UNIT-IV

Common People Issues

Perceptions and Misconception about Testing

- Testing is not Technically Challenging
- Testing does not provide me a career path or growth
- I am put in testing what is wrong with me?
- These folks are my adversaries.
- Testing is what i can do in the end if i get time.
- There is no sense of ownership in testing.
- Testing is only Destructive.

Testing is not Technically Challenging.

- Requires a holistic understanding of the entire product rather than just a single module. Typically, development engineers tend to focus on specific modules.
- Requires through understanding of multiple domains.
- Specialization in languages. Most development stakes their claim at expertise by gaining proficiency in some specific programming languages or platform.
- Use of tools: People in other activities of a software life cycle had access to a wide variety of tools CASE tools, debuggers, IDE's.

Testing does not provide me a career path or growth.

- Testing is not a devil and development is not an angel; opportunities abound equally in testing and development.

I am put in testing what is wrong with me?

- If a person is not suitable for development, for development, for the same or similar reason, he or she may not be suitable for testing either.
- Filling up positions for testing should not be treated as a second-string function. If the hiring and allocation policy is such that all the toppers in a class are allocated to development functions and testing functions get the leftovers.
- A person should be assigned to testing only when he or she has the right aptitude and attitude for testing.
- Appropriate recognition should be given to the engineers who participate in activities such as testing, maintenance, and documentation.

These folks are my adversaries.

- Testing and Development teams should reinforce each other and not be at loggerheads.
- Since the main function of testing is to find defects in the products, it is easy for an adversary attitude to develop between the testing team and the development team.

Testing is what i can do in the end if i get time.

- Testing is not what happens in the end of a project-it happens throughout and continues even beyond a release.
- Giving the testing team the freedom to mandate a minimum quality in the product before it can be released.

There is no sense of ownership in testing.

- Testing has deliverables just as development has and hence testers should have the same sense of ownership.

Testing is only Destructive

- There is a perception is that test engineers are only working towards "breaking the software" or testing to prove "the product doesn't work". This is not entirely true.
- The job of testing is a combination of pointing out what works in the product, "breaking the software and analyzing the risk of releasing the software with the defects.

Comparison between Testing and development functions

1. Testing is often a crunch time function:

Even though there are different phases of testing and different type of testing that get distributed throughout software life cycle.

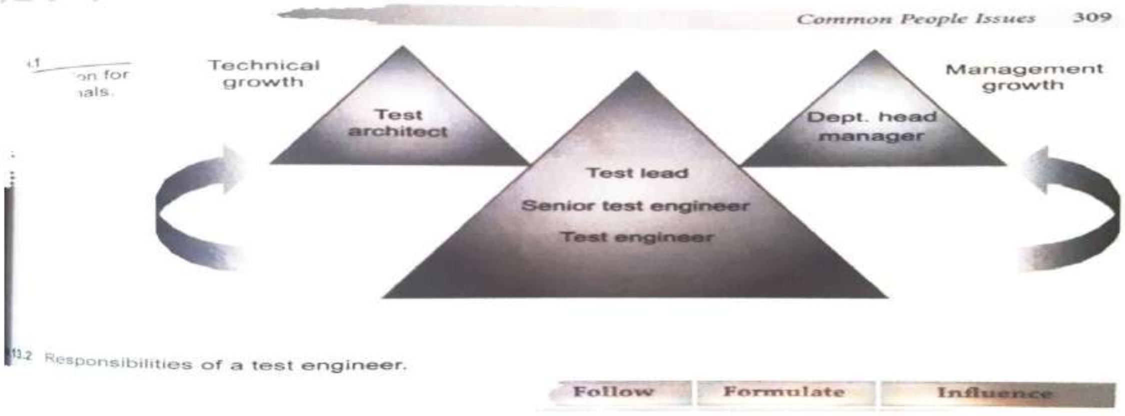
2. Generally more" elasticity" is allowed in projects in earlier phases.

It is considered almost normal to expect that the development functions will take longer than planned.

3. Testing functions usually carry more external dependencies than development functions.

There are two factors that contribute to this. The first factor is that testing comes at the end of a project life cycle.

Providing Career Paths for Testing Professionals



When people look for a career path in testing some of the areas of progression, they look for are

1. Technical challenge;
2. Learning opportunities;
3. Increasing responsibility and authority;
4. Increasing independence;
5. Ability to have a significant influence on an organization's success;
6. Rewards and recognition.

- The first stage is the **follow** stage. At the beginning of this stage, the trust, the trust and confidence level between the individual and the organization just starts to evolve.
- The individual is ready to move from just executing tasks by following instructions to **formulating** tasks, for him as well as for others.

