Assignment #4

Question:

Some people argue that developers should not be involved in testing their own code but that all testing should be the responsibility of a separate team. Give arguments for and against testing by the developers themselves:

Answer:

The team understands the problems that their program is tackling, and thus understands the problems that it might run into. They can write tests that accurately reflect the programs' future challenges. An outsider team may try to verify the code, without fully understanding the initial problem.

Depending on the team, and often pressured by time, they could write test units that insufficiently test the quality of the code, or are unable to think of challenges to their code beyond the solution that they've already tested.

Question:

Explain the differences between software reengineering and refactoring?

Answers:

Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code, yet improves its internal structure. It is a disciplined way to clean up code that minimizes the chances of introducing bugs. In essence when you refactor you are improving the design of the code after it has been written.

Although refactoring does not add features or functionalities in a software system, it is sharp weapon for developers in their maintenance activities. It makes a software system easier to understand and cheaper to modify without changing its observable behavior by changing its internal structure.

The purposes of refactoring:

1. Refactoring Improves the Design of Software
2. Refactoring Makes Software Easier to Understand
3. Refactoring Helps Finding Bugs
4. Refactoring Helps Programming Faster

The purpose of re-engineering:

1. To explain why software re-engineering is a cost-effective option for system evolution
2. To describe the activities involved in the software re-engineering process
3. To distinguish between software and data re-engineering and to explain the problems of data re-engineering