Empirical Methods in Software Engineering Research: How Well Do Developers Know Git?

Lauritz Timm

Kiel University, Kiel, Germany
lauritz.timm@stu.uni-kiel.de

Abstract. Version Control Systems (VCS) are an import part of modern software engineering and large software projects. The state-of-the-art tool Git is used by students, researchers and professional developers equally. While graphical interfaces, like GitHub, GitKraken or GitLab, offer a convenient way of interacting with code repositories, most developers use the command line interface of Git.
Most users do not struggle with the basic concepts of Git, but the practical application of commands seem to be inconvenient and prone to error. This results in many questions on Stack Overflow regarding syntax, semantics or applicability of Git commands. A vast majority of those posts are also related to complicated use cases, which are still part of basic workflows, for example, `clone` or `difftool`. On one hand, a lot of topics are about basic interactions with VCS, on the other hand questions related to advanced or niche commands may not get a suitable answer, when asked on Stack Overflow.
In addition to the quantitative analysis of forum posts, we looked into how developers learn and apply Git-commands. Most users are self-taught, therefore more training is needed to increase confidence in using advanced properties of VCS.

Keywords: VCS · Git · Stack Overflow.

1 Introduction

This section provides an overview and short introduction. We state our motivation and the goal of this paper.

Modern software projects often have a large codebase, which has to be developed, tested and maintained. Therefore, many developers work in teams but need to keep track of changes introduced by their peers, as well as the overall progress. The version control system Git has emerged as the state-of-the-art (SOTA) solution to provide a solution to this problem.

Unfortunately, it seems eminent, that users run into difficulties, when they try to use Git [11]. Often, they refer to the popular Q&A-platform Stack Overflow for help [25].

1.1 Goal of this Paper

With this paper we summarize the results from the survey conducted by Yang et al. [25] regarding Stack Overflow as online resource. They investigated, to which degree proficiency of experienced users with Git is reflected through quantity and quality of Stack Overflow questions and answers. In addition, we look into the overall demography of developers and combine the distribution with the prior observed results. To help users of Git or similar version-control tools, we compare strategies used to improve knowledge and confidence. We suggest how teachers, students and experienced users can tackle discrepancies and improve development workflow.

1.2 Structure

First, we introduce research questions and some basic terms, to provide the fundamentals for this paper. We continue with explaining our methodology and showing our results and observations. In the final part, we discuss these results and try to answer our research questions.

1.3 Research Questions

To guide this review, we introduce the following research questions:

RQ1 What is the level of expertise of Stack Overflow users with Git?
RQ2 Is Stack Overflow a good source for learning Git commands?
RQ3 How can we improve confidence and ability to use version control?
RQ3.1 What recommendations could be given to learn Git?

2 Background

In this section we introduce some basic terms, which are used in conjunction with version control systems (VCS), Git and Stack Overflow.

2.1 Distributed Version Control Systems

To coordinate software development, especially writing source code, programmers use VCS to keep track of changes and feature implementations. Git was introduced by Linus Torvalds in 2005 as tool for Linux kernel-development [7]. It provides an tree-like representation of version history and the ability to branch, merge and edit nodes, called commits, within a synchronized database. It is often used in conjunction with websites like GitHub or GitLab, which provide a remote repository and a convenient web-interface to view and interact with the source code directly. The distributed nature of such a repository allows several developers to work on the same project simultaneously. Recently, features like issue tracking, merge requests and continuous integration SOTA tools for code development used by millions of users [4,5].

2.2 Command Line Interface

While most users of Windows or similar popular operating systems (OS) may only encounter graphical user interfaces (GUI), computer scientists and software developers often use command line interfaces (CLI) to interact with the OS. Many, especially Linux and other Unix-based implementations, provide a command interpreter [7,14,17] to enter additional parameters for a program call.

While there are graphical front-ends for Git [7,13], most professional users opt to use the underlying command based interface, to customize their experience [7] to have the option to use features, not provided by any GUI, in a convenient way. For example, the documentation of Git currently lists around 150 different commands to interact with the CLI [12]. While this number seems large, only around ten different commands are needed for most use cases [7].

