



**TECH
STARTER**

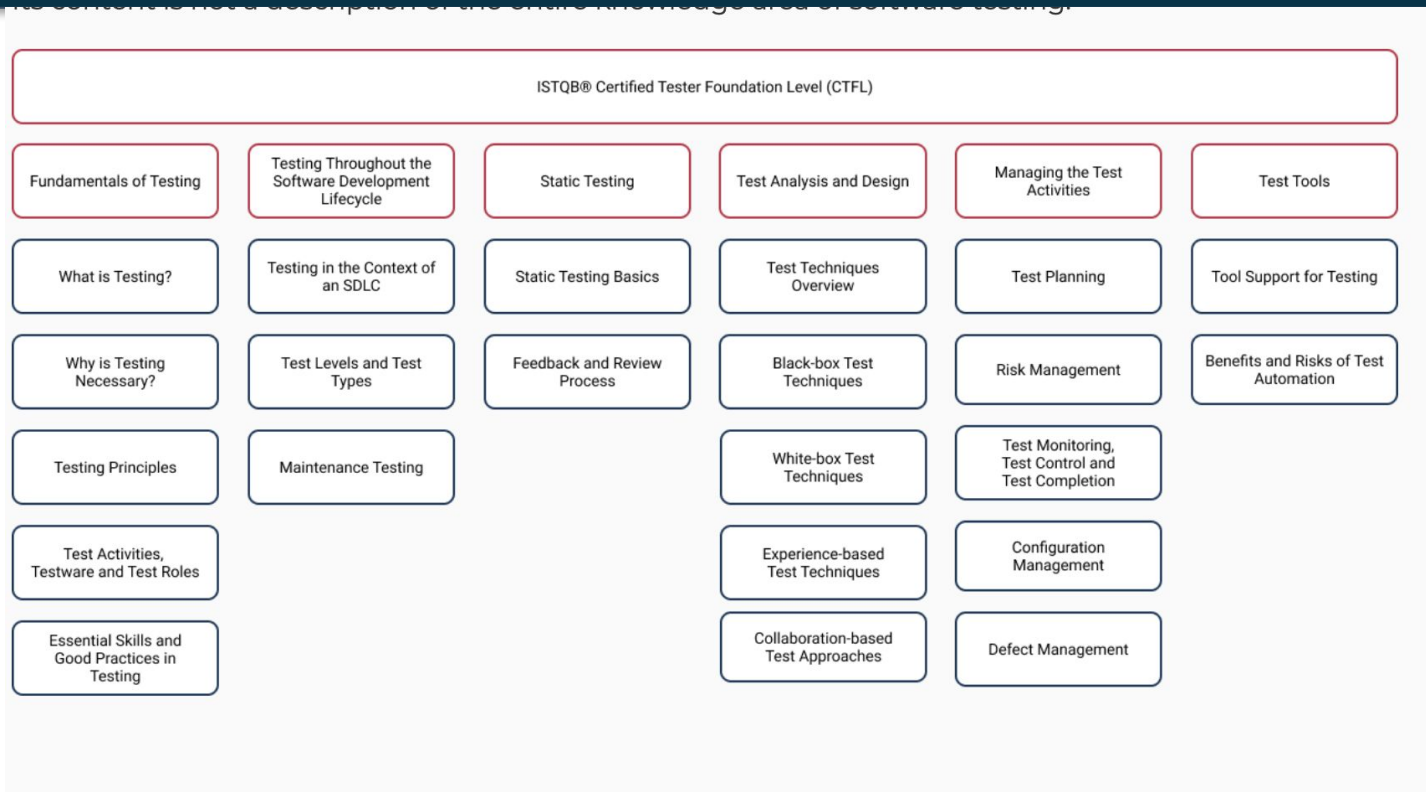
ISTQB -
Certified Tester
Foundation Level
v4.0

Fahrplan

- **Besprechung der HA**
- **Klausurinhalte**
- **Integration Tests in unserer Express App**



Inhalte



Inhalte nach Kapiteln

Ein Foundation Level zertifizierter Tester kann.

- FL-BO1 Verstehen, was Testen ist und warum es nützlich ist
- FL-BO2 Die grundlegenden Konzepte des Testens von Software verstehen
- FL-BO3 Den Testansatz und die anzuwendenden Aktivitäten in Abhängigkeit vom Kontext des Testens identifizieren
- FL-BO4 Die Qualität der Dokumentation bewerten und verbessern
- FL-BO5 Die Effektivität und Effizienz des Testens steigern
- FL-BO6 Den Testprozess an den Softwareentwicklungslebenszyklus anpassen
- FL-BO7 Grundsätze des Testmanagements verstehen
- FL-BO8 Klare und verständliche Fehlerberichte schreiben und kommunizieren
- FL-BO9 Die Faktoren, die die Prioritäten und den Aufwand für das Testen beeinflussen, verstehen
- FL-BO10 Als Teil eines funktionsübergreifenden Teams arbeiten
- FL-BO11 Risiken und Vorteile der Testautomatisierung kennen
- FL-BO12 Wesentliche Fähigkeiten, die für das Testen erforderlich sind, erkennen
- FL-BO13 Die Auswirkungen von Risiken auf das Testen verstehen
- FL-BO14 Über den Testfortschritt und die Qualität effektiv berichten

Inhalte nach Kapiteln

- Kapitel 1: Grundlagen des Testens (180 Minuten)
 - Der Lernende eignet sich die grundlegenden Prinzipien des Testens, die Gründe warum Testen notwendig ist und was Ziele des Testens sind, an.
 - Der Lernende versteht den Testprozess, die wichtigsten Testaktivitäten und Testmittel.
 - Der Lernende versteht die wesentlichen Fähigkeiten zum Testen.
- Kapitel 2: Testen während des Softwareentwicklungslebenszyklus (130 Minuten)
 - Der Lernende eignet sich an, wie das Testen in verschiedene Entwicklungsvorgehensweisen integriert wird.
 - Der Lernende eignet sich die Konzepte von Test-First-Ansätzen und DevOps kennen.
 - Der Lernende lernt die verschiedenen Teststufen, Testarten und den Wartungstest kennen.
- Kapitel 3: Statisches Testen (80 Minuten)
 - Der Lernende eignet sich die Grundlagen des statischen Testens, den Feedback- und den Reviewprozess an.

