

TEST PLAN EVOLUTION.

A test plan document will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from [test engineers](#)

The test plan document has evolved alongside the development of software engineering practices. Here's a brief history of its development:

Early Software Testing Practices (1950s - 1970s)

- In the early days of computing, testing was often informal and ad hoc.
- Test plans, if they existed at all, were typically informal notes or checklists created by individual testers.

Company: Early Computing Corporation

- Project: Development of Early Operating System
- In the early days of computing, testing was often informal and ad hoc. At Early Computing Corporation, testers created simple checklists to ensure basic functionality of the operating system being developed. These checklists, while informal, allowed testers to catch major bugs and ensure the system's basic operation.

Emergence of Structured Testing (1970s - 1980s)

- Test plans started to take a more formal shape, often detailing testing objectives, resources, schedules, and methodologies.
- The Waterfall model, popular during this time, formalized the process of creating test plans as part of a sequential software development process.

Company: Tech Solutions Inc.

- Project: Development of Accounting Software

- As structured testing emerged, Tech Solutions Inc. adopted a formal test plan for their accounting software project. The test plan detailed testing objectives, resources, schedules, and methodologies, ensuring a systematic approach to testing. By following the Waterfall model, the team completed each phase of testing before moving on to the next, resulting in well-tested and reliable accounting software.

Rise of Standards and Quality Assurance (1980s - 1990s)

- Software quality assurance (QA) became increasingly important, leading to the development of industry standards and methodologies like ISO 9000 and CMM (Capability Maturity Model).
- Test plans became more standardized and comprehensive, incorporating elements such as test cases, test procedures, test environments, and metrics for measuring testing effectiveness.

Company: Quality Software Solutions

- Project: Implementation of Enterprise Resource Planning (ERP) System
- Quality Software Solutions implemented the 829 Standard for Software Test Documentation for their ERP system project. The standardized test plan included test cases, procedures, environments, and metrics, ensuring comprehensive testing coverage. By adhering to industry standards like ISO 9000 and CMM, the team maintained high levels of quality throughout the development process, resulting in a robust and reliable ERP system.

Introduction of Agile and Iterative Development (2000s)

- The Agile Manifesto, introduced in 2001, shifted the focus from heavy documentation to working software and collaboration.
- Test plans adapted to the Agile methodology by becoming more lightweight and flexible. Instead of lengthy documents, Agile teams often rely on user stories, acceptance criteria, and test-driven development (TDD) for testing guidance.



Company: Agile Innovations Ltd.

Project: Development of Mobile Application

Agile Innovations Ltd. embraced Agile methodologies for their mobile application project. Instead of relying on traditional test plans, the team adopted lightweight, flexible Agile test plans. By focusing on user stories, acceptance criteria, and test-driven development (TDD), the team ensured that testing efforts were closely aligned with customer needs and delivered a high-quality mobile application on time and within budget.

Modern Test Planning (2010s - Present)

- Automation and continuous integration/continuous deployment (CI/CD) have influenced test planning, with more emphasis on automated testing frameworks and integration with development pipelines.

Company: Tech Solutions 2.0

- Project: Development of a Web-Based CRM Platform

- Tech Solutions 2.0 leveraged automation and continuous integration/continuous deployment (CI/CD) practices for their CRM platform project. Their test planning incorporated automated testing frameworks and integration with development pipelines. This approach enabled the team to achieve rapid feedback on code changes, ensuring that the CRM platform remained stable and reliable throughout its development and deployment phases.

IEEE 829 TEST PLAN STRUCTURE

[IEEE 829-2008](#), also known as the 829 Standard for Software Test Documentation, is an [IEEE](#) standard that specifies the form of a set of documents for use in defined stages of software testing, each stage potentially producing its own separate type of document.^[3] IEEE 829 is a standard for software testing documentation. *It was developed by the Institute of Electrical and Electronics Engineers (IEEE) and released in 1983.* This standard outlined various documents related to software testing, including the Test Plan document.

Project Plans and IEEE 829

The 16 clauses of the IEEE 829 test plan standard are:

1. Test plan identifier.
2. Introduction.
3. Test items.
4. Features to be tested.
5. Features not to be tested.
6. Approach.
7. Item pass/fail criteria.
8. Suspension criteria and resumption requirements.
9. Test deliverables.
10. Testing tasks.
11. Environmental needs.
12. Responsibilities.
13. Staffing and training needs.
14. Schedule.
15. Risks and contingencies.
16. Approvals.

The IEEE documents that suggest what should be contained in a test plan are:

- 829-2008 *IEEE Standard for Software and System Test Documentation*^[3]
 - 829-1998 *IEEE Standard for Software Test Documentation* (superseded by 829-2008)^[4]
 - 829-1983 *IEEE Standard for Software Test Documentation* (superseded by 829-1998)^[5]
- 1008-1987 *IEEE Standard for Software Unit Testing*^[6]
- 1012-2004 *IEEE Standard for Software Verification and Validation*^[7]
 - 1012-1998 *IEEE Standard for Software Verification and Validation* (superseded by 1012-2004)^[8]
 - 1012-1986 *IEEE Standard for Software Verification and Validation Plans* (superseded by 1012-1998)^[9]
- 1059-1993 *IEEE Guide for Software Verification & Validation Plans* (withdrawn)^[10]

References:

-
- https://en.wikipedia.org/wiki/Test_plan#cite_note-829-2008-3
 - <https://testsigma.com/blog/test-strategy-vs-test-plan/> AUTOMATION TESTING | Test Strategy vs Test Plan: Key Differences | August 1, 2023 by [Ankit Shrivastava](#)
 - Early testing in software testing: Building a Solid Foundation for Exceptional Software <https://medium.com/@softatdm1/what-is-early-testing-in-software-testing-c06b3092baf4#:~:text=Early%20testing%20refers%20to%20integrating,happens%20mainly%20at%20the%20end.>
 - CI/CD <https://about.gitlab.com/topics/ci-cd/>