

Question 1 Answer

To maintain code quality:

- code review process on every story(branch) before merging with master
- have code standards agreed in the team; There are many principles we need to keep in mind like S.O.L.I.D. (Single responsibility; Open-closed; Liskov substitution; Interface segregation; Dependency inversion); Don't Repeat Yourself (DRY); Keep It Simple Stupid (KISS); You Aren't Gonna Need It (YAGNI); Big Design Up Front; Avoid Premature Optimization; Principle Of Least Astonishment; CQS - Command Query Separation;
- code comments are not good idea as the code should be clear and what is doing can be traced in the tests - a documentation of the application for developers
- unit tests that cover the story
- integration tests (depending on the story)
- code test coverage
- coderules checks before commit, this could be configured to show errors or warnings while development too
- stories need to be clearly defined before the work on them starts; stories dependencies need to be clear so can help with the code design.

To maintain code stability:

- tests need to run as part of CI/CD
- before deployment test on mirror of live environment – staging – include performance testing here
- having role-back deployment process in place

Question 2 Answer

In the described situation is not allowed to have downtime.

There are many cloud providers that allow easily configurable zero downtime deployments; In Azure, kubernetes running on azure or AWS, the principle is the same.

The new version of the application is deployed and initially not used. Then the traffic gradually is redirected to this new version. Before switching, a health check needs to be performed. With Web Api, there is no session, but if there was, session in DB or Redis cache can be used. The tricky part is common resources – Database – the versions should share the same db or there should be synchronisation from old to new one.

As we need to deploy on different time-zones, we probably want to manage many pods(instances of the app) for different time-zones and the new version will replace the old version for the specific region. Independent of that depending on the traffic we will need more pods to add to the cluster or less.

The whole process is usually part of CD (Continuous Deployment) dev-ops config.