

3. The degree of independence avoids Author bias and is often more effective at finding defects and failures.

- There are several levels of finding independence in software testing which is listed from the lowest level of independence to the highest.

- i) Test by the person who wrote them.
- ii) Test by the another person works within the same team.
- iii) Test by the another person from different team.
- iv) Test by the another person from different company.

4. Clear and Courteous communication and feedback on defects between tester and developer.

- Since testing can be seen as destructive activity we need to take care while reporting defects and failures as politely as possible.

5. Code of Ethics:

- A written set of guidelines issued by an organization to its workers and management to help them conduct their actions in accordance with its primary values and ethic standards.

\* exercise honesty, diligence and objectivity in duty performance.

\* exhibit loyalty in all matters of organization's not engaged in acts or activities which are discreditable to the profession.

\* refrain from entering any activities that may be in conflict with interest of their organization.

- \* not accepting anything from of value from an employee, client etc that would impair their professional judgment.
- \* continually strive for improvement.
- \* not leak (release) sensitive information or news
- \* maintain/improve competency through continuing education.
- \* cooperate for mutual professional benefit.

Inf

### Software Industry Testing

- i) It is the process of executing a program or application with the intent of finding errors and bugs.
- ii) The requirement for software industry testing requires testing strategy, test plan, test data and test environment.

### Automobile Industry Testing

- i) It is the process of testing the automobiles with the intent of finding faults along with ensuring its reliability, durability and quality.
- ii) The requirement Automobile industry testing requires highest reliability and efficiency from research and development.

3) The  $\$n$  requires less budget for testing software.

3) The  $n$  requires more budget for testing automobiles.

4) The  $n$  performs the following tests:

4) The  $n$  performs following tests:

- i) whether the software meet the specifications or not.
- ii) whether the software response correctly to all kinds of inputs
- iii) whether the software performs its function within acceptable time or not.

- i) testing of alternative design including material selection, weight reduction functional performance
- ii) testing at extreme environmental conditions such as heat, cold, dampness, rain, etc.

5) Software testing is comparatively less costly as it requires only some testers, time and electricity cost.

5) automobile testing is comparatively more costly as it requires expensive parts and component along with high electricity bills.

6) If software testing fails it is less likely to have loss of life and properties.

6) If automobile testing fails it is more likely to have more loss of life and properties.



## 1st Software Issues

### #1 causes of Software Defects & clarification

#### 1. Human factors:

- The human factor beings develop software but they are prone to make mistakes.
- so, to rely on human intelligence to develop software have chances of leaving error on software.

#### 2. Communication failure:

- The communication failures includes miscommunication or lack of communication.
- The communication failure can happen at different levels (requirement gathering stage, documentation stage etc).
- This could lead to a situation where programmers would have to deal with problems that are not clearly understood, thus leading to errors.

#### 3. Unrealistic development timeframe:

- When there are unrealistic timeframes, then compromises are made in requirement, design to meet delivery schedule.
- so, the programmers do not give enough time to design, develop and test their code before handing it over to the testing team which likely to introduce errors.

#### 4. Poor design logic:

- lack of proper design and proper feasibility study before designing the architecture can invite errors.

#### 5. Poor coding practices:

- Bad coding practices, such as inefficient error handling, lack of proper validation (data type,