On the Effect of Discussions on Pull Request Decisions

Mehdi Golzadeh, Alexandre Decan, Tom Mens
Software Engineering Lab, University of Mons
Mons, Belgium
{mehdi.golzadeh, alexandre.decan, tom.mens}@umons.ac.be

Abstract

Open-source software relies on contributions from different types of contributors. Online collaborative development platforms, such as GitHub, usually provide explicit support for these contributions through the mechanism of pull requests, allowing project members and external contributors to discuss and evaluate the submitted code. These discussions can play an important role in the decision-making process leading to the acceptance or rejection of a pull request. We empirically examine in this paper 183K pull requests and their discussions, for almost 4.8K GitHub repositories for the Cargo ecosystem. We investigate the prevalence of such discussions, their participants and their size in terms of messages and durations, and study how these aspects relate to pull request decisions.

Index terms— collaborative development, pull requests, discussions, software repository mining, empirical analysis[1]

1 Introduction

Today's open source software development is increasingly relying on third-party contributors. Developers contribute to different projects on online distributed development platforms like GitHub. The collaborative nature of software development it an inherently social phenomenon [1, 2]. GitHub embraces this social nature by extending the traditional git workflow with collaboration mechanisms such as *pull requests* (PR) and commenting. The pull-based development process [3] constitutes the primary means for integrating code from thousands of developers. It allows developers to participate in many projects without having direct commit access. The primary advantage of a PR is the decoupling of the development effort from the decision to merge the result to the project's codebase. It helps developers to avoid frequent merge conflicts with other contributors.

Through a built-in commenting mechanism, project integrators can review the code submitted in a PR, and ask contributors to improve their code, add documentation and tests before deciding to integrate it [4, 5]. Therefore, the history of commenting activity on a PR (including all pull request comments and pull request review comments) provides a valuable source of information. It enables analysis of who was involved in the discussion about a PR (e.g. the PR creator, project integrators, or other contributors). The discussions that take place between the author of the PR and the project integrators may play a key role in the ultimate decision to merge the PR into the code base, if the concerns raised by the project integrators were properly addressed or discussed carefully by the PR author.

While many studies have focused on the importance of having successful PRs [6–9], there is much less research on understanding the effect of the presence of discussions on the decision to accept or reject a PR. Our research aims to empirically study the relation between the PR commenting history and the final PR decision. As preliminary steps, we focus in this paper on three research questions:

RQ_1 *How prevalent are discussions in PRs?* helps us to determine whether the research goal is worthwhile to pursue: if there is only a limited number of PRs with discussions, then we will not be able to draw statistically significant conclusions on their relation with PR decisions. We show that most PRs

In: D. Di Nucci, C. De Roover (eds.): Proceedings of the 18th Belgium-Netherlands Software Evolution Workshop, Brussels, Belgium, 28-11-2019, published at http://ceur-ws.org

[1]This research is supported by the joint FNRS / FWO Excellence of Science project SECO-ASSIST and FNRS PDR T.0017.18.

have at least a few comments and a few participants involved in their discussions, and that the presence of a discussion is related to the decision. In RQ_2 *Who is involved in PR discussions?* we identify and group participants based on their role in a PR. We report about their combined presence in discussions and exhibit a relation between a PR decision and the participants that are involved in its discussion. Finally, in RQ_3 *How long are discussions?* we measure discussion length in terms of time and of number of comments and show how they relate to a PR decision.

The remainder of this paper is organized as follows. Section 2 provides the necessary background of studies related to PRs and comments. Section 3 presents the data extraction and methodology. Section 4 presents the preliminary results for the above research questions. Section 5 discusses the threats to validity of our study. Section 6 summarises the main findings and outlines future work.

2 Background

Distributed software development on shared online GitHub repositories is very frequently following a pull-based development process [3–5]. Any contributor can create forks of a repository, update them locally by contributing code changes and, whenever ready, request to have these changes merged back into the main branch by submitting a PR [10]. This pull-based software development model offers a distributed collaboration mechanism that allows developers to contribute code in a way that makes code changes trackable and reviewable by version control systems. This review mechanism has the additional effect of increasing awareness of all changes and allows the developer community to form an opinion about the proposed changes and the ultimate merge decision [11]. Many empirical studies have targeted pull requests from different points of view, including evaluation of PRs through discussion [6], factors influencing acceptance or rejection [8, 9, 12, 13] and, predicting potential future contributors [14].

