Software Testing Overview

Objectives of testing:

1. Detect bugs or faults
2. Prevent bugs or faults
3. Measuring quality
4. Providing information

General principles:

1. Start testing as early as possible in the process (Requirements → Design specification → alpha version → beta version → product).
   Bugs are much easier to fix the earlier they are detected.
2. Exhaustive testing is impossible – except in simple cases or very high-risk applications.
   Tailor testing effort depending on circumstances / context.
3. An absence of bugs does not mean the product does what you want.
   Verification shows that the system meets the requirements, Validation shows that it meets
   the needs of the users, i.e. it is fit for the purpose for which it was built.
4. Pesticide Paradox
   Software will become “resistant” to tests, so vary your testing at each stage and avoid over-
   repeating tests.
5. Bugs occur in clusters – for instance, in complicated functions or requirements.
   Target high-risk areas for extra testing.

Distinguish between:

1. **Static Testing** – reviewing documentation and source code
   Can be used early in the process, cost-effective and gives a good measure of quality

2. **Dynamic Testing** – Actually running the code

Independence:

You must be emotionally detached to test software – a developer cannot reasonably test his own
work. Independence grades:

1. Testing your own work (not independent)
2. Peer review (independent, but still a developer so similar mind-set)
3. Internal tester – user or Quality Assurance
4. External third party – a professional tester, or the client company.