Is SE research addressing problems of multi-person multi-version development?

- Yes: 15%
- Partially: 73%
- No: 8%
- Doesn't matter: 4%
What areas of software engineering do you think should be studied more?

- Software evolution
- Software architecture
- Technology transfer
- Industrial software
- Software design process
- Software ecosystem
- Process management
- Human aspects
- Regression testing
- Developer communication
- Tool and support building
- Modern SQA practices
- Configuration management
- Continuous integration
- Engineering work
- Design decisions
- End user development
- Software specifications
- IoT and CPS development
- Version management
- Collaboration
- Knowledge management
- Defect reporting
- Software quality
Studying software ecosystems...

- Only helps open source projects (2.1)
- Is useful to understand history but doesn't help current projects (2.1)
- Teaches us about the past and is helpful for current projects (3.9)
What questions about multi-person multi-version development are interesting for individual projects?

- How can we ensure software quality?
- How do build up the most effective team/sub-teams and best support them?
- How to effectively share knowledge in development teams?
- What dependencies should you use? How to nudge developers of those dependencies to make changes if you need them? How to isolate yourself from their changes if you don't want them?

how the individual project's development process is effected by other projects or developers in the same ecosystem, in the context of multi-person multi-version development.
Can you think of interesting opportunities to automate parts of software development flow?
Is SE research addressing problems of multi-person multi-version development?

- Yes: 13%
- Partially: 88%
- No: 0%
- Doesn't matter: 0%