

COMMON SOFTWARE PRACTICE QUESTIONS

This section consists of questions that we will encounter in both the traditional and the Agile frameworks. These should be common and yet fundamental questions that we may ask ourselves every day. The practices mentioned in all the questions in this section are favoured in the Agile approach if done correctly. They should be considered as prerequisites to our Agile implementation.

1. Are your source codes and libraries version controlled?

- Yes
- No

Version control helps to track changes and is strongly encouraged.

2. Is development environment (e.g. versions of all libraries used, type of compiler, etc.) synchronized among all developers?

- Yes
- No

All developers should stick with the same development environment.

3. What types of sandbox do you use? (You may select more than one option)

- Development sandbox
- Project Integration sandbox
- Demo sandbox
- Pre-Production/ QA sandbox
- Do not use sandbox

All the sandboxes are good to have.

4. Do you use a pre-production/QA sandbox that replicates the product system and that cannot be changed?

- Yes
- No
- Not Applicable

A pre-production/QA sandbox mimics the production environment.

5. Where does your build take place?

- Developers' local machines
- Centralized build servers
- A mixture of both

We encourage that build should happen at a centralized build servers.

6. Are your deployment steps consistent for all environments?

- Yes
- No

Deployment steps should be consistent regardless of machines, developers, etc.

7. Does the team collect users' quality requirements such as acceptance criteria up-front?

- Yes all users' quality requirements are collected up-front
- Yes some users' quality requirements are collected up-front
- No users' quality requirements are collected up-front

We should collect all quality requirements on every feature that's to be worked on.

8. Do you use established coding standards?

- Yes
- No, I don't have established coding standards
- Somewhat

We recommend that the development team follow the same coding standard so that codes are more readable to all.

9. Do your developers cross review code changes?

- Yes always
- Yes sometimes
- Not at all

We encourage the team to cross check the codes before commit.

10. Which of the following test do you conduct to ensure quality of your system? (You may select more than one option. Choose only the ones that apply)

- Unit tests
- Integration tests
- Acceptance tests
- Functional tests
- Non-functional test (e.g. code analysis, performance/load test time etc.)
- Security scans
- Exploratory tests
- None of the above

All the tests are recommended.

11. Are your defects managed in a central repository?

- Yes
- No

We recommend that defects be managed in a centralized repository.

12. Do you use quality metrics as performance indicator?

- Yes
- No
- Not Applicable

In Agile term, we encourage the term 'Fail Fast'. We perceive failing fast as a good thing and we do not encourage making use of the quality test results as a performance indicator of the team.

13. To what extent do features rely on expert resources that are bottlenecked?

- Majority
- Some
- None

We encourage dedicated resources in the team and we avoid heavy multi-tasking

14. Progress updates are:

- Top-down
- Bottom-up
- Not Applicable

In Agile, we encourage that progress updates should be bottom up.

15. Which of the following would describe your management team?(You may select more than one option)

- Possesses command and control management style
- Caring leaders who possesses servant style leadership
- Listens to others' opinions
- Committed to growth of every individual within organization

We are against the command and control leadership style in Agile. We encourage a leadership which is more caring and respectful of people.

16. Are teams stable?

- Yes
- No

We recommend that teams should remain stable throughout the projects.

An initiative by

