**Agile Software Development Coach**

**Job Description**

Agile software development coaches are very experienced software developers, who help software development teams learn coding practices that lead to near-zero defects, code that is always shippable, and software that is easy to change and test. An agile software development coach is able to join the team in writing code, and mentor them in practices such as continuous integration and delivery, pair and mob programming, test-driven development, refactoring, and evolutionary design. The coach needs to be able to do more than just teach these practices on simple examples; they need to be able to guide the team in applying them in their real work.

Interview Questions

1. What are the business and customer facing outcomes that teams can realize as a result of agile software development practices?
2. What are the key agile software development practices you feel have the most positive impact on the team's ability to deliver value with high quality?
3. What is continuous integration?
4. What is trunk-based development? What are the benefits? Have you helped a team adopt this approach?
5. How would you contrast feature toggles with typical branching strategies?
6. How much experience do you have helping teams adopt pair and/or mob programming?
7. What are the 3 rules of test-driven development?
8. What coding exercises do you like to use, to introduce teams to test-driven development?
9. What is the testing pyramid?
10. What are the downsides of using mocking frameworks? Are you aware of any approaches to avoid them?
11. What is the definition of refactoring? How does it fit in with the red-green-refactor cycle?
12. If you have a body of passing unit tests, and then perform a refactoring, what will happen when you run those unit tests again (will they pass or fail)?
13. What are some of your most commonly used refactorings?
14. Have you ever done the Gilded Rose refactoring kata?
15. What is emergent design?
16. What are the 4 rules of simple design?
17. How would you contrast a "features before infrastructure" approach with architectural runway?
18. What is continuous delivery?
19. What does it mean to decouple deployment from release? Who owns each of those?
20. Who are the thought leaders in the agile software development space that you follow?
21. What books have you read recently related to agile software development practices?