2.3 Stack Overflow

Fig. 1. *2023 Developer Survey*: Main online resources to learn how to code. [23]

Due to the complexity of programming languages, frameworks and software systems, users and developers may refer to Stack Overflow, a Q&A and blogging website for computer-science-related topics. We see in Figure 1, that Stack Overflow is the main resources after documentation, developers use to write code. To ask questions, users can open threads which may be answered and discussed by peers. In addition to a simple question-answer-dialog, most times responses are also graded and discussed in detail. It is not uncommon, that many, each reasonable and applicable, answers exist for a single topic. Furthermore, wrong or displaced answers and comments are down-voted increasing the visibility of helpful ones. To ease navigation, Stack Overflow uses tags to categorize questions with corresponding topics.

Regarding Git, the platform is used as discussion board for syntax, semantic and functionality of commands [25] and the general approach on VCS [1,2].

3 Methodology and Related Work

This section describes the used methods by Yang et al. [25], as well as in other papers, divided into measurements focused on interactions on Stack Overflow and surveys how users work with Git as version control system.

The paper by Yang et al. [25], which is the starting point for this review, includes a survey on Git-related question-answer pairs on Stack Overflow. The main part of their work consists of a quantitative study working with the *Stack Overflow dataset*[1], where they analyze questions related to Git commands. In addition, they designed a online survey to gather information about professional developers' approaches to learning Git commands [25]. Therefore, their methodology is guided by the following seven steps:

1. Download the Stack Overflow dataset.
2. Identify relevant questions.
3. Determine question popularity trend.
4. Determine questioner distribution.
5. Determine command popularity.
6. Determine command difficulty.
7. Survey developers' learning approaches.

The identification of Git-related threads is done by filtering by the tag "git", and refining the results with focus on command-related questions [25]. All other metrics are then evaluated on this subset.

To provide our own context to the distribution of users on Stack Overflow, we look into the *2023 Developer Survey* [23] with around 90,000 participants, and the analysis of an older iteration of this survey[2] by Dada et al. [8]. We are especially interested in the level of expertise, education, occupation and learning approaches get an idea, how acquainted developers are, when they ask questions on popular Q&A-platforms. Fortunately, the survey also includes questions to which degree Stack Overflow is an important learning tool. Diyanati et al. [9] proposed an alternative approach of determining expertise level by looking into positive and negative comments on questions or responses of a user.

Furthermore, Treude et al. [24] and Rekha et al. [19] investigated the effectiveness of Stack Overflow as a learning platform. Meldrum et al. [16] looked into the code quality of snippets on the platform.

To look into Git as a software system, we look into a study by Eraslan et al. [10] focused on common errors and poor practices, and an analysis of the design of the underlying concepts by Perez De Rosso and Jackson [18] .

[1] Yang et al. used data from the *Stack Exchange Data Dump* received on the 23th May 2021 [21].

[2] Dada et al. used data from the *2020 Developer Survey* with nearly 65,000 participants [22].

Sproull [20] and Bonakdarian [6] did student surveys on whether providing a tutorial and additional exercises during computer-science classes improve the usage of Git.

The interesting idea of using crowd-sourced knowledge to recommend situational Git commands was proposed by Jia et al. [15]. They used the results of Yang et al. [25] to build a tool based on posts on Stack Overflow, which enables developers to receive recommendation for commands suiting a specific scenario.

4 Evaluation

In this part, we focus on our research questions and try to provide the data to answer those.

4.1 Stack Overflow and Git

Yang et al. observed, that even after years since Git was first introduced [7], the number of command-related questions and questioners is still large, showing, that a considerable amount of developers have issues with the usage of Git [25]. They also noticed, many questions regarding Git commands are popular and raised by different users, confirming, that many developers have faced difficulties when using this tool.

To tackle the underlying question regarding experience of users on Stack Overflow, Yang et al. found, that many of the questioners who have been registered for a long time on the Q&A-platform still post questions about the usage of Git [25].

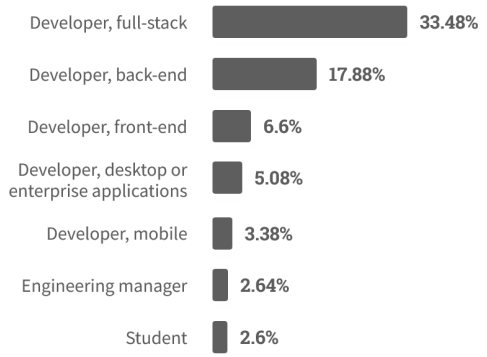

Fig. 2. *2023 Developer Survey*: Main developer types. [23]

Together with the recent *2023 Developer Survey* and analysis by Dada et al. [8], we can deduce, that a lot of users, who interact with the platform are

already experienced or even full-time employed developers [23], which can be seen in Figure 2.

