

Appendix C: Interview Questions

State of Practice of Medical Imaging Software

Information about these interview questions: This gives you an idea what I would like to learn about the development of Medical Imaging software. Interviews will be one-to-one and will be open-ended (not just “yes or no” answers). Because of this, the exact wording may change a little. Sometimes I will use other short questions to make sure I understand what you told me or if I need more information when we are talking such as: “*So, you are saying that ...?*”, to get more information (“*Please tell me more?*”), or to learn what you think or feel about something (“*Why do you think that is ...?*”).

- 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?
9. Currently, what are the most significant obstacles in your development process?
10. How might you change your development process to remove or reduce these obstacles?
11. How does documentation fit into your development process? Would improved documentation help with the obstacles you typically face?
12. In the past, is there any major obstacle to your development process that has been solved? How did you solve it?
13. What is your software development model? For example, waterfall, agile, etc.
14. What is your project management process? Do you think improving this process can tackle the current problem? Were any project management tools used?
15. 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?
16. 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?
17. Provide instances where users have misunderstood the software. What, if any, actions were taken to address understandability issues?
18. What, if any, actions were taken to address usability issues?
19. 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?
20. Do you have any concern that your computational results won’t be reproducible in the future? Have you taken any steps to ensure reproducibility?

END