

# Test Plan Evaluation Model

The answer to the question “How good is this test plan?” can only be given with reference to an idea of what a test plan should be. Although there are a number of public standards that specify test plan document formats, they provide little basis for distinguishing a better plan from a worse plan. This model identifies basic concepts, functions that a test plan serves, criteria that a test plan should satisfy, and some heuristics to assist in determining if the criteria are satisfied with respect to the functions:

## Terms and Concepts

- *Test Plan.* The test plan is the set of ideas that guide or represent the intended test process. Often those ideas are only partially documented, spread across multiple documents, and subject to change as the project evolves.
- *Test Plan Document.* A test plan document is any document intended to convey test plan information. However, test plan documents are not the only source of information about the test plan. Test plan information is also contained in the oral tradition of the project and the culture of the company.
- *Test Strategy.* The test strategy is the way tests will be designed and executed to support an effective quality assessment. Test strategy is the plan for what parts of the product will be covered by tests and what test techniques will be used. Test strategy is distinct from the logistics of implementing the strategy. Test strategy is essentially the “brains” of the test process.
- *Test Project.* The test project is the means by which the test strategy is implemented and results delivered. The test project is the “brawn” of the test process.

## Test Plan Functions

Test plan functions are what a test plan is supposed to do. Below is a list of functions served by an ideal test plan. However, a test plan document may only address a subset of these functions—the rest handled in other documents or managed directly by the test manager or individual tester without the support of any document. Thus, a test plan should be judged only with regard to those functions that it intends to serve, or are insufficiently served by other means.

- Support the development of a quality assessment that enables wise and timely decisions to be made concerning the product.
- Describe and justify the test strategy (including proposed test coverage) in relation to technical requirements and technical risk. Promote awareness of the benefits and limitations of the test strategy.

- Describe and justify any special requirements or entry criteria that must be met in order for the test project to proceed, as well as any exit or process for determining when to stop testing.
- Support the initiation and organization of the test project, including preparations, staffing, delegation of responsibilities, facility acquisition, task planning, and scheduling.
- Support daily management and evaluation of the test project and test strategy.
- Support effective coordination, collaboration, and other relations among members of the test team, and between the test team and the rest of the project.
- Identify and manage any risks or issues that may impact the project.
- Specify the deliverables of the test project, and the delivery process.
- Record historical information in support of process audits, process improvement and future test projects.

## Test Plan Quality Criteria

These criteria relate to how well a test plan performs its functions. A test plan is good to the extent that it satisfies these criteria. Exactly how good is “good enough” depends on situational factors and judgments.

- *Usefulness.* Will the test plan effectively serve its intended functions?
- *Accuracy.* Is it accurate with respect to any statements of fact?
- *Efficiency.* Does it make efficient use of available resources?
- *Adaptability.* Will it tolerate reasonable change and unpredictability in the project?
- *Clarity.* Is the test plan self-consistent and sufficiently unambiguous?
- *Usability.* Is the test plan document concise, maintainable, and helpfully organized?
- *Compliance.* Does it meet externally imposed requirements?
- *Foundation.* Is it the product of an effective test planning process?
- *Feasibility.* Is it within the capability of the organization that must perform it?

## Test Plan Heuristics

In order to determine how well the test plan meets the criteria, heuristics are used. That is to say, a test plan evaluation is based on generally accepted rules of thumb we have collected through experience and study.

Each of the heuristics in the table below relates to one or more of the criteria and functions identified above. The only criterion in the list for which there is no corresponding heuristic is *compliance*. This is because compliance to externally imposed requirements requires specific knowledge of those requirements.

Each heuristic is described in terms of a general rule, and a brief basis for that rule. The basis is intended to help determine when and where a heuristic applies.

