

We thank the editor and reviewers for their helpful comments, and have addressed their requests and questions here. Specific changes or additions can be found in the marked revision.

## Editor:

*First of all, I would like to thank the reviewers, who have done a fantastic job and provided very insightful comments for improving the paper. I hope that the authors will take them into serious consideration in the revision.*

*From my personal perspective, this is a very good contribution, which addresses one of the most important topics in the “citation” domain, i.e. software citation. I totally agree with the reviewers about the readability of the paper, and I think this paper will deserve publication in PeerJ CS.*

*However, there are some aspects that should be clarified in the revision. In particular:*

1. **Comment:** *I would like to see a new section that describes the presentation of these principles at FORCE 2016, and includes also a summary of the most important outcomes derived from the discussion the authors have done during the conference. Did such outcomes change some of the principles and/or other claims in this paper?*

**Our response:** \*\*\*AMS: This seems relatively straightforward? We had a workshop where we reviewed the principles with a group of FORCE11 participants (Dan - do we have a list of edits we made that day based on feedback?) and then presented (in 5 mins) to the main FORCE11 session. I don't think the presentations/review at FORCE 2016 changed the principles significantly as they'd already been through pretty substantial review during the drafting process. \*\*\*DSK: we would have to go back through git versions. I know that there were changes to the principles themselves in the group phone calls, but not after that - ie, not in the workshop or general presentation. there were some smallish changes to the high level text describing each principle later.

\*\*\*KEN: I will attempt this

2. **Comment:** *The link between the principles and the requirements should be introduced explicitly, since it is not entirely clear how ones are connected to the others.*

**Our response:** \*\*\*AMS: I think we just need to say that we thought it was important to draft the principles with the end-user(s) in mind, i.e. what different use cases/scenarios for software citation. The requirements table is an overview of these use cases and the metadata needs of each of these users/use cases.

\*\*\*KEN: Move paragraph at end of Section 1 to Section 3, and expand/merge with paragraph after mission statement; find text from FORCE2016 presentation about use cases and explaining procedure to get principles \*\*\*KEN: I will do this

3. **Comment:** *A clear research question should be announced explicitly from the very beginning, and accompanied by a text that explains how the authors would like to address it.*

**Our response:** We agree with Reviewer 2 on this issue, as this represents more of a position paper than an experimental paper. That said, we would be willing to add an explicit research question and discuss the “experimental” design if required, but feel this would be an artificial construct that doesn't add much to the paper.

4. **Comment:** *I suggest to include one or more appendixes with all the details about the use cases presented. This would make the paper more self-contained.*

**Our response:** \*\*\*KEN: I will add the full use cases text as an appendix, and revise to match the numbering in the table \*\*\*KEN: I will also add the FORCE16 comments and responses

\*\*\*KEN: Dan & Arfon: could you assist with adding descriptions to some of the use cases that don't have one?

5. **Comment:** *The reviewers did not agree on some aspects described in the paper, e.g. “if the software authors ask that a paper should be cited, that should typically be respected”. I would like to see an appropriate discussion on these points, so as to clearly address these disagreements.*

**Our response:** We have either made appropriate changes or discussed these points.

## Reviewer #1 (Tobias Kuhn):

*This paper argues for the adoption of software citations, basically for the same reasons that data citations have been advocated: Because software packages have become so important in science, they should be appropriately credited and referenced, which can be achieved if we extend the concept of citations in scientific articles.*

*The paper tackles an important problem and provides a very valuable foundation for further discussions and future implementations. In particular, the six software citation principles presented in the beginning are very valuable.*

*While I think this is a very useful paper, I see the following problems with the current version:*

1. **Reviewer comment:** *The logical connections between the different parts of the paper are not clear. For example, how do the “basic requirements” of Table 2 map to the “principles” in Section 1? And are the principles presented in the beginning the \*outcome\* of this research (this is what lines 29+ seem to suggest) or are they \*assumptions\* of the approach to be discussed and evaluated (this is how they seem to be treated in Section 5).*

**Our response:**

\*\*\*KEN: This will be addressed somewhat by the changes discussed above; but Dan, could you help with this?

2. **Reviewer comment:** *It is not clear whether the authors consider software citations to be just the first step towards a long-term vision, or whether they consider software citations the overall final solution. It seems that the first is the case, but the long-term vision is not discussed.*

**Our response:** \*\*\*AMS: Perhaps we should be adding a short paragraph outlining how we believe that ultimately academic publishing should be moving towards sharing all of the products of researcher activity. This work is primarily about making software a citable entity which is just one part of a much larger change in the scholarly ecosystem.