Moreover, there are studies which analyze the content of PR to recommend core member to review, analyze, evaluate and integrate PRs [15–19], recommend PRs with high priority [20], study the effect of geographical location of contributors on evaluation of PRs [21], and gender bias in PR acceptance or rejection [22]. Some studies targeted code reviews to study the reasons and impact of confusion in code reviews [23], linguistic aspects of code review comments [24], the impact of continuous integration on code reviews [25], the challenges faced by code change authors and reviewers [26], how developers perceive code review quality [27], how presence of bots and the effect of organization and developer profiles on the PR decision [7].

3 Methodology

To carry out our empirical investigation, we need a dataset containing a large number of repositories and PRs. The dataset should exclude git repositories that have been created merely for experimental or personal reasons, or that only show sporadic traces of activity and contributions [28]. Registries of reusable software packages (e.g., npm for JavaScript, Cargo for Rust, or PyPI for Python) are good candidates to find such repositories, as they typically host thousands of active software projects, and as one can expect most of them to have an associated git repository.

We selected the Cargo package registry for the Rust programming language, because it contains tens of thousands of projects, and a large majority of them (nearly 85%) is being developed on GitHub. As both Cargo and Rust are quite recent (Rust was introduced in 2011), they contain a large number of repositories, even after filtering out those that are inactive in terms of contributions and discussions related to these contributions.

We relied on libraries.io data dump to extract the metadata for more than 15K Cargo packages [29]. We filtered out 1,571 packages that did not have any associated git repository and 413 packages whose repository is not hosted on GitHub. Not all git repositories were still available at the time we extracted the data, and our final list of repositories is composed of 9,954 candidates. For each of these repositories, we retrieved using GitHub API its complete list of PRs and, for each PR, all related comments and PR review comments. We found that 5,210 repositories did not have any PRs, hence only 4,744 repositories were retained for further analysis, accounting for more than 188K PRs.

As our goal is to study the relation between discussions and PR decisions, we decided to remove all PRs for which no decision was (yet) taken. Such PRs represent a small fraction of our dataset (around 2.6%). Our final dataset contains more than 183K PRs, submitted by 13,623 contributors and accounting for nearly 1M comments.

For each PR in this dataset, we have access to its creation date, its decision date, its decision, the person that made that decision, the author of the PR, and all the comments that were made, including PR review comments. It is important to note that the very first comment visible in a PR corresponds to the PR description, and is not considered as a PR comment in this paper, following the distinction also made by GitHub. For each comment, we retrieved its creation

date and its owner. We distinguish between four categories of owners:

1. *author* corresponds to the contributor submitting the PR;
2. *integrator* refers to the person having accepted or rejected a previous PR in the same project;
3. *decider* refers to the integrator who accepted or rejected the PR currently under consideration; and
4. *other* corresponds to any other participant (e.g., users, bots, external contributors).

4 Research Results

RQ_1 How prevalent are discussions in PRs?

With this first research question, we aim to get insights into the prevalence of discussions in PRs. For each PR in the dataset, we computed its number of comments, its number of distinct participants and its number of *comment exchanges* between one of the integrators and the author, i.e., the number of times there is one comment from an integrator followed by an answer from the PR author. Fig. 1 shows the proportion of PRs having at least a given number of comments, participants, and comment exchanges.

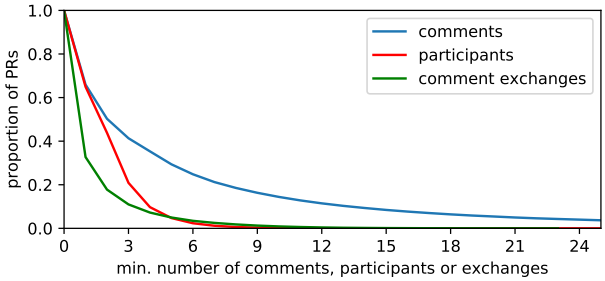

Figure 1: Proportion of PRs having at least a given number of comments, participants or comment exchanges.

