

# How To Evolve a Context-Driven Test Plan

This guide will assist you with your test planning. Remember, the real test plan is the set of ideas that actually guides your testing. We've designed the guide to be helpful whether or not you are writing a test plan *document*.

This is not a template. It's not a format to be "filled out." It's a set of ideas meant to jog your thinking, so you'll be less likely to forget something important. We use terse language and descriptions that may not be suited to a novice tester. It's designed more to support an experienced tester or test lead.

Below are seven task themes. Visit the themes in any order. In fact, jump freely from one to the other. Just realize that the quality of your test plan is related to how well you've performed tasks and considered issues like the ones documented below. The *Status Check* sections will help you decide when you have a good enough plan, but we recommend revisiting and revising your plan (at least in your head) throughout the project.

---

## 1. Monitor major test planning challenges.

*Look for risks, roadblocks, or other challenges that will impact the time, effort, or feasibility of planning a practical and effective test strategy. Get a sense for the overall scope of the planning effort. Monitor these issues throughout the project.*

### Status Check

- Are any product quality standards especially critical to achieve or difficult to measure?
- Is the product especially complex or hard to learn? (...then you will need more time)
- Will testers require special training or tools? (...then better go get that soon)
- Are you remote from the users of the product? (...then might have to work harder to understand them)
- Are you remote from any of your clients? (...then set up more disciplined process of communication)
- Is any part of the test platform difficult to obtain or configure? (...then warn your clients about that)
- Will you test unintegrated or semi-operable product components? (...then prep tools and environments)
- Are there any particular testability problems? (...then advocate for testability)
- Does the project team lack experience with the product design, technology, or user base?
- Does test execution have to start soon? (...then focus on what you really need to get started)
- Is any information needed for planning not yet available? (...then warn your clients you need it)
- Are you unable to review a version of the product to be tested (even a demo, prototype, or old version)?
- Is adequate testing staff difficult to hire or organize? (...then don't wait long before trying to do that)
- Must you adhere to an unfamiliar test methodology? (...then you better study it and see if you can)
- Are you being pressured to formalize too soon? (...then push back, because informality comes first)
- Are project plans made without regard to testing needs? (...then warn your clients that will impair testing)
- Is the plan subject to lengthy negotiation or approval? (...then start ASAP)
- Are project plans changing frequently? (...then establish a way to find out when they do)
- Will the plan be subject to audit? (...then find out what that audit process is and when it happens)
- Are your clients unsure of what they want from you? (...then tell them what they should want)

---

## 2. Know your mission.

*Any or all of the goals below may be part of your testing mission, and some more important than others. Based on your knowledge of the project, rank these goals. For any that apply, discover any specific success metrics by which you'll be judged.*

### Mission Elements to Consider

- Find important problems fast.
- Perform a comprehensive quality assessment.
- Certify product quality against a specific standard.
- Minimize testing time or cost.
- Maximize testing efficiency.
- Advise clients on improving quality or testability.
- Advise clients on how to test.
- Assure that the test process is fully accountable.
- Rigorously follow certain methods or instructions.
- Satisfy particular stakeholders.

### Possible Work Products

- Brief email outlining your mission.
- One-page test project charter.

### Status Check

- Do you know who your clients are?
- Do the people who matter agree on your mission?
- Is your mission sufficiently clear that you can base your planning on it?

---

### 3. Know the product.

*Get to know the product and the underlying technology. Learn how the product will be used. Steep yourself in it. As you progress through the project, your testing will become better because you will be more of a product expert.*

#### What to Analyze

- Users (who they are and what they do)
- Structure (code, files, etc.)
- Functions (what the product does)
- Data (input, output, states, etc.)
- Interfaces (user interfaces, APIs, connections to platform components)
- Platforms (external hardware and software)
- Operations (how product is used in real life)
- Timing (performance variables, periodic functionalities, race conditions)

#### Ways to Analyze

- Perform survey testing (testing with the primary goal of learning about the product).
- Apply the Heuristic Test Strategy Model product elements guidewords.
- Review product and project documentation.
- Interview designers and users.
- Compare w/similar products.

#### Possible Work Products

- Product coverage outline
- Annotated specifications
- Product bug list
- Project issue list

#### Status Check

- Do designers approve of the product coverage outline?
- Do designers think you understand the product?
- Can you visualize the product and predict behavior?
- Are you able to produce test data (input and results)?
- Can you configure and operate the product?
- Do you understand how the product will be used?
- Are you aware of gaps or inconsistencies in the design?
- Have you considered tacit specifications as well as explicit?