\*\*\*KEN: Dan, can you do this?

3. **Reviewer comment:** *This paper doesn't have an experimental design in the strict sense, as it is more of a position paper than a research article. The PeerJ guidelines state that “the submission should clearly define the research question”, which is currently not the case for this paper. However, I see a research question between the lines, which could be phrased as “how can citations be applied to software products to account for their increasing importance in scientific research?”. By making such a research question explicit, the PeerJ requirements can be met.*

**Our response:** \*\*\*AMS: The reviewer's suggested text sounds pretty good to me. Perhaps drop/change 'scientific' in that sentence?

\*\*\*KEN: Agreed on both counts

\*\*\*KEN: Let's say that We agree that this paper is more of a position paper than research article, and thus that creating a research question would be an artificial construct. We are willing to make this change if required, but feel it wouldn't add much to the paper.

4. **Reviewer comment:** *The use cases described in Section 3 form the main method to answer the implicit research question. Given the importance of these use cases for this paper, I think reference [18] with the details about these use cases should be published in a more official way, either on a site like ArXiv or FigShare, or maybe even as supplemental material or appendix for this paper. I think having these use cases in the appendix of this paper would be the best solution, which would also increase the impact of these use cases and would strengthen the paper in general.*

**Our response:** \*\*\*KEN: I will add the full use cases text as an appendix

5. **Reviewer comment:** *As noted above, the logical structure of the paper is unclear, and therefore it is not clear what exactly the main findings are. I think the main findings should be the principles stated in the beginning, and the use cases would then form the main method to arrive at these findings. Alternatively, one could label the principles as “assumptions” and apply the use case study to validate these assumptions. In any case, I think the findings are valid but they should be labeled as preliminary, as it is not possible to arrive at a final conclusion for such kinds of questions. Only the future can tell whether software citations really work out.*

**Our response:**

\*\*\*KEN: If we need to formulate the paper as a research paper, then describing the principles as main findings and use cases/discussion as the methods to arrive at them seems an interesting way to do it.

\*\*\*KEN: We agree that the main findings are the principles, and the use cases/discussion are the main to reach those. Don’t agree that the principles are assumptions or preliminary, but reflect current knowledge and experience

\*\*\*KEN: I will work on this, but may need help

6. **Reviewer comment:** *line 88: “which will be presented ...” to “which was presented ...”*

**Our response:** We agree, and have made this change.

7. **Reviewer comment:** *line 89: “We expect that this discussion may lead to a second, final version”: So this paper is not the final version? Please clarify.*

**Our response:** Thank you for pointing this error out—we have corrected this statement to indicate that the past discussion led to a second version (what was submitted).

8. **Reviewer comment:** *line 97: “the need for credit for application software underscores the need to overhaul the system of credit for all research products.”: Very true and very important. But then maybe the concept of a “citation” is obsolete altogether, and we shouldn’t be advocating “software citations”, but something new altogether? Some more discussion on this would be valuable, I think. This relates to my comment above about a long-term vision.*

**Our response:** \*\*\*AMS: We probably need to dig out some citations about improved/rich citation. I’m aware of some work that was going on in the PLOS Labs group about this but I’m not sure if that work was ever published.

\*\*\*KEN: Citations are going to go away—but format and how they fit in ecosystem may change.

\*\*\*KEN: Arfon: can you help here?

9. **Reviewer comment:** *line 238: “Citation is a record of software that is important to a research outcome”: This is another important issue, but not much more is said about this. What makes a software “important” to a research outcome? Or, more generally, what is the semantics of a software citation?*

**Our response:**

\*\*\*KEN: I think we have some citations about the importance of software to research we can add here (e.g., Hannay 2009), as well as the fact that research results depend on citation/proper version (Sufi 2014, Sandve 2013, Wilson 2014), and that errors in software or environment variations can effect results (Morin 2012, Soergel 2015). Although we probably want to avoid overlap with the challenge

paper. \*\*\*DSK: basically, what is important is up the authors to define, and then reviewers to agree with, all in the context of community practices and ‘standards’

\*\*\*KEN: Add discussion of both points; Dan’s note is already there in the text. I will work on this.

10. **Reviewer comment:** *line 254: “In addition, if the software authors ask that a paper should be cited, that should typically be respected”: I think I disagree with that. Whether a paper should be cited or not should \*not\* depend on whether the given software package tells you to do so. Such pointers can be useful as suggestions, but not more than that in my opinion.*

**Our response:** We have modified the text to make it clear that this is a recommendation from the working group; the statement reflects an agreed position of the entire working group. The modification should make it clear that this is a recommendation only.

