People Skills and Team Composition

1. **Individual Skills**
2. **Test Team Dynamics**
3. **Fitting Testing Within an Organization**
4. **Motivation**
5. **Communication**

1. Individual Skills:

- As **test managers**, we need to have a team with the appropriate balance of skills.
- As test managers, we can achieve the right balance of skills in a couple of ways.
  - One is by trying to ensure even distribution of skills across the entire team, which is referred to as the team of **generalists approach**.
  - We can also achieve that balance by allowing individuals to have particular strengths in some needed skills, with the team collectively strong in all necessary areas, which is referred to as the team of **specialists approach**.
  - There are innumerable variations on these approaches, as you can imagine. Whichever way you choose, a technique called **skills inventories** can help you manage this process.
- Tester skills grow through a combination of experience and training.
  - Using experience alone to grow your team is inefficient and leads to subtle weaknesses in your individuals and your team as a whole.
  - Using training alone to grow your team is ineffective and leads to a team with strong theoretical knowledge but little ability to apply the theory to real testing problems.
  - The ideal approach is to use targeted training followed by immediate application of the training concepts in real-world experience.
- When thinking about the skills to grow in your individual testers and across your team as a whole, keep in mind these three main areas of tester skills:
  - Use of software systems, including knowledge of the domain or business. In other words, does the tester understand the problem that the system must solve?
  - Understanding of software development, including analysis, development, technical support, and technical documentation. In other words, does the tester understand the practical issues related to creating, deploying, and supporting systems?
  - Understanding of software testing. In other words, does the tester know how to apply testing best practices to the problems they will face?

1.1. Testing Skills:

- Tester:
  - All testers, whatever their role, need good interpersonal skills.
  - They should be able to effectively give and receive criticism.
  - The nature of the test job requires a level of emotional toughness due to the pressures of the job and the fact that we are often the bearers of bad news, but that toughness must not cross the line into sadism, cynicism, or callousness.
  - Many testers in the business are technically competent—indeed, some are quite technically brilliant—but find themselves hampered on their projects and in their professional success by their poor interpersonal skills.
  - If you find yourself often ignored, involved in heated arguments with other project team members, viewed as an adversary, kept in the dark about important project details, put on people's email blacklists, or stymied in your career advancement, look at your interpersonal skills critically and honestly assess your role in your career problems.

- Testing is about generating information, and it tends to generate a lot of it. Testing often involves generating, editing, reviewing, absorbing, and analyzing a lot of technical documents and metrics. Therefore, testers need good organizational skills; this is especially true for disorganized and chaotic projects.
- Because testing generates information, your ability to be effective as a tester is only as good as your ability to convey that information. You need strong written and verbal communication skills.
- Testers must see the broader aspects of the project but at the same time focus on important details
- They should be able to read and analyze a requirements specification, a design specification, a use case, and other information about the system.
- They should be able to participate in product and project risk analysis activities.
- They should be able to apply the concepts discussed in the Foundation and Advanced syllabi to design an effective and efficient set of tests. These skills include creating test data, selecting test oracles, defining expected results, and, in some cases, participating in test automation.
- They should be able to run tests, evaluate the results, and record those results in a meaningful way. These skills include logging of test status, test incidents, and test-impacting events. It also includes capturing metrics.

- **Test Leader:**
  - To get work done and to have your findings accepted and used appropriately, a degree of skill in influencing and negotiating is typically required.
  - Testing generates information, your ability to be effective as a tester is only as good as your ability to convey that information. You need strong written and verbal communication skills. These skills become especially important for test managers.
  - They should be able to manage a project, because testing is often a sub-project within a larger project context. Test managers should have the skills required for all the usual project management tasks.
  - They should be able to carry out the various activities related to test planning. These skills include risk analysis, test estimation, and writing test plans. It also includes formulating effective test strategies and helping to set the organization's test policy.
  - They should be able to track test progress. These skills include capturing the appropriate metrics and logs. It also includes the ability to analyze those metrics, evaluate variance from the project plan and the test plan, and suggest and implement appropriate test control activities.
  - They should be able to report test results. These skills include creating, presenting, and explaining the kinds of charts and reports. The ability to effectively communicate the results to project stakeholders is critical. Without this ability, all the work done in planning, creating, and running the tests can be wasted when the project team rejects or misunderstands the findings.

