— Rebuttal —

Original Title	How to Analyze Build Logs: A Systematic Survey and Comparative Study of Chunk Retrieval Techniques
Authors	Carolin Brandt, Moritz Beller
Corresponding author	Carolin Brandt (c.e.brandt@tudelft.nl)
Revision date	TODO
Journal	IEEE Software

Dear Editor,
Dear Reviewers,

We thank reviewers for their in-depth and constructive feedback. All reviewers are positive about the relevance of the paper for IEEE Software's readership (assessing it Very Relevant or Relevant). Reviewer 2 calls it a "great paper" and Reviewer 1 "clearly written and easy to read," "highly relevant and fits well for IEEE Software and its target audience." Reviewer 3, while being perhaps most critical overall, notes that "[the paper] focuses on a much needed dual perspective integrating the developer's own perception to measure productivity. The paper is very well motivated and overall easy to read."

That said, there were a number of points that needed to be changed in the paper, in particular as regards scope and structure. We have made several improvements to the paper following your recommendations for this major revision. Key changes include:

- Re-structuring the manuscript, by putting visualizations into a side bar to make the article focused on the productivity study alone.
- Adding further details on study methodology and execution to the Telemetry and Regression Model sections while keeping the spirit of a light-weight, practitioner-oriented paper. In particular, we made the section on regression models easier to read for readers without a statistics background.
- Fixing the detailed remarks by the reviewers (throughout the paper).

Below we outline the remarks from the reviewers and our response in e-mail quotation and answer style.

Best regards,
Moritz Beller

Editor Comments

> The reviewers identified a number of severe limitations that must be addressed before the paper can be reconsidered for publication. The reviewers reported that the paper is too long and difficult to read; a general reorganization is needed to make the intended contribution more accessible. The reviewers also called for a discussion of motivations for this work and highlighted several conceptual limitations in the study. The presentation of results must be improved to better highlight strengths and weaknesses of the techniques; implications for researchers and practitioners must be better discussed. Comparison among techniques and literature review needs improvements, too. The accurate concerns of reviewers are so serious and numerous that addressing them means doing further substantial work. This is not to say that this work has no interest or value. Indeed, the reviewers agree that the paper deals with a relevant problem; some further work along the detailed directions indicated by the reviewers might result in a better contribution.

Reviewer 1

The authors conducted a literature review to better understand prior approaches that analyze build logs to identify build failures. Then, the authors proposed and evaluated three novel approaches to identify build failures.

The paper provides a good contribution to log analysis literature. However, I believe that the paper should benefit from a large body of improvements. The paper is very difficult to read, I had to go back and forth to understand many sections and many details appear too late in the paper. I have some concerns about the literature review, their 3 proposed approaches, as well as the evaluation of the approaches. More details are below.

Abstract: mention the exact number of studied log files and projects.

Introduction:

The authors mention that a guideline on when to use which of their three proposed techniques is a contribution. I think this is just a discussion of the evaluation and not a novel contribution.

The paper also mentions a set of ideas for future work. That is part of a future work section and not a contribution by itself.

The authors mention in Figure 1 that they propose 3 “promising” approaches. Why they are promising?

Section 2:

The authors performed backward snowballing to include more papers to their literature study. However, they missed the forward snowballing.

The authors mentioned that they used Google scholar to find their papers. They should instead use official scientific publications databases as suggested by Petersen et al. (the guideline the authors followed for the literature review) as well.

The paper mentions that the first author and the second author first studied five papers, then discuss when disagreement occurs to find a consensus. Then, each of the two authors studied a separate number of papers. I think that five papers are too low number and the disagreement should be discussed in the paper. That is because if the two authors disagree on all the classifications of the first 5 papers, they are more likely to disagree on the classification of the remaining papers. By the way, why 5? 5 is not even 10

The authors used just one keyword search (i.e., “build log”). What about other search keywords? E.g., build trace.

Could you provide more details on how you excluded the papers that are related to system logs? I would suggest adding a table that summarizes the inclusion, exclusion criteria and the rationale behind each of them.