11. **Reviewer comment:** *Lines 308+: You don’t imply that this list is complete, but wouldn’t an identifier of a software \*branch\* be another important type of identifier? For example, the Windows “version” of a software package. Also, I see in what situation one would use (1) or (2) with respect to software citations, but what is the purpose of (3) in the context of citations?*

**Our response:** Principle (6) states that “Software identification should be as specific as necessary”, and we consider that the specific version includes the branch.

We agree that (3) is not relevant in the context of citation, and have removed it from the discussion. However, it is a valid identifier that could be used in other contexts.

12. **Reviewer comment:** *Line 325: “other software may be available as a service”: This makes a huge difference, doesn’t it? I think some more discussion on this would be helpful. For example, with software services it can be hard to find out what the official name of the software is, let alone the developers or version numbers. Also, does a software that is only available as a service (e.g. REST API) violate the Accessibility principle (5)?*

**Our response:** Software as a service is a fact, and we need a method to cite it—even though some metadata may not exist or be available, we need to do the best we can. The same issues would hold for commercial software or that available only as an executable. If you can’t find the name of software, should be able to find the name of the service/endpoint or function being used.

Regarding accessibility, ideally we would like the source to be available, but if not we still need a way to cite. In this case, the Accessibility principle would apply to the service itself, rather than the source.

13. **Reviewer comment:** *I find it a bit weird to quote passages from the same paper like “The Accessibility principle (5) states that ?....?” (line 329 and others).*

**Our response:** We feel it is more clear to a reader to quote the basis by which we are making follow-on comments, rather than pointing back to a previous part of the paper.

14. **Reviewer comment:** *In the discussion about Unique identification (lines 292+), I miss the discussion of hash values as version identifiers, as applied by Git for instance. As the first author is affiliated with GitHub, I assume this aspect was left out intentionally, but why? As basically all software is nowadays maintained with version control systems such as Git, these identifiers come for free and are by design unique. So, I would think they are an important aspect to discuss for software citations.*

**Our response:**

\*\*\*KEN: Arfon: can you add some discussion here? We can add a sentence addressing git hashes and other version control identifiers, and point out that while they may be sufficient in some ways (globally unique), they aren’t necessarily resolvable on the Web, and different VCS use different identifiers, etc. In other words, they could be used (especially in the absence of other identifier like DOI), but not sufficient or recommended.

## Reviewer #2 (Graham Klyne):

1. **Reviewer comment:** *The two tables are relevant and helpful, though I did have a slight problem with the legibility of my printed copy of Table 2 (page 5). The distinction between solid and open circles was not immediately clear to me, and I might suggest using a slightly larger version of these symbols.*

**Our response:** We agree, and have changed the open symbol to a plus sign to improve legibility.

2. **Reviewer comment:** *The paper does not present a specific research finding, and as such does not feature a research question and experimental procedure that are expected of such papers. However, I think it makes an important contribution to the acceptance of software as part of the process and output of research activity, and reflects a consensus formed among a significant community of researchers. As such, I believe it justifies a place in the formally published literature.*

**Our response:** We agree with the reviewer's description of our paper, and appreciate the view that it should be formally published.

3. **Reviewer comment:** *Lines 32–35: as a justification for treating software differently from data, I found this passage raised more questions than answers, and that the comments here were unconvincing and unsupported. For example, some published data is updated on a frequency comparable with some software (e.g. the bi-monthly release schedule of FlyBase is comparable with many software projects - [http://flybase.org/static\\_pages/docs/release\\_schedule.html](http://flybase.org/static_pages/docs/release_schedule.html)). Also, I think the distinction between data and software is less clear-cut than the authors suggest (e.g. is a schema or stylesheet data or software?). I would see the difference between software and data as more akin to the difference between research methods and research findings, but that is just a personal opinion. My suggestion would be that this justification is not really necessary to the overall purpose of the paper, and might be dropped without significant loss of value.*

**Our response:**

\*\*\*KEN: I think we all would agree that software is a research method and data are research findings, but I do think we need some explanation of the differences—if only to justify why software citation may be different than data citation

\*\*\*KEN: Dan: work on this. Bring in text from CodeMeta paper and your blog post.

4. **Reviewer comment:** *Line 57, et seq: [Nit] I would find this easier to read if the sentence introducing the list of reasons were moved to the same page as the list itself.*

**Our response:** We agree that this would be easier to read, but believe the formatting will change in the final published paper.