The Role of the Ecosystem and a Call for Action

1. Role of Education System:

- The right values can only be more effectively caught by the students than be taught by the teachers!
- There are formal core courses on programming but only a few universities offer core courses on software testing =; most do not have a formal full course on testing.
- There are lab courses for various development tools but none or very few for common testing tools.
- Even during courses like operating systems and databases, the emphasis exercises and practical work is only on the programming aspects, not an appropriately testing the built product.

2. Role of senior management:

- Not allowing development engineers to look down upon test engineers.
- Encouraging and consciously ensuring that there is active job rotation among development, testing, and support functions.
- Demonstrating equity in recognition and reward schemes to the top performers in all the functions, including testing.

3. Role of the Community:

- As members of test community, do you have pride and sense of equality?
- Test engineers need not be just followers of technology-they can take an active part in shaping and using new technology.
- There is a misconception that new technology is difficult to understand for test engineers, and that only developers are better equipped to understand it.

Dimensions of organization structure:

Organization structures directly relate to some of the issues, in addition the study of the organization structure is important from the point of view of effectiveness because an appropriately designed organization structure can provide accountability to results.

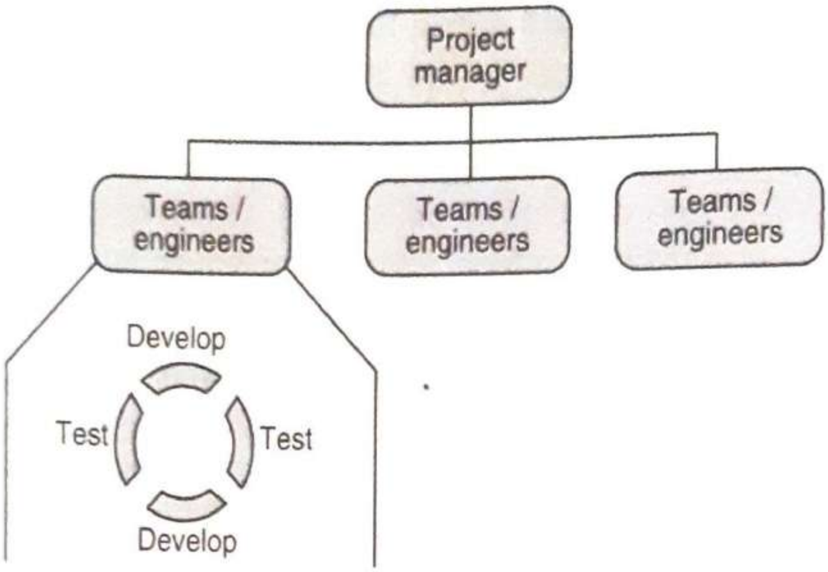
There are two types of organization structure:

- **Product organization**
- **Service Organization**

Product organization: Product organizations produce software products and have a "womb to tomb" (design, development, testing and maintenance of product) responsibility for the entire product.

Service organizations: Service organizations do not have complete product responsibility; in the testing context they are external organizations that provide testing services to other organization that require them.

Testing Team Structures for single product companies:



A small organization in the early stages of development, there is very little management hierarchy and people playing the roles of "managers", "leads" and so on actually are also "engineers" who are expected to act as individual contributors as well.

Ex: Start-up companies.

This model offers some advantages that are well suited to small organizations.

1. Exploits the rear-loading nature of testing activities:

Even though testing activities are distributed throughout the project life cycle, the major concentration and pressure points for testing arise during the latter part of the project life cycle.

2. Enables engineers to gain experience in all aspects of life cycle:

Since an engineer divides his or her time between development and testing, he or she can get to know what is involved in all the life cycle activities. This also makes them appropriate the importance of and difficulties in all the activities and prepares them for a better teamwork.

3. Some defects may be detected early:

Since the developers perform the testing functions there is possibility that they may catch the defects closer to the point of injection.

4. Accountability for testing and quality reduces:

Since the same persons perform both the development and testing functions, providing accountability for testing and quality becomes a challenge.

5. Developers do not in general like testing and hence the effectiveness of testing suffers:

In this model they may take up the tasks of testing as a price they have to do the “cool job” of development. This lack of intrinsic motivation for performing the testing function can have a telling effect on the effectiveness of testing.

6. Developers may not be able carry out the different types of tests:

A developer may not be equipped to carry out all the tests, because such testing may require specialized infrastructure or skills.

Ex: Performance testing may not be an area where general product developers can perform effectively because he or she would have to be well versed in ascertaining typical workloads.

- There is clear accountability for testing and development.
- Testing provides an external perspective. Since the testing and development teams are logically separated.

Precautions that must be taken to make this model effective:

- First the project manager should not twist under pressure and ignore the findings and ignore the findings and recommendations of the testing team by releasing a product that fails the test criteria.

- The project manager must ensure that the development and testing teams do not view each other as opponent(adversaries).

