



CERTIFICATION SYLLABUS

Certified Software Testing Foundations

Exam Code: CSTF™



A complete preparation guide covering exam objectives, core competencies, domain weightage, and topics to prepare.

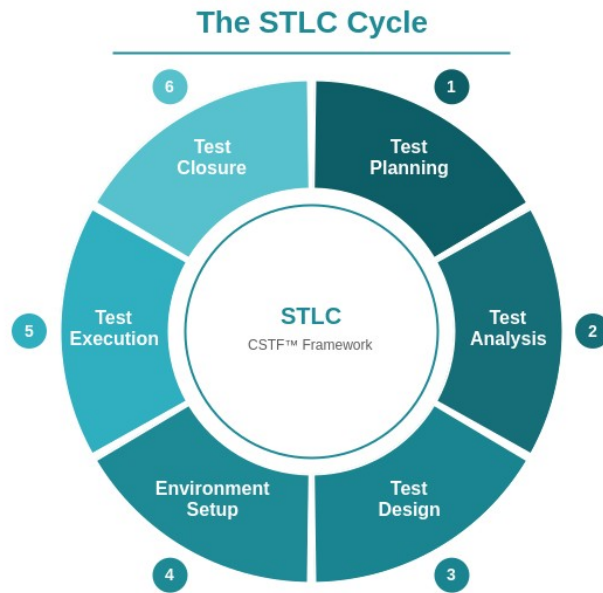
Table of Contents

1. Exam Name & Code.....	1
2. Table of Contents.....	2
3. Objective of the Certification.....	3
4. Core Competencies.....	4-6
➤ <i>Testing Fundamentals & SDLC Awareness</i>	
➤ <i>Software Testing Life Cycle (STLC) Execution</i>	
➤ <i>Test Design & Technique Application</i>	
➤ <i>Test Levels & Test Types</i>	
➤ <i>Defect Management</i>	
➤ <i>Static Testing & Early Defect Detection</i>	
➤ <i>Test Management Awareness</i>	
➤ <i>Tooling & Automation Awareness</i>	
5. Who Should Take This Certification.....	7
6. Exam Information.....	7
7. Examination Domain & Weightage.....	8
8. Topics to Prepare.....	9-10
➤ <i>Testing Fundamentals & SDLC Awareness</i>	
➤ <i>STLC Execution</i>	
➤ <i>Test Design & Technique Application</i>	
➤ <i>Test Levels & Test Types</i>	
➤ <i>Defect Management</i>	
➤ <i>Static Testing & Early Defect Detection</i>	
➤ <i>Test Management Awareness</i>	
➤ <i>Tooling & Automation Awareness</i>	

Objective of the Certification

- **Build a solid understanding of software testing fundamentals** — what testing is, why it matters, and its role within the software development lifecycle (SDLC).
- **Develop familiarity with the Software Testing Life Cycle (STLC)** — from test planning and analysis through test design, execution, and closure.
- **Introduce core test design techniques** — including black-box techniques (equivalence partitioning, boundary value analysis, decision tables) and basic white-box concepts (statement and branch coverage).
- **Establish competency in identifying test levels and test types** — unit, integration, system, and acceptance testing; functional vs. non-functional testing.
- **Build skills in defect management** — recognizing, documenting, prioritizing, and tracking defects through their lifecycle.
- **Introduce static testing techniques** — reviews, walkthroughs, and inspections as early defect-detection methods.
- **Familiarize candidates with basic test management concepts** — test planning, estimation, monitoring, and reporting.
- **Provide exposure to common testing tools and automation concepts** — at an awareness level, without requiring hands-on scripting expertise.
- **Prepare candidates to contribute effectively to QA teams** — equipping them with a shared vocabulary and foundational mindset for collaborating with developers, business analysts, and test leads.
- **Create a foundation for progression toward advanced or specialized testing certifications** — such as test automation, performance testing, or test management certifications.

Core Competencies — CSTF™ (Software Testing Foundations)



1. Testing Fundamentals & SDLC Awareness

- Explain what software testing is, why it's necessary, and how it fits within the Software Development Lifecycle (SDLC).
- Distinguish between verification and validation, and between testing and debugging.
- Understand common SDLC models (Waterfall, V-Model, Agile/Scrum) and how testing activities align with each.

2. Software Testing Life Cycle (STLC) Execution

- Follow the STLC phases: test planning, test analysis, test design, test environment setup, test execution, and test closure.
- Contribute to writing clear, traceable test cases and test scripts from requirements or user stories.
- Understand entry and exit criteria for each testing phase.

3. Test Design & Technique Application

- Apply black-box test design techniques: equivalence partitioning, boundary value analysis, decision tables, and state transition testing.
- Understand basic white-box concepts: statement coverage and branch/decision coverage.
- Select an appropriate test design technique based on the feature or requirement being tested.

4. Test Levels & Test Types

- Differentiate between unit, integration, system, and acceptance testing.
- Distinguish functional testing from non-functional testing (performance, usability, security, compatibility at an awareness level).
- Understand where each test level fits within a typical project timeline.

5. Defect Management

- Identify, document, and report defects clearly and reproducibly.
- Understand defect lifecycle states (new, assigned, open, fixed, retested, closed, reopened).
- Prioritize and classify defects by severity and priority.

6. Static Testing & Early Defect Detection

- Understand the value of static testing (reviews, walkthroughs, inspections) versus dynamic testing.
- Participate in review processes to catch defects before code execution.