We observe that while 48.8% of all PRs have at least two comments and 42.4% of all PRs have at least two participants, only 31.9% of them have comment exchanges. We also observe that all curves exhibit power law behaviour: the proportion of PRs is exponentially decreasing as the required number of comments, participants or exchanges increases. For instance, around 80% of all PRs have less than 8 comments, 3 participants and 2 comment exchanges.

Since the presence of comments, participants and/or comment exchanges could affect the acceptance or rejection of a PR, we computed the proportion of accepted (resp. rejected) PRs that have at least one comment (*has comments*), at least two participants (*has participants*) and at least one comment exchange (*has exchange*). Fig. 2 reports on these proportions. Note that by definition a comment exchange implies at least 2 participants, hence we have *has exchange* $\Longrightarrow$ *has participants* $\Longrightarrow$ *has comments*.

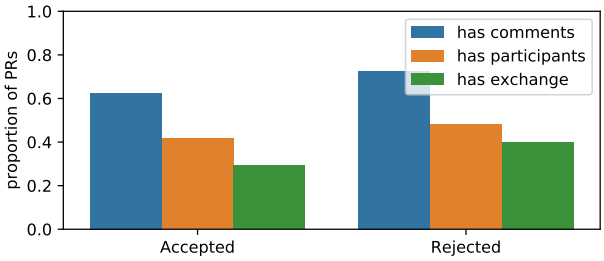

Figure 2: Proportion of accepted and rejected PRs w.r.t. the presence of comments and participants.

While we observe that a majority of PRs (regardless of their decision) have comments, proportionally more PRs have comments for rejected PRs (72.5%) than for accepted ones (62.4%). Similar observations can be made for the other criteria, suggesting a relation between PR acceptance and the presence of a comment/participant.

RQ_2 Who is involved in PR discussions?

This research question focuses on the participants that are involved in PR discussions. We distinguish between four categories of participants, as explained in Section 3. For each PR, each participant involved in the discussion was classified in *author, integrator, decider* or *other*. Fig. 3 shows the proportion of PR discussions in function of the presence of categories of participants.

We observe that the author of a PR is involved in most discussions (64%=6+12+3+3+3+4+20+13), as is the case for deciders (62%=11+9+20+12+3+4+1+2) and integrators (57%=6+9+1+1+3+4+20+13). Other participants are involved in only 23% (=2+1+4+3+3+3+1+6) of the discussions. We

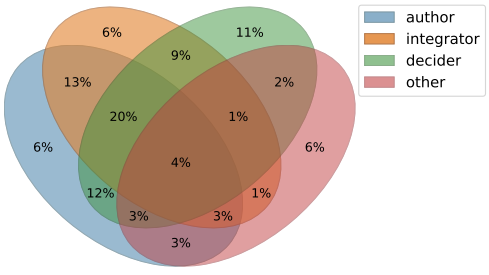

Figure 3: Proportion of PR discussions w.r.t. the presence of participants.

observe that the most frequent combinations of participants involve the author and some integrator/decider. For instance, the pair composed of author/integrator is the most frequent one (40%=13+20+4+3) followed by the pair author/integrator (39%=20+12+4+3). 24% (=20+4) of the discussions involve the author, an integrator and the decider. 29% (=6+6+11+6) of all cases involve a single participant only.

Similar to what was done for RQ_1, we grouped PRs according to their decision, and we computed the proportion of PRs with respect to the presence of participants of each category. Fig. 4 reports on these proportions.

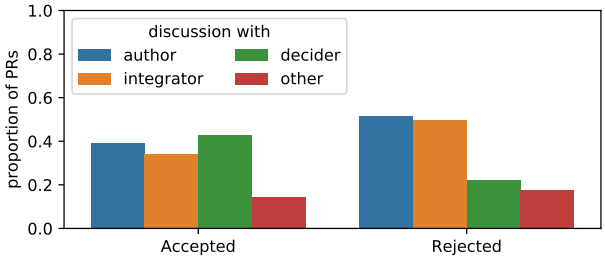

Figure 4: Proportion of PRs w.r.t. participants, grouped by PR decision.