1.2. Technical and Software Skills

- It is true that testers over time, with repeated exposure to a system, will become more tolerant of its failures.
- Testers who understand requirements analysis, design, and coding have insight into the bug lifecycle. In other words, they know how programmers introduce defects, where they can detect defects, and, better yet, how to prevent the introduction of defects in the first place.
- Testers who understand technical support issues have insight into the user experience, user and customer expectations, and usability issues.
- In fact, we have clients that populate their test teams almost exclusively with former technical support staff. That staffing technique works well when the erstwhile technical support people are given the skills needed in other areas.

- When they are not, they tend to retain the firefighting, solve-the-problem-immediately attitude of technical support staff, which can create real problems for the test team.

1.3. User, Business, and Domain Skills

- At the risk of stating the obvious, people who spend their time getting real work done with the system have the essential perspective of the user. They know how to use and operate the system. They have a good understanding of failure priority, which helps during both quality risk analysis and incident management. They have a clear grasp of what "correct behavior" looks like, or at least what they want it to look like, which makes them and their opinions useful inputs for test oracles.
- Business and domain experts, whether users or not, also have valuable insight. They know the importance of various features to the business. They know how certain behaviors would affect the business. Again, this makes them valuable participants in both quality risk analysis and incident management.
- In addition to quality risk analysis and incident management, both users and domain experts can help with various other test activities. They can help to prioritize testing activities, particularly if various project constraints make it difficult to run tests in purely risk order. They can provide or help to create realistic test data and test cases. They can verify or, better yet, supply use cases.
- As a test manager, you want to draw on the abilities of users and domain experts sparingly. For one thing, these people usually have other work assignments, and participating in testing tasks distracts them from that real work and might prove irritating to them. I made some political trouble for myself early in my career as a test manager by relying too heavily on domain experts who worked for another manager, leading to complaints that I was practicing "vampire management" and also neglecting the skills deficits in my team.
- For another thing, users and domain experts sometimes feel they have the right to dictate to members of the project team. In that case, without careful management of the involvement of these folks, you can easily find yourself in a situation where outside staff and managers are second-guessing, micromanaging, and commanding your test effort. As a manager, you must know how to tactfully solicit help without abdicating ownership of your team.

2. Test Team Dynamics:

- As a test manager, test team development is probably the most critical testing process of all. If you develop your team properly, you can often survive other classic test manager mistakes because your strong team will cover for you.
- New test team members are often hired, of course, but they can also be transferred from within the organization.
- The issue of the right team members is not only one of individual skills. You also must consider team dynamics.
    - It’s easy to determine how to select a person with complementary skills; someone who can help fill weak spots in your team and cross-train others.
    - You also need to consider personality type.
    - As a test manager, you should see the whole test team development process as an ongoing effort to build the best possible test team. The team should be suitable for various projects. In addition, they should work well with the various project team members with whom they will interact
    - When you add people to the team—whether by hiring them or by transferring them from within the organization—you want to ensure that the experience of joining the team is positive from the beginning.

