**Common Sections of Test Strategy Document**

Step #1: Scope And Overview

Project overview along with information on who should use this document. Also, include details like who will review and approve this document. Define testing activities and phases to be carried out with timelines with respect to overall project timelines defined in the test plan.

Step #2: Test Approach

Define the testing process, level of testing, roles, and responsibilities of every team member.

For every [test type](https://www.softwaretestinghelp.com/category/types-of-testing/) defined in the Test plan (**For Example,** [Unit](https://www.softwaretestinghelp.com/unit-testing/), Integration, System, Regression, [Installation/Uninstallation](https://www.softwaretestinghelp.com/software-installationuninstallation-testing/), Usability, Load, Performance, and Security testing) describe why it should be conducted along with details like when to start, test owner, responsibilities, testing approach and details of automation strategy and tool if applicable.

In test execution, there are various activities like adding new defects, defect triage, defect assignments, re-testing, regression testing and finally test sign-off. You must define the exact steps to be followed for each activity. You can follow the same process that worked for you in your previous test cycles.

A Visio presentation of all these activities including a number of testers and who will work on what activities would be very helpful to quickly understand the roles and responsibilities of the team.

**For example,** defect management cycle – mention the process to log the new defect. Where to log in, how to log new defects, what should be the defect status, who should do defect triage, whom to assign defects after triage etc.

Also, define the change management process. This includes defining change request submissions, templates to be used, and processes to handle the request.

Step #3: Test Environment

The test environment setup should outline information about the number of environments and the required setup for each environment. **For example,** one test environment for the functional test team and another for the UAT team.

Define the number of users supported in each environment, access roles for each user, software and hardware requirements like operating system, memory, free disk space, number of systems, etc.

Defining test data requirements is equally important. Provide clear instructions on how to [create test data](https://www.softwaretestinghelp.com/tips-to-design-test-data-before-executing-your-test-cases/) (either generate data or use production data by masking fields for privacy).

Define test data backup and restore strategy. The test environment database may run into problems due to unhandled conditions in the code. I remember the problems we faced on one of the projects when there was no database backup strategy defined and we lost all the data due to code issues.

Backup and restore process should define who will take backups when to take a backup, what to include in backup when to restore the database, who will restore it and the data masking steps to be followed if the database is restored.

Step #4: Testing Tools

Define test management and automation tools required for test execution. For performance, load and security testing, describe the test approach and tools required. Mention whether it is an open source or commercial tool and how many users are supported on it and plan accordingly.

Step #5: Release Control

As mentioned in our [UAT article](https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/), unplanned release cycles can result in different software versions in test and UAT environments. The release management plan with proper version history will ensure test execution of all modifications in that release.

**For example,** set build management process which will answer – where new build should be made available, where it should be deployed, when to get the new build, from where to get the production build, who will give the go, the no-go signal for production release, etc.

Step #6: Risk Analysis

List all the risks that you envision. Provide a clear plan to mitigate these risks along with a contingency plan in case you see these risks in reality.

Step #7: Review And Approvals

When all these activities are defined in the test strategy 1plan, they need to be reviewed for sign-off by all entities involved in project management, business team, development team, and system administration (or environment management) team.

A summary of the review changes should be tracked at the beginning of the document along with the approver’s name, date and comment. Also, it’s a living document meaning this should be continuously reviewed and updated with testing process enhancements.

**Simple Tips to Write a Test Strategy Document**

1. Include product background in the test strategy document. Answer the first paragraph of your test strategy document – Why do stakeholders want to develop this project? This will help us understand and prioritize things quickly.
2. List all the important features you are going to test. If you think some features are not a part of this release then mention those features under “Features not to be tested” label.
3. Write down a test approach for your project. Clearly, mention what type of testing you are going to conduct?  
   i.e., Functional testing, UI testing, Integration testing, Load/Stress testing, Security testing, etc.
4. Answer questions like how you are going to perform functional testing? Manual or automation testing? Are you going to execute all the test cases from your test management tool?
5. Which bug tracking tool are you going to use? What will be the process when you find a new bug?
6. What are your test entry and exit criteria?
7. How will you track your testing progress? What metrics are you going to use for tracking test completion?
8. Task distribution – Define the roles and responsibilities of each team member.
9. What documents will you produce during and after the testing phase?
10. What risks do you see in Test completion?

**Conclusion**

Test Strategy is not a piece of paper. It’s the reflection of all QA activities in the software testing life cycle. Refer to this document from time to time during the test execution process and follow the plan till the software release.

When the project nears its release date, it’s fairly easy to cut down on testing activities by ignoring what you have defined in the test strategy document. However, it is advisable to discuss with your team whether or not cutting down on any particular activity will help for release without any potential risk of major issues post-release.

Most agile teams cut down on writing strategy documents as team focus is on test execution rather than documentation.

But having a basic test strategy plan always helps to clearly plan and mitigate risks involved in the project. Agile teams can capture and document all high-level activities to complete test execution on time without any issues.