Quiz 10

Due Nov 15 at 10pm	**Points** 10	**Questions** 5	**Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	14 minutes	8 out of 10

Correct answers are no longer available.

Score for this quiz: **8** out of 10
Submitted Nov 15 at 9:27pm
This attempt took 14 minutes.

Question 1 — 2 / 2 pts

Select all of the correct statements

- [x] Refactoring is an important tool for software development
- [x] Refactoring can help to improve code readability
- [ ] Only code should be refactored, not design
- [x] Test Driven Development makes refactoring easier
- [x] Refactoring helps you to find bugs

Question 2 — 2 / 2 pts

True or False and why?: Configuration Management is only important for big projects so I don't need to worry about it. Why?

Your Answer:

False, Whether the business or project is small but the configuration management is mandatory because it helps building and identifying the configurations of a project and helps to keep track for controlling changes do the needful changes in the configuration

Configuration management is important for all projects, both large and small.

A good CM solution has many advantages that are important to all teams including:

- each access to all versions of important documents, e.g. code

- CM helps to facilitate collaboration across teams by allowing different people to work on the same files simultaneous

- CM provides backup across devices

Question 3

2 / 2 pts

What information should we track in configuration management for your current project?

- [x] Source code files
- [x] Code Documentation
- [x] Architecture and Design documents
- [x] Issues and design decisions

Question 4

0 / 2 pts

You've been assigned to work on a new release of a development project with three other developers. The project has a core set of classes, developed in the previous release, that are shared by all developers and each developer is responsible for implementing independent features for the next release. The project GitHub repository currently has only a "master" branch. Describe an optimal branching solution for the upcoming release. Will all four developers work and commit directly to the "master" branch? Will you create one or more sub-branches? What are the benefits of your solution?

Your Answer:

I would recommend creating 4 branches in this case since there are four developers. As and when needed they can merge it to the dev branch which can be tested for integration properly before merging to master. Forming different branches will help the team members to work on their given tasks independently without any conflicts in the code.

One good solution is to fork a new "dev" branch from the "master" branch where the developers will push their changes. It's best to have everyone working in the same branch so the branches don't get too far out of sync and any integration problems can be identified and fixed quickly.

Branches should be created by feature, not by developer. Too many branches arent useful because it grows complexity too quickly

Question 5

2 / 2 pts

How is the merge done by a pull request different from the merge that is done when pushing or pulling code from a branch in GitHub? Hint: who is involved in a pull request vs a push/pull.

Your Answer:

In pull request, you make the change and push it to the repo and wait for feedback and approval of other developers working with you, whereas in push pull we just have to merge after making change without having to wait for others approval. Anyone can merge changes into a branch.

Anyone can merge changes into a branch. Pull requests are a request from contributors for the project leaders to review the changes and either to approve or reject those changes.

Quiz Score: **8** out of 10