SDLC Assessment Questions

Feb 2007

1. How do you gather/author requirements?
2. How do you store/manage gathered requirements?
3. Can you trace a requirement all the way down to the units of code built to satisfy the requirement?
4. On a scale of 1 to 10 (high), how "professional" would you rate your software development team? How do you define 'professional'?
5. Are you currently asked to provide regular reports on progress or quality?
6. What is your development team's velocity? Does a decrease in velocity result in a corresponding decrease in bugs?
7. What Source Code Control (SCC) system do you use? Does it have any features that you really like? Does it have any 'features' you hate?
8. How do you estimate the effort and time involved in a software project? How accurate are you?
9. Is your developer code peer reviewed? For performance? For security?
10. Are your performing automated and/or manual tests? Are they peer reviewed?
11. Do you have a coding standards (i.e. naming conventions)?
12. Do you use unit testing? If so, is it an individual level? Or an organizational level?
13. What metrics do you use to manage your resources (including developers)? (Examples: hours worked, dollars of functionality created, tasks completed)
14. How often do you run a complete compile of the code for a single project?
15. How often do customers/users get to see code in progress?
16. How do you currently track and mitigate risk?
17. How many hours a day can your developers focus on writing code?
18. Can your developers currently trigger an integration build without calling the CM team?
19. How are you tracking and triaging your bugs/defects?
20. Does your software project have a Vision / Elevator Pitch that the developers know?
21. If I were to ask your customers/users about your service, would they say (A) development is too *slow*, (B) development is too *expensive*, (C) developed software doesn't meet our *needs/requirements*