Questions to Developers

Ao Dong

November 29, 2020

[A pilot will be conducted with these questions to test them. Test subjects may include PhD students. —PM]

1 Information about the developers and users

[Jegatheesan, 2016]

1. Interviewees' current position/title? degrees?
2. Interviewees' contribution to/relationship with the software?
3. Length of time the interviewee has been involved with this software?
4. How large is the development group?
5. Do you have a defined process for accepting new contributions into your team?
6. What is the typical background of a developer?
7. What is your estimated number of users? How did you come up with that estimate?
8. What is the typical background of a user?

2 Information about the software

[3 aspects: dev process, software quality, user feedback —AD]

[consider adding 'What, if any, actions were taken to address x?' to most/all questions, similar to question 10 —PM]

Square brackets are used for traceability to qualities and relevant research questions found in the methodology document.

1. Currently, what are the most significant obstacles in your development process?
2. How might you change your development process to remove or reduce these obstacles?

3. How does documentation fit into your development process? Would improved documentation help with the obstacles you typically face? [traceability - research question 5b, visibility/transparency - research question 5i]

4. In the past, is there any major obstacle to your development process that has been solved? How did you solve it?

5. What is your software development model? For example, waterfall, agile, etc.

6. What is your project management process? Do you think improve this process can tackle the current problem? Were any project management tools used?

7. ~~Can you name several software qualities that are the most important to this project?~~

8. Was it hard to ensure the correctness of the software? If there were any obstacles, what methods have been considered or practiced to improve the situation? If practiced, did it work? [correctness - research question 5e]

9. When designing the software, did you consider the ease of future changes? For example, will it be hard to change the structure of the system, modules or code blocks? What measures have been taken to ensure the ease of future changes and maintains? [maintainability - research question 5d, modifiability - research question 5c]

10. Provide instances where users have misunderstood the software. What, if any, actions were taken to address understandability issues? [understandability - research question 5f]

11. What, if any, actions were taken to address usability issues? [usability - research question 5a]

12. Do you think the current documentation can clearly convey all necessary knowledge to the users? If yes, how did you successfully achieve it? If no, what improvements are needed? [unambiguity - research question 5g]

13. Do you have any concern that your computational results won't be reproducible in the future? Have you taken any steps to ensure reproducibility? [reproducibility - research question 5h]

2.1 Old notes from methodology: Information about the software

Compare to questions above and then delete this subsection

1. [General —AD] What is the most important software quality(ies) to your work? (set of selected qualities plus "else")

2. [General —AD] Are there any examples where the documentation helped? If yes, how it helped. (yes*, no)

3. [General —AD] Is there any documentation you feel you should produce and do not? If yes, what is it and why? (yes*, no)

4. [Completeness —AD] Do you address any of your quality concerns using documentation? If yes, what are the qualities and the documents. (yes*, no)

5. [Visibility/Transparency —AD] Is there a certain type of development methodologies used during the development? ({Waterfall, Scrum, Kanban, else})

6. [Visibility/Transparency —AD] Is there a clearly defined development process? If yes, what is it. ({yes*, no})

7. [Visibility/Transparency —AD] Are there any project management tools used during the development? If yes, what are they. ({yes*, no})

8. [Visibility/Transparency —AD] Going forward, will your approach to documentation of requirements and design change? If not, why not. ({yes, no*})

9. [Correctness and Verifiability —AD] During the process of development, what tools or techniques are used to build confidence of correctness? (string)

10. [Correctness and Verifiability —AD] Do you use any tools to support testing? If yes, what are they. (e.g. unit testing tools, regression testing suites) ({yes*, no})

11. [Correctness and Verifiability —AD] Is there any document about the requirements specifications of the program? If yes, what is it. ({yes*, no})

12. [Portability —AD] Do you think that portability has been achieved? If yes, how? ({yes*, no})

13. [Maintainability —AD] How was maintainability considered in the design? (string)

14. [Maintainability —AD] What is the maintenance type? (set of {corrective, adaptive, perfective, unclear})

15. [Reusability —AD] How was reusability considered in the design? (string)

16. [Reusability —AD] Are any portions of the software used by another package? If yes, how they are used. (yes*, no)

17. [Reproducibility —AD] Is reproducibility important to you? (yes*, no)

18. [Reproducibility —AD] Do you use tools to help reproduce previous software results? If yes, what are they. (e.g. version control, configuration management) (yes*, no)

19. [Completeness —AD] Is any of the following documents used during the development? (yes*, no)

20. [General —AD] Will this experience influence how you develop software? Do you see yourself maintaining the same level of documentation, tool support as you go forward? (string)

    - Module Guide
    - Module Interface Specification
    - Verification and Validation Plan
    - Verification and Validation Report

References

Thulasi Jegatheesan. Case studies in document driven design of scientific computing software. Master's thesis, McMaster University, Hamilton, Ontario, Canada, July 2016.