Heuristic	Basis for heuristic
1. Testing should be optimized to find important problems fast, rather than attempting to find all problems with equal urgency.	The later in the project that a problem is found, the greater the risk that it will not be safely fixed in time to ship. The sooner a problem is found after it is created, the lesser the risk of a bad fix.
2. Test strategy should focus most effort on areas of potential technical risk, while still putting some effort into low risk areas just in case the risk analysis is wrong.	Complete testing is impossible, and we can never know if our perception of technical risk is completely accurate.
3. Test strategy should address test platform configuration, how the product will be operated, how the product will be observed, and how observations will be used to evaluate the product.	Sloppiness or neglect within any of these four basic testing activities will increase the likelihood that important problems will go undetected.
4. Test strategy should be diversified in terms of test techniques and perspectives. Methods of evaluating test coverage should take into account multiple dimensions of coverage, including structural, functional, data, platform, operations, and requirements.	No single test technique can reveal all important problems in a linear fashion. We can never know for sure if we have found all the problems that matter. Diversification minimizes the risk that the test strategy will be blind to certain kinds of problems.
5. The test strategy should specify how test data will be designed and generated.	It is common for the test strategy to be organized around functionality or code, leaving it to the testers to concoct test data on the fly. Often that indicates that the strategy is too focused on validating capability and not focused enough on reliability.
6. Not all testing should be pre-specified in detail. The test strategy should incorporate reasonable variation and make use of the testers' ability to use situational reasoning to focus on important, but unanticipated problems.	A rigid test strategy may make it more likely that a particular subset of problems will be uncovered, but in a complex system it reduces the likelihood that <i>all</i> important problems will be uncovered. Reasonable variability in testing, such as that which results from interactive, exploratory testing, increases incidental test coverage, without substantially sacrificing essential coverage.
7. It is important to test against implied requirements—the full extent of what the requirements mean, not just what they say.	Testing only against explicit written requirements will not reveal all important problems, since defined requirements are generally incomplete and natural language is inherently ambiguous.

Heuristic	Basis for heuristic
8. The test project should promote collaboration with all other functions of the project, especially developers, technical support, and technical writing. Whenever possible, testers should also collaborate with actual customers and users, in order to better understand their requirements.	Other teams and stakeholders often have information about product problems or potential problems that can be of use to the test team. Their perspective may help the testers make a better analysis of risk. Testers may also have information that is of use to them.
9. The test project should consult with development to help them build a more testable product.	The likelihood that a test strategy will serve its purpose is profoundly affected by the testability of the product.
10. A test plan should highlight the non-routine, project-specific aspects of the test strategy and test project.	Virtually every software project worth doing involves special technical challenges that a good test effort must take into account. A completely generic test plan usually indicates a weak test planning process. It could also indicate that the test plan is nothing but unchanged boilerplate.
11. The test project should use humans for what humans do well and use automation for what automation does well. Manual testing should allow for improvisation and on the spot critical thinking, while automated testing should be used for tests that require high repeatability, high speed, and no judgment.	Many test projects suffer under the false belief that human testers are effective when they use exactly specified test scripts, or that test automation duplicates the value of human cognition in the test execution process. Manual and automated testing are not two forms of the same thing. They are two entirely different classes of test technique.
12. The test schedule should be represented and justified in such a way as to highlight any dependencies on the progress of development, the testability of the product, time required to report problems, and the project team's assessment of risk.	A monolithic test schedule in a test plan often indicates the false belief that testing is an independent activity. The test schedule can stand alone only to the extent that the product the highly testable, development is complete, and the test process is not interrupted by the frequent need to report problems.
13. The test process should be kept off of the critical path to the extent possible. This can be done by testing in parallel with development work, and finding problems worth fixing faster than the developers fix them.	This is important in order to deflect pressure to truncate the testing process.

Heuristic	Basis for heuristic
<p>14. The feedback loop between testers and developers should be as tight as possible. Test cycles should be designed to provide rapid feedback to developers about recent additions and changes they have made before a full regression test is commenced. Whenever possible testers and developers should work physically near each other.</p>	<p>This is important in order to maximize the efficiency and speed of quality improvement. It also helps keep testing off of the critical path.</p>
<p>15. The test project should employ channels of information about quality other than formal testing in order to help evaluate and adjust the test project. Examples of these channels are inspections, field testing, or informal testing by people outside of the test team.</p>	<p>By examining product quality information gathered through various means beyond the test team, blind spots in the formal test strategy can be uncovered.</p>
<p>16. All documentation related to the test strategy, including test cases and procedures, should be undergo review by someone other than the person who wrote them. The review process used should be commensurate with the criticality of the document.</p>	<p>Tunnel-vision is the great occupational hazard of testing. Review not only helps to reveal blind spots in test design, but it can also help promote dialog and peer education about test practices.</p>