## 4. Know the risk.

*How might this product fail in a way that matters? At first you'll have a general idea, at best. As you progress through the project, your test strategy, your testing will become better because you'll learn more about the failure dynamics of the product.*

### What to Analyze

- Threats (challenging situations and data)
- Vulnerabilities (where it's likely to fail)
- Failure modes (possible kinds of problems)
- Victim impact (how problems matter)

### Ways to Analyze

- Perform survey testing or general shallow testing to identify risk hotspots.
- Review product against risk heuristics and quality criteria categories.
- Review requirements and specifications.
- Review actual failures in the lab or in the field.
- Review code and architecture to understand failure modes and fault propagation pathways.
- Interview designers and users.
- Use a risk catalog to identify problems you want to specifically test for.

### Possible Work Products

- Component/risk matrix (outline of the parts of the product and risk factors associated with them)
- List of risk areas (clusters of related suspected risks)
- List of risk factors (threats and vulnerabilities)
- Risk catalog (outline of all the kinds of problems that typically occur with that technology)

### Status Check

- Do the designers and users concur with the risk analysis?
- Will you be able to detect all significant kinds of problems, should they occur during testing?
- Do you know where to focus testing effort for maximum effectiveness?
- Can the designers do anything to make important problems easier to detect, or less likely to occur?
- What makes you think your risk analysis is any good? Have a compelling story about that.

---

## 5. Decide the test strategy.

*What can you do to test rapidly and effectively based on the best information you have about the product? Make the best decisions you can, up front, but let your strategy improve throughout the project.*

### Consider Techniques From Five Perspectives

- Tester-focused techniques.
- Coverage-focused techniques (both structural and functional).
- Problem-focused techniques.
- Activity-focused techniques.
- Oracle-focused techniques.

### Ways to Plan

- Match techniques to risks and product areas.
- Visualize specific and practical techniques.
- Diversify your strategy to minimize the chance of missing important problems.
- Look for ways automation could allow you to expand your strategy.
- Don't overplan. Testers must stay awake by using their brains.

### Possible Work Products

- Itemized statement of each test strategy chosen and how it will be applied.
- Risk/task matrix.
- List of issues or challenges inherent in the chosen strategies.
- Advisory of poorly covered parts of the product.
- Test cases (only if required)

### Status Check

- Do your clients concur with the test strategy?
- Is everything in the test strategy necessary?
- Can you actually carry out this strategy?
- Is the test strategy too generic—could it just as easily apply to any product?
- Is there any category of important problem that you know you are not testing for?
- Has the strategy made use of available resources and helpers?

## 6. Know the logistics.

*How will you implement your strategy? Your test strategy is profoundly affected by logistical constraints or mandates. Try to negotiate for the resources you need and exploit whatever you have.*

### Logistical Areas

- Making contact with users
- Making contact with your clients
- Test effort estimation and scheduling
- Testability advocacy
- Test team staffing (right skills)
- Tester training and supervision
- Tester task assignments
- Product information gathering and management
- Project meetings, communication, and coordination
- Relations with all other project functions, including development
- Test platform acquisition and configuration
- Agreements and protocols
- Test tools and automation
- Stubbing and simulation needs
- Test documentation management and maintenance
- Build and transmittal protocol
- Test cycle administration, especially after changes
- Bug reporting system and protocol
- Test status reporting protocol
- Code freeze and incremental testing
- Pressure management in the end game
- Sign-off protocol
- Evaluation of test effectiveness, including escaped bug analysis

### Possible Work Products

- Project agreements
- Project issues list
- Project risk analysis
- Responsibility matrix
- Test schedule

### Status Check

- Do the logistics of the project support the test strategy?
- Are there any problems that block testing?
- Are the logistics and strategy adaptable in the face of foreseeable problems?
- Can you start testing now and sort out the rest of the issues later?

---

## 7. Share the plan.

*You are not alone. The test process must serve the project. So, involve the project in your test planning process. You don't have to be grandiose about it. At least chat with key members of the team to get their perspective and implicit consent to pursue your plan.*

### Ways to Share

- Engage designers and stakeholders in the test planning process.
- Actively solicit opinions about the test plan.
- Do everything possible to help the developers succeed.
- Help the developers understand how what they do impacts testing.
- Talk to technical writers and technical support people about sharing quality information.
- Get designers and developers to review and approve reference materials.
- Record and track agreements.
- Get people to review the plan in pieces.
- Improve reviewability by minimizing unnecessary text in test plan documents.

### Goals

- Common understanding of the test process.
- Common commitment to the test process.
- Reasonable participation in the test process.
- Management has reasonable expectations about the test process.

### Status Check

- Is the project team paying attention to the test plan?
- Does the project team, especially first line management, understand the role of the test team?
- Does the project team feel that the test team has the best interests of the project at heart?
- Is there an adversarial or constructive relationship between the test team and the rest of the project?
- Does anyone feel that the testers are “off on a tangent” rather than focused on important testing?