Structures in Multi-Product Company:

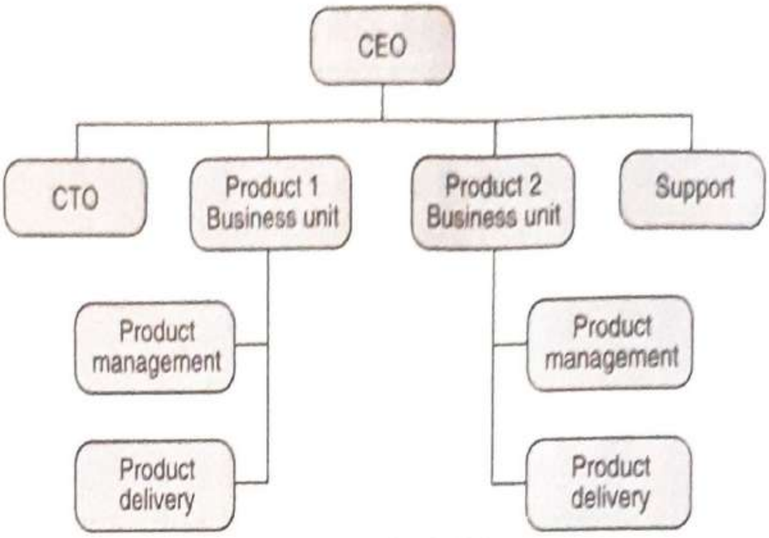


Figure 14.1

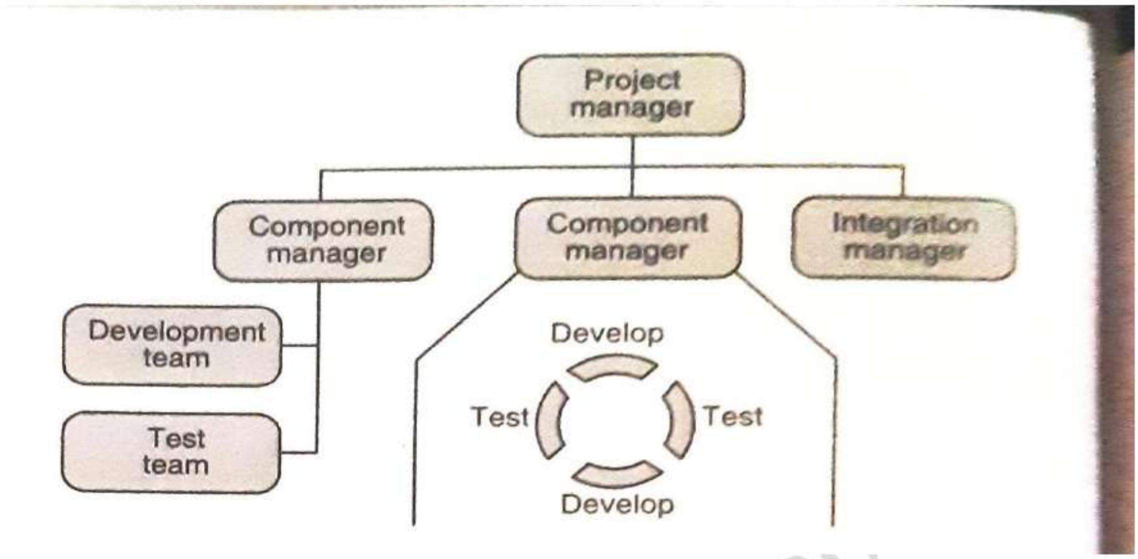
Organization structure of a multi-product company

The CTO's (Chief Technology Officer) office sets the high-level technology directions for the company. A business unit is in charge of each product that the company produces. A product business unit is organized into a product management group and a product delivery group. The product management group has the responsibility of merging the CTO's directions with specific market needs to come out with product road map. The product delivery group is responsible for delivering the product and handles both the development and testing functions.

Component-wise system Testing

If a company produces only one product, the product is made up of a number of components that fit together as a whole. Each component may be developed and tested by separate teams and all the components integrated by a single integration test team reporting to the project manager.

Component testing is defined as a software testing type, in which the testing is performed on each individual component separately without integrating with other components. It's also referred to as Module Testing when it is viewed from an architecture perspective. Component Testing is also referred to as Unit Testing, Program Testing or Module Testing.



When to perform Component testing

Component testing is performed soon after the Unit Testing is done by the developers and the build is released for the testing team. This build is referred as UT build (Unit Testing Build). Major functionality of all the components are tested in this phase,

Entry criteria for component testing

- Minimum number of the component to be included in the UT should be developed & unit tested.

Exit criteria for component testing

- The functionality of the entire component should be working fine.
- There should not presence of any Critical or High or Medium severity & priority defect log.

Example: Consider the home page it contains many components such as Home, Testing, SAP, Web.