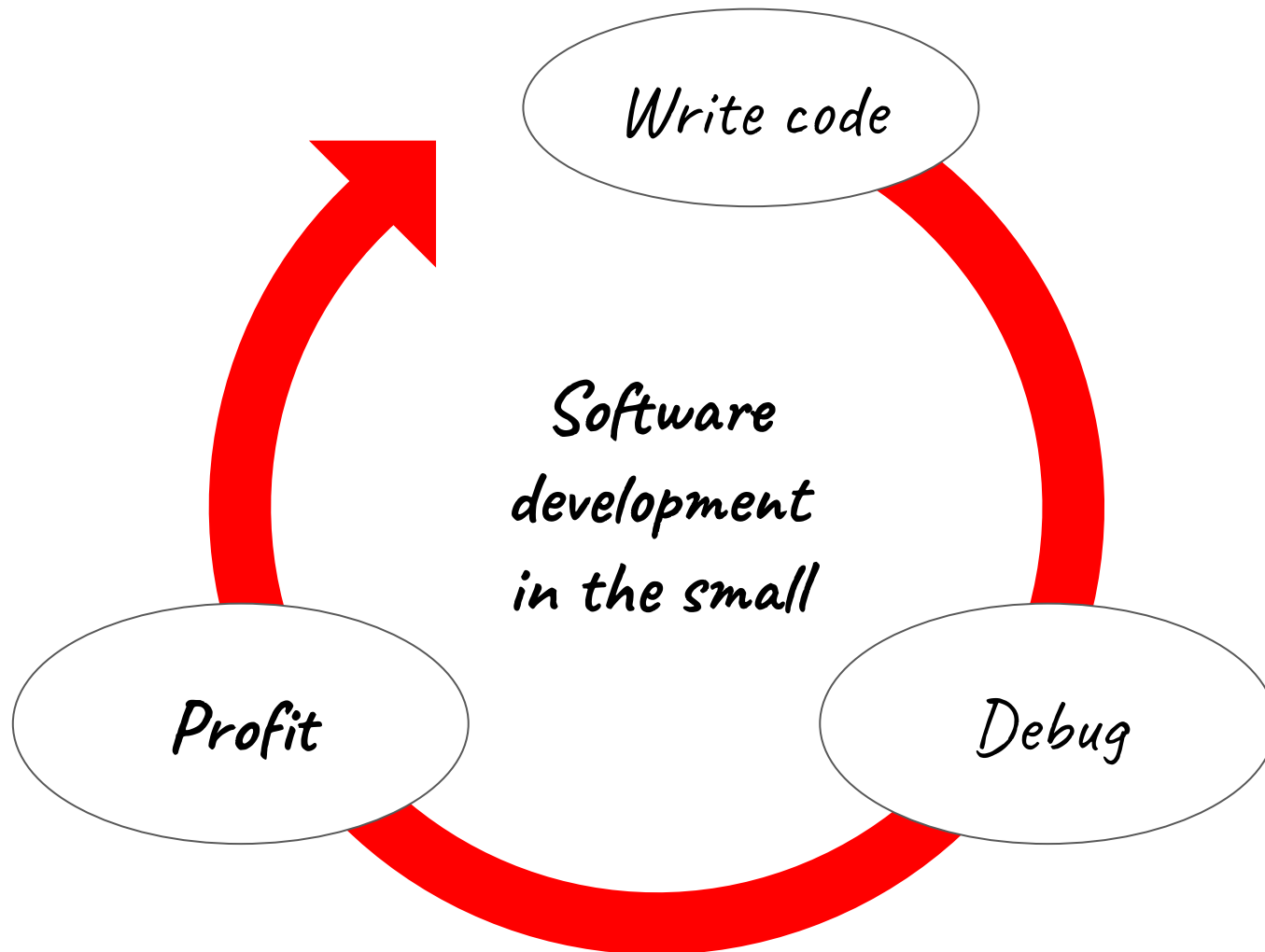
(on your own, see  
exercises)

