State of Practice of Medical Imaging Software

Interview Answers

Alark Joshi, USF

1. **Interviewees' current position/title? degrees?** Associate Professor and Department Chair of Computer Science at the University of San Francisco. I got my PhD from the University of Maryland Baltimore County in 2007.
2. **Interviewees' contribution to/relationship with the software?** I worked on BioImage Suite between 2007-2012 and continue to contribute to the software a bit from time to time. I was a Postdoctoral Associate at the Yale School of Medicine between 2008-2010 when I worked on it.
3. **Length of time the interviewee has been involved with this software?** Actively for 5 years and intermittently for the last 13 years.
4. **How large is the development group?** The group has grown and shrunk over the years. As of now, there are about 4 active developers made up of students, postdocs, and Prof. Xenophon Papademetris (who created and sustains BioImage Suite).
5. **Do you have a defined process for accepting new contributions into your team?** We have been using a combination of Github and email/newsgroup communications to accept new contributions to the software.
6. **What is the typical background of a developer?** The typical background of a developer is an undergraduate/graduate degree in Computer Science/Biomedical Engineering.
7. **What is your estimated number of users?** How did you come up with that estimate? We have approximately 100 active users at Yale School of Medicine in the Radiology, Cardiology, Neurosurgery, Pediatrics, and Biomedical Engineering departments. We have some active users at other universities such a Harvard Medical School and Vanderbilt University.
8. **What is the typical background of a user?** A typical user is a domain expert who uses medical images for analysis and treatment, in some cases. Our users could have a variety of backgrounds from Physics, Biomedical engineering, Neurology, Radiology, and so on.
9. **Currently, what are the most significant obstacles in your development process?** The most significant obstacle in our development process is building software for our end users who may have a variety of computing power on their respective devices. It is particularly challenging since medical imaging software manages large files and includes extensive computation.

10. **How might you change your development process to remove or reduce these obstacles?** As of now, we are working on a web-based approach that will move some of the computing burden to the servers and all users to explore and analyze their data.
11. **How does documentation fit into your development process? Would improved documentation help with the obstacles you typically face?** Documentation is an important part of our overall process. In addition to documenting our code, we have also written books with chapters that end users can refer to in order to learn more about the features in our software. Any code that is merged into the main branch must have extensive documentation and goes through at least one code review. Unfortunately, documentation will not address our obstacles mentioned above.
12. **In the past, is there any major obstacle to your development process that has been solved? How did you solve it?** Working closely with our end users is an important part of our software's adoption and success. Most major obstacles are identified by discussing challenges and tasks that users want to perform. We follow an iterative, agile approach to building software and that has helped us overcome obstacles in the past.
13. **What is your software development model? For example, waterfall, agile, etc.** As per my answer above, we follow the agile software development process.
14. **What is your project management process? Do you think improving this process can tackle the current problem? Were any project management tools used?** We have used a combination of Github, bug trackers, and issue trackers, to manage the progress of a project. We also have nightly tests that allow us to examine the impact of any changes on the overall software.
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?** Our approach has always been test-driven development. We add nightly tests for all the code that we write. The nightly tests generate a dashboard that allows developers to examine the impact their code has on the overall software. This approach has worked very well for us.
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?** We have designed and re-designed our software multiple times in the last 10+ years. The goal is always maintainability with an eye on modularity. The modular approach was published in a paper that describes our unified framework for prototyping, development, and testing - https://scholarworks.boisestate.edu/cs_facpubs/5/.
17. **Provide instances where users have misunderstood the software. What, if any, actions were taken to address understandability issues?** Misunderstandings are common with our users. Our small group of active users (~3/4) test every software release and every feature within. Frequently, features are built to solve a problem that they may be facing. This iterative process helps with eliminating most of the usability bugs as well as misunderstandings. As mentioned before, we also have a user manual to help potential users accomplish their tasks - https://medicine.yale.edu/bioimaging/suite/manual/guide/ .

18. **What, if any, actions were taken to address usability issues?** We have performed some usability testing with our end users, but not enough. That is one of the weakness in our software. Our end users learn to use the software, but I think it could be much more streamlined to help users complete their tasks quicker.
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?** Our documentation has always been a work-in-progress that gets updated frequently. The user manual is one of the most commonly used resource by our users. Additionally, we have also conducted workshops for 15-20 users to go over any new features that we think would be relevant to certain user groups - brain connectivity, cardiology, neurosurgery, tissue engineering, and so on.
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?** We have always been active proponents of open source software and have contributed to and build bridges to Harvard's Slicer software. With respect to reproducibility, I believe that making code, data, and documentation available is the best way to facilitate reproducibility. We still run into challenges when communicating with individuals all over the world and use a newsgroup for the same.