We observe some interesting differences between accepted and rejected PRs mainly based on the presence of *authors* and *integrators*. 51.4% of rejected PRs involve the author of that PR and 49.6% involve an integrator, while for accepted PRs only 39.1% involve the author and 34.3% involve an integrator. While *integrators* are proportionally more involved in rejected than accepted PRs, the opposite is true when it comes to the *decider* of a PR: a *decider* is involved in 42.6% of accepted PRs but "only" in 22.0% of the rejected ones. Finally, when considering all other participants there is only a slight difference between accepted PRs (14.4%) and rejected PRs (17.4%).

RQ_3 How long are discussions?

The last research question focuses on the length of discussions in terms of number of comments and time between the first and last comment. We computed these two characteristics for discussions having at least 2 comments. These account for 49% of all PRs considered so far. The results are reported in Fig. 5, combining a scatter plot and two density plots (one for each considered characteristic).

We observe from the density plots that most discussions have a few comments and last for a short period of time. For instance, the median number of comments is 5 and the median duration is 0.7 days. We observe from the scatter plot a difference between discussions in accepted and rejected PRs, both for the

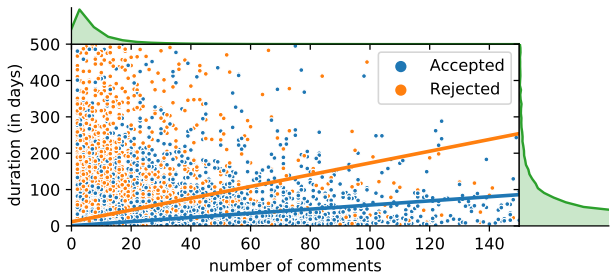

Figure 5: Scatter plot and density plots of discussion duration and number of comments.

number of comments and the duration. We statistically compared these distributions by means of Mann-Whitney-U tests. The null hypothesis was rejected in both cases ($p < 0.001$), indicating a statistically significant difference between these distributions. However, we found this difference to be *negligible* (Cliff's delta $|d| = 0.025$) for the number of comments [30,31], and *small* ($|d| = 0.219$) for the duration of these discussions, indicating a higher duration in rejected PRs than in accepted ones. For instance, the median duration is 1.69 days for rejected PRs and 0.6 for accepted ones.

The two regression lines superposed on the scatter plot reflect the average time between comments (i.e., the ratio between duration and comments). We computed this ratio for all considered discussions, and we statistically compared their distributions for accepted and rejected PRs using a Mann-Whitney-U test. We found a statistically significant difference between the two distributions ($p < 0.001$) and a *small* effect size ($|d| = 0.258$), indicating a higher discussion ratio in accepted PRs than in rejected PRs. For instance, the median average time between comments is 0.08 for accepted PRs, and 0.26 for rejected PRs.

5 Threats to Validity

Since our analyses are based on data from git repositories on GitHub, our results may be exposed to the usual threats related to mining data from GitHub such as *"a large portion of repositories are not for software development"* and *"two thirds of projects are personal"* [28]. However, given that our dataset is composed of git repositories related to Cargo projects, it is unlikely to be affected by such threats. On the other hand, the selection bias induced by our dataset being exclusively based on repositories related to Cargo projects is a threat to *external validity* [32], since the results and conclusions cannot be generalized outside the scope of this study.

The main threat to *construct validity* is that *"most pull requests appear as non-merged even if they are actually merged"* [28], potentially leading to an over-

estimation of the number of rejected PRs to the detriment of accepted ones. Fully addressing this threat is not possible, but we could rely on heuristics to detect whether PR commits are actually part of the main branch. Such heuristics are likely to change the figures reported in this paper, but are unlikely to affect the findings we obtained. Indeed, even if some PRs were wrongly identified as non-merged (=rejected), we already exhibited differences in PR discussions between accepted and rejected PRs.

Another threat to construct validity stems from the presence of bots and contributors with multiple identities. We mitigated the problem of multiple identities by relying on GitHub usernames to identify contributors instead of the "author" field values. We did not consider the presence of bots in this work. This may have led to an overestimation of the number of comments and participants, but our findings should not be significantly affected, assuming that bots represent only a fraction of the considered comments. In our future work, we will study heuristics to detect bot comments in order to take them into account in our analyses.