- SSH keys
- GPG signatures
- `git range-diff`
- `git send-email`
- `git rerere`

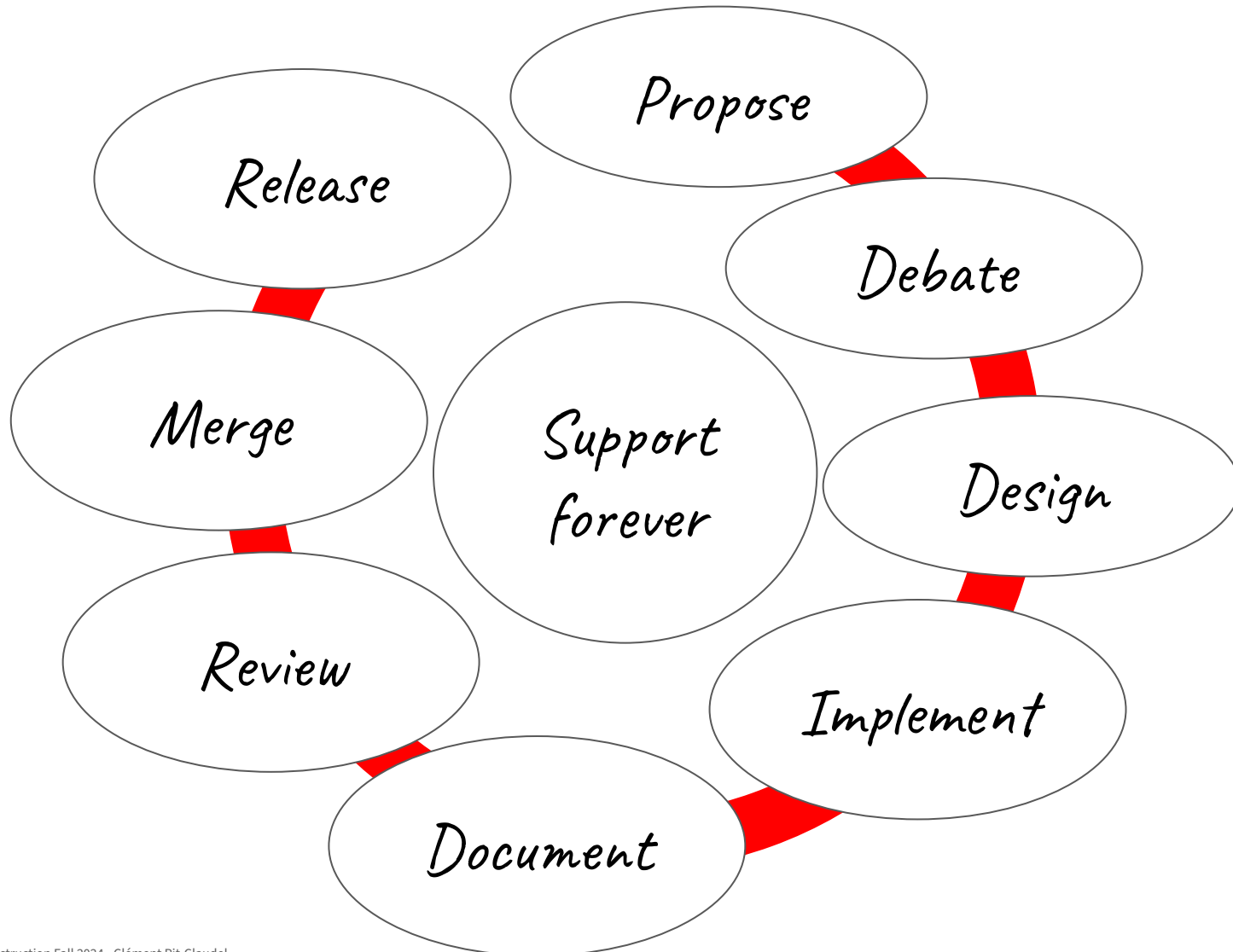
# Social aspects of distributed development

- The lifetime of a software feature
- The messy reality of software development
- Fostering community and collaboration
- Tools for distributed development
- Ethics and software

# The lifetime of single-user software



# The lifetime of a software feature in the large





Azure



ATLASSIAN



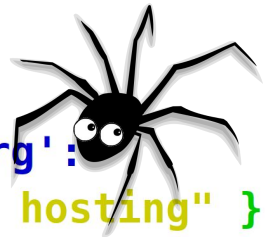
# GitLab



Savannah



{ 'NotABug.org' :  
"Free code hosting" }



APACHE



ALLURA™

plain HTTP

# Three dominant workflows

## Patches

- `git clone`
- `git branch, commit`
- `git format-patch`
- `git send-email`
- `git am`

## Who?

- Larger, older projects with complex workflows

## Examples

- [torvalds/linux](https://torvalds/linux)
- [GCC - GNU Project](https://gcc.gnu.org/)

## Public forges

- `fork/branch, clone`
- `branch, commit, push`
- Web UI for review
- `git rebase`
- `git merge`