3. Fitting testing within an organization:

3.1. The degrees of independence

- **Self-testing** occurs when the developer tests their own code. There is no independence here, of course.
  - The author bias problem is significant, and the developer—even if given enough time to do unit testing—often misses the important bugs because they determine that the code works as they intended. Of course, that might not match the actual requirements.
  - The advantages are that the developer can fix any defects they find quite quickly and, being quite technical, understands the software being tested.
- **Buddy testing** occurs when developers test each other's code but not their own code. Pair programming, which is a practice in some agile techniques, is a special form of this, where development of code, continuous code review, and development and execution of unit tests by a team of two programmers evolves the code.
  - While the author bias problem is not so acute here, when two people work closely together, it's hard to say there is much independence between the developer and the tester.
  - In addition, there tend to be few if any usable defect metrics captured in this situation without careful cultivation of the proper mindset since peers testing each other's code might not want to report defects.
  - Finally, because the average programmer has little training or formal experience with testing, the mindset is usually focused on positive tests.
  - Once again, the advantages include a quick repair of defects and good tester understanding of the software being tested.
- **Self-editing** means that the tester does not report—or reports only informally to the developer—those problems found, leaving no official trail in a bug tracking system.
  - Self-editing is the equivalent of an organization tearing out its eyes and flying blind with respect to quality.
- Testing by business, users, and technical support occurs, often in the context of acceptance testing and beta testing.
  - This approach has the advantage of a truly independent outlook, motivated to report findings truthfully to the stakeholders.
  - What these folks typically care about is the ability to get their job done, and if quality's not there, they'll suffer. This is a great approach for the final levels of testing.
- **Test specialists** in an independent test group occur in many thoughtful organizations, with the independent test team responsible for system test, system integration test, and, in some case, component integration test.
  - In this case, we have all the advantages of true, professional testers testing against specific test targets. We often see test targets beyond functionality, including usability, security, and performance.
  - For all the advantages of an independent test team, it should be kept in mind that the formality usually associated with such teams does tend to slow down the process. It's also possible that reporting structures or poor management can lead to perverse incentives and a lack of focus on quality.
- Testing by an **external test organization** occurs in a number of settings.
  - For example, in certain military contracts, independent verification and validation by a team not in any way associated with the prime contractor is required. As another example, you might hire a test lab to do compatibility testing for an e-commerce website to save the expense of having all the configurations in-house.

- Here, the maximum level of independence is achieved.
- Of course, the separation of test and development duties might mean that the knowledge transfer necessary for thorough testing might not occur.
- To make up for these disconnections, the organizations must put in place very clear requirements and well-defined communication structures.

3.2. On Using Different Mixes of Independence

- Increasing independence of testing is not without risks, as I discussed earlier. More independence can result in more isolation.
- It can reduce the level of insight and understanding of what is going on in the project.
- It can also lead to a loss of ownership and responsibility for quality on the part of those developing the code.
- These are not necessary outcomes of independent testing, as some of those who argue a dualistic view of this question suggest, but simply project risks that the manager of an independent test team must mitigate.
- Decreasing independence of testing is not without risk either. It can increase insight and understanding of the project—and this is the outcome touted by many of the dualists—but it can and often does introduce conflicting goals.
- Decreasing independence can lead to blinds spots as to what the requirements really are.
- Decreasing independence also decreases the degree to which testing involves people who possess focus and specialization on testing, and thus teams have an imperfect, skewed skills mix.
- When you do split up the testing across various entities with various degrees of independence, the usual rules of pervasive testing apply.
- Make sure that you define the responsibilities and expectations for each test level and entity doing testing.
- A concise, clear test policy document, developed with the participation of all the entities and approved by senior management, can accomplish this.
- By doing this, you'll be setting up a mix of different filters deployed at the ideal spot in the lifecycle, which can maximize quality within the schedule and budget constraints of the project.

3.3. Outsourcing as Independent Testing

- Outsourcing of testing is one form of external, independent testing. This can take a number of forms. One is hiring an outside testing company to provide collocated testing services (which is sometimes called insourcing).
- Another is to have the testing done at an external facility located close to the development team.
- Yet another is to have the testing done at an external facility that is some distance away, perhaps even in different time zones.
- Some challenges of outsourcing of testing:
    - The outsource testing team might have cultural differences with your test team, your development team, or both.
    - The project team and local test team might have difficulty providing timely, adequate supervision and direction, particularly on chaotic, constantly changing projects.
    - Due to a lack of foresight, significant communication problems can exist between the local project team and the outsource test entity. Poor communication can compound the supervision problem.
    - Without careful contracting, you can have problems with protection of intellectual property. Even with good contracts, in some countries your legal recourse might be quite limited.
    - Again, if insufficient care is taken during the contracting of the testing, including especially the selection of the outsource test vendor, the skills of the testers can be questionable

- Exacerbating this skills problem can be the problem of employee turnover. Again, proper contracting and vendor selection can help reduce this
- Because companies pursuing outsourcing often forget to include their own costs of managing the relationship in the overall budget, outsourcing does not always involve accurate cost estimation.
- Finally, quality of the work can suffer.

