As a service to the reader, here are are the threats identified by reviewers to the validity of the paper's conclusions that could not handled in the space available, by an already full paper or that the third author has not the knowledge to handle the question.

Comments by the third author are set off by "[" and "]"

[The history of the paper is important to understand:

The first author was a master's degree student of the second author. The first author graduated in 2015, and went to work in industry, not at the company whose software is the subject of the paper. The second author, submitted a paper in 2015 by the first author and himself to a conference, but it was rejected. He fell ill very soon after the rejection, and never recovered enough to to begin to revise the paper. The third author noticed the first author's thesis and needed to cite it or a paper derived from it in his work. He offered to the second author to revise the rejected paper to bring it to publication quality, at the price of becoming the third author. The third and second author met a few times to allow the third author to learn what he needed to know what to write. Just as the paper was finished for this submission, the second author passed away. May his memory be a blessing.

The third author thus lacks the expertise in the data to deal with some of the issues that did not come up during his conversations with the second author. The first author does not remember all the details, having moved on to another job.]

```
==================================================================
                    REVIEWER #1
==================================================================
```

Detailed Comments

...

- a threat to validity is that if a project started earlier is more likely to have less defects, as these may have been solved. This is the main weakness of the paper, so please address this threat. It is unclear why the authors selected ongoing projects. At page 6 the authors speak about project duration and time of completion, so I am a bit confused.

- the evaluation is made as a two-tailed test, while I would expect a one-tailed test, as one wants to see if one method generates more/less defects than the other, otherwise the authors can only state that there is a difference but not that one method is better than the other. This is another critical point, which need to be addressed for the paper to be accepted.

- the paper mixes introduction background and related work in a single section of three pages. It would be better to have a stand alone introduction section, with background and related works in other sections. The introduction should summarise what is the context, what is the approach and what are the results. Currently, most of the introduction is dedicated to background.

- it would be useful to have a comparative table in the background section, in which WM and

AM are compared.

- the background section does not differentiate between different agile methods, but Kanban and SCRUM could have differences that may affect the discussion. Please discuss this aspect.

- I think that the hypotheses are better expressed with reference to the mean, e.g., the average number of defects is the same.

- the authors should clarify how they decided whether a defect was requirements-related or not. It seems that most of the defects are requirements related, so I would like to see example of the two classes.

[Footnote 2 gives some sanitized examples of RRDs and implementation defects.]

....

- the evaluation appear to use the t-test, but it is not clear if the authors tested the hypotheses of the t-test, e.g., normality of the data.

[The first's author's filed thesis uses the Mann-Whitney U test, which does not require a normal distribution. Evidently (only the deceased second author knew for sure), the submitted paper, written mainly by the second author from the first author's thesis, is based on an earlier version of the thesis that incorrectly used the t-test. Therefore, we changed the paper to use the thesis's Mann-Whitney U test. Thanks for catching that.]

==
REVIEWER #2
==

Detailed Comments

...

A study like this comes with a very large number of confounds, i.e., things to be controlled for in order for internal validity to not be compromised. The authors do a good job summarizing many of them. The following, however, are additional concerns that came to my mind and I find also relevant:

1. The staring date of WM projects precedes that of AM projects. This normally means different people and leadership, organizational maturity, developer skill, project complexity, technologies and tool support, etc.. The authors seem to address this by focussing on projects that start around 2012 (the transition year) so that they are all contemporaneous -- but this seems difficult if taken literally: the WM projects must either start close enough to 2012 or be long enough to be active long past 2012. Some more details and justifications are probably pertinent. In addition, this includes including whether the projects are independent and what kinds of projects they are (e.g. development from scratch, development of an extension, configuration of a custom solution,

etc.). Further, when the company transitions to AM, it is likely to be in a learning mode for a period, i.e., they may not be doing AM properly for a period.

2. The company *claims* to have undergone a transition from WM to AM. But how do we know that the exact practices they followed actually match common understanding of the two models, as also nicely surveyed in the paper? Have the authors investigated any such contextual information?

3. From a construct validity viewpoint, numbers of work items and stories are considered not only measures of size but also mutually commensurable. Both assumptions seem problematic. The estimation community, for example, has well established tangible or conceptual units or size measurement (LOCs, function points). Is there literature that motivates measuring size using numbers of requirements statements? It would be counterintuitive because any two requirements statements may describe chunks of software of very different size, depending also on who writes them. Assuming that size is a measure of development/maintenance effort (is it?). I note that the authors seem to be ambivalent with regards to the need to take size into account and whether their ad hoc ratio constructions are the best way to do this. I return to the latter question below, but it seems clear to me that size is a major factor. Meaning that comparisons of the raw scores do not seem meaningful.

The second issue I have with the methodology is the statistical tests. From a technical standpoint: (a) a parametric test (versus a non-parametric equivalent) seems to be performed, but it is unknown if statistical assumptions are met, e.g., normality, homogeneity, etc. -- noting also that (i) independence among data points is questionable here, (ii) the sample is ultimately not random, (b) there does not seem to be regard for family-wise error which would require that the alpha threshold is corrected downwards becoming stricter (e.g., divided by the number of tests). These could be checked with the person who reportedly advises the authors on statistics and empirical methods.

