Q1

- Explain how standards may be used to capture organizational wisdom about effective methods of software development. Suggest four types of knowledge that might be captured in organizational standards.
Standards encapsulate organizational wisdom because they capture good practice that have evolved over the years. Knowledge that might be captured in organizational standards include:

1. Knowledge of specific types of fault that commonly occur in the type of software developed by an organization. This might be encapsulated in a standard review checklist.

2. Knowledge of the types of system model that have proved useful for software maintenance. This can be encapsulated in design documentation standards.

3. Knowledge of tool support that has been useful for a range of projects. This can be encapsulated in a standard for a development environment to be used by all projects.

4. Knowledge of the type of information that is useful to include as comments in code. This can be encapsulated in a code commenting standard.
Q2

- Assume you work for an organization that develops database products for individuals and small businesses. This organization is interested in quantifying its software development. Write a report suggesting appropriate metrics and suggest how these can be collected.
Answer:

- The organization is interested in quantifying its software development so may collect metrics about its products and about its processes.
- The type of software which is developed is important as the metrics should take into account its characteristics. In this case, the company is developing database products for microcomputers so:
  - As they are shrink-wrapped products, they will run on many different system configurations. Configuration dependent problems may occur. It is important that the system should not hang the machine on which it is running.
  - As they are database products, it is important that the system does not corrupt the database.
Product metrics

Product metrics should be used to judge the quality and efficiency of the software.

1. Total number of measured faults detected by testing
2. Total number of faults which resulted in database corruption
3. Total number of system failures which forced a system restart
4. Number of database transactions processed per unit time.
5. Time to read/write large DB records
Answer:

- Process metrics
- 1. Number of different configurations used for system testing
- 2. Number of fault reports submitted
- 3. Average time required to clear fault after it is reported
- 4. Time required to run system regression tests
Q3

- Explain why program inspections are an effective technique for discovering errors in a program. What types of error are unlikely to be discovered through inspections?
Program inspections are effective for the following reasons:
1. They can find several faults in one pass without being concerned about interference between program faults.
2. They bring a number of people with different experience of different types of errors. Hence, the team approach offers greater coverage than any individual can bring.
3. They force the program author to re-examine the program in detail – this often reveals errors or misunderstandings.

The types of errors that inspections are unlikely to find are specification errors or errors that are based on a misunderstanding of the application domain (unless there are domain experts in the team).
Q4

- Explain why it is difficult to validate the relationships between internal product attributes, such as complexity and external attributes, such as maintainability.
The basic difficulty arises because the external attributes such as maintainability are not just dependent on a small number of internal product attributes. While the complexity of a system influences its maintainability, other issues such as the use of variable names, the system documentation and, particularly, the skills of the people doing the maintenance have such a large effect on the process that they may mask any maintainability differences arising from different levels of complexity. This does not contradict experiments where a relationship between maintainability and complexity was discovered – however, we don’t have enough evidence at the moment to generalize this.
Thank you