4. Motivation:

- As test managers, we need to motivate our team members
- There are number of ways:
  - **Recognition.**
    When someone does good work, tell them so. Public praise for a job well done is a major motivator for many people.
    The converse, by the way, is not true: Public criticism for a job poorly done will not motivate someone to do better.
    You should give criticism privately, and make sure that criticism is constructive and guides the tester toward the desired improvement.
  - **Management approval**: This doesn't just mean your approval. As a test manager, promote your team and get upper-management approval for your work. Then, share that with your team.
  - **Respect**: As a test manager, you want to build a culture of mutual respect in your team. You must also ensure that the project team treats your testers with respect. Of course, that respect must be earned.
  - **Adequate rewards**: This includes financial elements, such as salary, merit increases, and bonuses. It also includes nonfinancial elements such as training and career growth.
- It happens that project realities constrain the motivational tools available to you. If the team is subject to impossible deadlines, that makes the job hopeless. In addition, tight project deadlines usually mean poor quality when testing starts because everyone is taking shortcuts.
- Testers should not take bug deferral personally, if it is done thoughtfully and with an eye toward what matters to customers
- Crunch time happens on projects, and testers should be ready to put in extra hours to accomplish important tasks. However, mandatory overtime that clearly accomplishes nothing is demotivating
- Finally, being held responsible for quality when everyone else on the project is taking steps that actively undermine quality—being the quality scapegoat—is demotivating.

4.2. Metrics and Motivation:

- Properly used, though, product, process, and project metrics can demonstrate the value of testing.
- Define an assessment program for your test team.
- In consultation with the test team and other project stakeholders, and based on what you want to see, set goals and publish progress toward those goals.
- This will help you demonstrate value and progress to management and will also help show testers that, in spite of some daily frustrations, you are collectively, as a team, making progress.
- Recognition takes various forms. People want respect from their peers and managers and peer and manager approval of their work. They want promotional opportunities. They want fair pay, relative to their peers and the industry. They want a career path.
- A test team that is not respected will not be recognized. One that is respected can make meaningful contributions and will be recognized; that is a virtuous cycle, and your job as a test manager is to create and sustain that cycle.
- To create and sustain this cycle of recognition and respect, you must be able to demonstrate value. Metrics are essential to demonstrate value. Test managers who insist on respect and recognition for

their test teams based purely on subjective reports of good work done—or worse yet, based on an assertion that the test team deserves respect—tend to be disappointed in the organization's reaction.

5. Communication:

- There are three levels of communication for most test teams:
  - First, we communicate, mostly internally but also with others, about the documentation of test products. These communications include discussions of test strategies, test plans, test cases, test summary reports, and defect reports.
  - Second, we communicate feedback on reviewed documents, typically on a peer level both inside and outside the test group. These communications include discussions about requirements, functional specifications, use cases, and unit test documentation.
  - Third, we communicate as part of information gathering and dissemination. These communications include not just peer level communications but communications to managers, users, and other project stakeholders. It can be sensitive, as when test results are not encouraging for project success.
- Both internal and external communications are important to the professionalism of the tester and test manager
- It's important to be professional, objective, and effective. When communicating about test results, giving feedback on issues with documents, or delivering any other potentially touchy news, make sure to use diplomacy.
- It's easy to become caught up in emotions at work, especially during test execution when things are often stressful. Remember to focus on achieving test objectives. Remember also that you want to see the quality of products and processes improved. Don't engage in communication that is contrary to those goals.
- We testers often speak in a sort of shorthand about very fine-grained details of our work and findings, and with a certain degree of skepticism.
- You have to remember to tailor communication for the target audience.
- When talking to users, project team members, management, external test groups, and customers, you need to think carefully about how you are communicating, what you are communicating, and whether your communications support your goals.