CORE SOFTWARE DEVELOPMENT PRACTICES QUESTIONNAIRE

Quality Control

Quality control ensures that developed software meets and complies with defined or standardized quality specifications. It is an ongoing process within the software development life cycle that routinely checks the developed software to ensure it meets desired quality measures.

1. Which code standards do you use?

2. Do you reviewed code according to the coding standard?

3. Are code conventions checked automatically? Which tools do you use?

Integrated Development Environments (IDEs)

1. Which IDEs do you use? Do all teams use same IDEs?

Version Control

1. Which version control system do you use?
2. Do you have different repositories for code, configuration and third party components?
3. Do you check in things (such as generated code) that can be build form other source?
4. Does your version control system have a commit hook to the issue tracking systems?
5. Do you store very sensitive data (e.g. passwords) in the version control system?
6. Do you use branching? What is your branching policy?

Issue Tracking

1. Which issue tracking system do you use?
2. Are all issues (feature requests, bugs, non-functional...) logged?
3. Do you use one issue tracking system for all stakeholders?

Build Process

1. Is your build process fully automated?

2. What tools do you use for building?

3. How is a build triggered? Does it start automatically after every commit?

4. Does the build stop on any failing test?

5. Is the build status visible to everyone? How?

Test Automation

1. How do you do unit testing? Which tools do you use?

2. Do you measure unit test coverage? What is the current coverage? Which tools to you use to measure coverage?

3. How do you do component / integration testing? Which tools do you use?

4. Do you measure component / integration test coverage? What is the current coverage? Which tools to you use to measure coverage?

5. How do you do end-to-end testing? Which tools do you use?

6. Do you measure end-to-end test coverage? What is the current coverage? Which tools to you use to measure coverage?

Capacity Testing

1. Do you do load testing? How often? Are these tests automated? Are they a part of the build pipeline?

2. Do you do stress testing? How often? Are these tests automated? Are they a part of the build pipeline?

3. Do you do endurance testing? How often? Are these tests automated? Are they a part of the build pipeline?

4. Do you do spike testing? How often? Are these tests automated? Are they a part of the build pipeline?

Test Environment

1. Is you test architecture and infrastructure representative (production-like)?

2. How do you create test data? Is test data representative?

3. Are sensitive data protected in test data?

Deployment Process

1. Is the deployment process repeatable?

2. Is the deployment process (fully) automated?

3. How much time a typical deployment takes?

4. Is continuous delivery used?

5. Do you have automated smoke tests after every deployment to all environments?

6. What is your patching policy?

7. Can you deploy the system independently? Does the deployment dependents on deployment of other systems?

Continous Integration

1. Which integration server do you use?

2. Is everything is checked in into a single version control repository?

3. Does everyone check in code changes at least once a day?

4. Is the codebase in a shared ownership?

5. Does fixing a broken build have the highest priority for everyone in the team?

6. Do automated tests run after every commit?

7. Can everyone see the results of integration? How?

Third-Party Dependencies Management

1. How do you manage dependecies to 3rd party libraries and components?

2. Are used 3rd party libraries are kept up-to-date? What is you policy regarding library versions?

3. Do you check if used libraries are still supported?

Monitoring

1. Do you continuously monitor the resource usage? Which tools do you use?

2. Is resource usage monitoring centralized? Which tools do you use?

3. Do you monitor at the level of business transactions? Which variables are you monitoring? Do you use corelation IDs? Which tools do you use?

4. Do you collect log files centrally? Which tools do you use?

5. Do you analyze log files pro-actively? Which tools do you use?

Iterative Development

1. Are you iterations time-boxed? What is time-box size?

2. Is software tested and working at the end of each iteration?

3. Does an iteration start before the specification is complete?

Scrum Practices

1. Is the product owner known to all stakeholders? Is she/he easily accessible?

2. Is the product backlog prioritized by business value?

3. Does the product backlog includes the estimates created by the team?

4. Are project managers (and other) disrupting the work of the team?