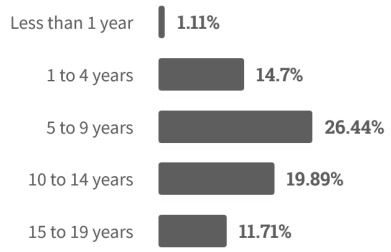

Fig. 3. *2023 Developer Survey*: Years since users started coding (not necessarily professionally). [23]

Furthermore, most developers have more than five years practice in coding, as seen in Figure 3, not matching the idea of novice questioners. This is also suggested by Yang et al., who conclude "Git commands are not easy to learn and master, even for developers with years of programming experience" [25].

In the next part, Yang et al. analyzed question popularity and listed `revert`[3], `reflog`[4], `stash`[5], `clean`[6] and `reset`[7] as the five most popular commands asked about, while common commands like `push`[8], `pull`[9] or `add`[10] have more individual questions associated, but less views overall [25]. According to the documentation, most of the popular talked about commands are used to tackle issues encountered when working with Git as a VCS [7], are not part of the day-to-day workflow, and can easily corrupt the existing database if executed incorrectly, which is also eminent from the design analysis of Perez De Rosso and Jackson [18]. They discuss in detail, why a lot of concepts of Git are unintuitive and simple workflows often times fail, due to unexpected side-effects [18].

Regarding difficulty of singular commands, Yang et al. [25] looked into commands with high percentage of associated questions with no accepted answer, for example, threads asking about `pack-redundant`[11], `http-push`[12] or `citool`[13]. Those commands are very specific to a certain scenario, only have a small number of associated threads, and are not applicable to a lot of use cases in general.

[3] "Revert some existing commits." [12]
[4] "Manage reflog information." [12]
[5] "Stash the changes in a dirty working directory away." [12]
[6] "Remove untracked files from the working tree." [12]
[7] "Reset current HEAD to the specified state." [12]
[8] "Update remote refs along with associated objects." [12]
[9] "Fetch from and integrate with another repository or a local branch." [12]
[10] "Add file contents to the index." [12]
[11] "Find redundant pack files." [12]
[12] "Push objects over HTTP/DAV to another repository." [12]
[13] "Graphical alternative to git-commit." [12]

Moreover, the time to get an answer to those posts is very long compared to other commands.

In contrast, frequently occurring questions without accepted answer seem to be about commands more commonly used by developers. Examples include `clone`[14], `blame`[15] and `difftool`[16] [25]. Therefore, those commands are not only difficult to use, but are also asked a lot about. This behavior was also observed by Treude et al. [24], who concluded, "that Stack Overflow is particularly effective at code reviews, for conceptual questions and for novices", not matching the nature of Git commands. Interestingly, the time needed to receive a satisfying answer to those threads is also way longer than the average time a question gets answered on Stack Overflow [25].

On a general scope, Alghamdi et al. [3] looked into how websites, especially Stack Overflow, are used during development. They noted, that "failure to fully understand code snippets also impacts the propagation of problematic code" [3]. This issue also applies to Git commands, while smaller in scope, they have similar potential to damage a repository or not yield the expected results [18]. Alghamdi et al. [3] use the term "Google effect" to describe offloading the responsibility for retaining information to the internet, resulting in the false assumption of knowing something. This may lead to the observed issues with proficiency deficits regarding Git, as most developers regularly consult online-resources for help instead of knowing commands by heart [19]. Fortunately, Meldrum et al. [16] found, that code snippets on Stack Overflow were not of "alarmingly poor quality". Alghamdi et al. [3] note, that the role of memory in successful programming is relatively under-explored and research needs to be conducted, whether the "Google effect" may be evident, when using platforms like Stack Overflow.

4.2 Learning Git

As second part of their work, Yang et al. [25] investigated, how developers learn Git commands. They recorded, that all of their survey participants have at least a bachelor's degree and more than two thirds of them have more than five years of experience in using Git. Regarding the self-evaluation of the expertise level of Git-command usage, they noticed, that only a small number of respondents thought of themselves as proficient, while the majority considered themselves as "advanced beginners" or only "competent", many of whom having at least five years of experience in using the tool [25]. They concluded, that these results coincide with the prior observed fact, that even experienced developers still encounter issues with Git. In addition, their results show, that most users learn commands from the internet or technical documentation, and only a small fraction is learned during classes. This is also eminent from the work of Alghamdi et al. [3], who noted, that "both students and professionals make significant

[14] "Clone a repository into a new directory." [12]
[15] "Show what revision and author last modified each line of a file." [12]
[16] "Show changes using common diff tools." [12]

use of online resources", which is also supported by the results of Rekha and Venkatapathy [19].

