The Cathedral & the Bazaar Mediation Q’s

- Do you think it is better to build software like a cathedral (in isolation with each person crafting only one thing at a time), or like a bazaar (busy, bustling, with people joining and leaving the project as they see fit)?
- Do you agree with the statement that “Every good work of software starts by scratching a developer's personal itch?” What pieces of software do you already know that may have satisfied an itch?
- The paper makes the comment “Good programmers know what to write. Great ones know what to rewrite (and reuse).” Discuss some software that you may have had to re-write, or software that you wish you could have re-written.
- Discuss a time where you have made use of “constructive laziness” in your programming career.
- Discuss a time where you thought you knew the solution to a problem that you had implemented, only to find that your “solution” was complete garbage and had to start again. How did it turn out with a fresh perspective?
- The paper states “If you have the right attitude, interesting problems will find you.” Discuss with each other a time in your life where this has occurred, in the coding world, or everyday life.
- The paper states “When you lose interest in a program, your last duty to it is to hand it off to a competent successor.” Have you ever had to give up on a project, and pass your work off to another individual? If so, describe how it turned out.
- The paper states “Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.” Have you ever shown your code to a friend/colleague only for them to break it, or show you its hidden flaws? Describe such a time, and how it may or may not have improved your program in the long run.
- Discuss the benefits and drawbacks of releasing early and releasing often.
- Linus Torvalds stated, “Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.” Discuss the different types of people you have met over your coding career. Was one person good at everything? Or do most of us have a niche that caters to what we enjoy doing?
- “Smart data structures and dumb code works a lot better than the other way around.” Describe how your experience in programming thus far may support or challenge this claim.

- “If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource.” Discuss the validity of this statement, and how it may or may not be true to you.
- “The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.” Describe a time over the course of your programming career where you have given a good idea to someone or have provided a good idea to someone else. Explain the repercussions of your/their advice.
- “Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.” Discuss how this may or may not be true in today’s world.
- “When you hit a wall in development—when you find yourself hard put to think past the next patch—it's often time to ask not whether you've got the right answer, but whether you're asking the right question. Perhaps the problem needs to be reframed.” → I just thought this was an interesting quote to discuss.
- “Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away.'' → Same here. Thoughts?
- “To solve an interesting problem, start by finding a problem that is interesting to you.” What is a problem that is interesting to you?