Extension papers should not be excluded. That is, in particular, because the authors discuss the approaches that are proposed in the literature and their limitations. Maybe, some limitations are addressed in the extension and not in the original paper.

Subsection 2.4.1 is not part of the results of the paper. In other words, where the papers are published is not the goal of the paper or the analysis, so it’s not a result. You might include 2.4.1 in 2.3.

Change figure 5 to a table that shows which papers focus on which category so readers can go back to the papers that consider a given category. That table should also include a description of each category.

The following finding is not clear: “The majority of works (46

“31

Please add a description to each log analysis technique to Table 1, and no need for a barplot to show the frequency in the last column of the table. You can put just a number and save some space for the description. That is in particular since the definition of the categories of Table 1 is scattered all over the Section.

You can’t make the following association: “we rarely saw works justifying why they use a certain technique to analyze build logs as there is a general lack of guidance on when to use which technique”. That is because you can’t know why the authors didn’t justify why they use a technique.

Section 3:

What is the difference between the existing approaches and your three approaches?

From where are your 3 proposed approaches coming from? Although the authors claimed that “previous work seemed to choose a log analysis technique arbitrary ...” in Section 4, they didn’t justify their approaches neither.

The authors claim that some of the existing approaches are not generalizable to all the projects. The question is why the generalizability matters. One doesn’t need for her project something that is generalizable to others’ projects.

The authors mention “techniques … based on deep learning, are thinkable, but we did not evaluate them in this article since they did not emerge from the literature survey”. I do not fully understand this.

Does it mean that these techniques exist, but you missed them on the literature review? So the literature review is incomplete. Or are the deep learning techniques not used at all in the build log analysis, so no need to even mention them in the paper.

3.3., 3.4., 3.5 are so difficult to understand. Support these sections with examples (instead of having a subsection of examples that comes very late, i.e., 3.5).

Fig. 7.a is no explaining "why" Android build failed, it's showing just the symptom of an error. So update the caption of the figure accordingly.

KWS comes after CTS in Table 2, while CTS comes before KWS in the textual explanation; so switch the order of KWS and CTS either in the text or in the Table.

A major concern I have about the baseline approach that is used in the evaluation. It's not clear why the authors compared their three novel approaches to a random-based approach instead of comparing their approaches to the existing studies or at least to state of the art approaches?

Section 4:

Section 4.3 is almost just repeating figures with very few discussions.

The authors didn't define what they mean by the "structural category" when most of the evaluation is around that word. I had a hard time to deduce what it means.

The authors classified the recalls into three categories: successful (i.e., recall = 1), unsuccessful (i.e., recall = 0), and everything in between is classified as partially successful. That is not fair since the recall of 0.001

Do you measure precision and recall at the file or project level? By the way, what is the distribution of log files across the projects that you studied? Does the evaluation consider projects separately?

Do you consider a training set from different projects or do you train and test within the same project? That should be discussed and justified.

For Fig. 12, I'm wondering what one example means? Are you able to accurately identify the chunks from one example? It's surprising. How come the approach is that good from one example? That comes to another concern, can you discuss the why of your evaluation. Why the approach works well and why it doesn't work well?

For Fig. 15, why do you use a barplot instead of a violin plot similar to the other figures? That what also triggered my comments about how do you conduct the evaluation and which data do you consider in the training and testing (i.e., do you train and test the approach on the whole data level, at a project level, or at a file level).

You can't report the execution time since you are using different infrastructures for each of your approaches. E.g., you used C# for one of the three approaches, and R for another approach. So, you can't know if one approach is slower than the other approaches because of the approach itself or because it's developed on a given programming language (e.g., R). For example, R loops are much slower than C# loops.

A lot of terms are used but never defined, which makes the paper difficult to read. E.g., what OR-based programs are? Please double-check this problem throughout the whole paper.

Fig. 16, use a statistical approach (e.g., Wilcoxon test) to compare the four approaches (i.e., the three approaches and the baseline).

Section 5:

I don't get what is the threat to validity on the implementation and the dataset? E.g., "text2vec ... is influencing our implementation of CTS". How it is influencing? What is the threat here and how did you mitigate it?