The results from the developed tool by Jia et al. [15] show, that they can provide recommendations based on keywords entered by a user, but are unable to provide sequences of commands, as more research into sequential relationships is needed.

Eraslan et al. [10] list several errors and poor practices by students using Git, for example, forgetting to branch, having infrequent commits or merging incorrectly. To address this shortcoming, Sproull [20] used a Git tutorial and lightweight interventions in assignments during an computer science course, resulting in improved knowledge about VCS concepts. A similar approach was applied by Bonakdarian [6], consisting of four incremental stages, introducing and using different topics rereading CLI usage, Git and GitHub. He observed, that students finished the course with a solid understanding of the command line and the capabilities of VCS. Moreover, he noticed, that students seem more confident in using Git, and suggested, that using the tool locally in the first stages eliminates a potential source of confusion, in contrast to working on a shared repository.

5 Discussion

> In this section we discuss the results and try to reason, why our results are valid in a sense of applicability to the research questions. We try to come up with recommendations for students and developers to further improve their usage of Git.

Regarding RQ2, we wanted to determine the level of expertise of Stack Overflow users and our results show, that the site seems to be a good source for answers to questions regarding basic usage of Git or similar applications, as a lot of users are experienced developers themselves with SOTA knowledge acquired through recent studies or continued exposure to the topic. On the other hand, if such a user encounters an uncommon problem and asks about it, the probability of an applicable answer seems unlikely. Unfortunately, this seems to be the case for most advanced features of Git, as users tend to be proficient in basic usage, but rarely encounter unconventional use-cases. RQ1, asking whether Stack Overflow a good source for learning Git commands, can be addresses in a sense, that most frequent users of VCS are familiar with both, basic concepts and application in their development cycle, but struggle with many features due to lack of familiarity and documentation.

The large number of questions related to basic features also show, that more focus needs to be laid in educating students in the basic properties, syntax and semantics of VCS. To answer RQ3, asking how we can improve confidence and ability to use version control, we want to increase proficiency with Git and similar software. The results from student surveys [6,20] show, that with additional exercises and tutorials, the hurdles of getting into being able to use

Git as development tool are easier to overcome. RQ3.1, where we wanted to come up with recommendations to learn Git, can be summed up, that to learn basic features of Git by heart, students and self-taught developers need practice becoming proficient in applying version control efficiently without needing to look up syntax or semantics.

Unfortunately, this approach does not cover advanced and uncommon features of Git, which itself needs better documentation aiming at improved accessibility of VCS tools in general. A possible improvement could be achieved by including use cases and extensive examples in the technical documentation.

6 Conclusion

To close this review, we repeat our main findings and conclude with possible guidelines to improve the usage of Git.

We observed, that many experienced developers struggle with using Git to its full potential and often refer to Stack Overflow for help. On the other hand, we noted, that Stack Overflow seems like a good resource to look up syntax and semantics of commands and discuss VCS related topics. Unfortunately, the tool Git itself has conceptional flaws, leading to continued inquiries about its features. To tackle this, the proficiency of students needs to be improved by including tutorial and exercises about VCS in basic courses, so that student get an early introduction and are able to build on a solid knowledge-base about Git.