7. Test Management Awareness

- Understand the basics of test planning, effort estimation, and progress monitoring.

- Interpret basic test reporting metrics (test case pass/fail rates, defect density, coverage).

8. Tooling & Automation Awareness

- Understand the role of test management tools (e.g., test case repositories, defect trackers) at a conceptual level.
- Understand basic automation concepts and when automation is/isn't appropriate — without requiring scripting proficiency.

Who Should Take This Certification

- Aspiring QA/Software Testing professionals with little to no prior formal testing experience.
- Fresh graduates or career-changers looking to enter the software testing/QA field.
- Manual testers seeking to formalize and validate their existing practical knowledge.
- Business Analysts, Developers, and Product Owners who want a working understanding of testing processes to collaborate more effectively with QA teams.
- Project Managers and Scrum Masters who oversee testing activities and need foundational fluency in QA terminology and processes.
- Support/Helpdesk professionals transitioning into a QA or testing role.
- Individuals preparing to pursue more advanced or specialized testing certifications (e.g., test automation, performance testing, test management).

Prerequisite: None (Entry-level certification — no prior testing experience required)

Exam Information

Exam Name	Software Testing Foundations
Exam Code	CSTF™
Exam Format	Multiple Choice Questions (MCQ)
Duration	90 minutes
Number of Questions	60
Passing Score	65%
Mode of Exam	Test Center Based Testing / Online Testing
Language	English
Validity	Lifetime
Open Book	No

Examination Domain & Weightage

Domain	Core Competencies Tested	Weightage
Testing Fundamentals & SDLC Awareness	Purpose of testing, verification vs. validation, testing vs. debugging, SDLC models (Waterfall, V-Model, Agile)	15%
STLC Execution	Test planning, analysis, design, environment setup, execution, closure; entry/exit criteria	15%
Test Design & Technique Application	Equivalence partitioning, boundary value analysis, decision tables, state transition testing, statement/branch coverage	20%
Test Levels & Test Types	Unit, integration, system, acceptance testing; functional vs. non-functional testing	15%
Defect Management	Defect identification, documentation, lifecycle states, severity vs. priority	15%
Static Testing & Early Defect Detection	Reviews, walkthroughs, inspections; static vs. dynamic testing	10%
Test Management Awareness	Test planning, estimation, progress monitoring, basic reporting metrics	5%
Tooling & Automation Awareness	Test management/defect tracking tools, automation concepts and suitability	5%
Total		100%

Topics to Prepare

The topics below represent the core areas candidates should focus on for each domain, reflecting the foundational, entry-level depth expected at the CSTF™ level.

1. Testing Fundamentals & SDLC Awareness

- **What is Software Testing:** Purpose, objectives, and importance of testing in software quality.
- **Verification vs. Validation:** Distinguishing “building the product right” from “building the right product.”
- **Testing vs. Debugging:** Understanding the distinct roles testers and developers play.
- **SDLC Models:** Waterfall, V-Model, and Agile/Scrum, and how testing activities integrate into each.
- **Cost of Defects:** Why finding defects earlier in the lifecycle reduces cost and risk.

2. STLC Execution

- **STLC Phases:** Test planning, test analysis, test design, test environment setup, test execution, and test closure.
- **Entry & Exit Criteria:** Defining readiness to start and complete each testing phase.
- **Test Case Writing:** Structuring clear, traceable test cases and test scripts from requirements or user stories.
- **Requirement Traceability:** Mapping test cases back to requirements to ensure coverage.

3. Test Design & Technique Application

- **Equivalence Partitioning:** Dividing input data into valid and invalid partitions to reduce redundant tests.
- **Boundary Value Analysis:** Testing at the edges of input ranges where defects commonly occur.
- **Decision Tables:** Modeling combinations of conditions and actions for complex business logic.
- **State Transition Testing:** Testing system behavior across different states and transitions.
- **White-Box Basics:** Statement coverage and branch/decision coverage concepts.

4. Test Levels & Test Types

- **Unit Testing:** Testing individual components or functions in isolation.
- **Integration Testing:** Testing interactions between integrated components or systems.
- **System Testing:** Testing the complete, integrated system against specified requirements.
- **Acceptance Testing:** Validating the system meets business needs (UAT).

- **Functional vs. Non-Functional Testing:** Functional correctness vs. performance, usability, security, and compatibility (awareness level).

5. Defect Management

- **Defect Identification & Reporting:** Writing clear, reproducible defect reports.
- **Defect Lifecycle:** New, assigned, open, fixed, retested, closed, and reopened states.
- **Severity vs. Priority:** Classifying defects by technical impact versus business urgency.
- **Defect Tracking Tools:** Conceptual understanding of how defect trackers support the lifecycle.

6. Static Testing & Early Defect Detection

- **Reviews, Walkthroughs & Inspections:** Differences and purposes of each static technique.
- **Static vs. Dynamic Testing:** Detecting defects without executing code vs. through execution.
- **Benefits of Early Detection:** Reducing downstream cost and rework.

7. Test Management Awareness

- **Test Planning & Estimation:** Basics of scoping test effort and timelines.
- **Progress Monitoring:** Tracking test execution status against plan.
- **Basic Reporting Metrics:** Test case pass/fail rates, defect density, and test coverage.

8. Tooling & Automation Awareness

- **Test Management Tools:** Conceptual role of test case repositories and defect trackers.
- **Automation Concepts:** When automation is appropriate vs. when manual testing is preferred, at an awareness level (no scripting required).