Finally, the lack of distinction between the different types of comments in our dataset represents a threat to *internal validity*. Not all comments are equal, but have been treated as such in this work. We did not differentiate based on the size or content of the comments. Similarly, we did not distinguish between PR comments and PR review comments, even if they do not serve the same purpose. Making such distinctions can potentially lead to different results, and will be explored in future work to gain additional insights.

6 Conclusion

In this preliminary research, we empirically studied 183K PRs and their discussions, accounting for around 1M comments. We showed that discussions are prevalent in PRs and there are proportionally more comments, participants and comment exchanges for rejected PRs than for accepted ones. We identified and grouped participants based on their role in a PR, and showed that a majority of discussions involved the author, the decider or one of the integrators. We showed that the presence of these participants is related to PR decisions.

Finally, we considered discussion length in terms of duration and number of comments. We observed that most discussions have only a few comments and do not last for long. While we have not found large differences between accepted and rejected PRs based on their number of comments, we found that discussions in rejected PRs are longer, and that discussions in accepted PRs are more intense.

This paper is part of a broader study and our intention is to gain a deeper understanding of the dynamics and patterns of discussions in pull requests, and their impact on PR decisions. Our goal is to provide techniques and tools to allow the community to perform better. Reducing the time to make decisions for pull requests can help the community to encourage better contributions by reducing the time required to reject contributions of insufficient quality or relevance, and by reducing the time to review and accept positive contributions. Moreover, based on the insights obtained during this study we aim to develop techniques to increase the productivity of contributions in terms of code quality and contribution time.

References

[1] Laura A. Dabbish, H. Colleen Stuart, Jason Tsay, and James D. Herbsleb. Social coding in GitHub: transparency and collaboration in an open software repository. In *Int'l Conf. Computer Supported Cooperative Work*, pages 1277–1286, 2012.

[2] Tom Mens, Marcelo Cataldo, and Daniela Damian. The social developer: The future of software development. *IEEE Software*, 36, January–February 2019.

[3] Georgios Gousios, Martin Pinzger, and Arie van Deursen. An exploratory study of the pull-based software development model. In *International Conference on Software Engineering*, pages 345–355. ACM, 2014.

[4] G. Gousios, A. Zaidman, M. Storey, and Arie van Deursen. Work practices and challenges in pull-based development: The integrator's perspective. In *International Conference on Software Engineering*, volume 1, pages 358–368. IEEE, May 2015.

[5] Georgios Gousios, Margaret-Anne Storey, and Alberto Bacchelli. Work practices and challenges in pull-based development: The contributor's perspective. In *International Conference on Software Engineering*, pages 285–296. ACM, 2016.

[6] Jason Tsay, Laura Dabbish, and James Herbsleb. Let's talk about it: Evaluating contributions through discussion in github. In *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2014, pages 144–154, New York, NY, USA, 2014. ACM.

[7] Olga Baysal, Oleksii Kononenko, Reid Holmes, and Michael W. Godfrey. Investigating techni-

cal and non-technical factors influencing modern code review. *Empirical Software Engineering*, 21(3):932–959, Jun 2016.

[8] Mohammad Masudur Rahman and Chanchal K. Roy. An insight into the pull requests of GitHub. In *Working Conference on Mining Software Repositories*, pages 364–367. ACM, 2014.

[9] Di Chen, Kathryn T. Stolee, and Tim Menzies. Replication can improve prior results: A github study of pull request acceptance. In *Proceedings of the 27th International Conference on Program Comprehension*, ICPC '19, pages 179–190, Piscataway, NJ, USA, 2019. IEEE Press.

[10] Y. Yu, H. Wang, V. Filkov, P. Devanbu, and B. Vasilescu. Wait for it: Determinants of pull request evaluation latency on GitHub. In *Working Conference on Mining Software Repositories*, pages 367–371, May 2015.

[11] Jason Tsay, Laura Dabbish, and James Herbsleb. Influence of social and technical factors for evaluating contribution in GitHub. In *International Conference on Software Engineering*, pages 356–366. ACM, 2014.

