Reviewer's report

Title: Methods for enhancing the reproducibility of biomedical research findings using electronic health records.

Version: 0 Date: 30 Jun 2017

Reviewer: Frank Lesske

Reviewer's report:

In this work the authors argue for methods an tools dealing with EHR data. They focus on the development of software in this field.

In generally I agree to the statement that coding analytical software for EHR data should be carried out under the same aspects of quality assurance as any other software.

But from this work I did not understand what are the specific requirements and constraints in the area of EHC software, that would distinguish it from other software development projects. In the field of software engineering there exist in the meantime a whole bunch of techniques and methods in order to produce software of good quality. This includes methods for analysis, design, coding testing and verification, and documentation. So why no just require the use of standard software development techniques (to phrase it in a short way)? Why are the selected methods (modularity, revision control, tests) are more important in the field of EHC data, than others are? Why are design techniques, e.g. the use of standardized diagrams as in UML, not mentioned at all, although they could help a lot to improve the quality of documentation? Is testing alone sufficient, or are more rigorous methods needed, as e.g. formal verification?

In summary, my impression is that a set of methods and tools is suggested, but it is not really described what are the specific advantages of exactly these methods and tools for the field of EHC. There is no description of what are the specific challenges of EHC software development that makes if different from software development in other areas. This gives the impression of a somehow arbitrary choice.

Some more detailed comments:

- page 4: modular programming has been studied thorougly in computer science leading to terms like encapsulation, abstraction (abstract data types), object oriented programming etc. Why is this not mentioned?

- page 4/5: The use of libararies in R is not the crucial point. You should consider the use of interfaces and libraries that implement certain interfaces.
Testing software should be a natural course of action. But wouldn't TDD, in its original meaning, give a benefit. Why do you define it more broader? You should also distinguish unit tests, system tests (tests of integration) and acceptance tests. Units test alone, are not sufficient.

Is testing sufficient at all?

- I didn't understand the intention of Box 1? It occurs to me as a simple R function with some simple case-statement? Checking function arguments using if-statements is not really something new, did you ever try using invariants (preconditions, postconditions)?

- Using a source control system in software development is, to my opinion, not worthwhile mentioning. If you mention it you should also consider the problems of parallel development and branching.

- page 10: In terms of documentation you should consider modelling techniques that use different kinds of diagrams (e.g. stat charts etc.) that could support the traceability and reproducability of results.

Why do workflow systems enable the development of scalable tools, as opposed to other approaches?

- page 11: Again, there exist many modelling techniques to describe the algorithmic logic behind a solution. In software engineering MDA (model driven architecture) is an approach to develop a solution independently from the particular implementation environment.

- page 11: Talking about literate programming: how is about using pseudo-code? This is used in many publications.

Level of interest
Please indicate how interesting you found the manuscript:

An article whose findings are important to those with closely related research interests

Quality of written English
Please indicate the quality of language in the manuscript:

Acceptable
Declaration of competing interests

Please complete a declaration of competing interests, considering the following questions:

1. Have you in the past five years received reimbursements, fees, funding, or salary from an organisation that may in any way gain or lose financially from the publication of this manuscript, either now or in the future?

2. Do you hold any stocks or shares in an organisation that may in any way gain or lose financially from the publication of this manuscript, either now or in the future?

3. Do you hold or are you currently applying for any patents relating to the content of the manuscript?

4. Have you received reimbursements, fees, funding, or salary from an organization that holds or has applied for patents relating to the content of the manuscript?

5. Do you have any other financial competing interests?

6. Do you have any non-financial competing interests in relation to this paper?

If you can answer no to all of the above, write 'I declare that I have no competing interests' below. If your reply is yes to any, please give details below.

I declare that I have no competing interests.

I agree to the open peer review policy of the journal. I understand that my name will be included on my report to the authors and, if the manuscript is accepted for publication, my named report including any attachments I upload will be posted on the website along with the authors' responses. I agree for my report to be made available under an Open Access Creative Commons CC-BY license (http://creativecommons.org/licenses/by/4.0/). I understand that any comments which I do not wish to be included in my named report can be included as confidential comments to the editors, which will not be published.

I agree to the open peer review policy of the journal.

Do you want to get recognition for reviewing this manuscript? Add a record of this review to Publons to track and showcase your reviewing expertise across the world’s journals. Signing up is quick, easy and free!

Yes