## Who?

- Most free software, various companies

## Examples

- [lampepfl/dotty](https://lampepfl.dotty.ch)
- [microsoft/vscode](https://microsoft.com/vscode)

## Custom forge

- `<clone>`
- `<branch, commit>`
- Custom patching
- Custom review

## Who?

- Large projects and companies

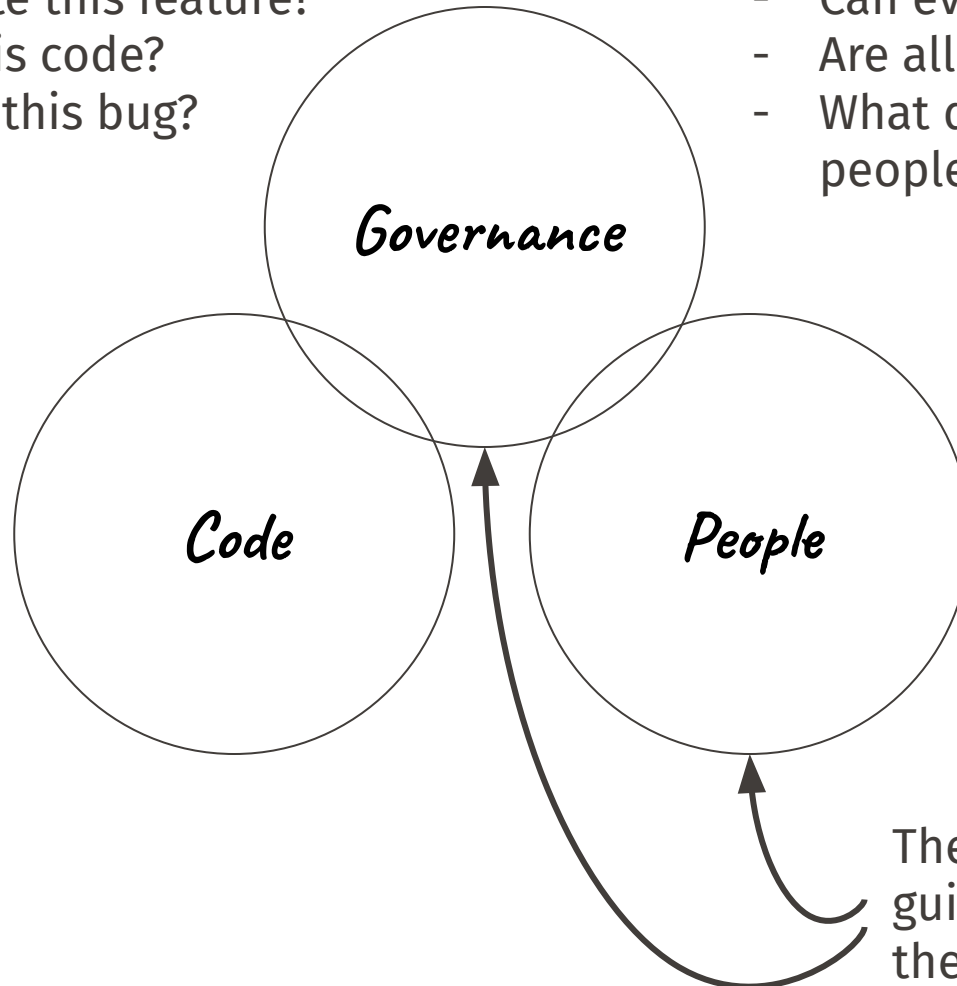
## Examples

- [salsa.debian.org](https://salsa.debian.org)
- AWS, [Google](https://google.com/), ...

# 3 aspects of distributed software construction

- Which libraries do we use?
- Can we deprecate this feature?
- Do we accept this code?
- Do we prioritize this bug?

- Who makes decisions?
- Can everyone contribute?
- Are all discussions public?
- What do you do when people clash?



The CS214 debugging guide doesn't work on these :(

# Fostering community and collaboration

- Communicating with users
  - Release notes, version numbers
  - Bug trackers
- Collaborating with developers
  - Code review
  - Automated checks
  - Milestones, task tracking, project planning
- Onboarding contributors and maintaining a community
  - [Codes of conduct](#)
  - [Mailing lists](#), meetups, IRC/chat
  - Guides for [committing](#) and [patching](#)

# Software ethics

- Environmental impacts (energy consumption, device fabrication)
- Social impacts (access to information, online harassment)
- Economic impact (scams, malware, cryptocurrencies)
- Fairness and equality (software licenses, access to technology)

→ BA5 course: [Responsible software!](#) ←

**Be a force for good!**