[12] Igor Steinmacher, Gustavo Pinto, Igor Scaliante Wiese, and Marco A. Gerosa. Almost there: A study on quasi-contributors in open source software projects. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, pages 256–266, New York, NY, USA, 2018. ACM.

[13] M. Wessel, I. Steinmacher, I. Wiese, and M. A. Gerosa. Should i stale or should i close? an analysis of a bot that closes abandoned issues and pull requests. In *2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE)*, pages 38–42, May 2019.

[14] Damien Legay, Alexandre Decan, and Tom Mens. On the impact of pull request decisions on future contributions. *arXiv e-prints*, page arXiv:1812.06269, Dec 2018.

[15] Y. Yu, H. Wang, G. Yin, and C. X. Ling. Reviewer recommender of pull-requests in GitHub. In *International Conference on Software Maintenance and Evolution*, pages 609–612. IEEE, Sep. 2014.

[16] Y. Yu, H. Wang, G. Yin, and C. X. Ling. Who should review this pull-request: Reviewer recommendation to expedite crowd collaboration. In *Asia-Pacific Software Engineering Conference*, volume 1, pages 335–342, Dec 2014.

[17] Manoel Limeira de Lima Júnior, Daricélio Moreira Soares, Alexandre Plastino, and Leonardo Murta. Developers assignment for analyzing pull requests. In *ACM Symposium on Applied Computing*, pages 1567–1572. ACM, 2015.

[18] Jing Jiang, J.-H He, and X.-Y Chen. Coredevrec: Automatic core member recommendation for contribution evaluation. *Journal of Computer Science and Technology*, 30:998–1016, 09 2015.

[19] Manoel Limeira de Lima Júnior, Daricélio Moreira Soares, Alexandre Plastino, and Leonardo Murta. Automatic assignment of integrators to pull requests: The importance of selecting appropriate attributes. *Journal of Systems and Software*, 144:181 – 196, 2018.

[20] E. v. d. Veen, G. Gousios, and A. Zaidman. Automatically prioritizing pull requests. In *Working Conference on Mining Software Repositories*, pages 357–361. IEEE, May 2015.

[21] Ayushi Rastogi, Nachiappan Nagappan, Georgios Gousios, and André van der Hoek. Relationship between geographical location and evaluation of developer contributions in github. In *International Symposium on Empirical Software Engineering and Measurement*. ACM, 2018.

[22] Josh Terrell, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson Murphy-Hill, and Chris Parnin. Gender bias in open source: Pull request acceptance of women versus men. 01 2016.

[23] Felipe Ebert, Fernando Castor, Nicole Novielli, and Alexander Serebrenik. Confusion in code reviews: Reasons, impacts, and coping strategies. pages 49–60, 02 2019.

[24] Vasiliki Efstathiou and Diomidis Spinellis. Code review comments: Language matters. *CoRR*, abs/1803.02205, 2018.

[25] M. M. Rahman and C. K. Roy. Impact of continuous integration on code reviews. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pages 499–502, May 2017.

[26] L. MacLeod, M. Greiler, M. Storey, C. Bird, and J. Czerwonka. Code reviewing in the trenches: Challenges and best practices. *IEEE Software*, 35(4):34–42, July 2018.

[27] O. Kononenko, O. Baysal, and M. W. Godfrey. Code review quality: How developers see it. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, pages 1028–1038, May 2016.

[28] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. The promises and perils of mining GitHub. In *Int'l Conf. Mining Software Repositories*, pages 92–101. ACM, 2014.

[29] Jeremy Katz. Libraries.io open source repository and dependency metadata (version 1.4.0) [data set]. http://doi.org/10.5281/zenodo.2536573, 2018.

[30] N. Cliff. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3):494–509, 1993. cited By 364.

[31] Jeanine Romano, Jeffrey D Kromrey, Jesse Coraggio, Jeff Skowronek, and Linda Devine. Exploring methods for evaluating group differences on the NSSE and other surveys: Are the t-test and Cohen's d indices the most appropriate choices? In *Annual Meeting of the Southern Association for Institutional Research*, 2006.

[32] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, and A. Wesslen. *Experimentation in Software Engineering - An Introduction*. Kluwer, 2000.