Reviewer 2

1. Paper summary ———————————— This paper provides a literature mapping survey in the emerging field of build log analysis. In a subsequent phase, the authors have developed three prototypical implementations for analyzing build logs. Their implementations have been validated through an empirical study on 797 build logs extracted from different projects written with a variety of programming languages and using different build automation tools. Their findings point out that there is no technique that is always better than the others. Indeed, based on the strengths and weaknesses of each technique, they provided guidelines on when to choose a specific technique.

2. General Evaluation ———————————— The paper addresses a very hot topic in the software engineering field; the study methodology for the literature mapping study is sound while the methodology adopted in the second phase is not completely clear and have some weaknesses. Finally, there are no practical implications. Most of the strengths and weaknesses seem to be related to the technique itself and are not related to their application to the emerging field of build logs analysis. Looking at the comparison among the different techniques, the methodology used is completely hidden, e.g., have you

trained your implementations by project? and how many build logs for each project do you have? or How different are in terms of information the build logs in your dataset per each project? and are your approaches aimed only at analyzing build failures? Finally, looking at the decision tree constructed for identifying the best technique to adopt, it seems to follow the technique and not the technique applied to the build logs. So, I do not think that your result as is, is ready for publication and is ready to be used by practitioners and/or researchers.

3.Points in Favor ———————————————— -Interesting literature mapping survey. -Appropriate identification of related literature. -Interesting topic.

4. Points against the paper ———————————————— -No practical usage of your findings. -No novelty in the provided results, strengths and weaknesses due to the techniques and not to their application on build logs (obvious results). -Study methodology not completely clear with some hidden details.

5. Suggestions for improvement ————————————————- -Looking at the literature review, among the 61 works are there some works that justify the reasons behind the adoption of a specific technique? This information has to be reported in the paper, and in presence of this information, it may be of interest to see how the results of the second phase are in line with what already stated in the literature. You state only that this occurs only rarely but I think that you need to go deeper on this, instead of only looking at which kind of information are developers/researchers looking at. -Your results are provided without discriminating among different build automation tools. Since the community knows that different build tools generate differently structured build logs it may be of interest to check whether there exists a technique that is able to work the best while applied to a specific kind of build automation tool. -Among the RQ1 results, you state that there is a need for defining a more generic solution for extracting useful information from build logs. However, both the design and the results in RQ2 do not seem to go in this direction. -Another unclear point while reporting the study methodology for RQ2 is how are your prototypical implementations generalizable? Each technique requires the identification of training examples. These ones have to be changed each time the build process changes into the system. -You stated that KWS is automatized, but how? How do the keywords have been defined? How the keywords can be generalized to different build automation tools and for different purposes? -Even if you have provided a section with a concrete example, it is unclear to me how your approach effectively works. In order to create regular expressions, PBE needs to have input/output chunks so can you be more detailed on how to run your techniques? What is the current input being provided? The same happens for the CTS where you talk about a search query to be compared with each line inside your build log but what is the search query you have used in your study? -In Section 3.6 you have defined a random baseline, however, a more realistic case to compare with is the one that only looks at the lines in the bottom part of the build logs. Usually, there you can find a possible reason for the failure. So think about using a different baseline. -The context of your study misses important information: how many different build automation tools are in your dataset? How many build logs do you have for each different build tool? What is the distribution of build logs by project in your dataset? -Clarify whether while running your techniques you have done a training phase per each project. The latter requires the answer to my previous point. -While comparing the results provided by the different techniques, did you have computed the metrics by project and then you have averaged the results? Please be more specific here otherwise it is not possible to follow the provided results. -As regards the presentation of your results, please clarify and discuss the results in each figure being attached to the paper, as well as make them more readable. The discussion phase may help the readers in understanding the guidelines you provide as the outcome of this study. -It may be of interest to know in how many of the 61 works the technique being used does not follow your guidelines. If the guidelines are always met probably researchers/developers do not need more guidance on this.