References

1. Git workflow and rebase vs merge questions (2023), https://stackoverflow.com/questions/457927/git-workflow-and-rebase-vs-merge-questions, last accessed 8 Jan 2024
2. Why should i use version control? (2023), https://stackoverflow.com/questions/1408450/why-should-i-use-version-control, last accessed 8 Jan 2024
3. Alghamdi, O., Clinch, S., Skeva, R., Jay, C.: How are websites used during development and what are the implications for the coding process? Journal of Systems and Software **205** (2023). https://doi.org/10.1016/j.jss.2023.111803
4. Bertino, N.: Modern version control: creating an efficient development ecosystem. In: Proceedings of the 40th annual ACM SIGUCCS conference on User services. SIGUCCS '12, ACM (2012). https://doi.org/10.1145/2382456.2382510
5. Bock, T., Alznauer, N., Joblin, M., Apel, S.: Automatic core-developer identification on GitHub: A validation study. ACM Transactions on Software Engineering and Methodology **32**(6) (2023). https://doi.org/10.1145/3593803
6. Bonakdarian, E.: Pushing Git & GitHub in undergraduate computer science classes. Journal of Computing Sciences in Colleges **32**(3) (2017)
7. Chacon, S., Straub, B.: Pro Git. Apress (2014). https://doi.org/10.1007/978-1-4842-0076-6, https://git-scm.com/book/en/v2
8. Dada, O.A., Obaido, G., Sanusi, I.T., Aruleba, K., Yunusa, A.A.: Hidden gold for it professionals, educators, and students: Insights from Stack Overflow survey. IEEE Transactions on Computational Social Systems **10**(2) (2023). https://doi.org/10.1109/tcss.2022.3151130

9. Diyanati, A., Sheykhahmadloo, B.S., Fakhrahmad, S.M., Sadredini, M.H., Diyanati, M.H.: A proposed approach to determining expertise level of Stack Overflow programmers based on mining of user comments. Journal of Computer Languages **61** (2020). https://doi.org/10.1016/j.cola.2020.101000
10. Eraslan, S., Ríos, J.C.C., Kopec-Harding, K., Embury, S.M., Jay, C., Page, C., Haines, R.: Errors and poor practices of software engineering students in using Git. In: Proceedings of the 4th Conference on Computing Education Practice 2020. CEP 2020, ACM (2020). https://doi.org/10.1145/3372356.3372364
11. Fay, R.: Avoiding git disasters: A gory story (2011), https://www.randyfay.com/node/89, last accessed 20 Feb 2024
12. Git: Git reference (2023), https://git-scm.com/docs/git, last accessed 8 Jan 2024
13. GitKraken: GitKraken client (2023), https://www.gitkraken.com/git-client, last accessed 8 Jan 2024
14. GNU: GNU bash (2023), https://www.gnu.org/software/bash/, last accessed 8 Jan 2024
15. Jia, H., Yang, W., Shen, C., Pan, M., Zhou, Y.: Git command recommendations using crowd-sourced knowledge. Information and Software Technology **159** (2023). https://doi.org/10.1016/j.infsof.2023.107199
16. Meldrum, S., Licorish, S.A., Owen, C.A., Savarimuthu, B.T.R.: Understanding stack overflow code quality: A recommendation of caution. Science of Computer Programming **199** (2020). https://doi.org/10.1016/j.scico.2020.102516
17. Microsoft: cmd (2023), https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/cmd, last accessed 8 Jan 2024
18. Perez De Rosso, S., Jackson, D.: What's wrong with Git?: a conceptual design analysis. In: Proceedings of the 2013 ACM international symposium on New ideas, new paradigms, and reflections on programming & software. SPLASH '13, ACM (2013). https://doi.org/10.1145/2509578.2509584
19. Rekha, V.S., Venkatapathy, S.: Understanding the usage of online forums as learning platforms. Procedia Computer Science **46** (2015). https://doi.org/10.1016/j.procs.2015.02.074
20. Sproull, T.: Do students Git it?: A lightweight intervention to increase usage of advanced Git features. In: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2. SIGCSE 2022, ACM (2022). https://doi.org/10.1145/3478432.3499053
21. Stack Exchange Inc.: Stack Exchange data dump (2023), https://archive.org/details/stackexchange, last accessed 8 Jan 2024
22. Stack Overflow: 2020 developer survey (2023), https://survey.stackoverflow.co/2020/, last accessed 8 Jan 2024
23. Stack Overflow: 2023 developer survey (2023), https://survey.stackoverflow.co/2023/, last accessed 8 Jan 2024
24. Treude, C., Barzilay, O., Storey, M.A.: How do programmers ask and answer questions on the web? (NIER track). In: Proceedings of the 33rd International Conference on Software Engineering. ICSE11, ACM (2011). https://doi.org/10.1145/1985793.1985907
25. Yang, W., Zhang, C., Pan, M., Xu, C., Zhou, Y., Huang, Z.: Do developers really know how to use Git commands? a large-scale study using stack overflow. ACM Transactions on Software Engineering and Methodology **31**(3) (2022). https://doi.org/10.1145/3494518