My main question, however, is whether all this is even needed. We perform inferential statistics in order to generalize a finding to an entire population. But what is the population here? All projects of the specific company? All projects in Canada? All projects relating to transaction management for mortgage processing? It seems to me that a kind of analytical generalization (see Yin's Case Study Research) is more pertinent than the statistical one here. Thus, the authors can simply avoid t-tests and do a nice descriptive analysis and stick to and expand on the (good) comparison they have with relevant literature.

Regardless, assuming t-test are pertinent, the concern whether to relax the threshold from 0.05 to 0.1 seems irrelevant; one can report a confidence interval and effect sizes. On the question of the ratio of defects over size: to avoid constructing a ratio, one could probably perform a one-way ANCOVA with size as a covariate. But the statistics expert should be consulted, as sample size may be too small for such analysis. Again, I am not sure inferential statistics are important here; and, in my view, their absence does not make the study less "quantitative".

==
REVIEWER #3
==

Detailed Comments

It is unclear the motivation for a comparative study between these two methods in particular: agile and waterfall. I understand that AMs are widely used in industry today, but why study waterfall, why not compare against prototyping or iterative-incremental or other methods? I suggest improving the reason for selecting both types of methods.

I think the authors should better explain why they chose requirements defects detected after implementation as a quality metric to compare the two types of methods. Additionally, it is not clear enough if the defects to be counted are those identified by users during software operation, or during a so called "post-deployment" phase. Thus, it is not clear how the duration of that period or phase is established. I think the authors should clarify this.

There are several other issues not clear enough:

- Why were these three attributes project size (number of work items/use stories), team size and project duration used for project selection? Why not other attributes, such as software size (for example, number of lines of code or components)?

- How can authors estimate project duration using the time and effort spent implementing change requests? This question arises from what was said in point (2) of section 2.6. However, the calculation of the duration according to the first paragraph of section 2.7.3., contradicts what was said in point (2) of section 2.6.

- The justification given in the beginning of section 2.7.3 for using the project duration attribute to select projects is too weak.

An important concern is whether it is really a good comparison between AMs and WMs to select all projects with similar characteristics (duration, team size, number of requirements items). This can lead to a very restricted conclusion, since the results apply only to projects with a small to medium duration (3-9 months) with a moderate team size (5-11 members) and a range of 7 to 69 requirement items. Is it possible to achieve very different results when the projects are large in size and obviously with larger teams and perhaps longer duration?

Another concern is the production of statistical data ("scaled values") that the authors themselves argue about the meaning of that data. What does the following phrase in section 3.1 mean "the conclusion is that these scaled values must be used with caution and full disclosure."?

Most references are from the previous decade. The most updated reference is from 2015 [35]. The references in sections 1.1 and 1.2 (Past Work) and in section 4 (Discussion) are from 2003 to 2014. I believe that the authors should search the literature more systematically to find more recent articles on AMs and NFRs and those that address comparative studies between AMs and WMs. The authors must take into account that mainly in the last decade the use of MAs has proliferated, adapting their practices to the current needs of the software industry.

Some updated references about agile vs waterfall:

* Andrei, B. A., Casu-Pop, A. C., Gheorghe, S. C., & Boiangiu, C. A. (2019). A study on using waterfall and agile methods in software project management. Journal of Information Systems & Operations Management, 125-135. (Qualitative research)

* Theo Thesing, Carsten Feldmann, & Martin Burchardt, Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project, Procedia Computer Science, Volume 181, 2021, pp. 746-756, https://doi.org/10.1016/j.procs.2021.01.227. (based on qualitative research, it shows how to select a method considering 15 criteria, such as scope, time, costs, organization context, and project-team characteristics)

* M. Kassab, J. DeFranco and V. Graciano Neto, "An Empirical Investigation on the Satisfaction Levels with the Requirements Engineering Practices: Agile vs. Waterfall," 2018 IEEE International Professional Communication Conference (ProComm), Toronto, Canada, 2018, pp. 118-124, doi: 10.1109/ProComm.2018.00033.

When the authors state that "We found no empirical research studies that compared the numbers of NFRRDs in the CBSs resulting from different CBS development methods" (page 9), they should mention which literature search method and databases they used, since this statement is too conclusive.

The issues discussed in subsection 1.1.5 referring to NFR in AMs are based on articles [1,15, 17-20] from past decades (2004, 2006, 2011, 2012). Nothing has changed since then? There are more updated articles:

* Rahy, S. and Bass, J. M. (2021). Managing non-functional requirements in agile software development. IET Software, 16(1), 60-72. https://doi.org/10.1049/sfw2.12037

* Jarzebowicz, A., & Weichbroth, P. (2021). A qualitative study on non-functional requirements in agile software development. IEEE Access, 9, 40458-40475.

* Kopczynska, S., Ochodek, M., & Nawrocki, J. (2020). On importance of non-functional requirements in agile software projects---a survey. Integrating Research and Practice in Software Engineering, 145-158.

There is a subsection 1.1.6. dedicated to comparing AMs and WMs from the perspective of SQA. Although the authors mention the activities involved in SQA, in the subsequent descriptions (1.1.6.1 to 1.1.6.3) the comparison focuses only on testing, leaving aside the other SQA activities. No justification is provided as to why other SQA activities are not considered. This should be fixed.