Reviewer 3

Build logs contain important information about the build process, and it is essential for software engineers to analyze these logs to investigate build errors and test failures. However, understanding the extensive build logs generated is a tedious task. Therefore, a foremost step in build log analysis is the retrieval of the relevant information. The authors' goals are to provide a comprehensive view of the existing techniques to extract relevant information from build logs and evaluate them. In order to achieve goals, the authors performed a Systematic Literature Review in the general area of build log analysis then implemented and evaluated three of the ten identified techniques.

The manuscript is well-written and presents substantial effort and different methods to convince readers about the importance of building log analysis. However, the manuscript presents some major issues.

Motivations issues

One of the soundness issues in the manuscript is the lack of consistent motivation. The motivation is supposedly motivated by two references. From reference [5], the authors claim that

“This manual activity is tedious and prone to errors [5].”

However, the reference [5] is a non-peer-reviewed manuscript of

M. Santolucito, J. Zhang, E. Zhai, and R. Piskac, “Statically verifying continuous integration configurations,” arXiv preprint arXiv:1805.04473, 2018.

And analysing Santolucite et al. [5] draft, we found that

“We demonstrate the functionality of VeriCI by using a real-world example of a TravisCI build failure from thesferik/railsadmin repository (bbenezech 2016) on GitHub. [...] Manually checking the log information is a tedious process and it is very difficult to gain any helpful information for understanding or correcting the issue.”

Thus, it is possible to conclude that the motivation claim is an anecdotal observation from a non-peer-reviewed article.

The second reference is from a previous work from one of the authors

[2] M. Beller, G. Gousios, and A. Zaidman, “Oops, my tests broke the build: An explorative analysis of travis ci with github,” in Proceedings of the 14th IEEE/ACM International Conference on Mining Software Repositories (MSR). IEEE, 2017, pp. 356–367.

The manuscript presents the fact from the reference that a single build log can be over 50 megabytes Large [2]. However, in paper [2], there is any discussion about the size of build logs. There is only

“As a consequence, the amount of information we can extract from a build log varies perbuild technology used.”

Thus, the manuscript lacks soundness and reliability when 1) it uses as motivation a non-peer-reviewed paper to present a general statement from anecdotal observation, and 2) it uses a previous work to present an incorrect observation.

Unfortunately, the manuscript presents essential soundness issues that compromise the main motivation and reliability, and it could not be relevant if the motivation arguments are false. Please, address those issues with more robust evidence about the problems with build log analysis.

Assuming that build log analysis is not an actual problem, the main RQ “What is the state-of-research to analyze build logs?” is irrelevant. Moreover, the claim “In summary, the outcome of our literature survey shows the need for a new generation of automated techniques to analyze build logs,” could be false, because if build log analysis is not an actual problem to engineers, it is evident that there is not in the literature solutions for a possible irrelevant problem.

However, I will assume the authors will provide strong evidence to support the claim that build log analysis is an important issue that justifies automatic mechanisms of log analysis in a new version of the manuscript to continue this report.

Log conceptual issues

Log mining/management solutions include three phases log collection, log abstraction/parsing and log analysis. The log abstraction component is responsible for abstracting the unstructured log messages in a log dataset into a structured log event list that is then inputted in the log analysis component. Also, log abstraction is a two-tier process - first, log message templates (or another form of matching rules) must be found (discovered), and only after that can be the incoming log messages abstracted into higher-level representations (matched).

Thus, the authors fail (1) to clearly distinguish between the log abstraction step and the log analysis step and (2) to distinguish between the log abstraction two separate phases. Therefore, we suggest the authors (1) clearly distinguish between the log abstraction step and the log analysis step; (2) highlight the two phases of log abstraction separately. I suggest the authors observe the discussion from the paper “A systematic literature review on automated log abstraction techniques, Information and Software Technology 122 (2020)” for further detail and description on log abstraction/parsing and as a related paper.

Besides, it would be beneficial for the readers to give an illustrative example of a build log to show how a system log differs from a build log. I suggest the authors add a section on build logs format and where/how they are created.

Systematic mapping issues

I am concerned about the use of Google scholar alone for an SLR. The authors could complement their search strategy with at least another source such as Scopus, Engineering village, IEEE xplore or ACM library to ensure the rigour, completeness, and repeatability of the process.