5. **Reviewer comment:** *Line 88: “will be presented” refers to an event now in the past: change to “was presented”?*

**Our response:** We agree, and have made this change.

6. **Reviewer comment:** *Line 113: [Nit] “work using software” appears to be very broadly scoped, and could apply to just about any activity using a computer. Suggest “research using software”.*

**Our response:** We agree, and have made this change.

7. **Reviewer comment:** *Table 2, page 5: I had a problem with the legibility of my printed copy of this table. The distinction between solid and open circles was not immediately clear to me. I might suggest using a slightly larger version of these symbols, or maybe more distinct symbol shapes.*

**Our response:** We agree, and have replaced the open bullet with a plus sign for better legibility.

8. **Reviewer comment:** *Table 2, page 5: [Comment] This table contains a lot of information, and much food for debate. I was surprised by some omissions, which may mean that I wasn't fully understanding the concepts being presented:*

- (a) *I find it hard to imagine any of the use cases not benefiting from availability of a “Description”, especially 1, 2, 9, and 16.*
- (b) *I would have expected “Indexed citations” to be useful for 5: getting credit.*
- (c) *For publishing a software paper, I would expect all of the basic requirements other than “Indexed citations” to be at least beneficial.*
- (d) *For 12: build a software catalog/registry, would there not be a role for indexed citations?*
- (e) *For 16: Store software entry, I’d expect pretty much all of the Basic requirements other than “Indexed citations” to be relevant, and particularly “Description” and “Keywords”.*
- (f) *I think there is one use-case missing, which I think is commonly faced by researchers: viz. “Evaluate suitability of software for a given task” - I think this is distinct from discovery, and typically involves making a choice between different available software packages, or to use an existing package, modify an existing package or develop something new.*

**Our response:**

- (a) Agreed for 1 and 2.
- (b) \*\*\*KEN: I think I agree, or at least benefit
- (c) \*\*\*KEN: I think what we had originally makes sense, but perhaps there needs to be more explanation?
- (d) \*\*\*KEN: no, indexed citations not really relevant for catalog/registry, unless it wants to display citations of software—but that is use case 4
- (e) \*\*\*KEN: no, I don’t think those things would be stored—because they aren’t included in the citation itself
- (f) We intended “find” to mean more than discovery, but since the appendix was not present in the first submission this was not clear. We have altered the use case to be “Find/choose software to implement task” to make it clear that this use case goes beyond discovery.

9. **Reviewer comment:** *Line 175: [Nit] “in(1)” Missing space?*

**Our response:** This has been fixed.

10. **Reviewer comment:** *Lines 241–242: “Similarly, the software metadata that is recorded as part of data provenance should be a superset of the metadata recorded as part of software citation.” I found this non sequitur (or am not seeing the logical connection). I can imagine reasons to cite software that does not form part of the provenance chain of results presented (e.g. alternatives considered).*

**Our response:**

\*\*\*DSK: this is a good point - I probably need to change our text a bit

11. **Reviewer comment:** *Lines 245–247: “In general, we intend the software citation principles to cover the minimum of what is necessary for software citation for the purpose of software identification. Other use cases (e.g., provenance, reproducibility) may lead to additional requirements (i.e., enhanced metadata).” This statement feels to me to be at odds with table 2, which presents “Use cases and basic metadata requirements for software citation”, which includes “reproducibility” as a use-case. This leaves me wondering if the use cases in table 2 are going beyond what is strictly needed for the purposes of software citation.*

**Our response:**

\*\*\*KEN: The table collects use cases that relate to or depend on software citation, but certainly each use case has additional dependencies that are not and cannot be contained in the table. So, perhaps we can rephrase somewhere to emphasize that the table requirements are only for the citation of software related to each use case, and not meant to be exhaustive about everything actually required for the use case beyond citation? The requirements in the use cases table only indicate the metadata needed for citation related to each use case, and not necessarily everything needed to complete each use case. For example, use case 7 “Benchmark software” does not include information about actual benchmarking, or the need for similar hardware, etc.

12. **Reviewer comment:** *Line 327–328: “When software exists as both source code and another type, we recommend that the source code be cited”. The term “exists” here doesn’t seem quite right to me. E.g., proprietary software may “exist” as source code, but be unavailable. Suggest “exists and is accessible” or just “is accessible”?*

**Our response:** Agreed; this change has been made.

13. **Reviewer comment:** *Line 348: “Updates to this document”. This paper is being presented for consideration as a part of the formal academic record - as such, I don’t think “this document” is something which may be updated. Suggest “Updates to these principles”.*

**Our response:** Agreed; this change has been made.