

# Challenges and Confusions in Learning Version Control with Git

Ville Isomöttönen and Michael Cochez

University of Jyväskylä

ICTERI 2014

# Contents

- 1 Background & research target
- 2 Related work
- 3 The study
- 4 Results

# Background

- Version Control Systems have a decades long history in computing (SCCS, RCS, CVS, SVN...)
- Their educational value is commonly agreed
- $\Rightarrow$  version controlling is a useful professional skill in today's distributed workplaces

# Research target

- Research topic: teaching version control with the distributed version control system *Git*
- Motivation: parsimony of empirical studies on student use and learning of VCSs
- The main research question: What challenges do students face and what confusions emerge?
- Studied with a qualitative survey and through identification of our own teaching challenges
- The present study extends our previous work where we have analyzed student Git repositories

# VCSs in educational settings

From the literature we found that VCSs have been used for

- course management and supporting teaching practices (see [1])
- monitoring learning processes and contributions of individual students (see [2])
- understanding student code (see [3])
- promoting realistic software development experiences (see [4])
- plagiarism detection and personalizing exams (see [5])
- etc.

The related research findings suggests that

- students may mix up different functionalities of a VCS
- students have difficulties in understanding inner workings of the system used
  - ▶ ...thus they may not have a proper mental model of the system
- there may be management overhead arising from setting up and managing the student repositories
- etc., see [6, 7, 5]

The present study was based on

- a survey issued to bachelor level Project course students who used Git in their project (group) work during spring 2013
- a synthesis of our own teaching experiences from several courses where Git has been used

# Survey

Survey was answered by 21/26 project students, consisting of the following questions:

- Q1 (open-ended): Do you think that using the VCS was useful?
- Q2 (open-ended): What difficulties did you encounter regarding version control?
- Q3 (5-point Likert scale): Do you think that your group used the VCS in an efficient manner?
- Q4 (5-point Likert scale): Evaluate how actively you used the system.
- Q5 (open-ended): Described how you used the VCS?
- Q6 (open-ended): Did you read the commit messages of others?
- Q7 (open-ended): Did you find the messages beneficial?



# Sources of teaching experiences

- The Project course (source of the survey answers)
- Introductory software engineering course which included supervised sessions on version controlling with Git, a command line client was used
- Master's level programming-oriented courses where Git has been used for submitting work and suggested for students as a group work tool

# Summary of the survey results I

- A VCS is experienced useful and necessary, but the limited resources of a project assignment may discourage students to experiment with and take advantage of the advanced features of a VCS.
- Authentic project assignment makes students to experience benefits so that they see a VCS advantageous and are willing to use it.
- The routine of requesting the changes at the beginning of a work session and sharing changes in the remote machine at the end of the session appears to be well absorbed, while there seems to be variation in how frequently students interact with the remote machine.

## Summary of the survey results II

- Students need not only conceptual explanation of the inner workings of Git, but also of what a distributed VCS implies and enables in terms of usage patterns. Students would have benefited from encouragement towards and information on the benefits of frequent commits and that branching can be useful for committing the work in progress.
- Regarding commit message writing, we interpreted from the data that their writing may naturally become dependent on group size, communication distance, and the current complexity of the shared product development. The finding that students considered their presence in the work room removing a need for informative commit messages is likely to explain our previous observation of nonsensical commit messages [8], while this appears to be less prevailing in a project-based course as compared to exercise-based courses [9].





# Summary of teaching experiences

- branches may be conceived as folders or files
- mixing of Git and Bash
- `'git add file'` interpreted to mean creation of a new file
- challenges with Git's presentational conventions, e.g. `origin/master` vs `origin master`
- confusion from no default branch on a remote machine
- accidental sub-repositories, e.g., by experimenting with both `'git init'` and `'git init repo-name'` commands
- blind-testing effect: student lacking knowledge of inner workings of Git try to proceed by giving commands arbitrarily
- absence of authentic scenarios causes association of branching to rather unorthodox use cases
- attitudinal barriers arising when Git is compared with other, less powerful, systems and considered difficult




## Concluding remarks

- Use pedagogy that explains theory together with practice (explain inner workings of Git in conjunction with practicing)
- Introduce authentic/relevant use cases to motivate the use and learning of the system
- Authentic practical courses cause students to see the value of VCSs
- However, use VCSs throughout the curriculum, as, for instance, limited resources available during authentic project courses may limit how the system is adopted
- Future research could compare Git usage between professional and educational setting (effect of group size and communication distance on the use of a VCS)
- Future research should also compare between different VCSs in educational setting

# References I

-  C. Clifton, L. C. Kaczmarczyk, and M. Mrozek, “Subverting the fundamentals sequence: Using version control to enhance course management,” *SIGCSE Bull.*, vol. 39, pp. 86–90, Mar. 2007.
-  Y. Liu, E. Stroulia, K. Wong, and D. German, “Using CVS historical information to understand how students develop software,” in *MRS 2004: International Workshop on Mining Software Repositories*, 2004.
-  L. Glassy, “Using version control to observe student software development processes,” *J. Comput. Sci. Coll.*, vol. 21, pp. 99–106, Feb. 2006.
-  K. T. N. Hartness, “Eclipse and CVS for group projects,” *J. Comput. Sci. Coll.*, vol. 21, pp. 217–222, Apr. 2006.

## References II

-  G. Robles and J. Gonzalez-Barahona, “Mining student repositories to gain learning analytics. an experience report,” in *Global Engineering Education Conference (EDUCON)*, pp. 1249–1254, IEEE, March 2013.
-  K. L. Reid and G. V. Wilson, “Learning by doing: Introducing version control as a way to manage student assignments,” in *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, SIGCSE '05, (New York, NY), pp. 272–276, ACM, 2005.
-  J. Lawrance, S. Jung, and C. Wiseman, “Git on the cloud in the classroom,” in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, (New York, NY), pp. 639–644, ACM, 2013.

## References III



M. Cochez, V. Isomöttönen, V. Tirronen, and J. Itkonen, “The use of distributed version control systems in advanced programming courses,” in *Proceedings of the 9th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer*, vol. 1000, pp. 221–235, CEUR, 2013.



M. Cochez, V. Isomöttönen, V. Tirronen, and J. Itkonen, *How Do Computer Science Students Use Distributed Version Control Systems?*, pp. 210–228. Springer, 2013.