The authors did not perform a quality assessment filtering on the retrieved papers. If they did, I suggest the authors should clearly describe their quality assessment strategy and their quality assessment criteria. It is particularly important because the authors are considering grey literature in their SLR.

Why did the authors not perform forward sampling as well? The authors should add a justification for their decision. Also, please, describe in the manuscript the number of backward snowball levels were applied.

What are the inclusion/exclusion criteria for the third filter?

To evaluate the search string, I performed a simple search in Google Scholar using [“build log analysis”]. Surprisingly, I found several papers in which should be evaluated in systematic mapping. For example, why the papers

“Hirebuild: An automatic approach to history-driven repair of build scripts F Hassan, X Wang - 2018 IEEE/ACM 40th International . . . , 2018 - ieeexplore.ieee.org”

What are the factors impacting build breakage? Y Luo, Y Zhao, W Ma, L Chen - 2017 14th Web Information . . . , 2017 - ieeexplore.ieee.org

A quantitative study of java software buildability M Sulír, J Porubän - Proceedings of the 7th International Workshop on . . . , 2016 - dl.acm.org

are not in the mapping dataset?

I strongly recommend that authors perform a careful review of the search strings and present inclusion/exclusion criteria. Besides, authors should perform a card sort to analyze the works systematically.

Also, include, in the replicability kit, all raw data from each analysis step. There is no evidence about the intermediate analysis/filtering steps.

Minor: - The title of section 2 “systematic literature mapping survey” might lead to confusion, I advise to not use “survey” in this title. Same comment for the use of the term “systematic survey” in the paper can be replaced by “systematic mapping” Fig1. is useful but its size could easily be reduced without a negative impact. - Fig. 2 shows the 259 paper as the output of the first filter but in section 2.2 the output of the first filter is 256 papers.

The authors merge the results from peer-reviewed literature and grey literature. I can understand that the authors need to use grey literature in their SLR since the number of peer-reviewed papers is low. However, the authors should present the results from grey literature separately from the results from peer-reviewed literature.

- In Table 1, the authors present the identified build log analysis techniques. However, having an “analysis” technique for “build log analysis” is not informative and confusing. The same goes for “ scan”, “others” and “non identified”. The “machine learning” and “information retrieval” are very general, which ML techniques were used? I suggest the authors provide a clear and more detailed description/definition of each identified technique in Table 1.

The authors chose not to include and evaluate the machine learning techniques because they did not emerge from their literature survey and labeling is “unfeasible”. However, machine learning was the fourth most used technique in Table 1. Not considering at least one machine learning technique is a weak part in this work. I strongly suggest the authors expand their study and include ML techniques.

Chunk retrieval comparison issues

In section 4, authors claim that

“In the literature survey, we noted that previous work seemed to choose a log analysis technique arbitrarily and that is a lack of guidance on when to use which technique.”

As the literature mapping presents several issues, I can not assume that claim is true. However, I discuss some observations assuming that there is a lack of guidance.

The authors evaluate their chosen techniques according to precision and recall. However, many other characteristics are important for software engineers and practitioners to make an informed decision on the best techniques for their case such as scalability and efficiency. I suggest the authors discuss and consider these important metrics.

I do not understand the value of the decision tree for the given paper, given that it is the second classification of the same techniques after the classification introduced in Section 4.3 and Figures 16. I would suggest avoiding this section altogether.

I am not convinced that the orientation of the use-cases on different human-roles is a way to go. Why would a researcher need different information retrieval techniques than a software engineer? For example, is there a reason why one of them would want low recall or low precision of their information retrieval technique when having other structural category files? What if they have multiple structures and need to have both high recall and high precision? Wouldn’t it be better to map the decision tree to some real needs and the input data characteristics? Or drop it altogether?

Despite great effort from authors, the manuscript presents essential soundness issues that compromise the main motivation and reliability. It is not relevant if the motivation arguments are false. Please, address these issues carefully with more robust evidence about the problem with build log analysis. I recommend a revision on this paper due to the following major and minor concerns or suggestions.