Inhalte nach Kapiteln

- Kapitel 4: Testanalyse und -entwurf (390 Minuten)
 - Der Lernende lernt Black-Box-, White-Box- und erfahrungsbasierte Testverfahren anzuwenden, um Testfälle aus verschiedenen Arbeitsergebnissen der Softwareentwicklung abzuleiten.
 - Der Lernende lernt den auf Zusammenarbeit basierenden Testansatz kennen.
- Kapitel 5: Management der Testaktivitäten (335 Minuten)
 - Der Lernende eignet sich an, wie man Tests im Allgemeinen plant und wie man den Testaufwand schätzt.
 - Der Lernende eignet sich an, wie Risiken den Umfang des Testens beeinflussen können.
 - Der Lernende eignet sich an, wie man Testaktivitäten überwacht und steuert.
 - Der Lernende eignet sich an, wie das Konfigurationsmanagement das Testen unterstützt.
 - Der Lernende eignet sich an, wie man Fehlerzustände klar und verständlich berichtet.
- Kapitel 6: Testwerkzeuge (20 Minuten)
 - Der Lernende lernt, Testwerkzeuge zu klassifizieren und die Risiken und Nutzen von Testautomatisierung zu verstehen.

Zu beantwortende Fragen nach Kapitel 1

1.1 Was ist Testen?

1.2 Warum ist Testen notwendig?

1.3 Grundsätze des Testens

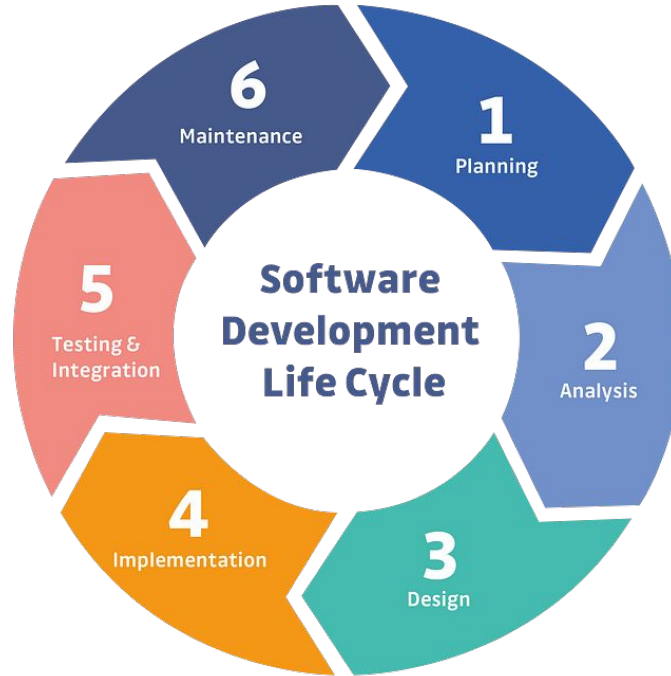
1.4 Testaktivitäten, Testmittel und Rollen des Testens

1.5 Grundlegende Kompetenzen und gute Praktiken beim Testen

1.1 Was ist Testen?

- Softwaretests bewerten die Qualität der Software und helfen, das Risiko einer Fehlerwirkung im Betrieb zu verringern.
- Bewertung der Qualität von Softwareartefakten. Werden diese Artefakte getestet, werden sie als Testobjekte bezeichnet
- muss auf den Softwareentwicklungslebenszyklus (Software Development Life Cycle, SDLC) abgestimmt sein

Software Development Lifecycle (SDLC)



Quelle: <https://medium.com/>

Dynamisch vs. statisches Testen

- Beim dynamischen Test wird die Software ausgeführt
- beim statischen Test hingegen nicht
- Zum statischen Test gehören Reviews und statische Analysen
- Beim dynamischen Test werden verschiedene Testverfahren und Testansätzen verwendet, um Testfälle abzuleiten

Testziele

- Evaluieren von Arbeitsergebnissen wie Anforderungen, User Storys, Entwürfe und Code
- Auslösen von Fehlerwirkungen und Finden von Fehlerzuständen
- Sicherstellen der erforderlichen Überdeckung eines Testobjekts
- Verringern des Risikos einer unzureichenden Softwarequalität
- Verifizieren, ob spezifizierte Anforderungen erfüllt wurden

Testziele

- Verifizieren, ob ein Testobjekt den vertraglichen, rechtlichen und regulatorischen Anforderungen entspricht
- Bereitstellen von Informationen für die Stakeholder, damit diese fundierten Entscheidungen treffen können
- Aufbauen von Vertrauen in die Qualität des Testobjekts
- Validieren, ob das Testobjekt vollständig ist und aus Sicht der Stakeholder wie erwartet funktioniert.

Testing vs. Debugging

Testing

trigger_error
trigger_error
trigger_error

VS

Debugging



Quelle: <https://www.interviewbit.com/>

Debugging Prozess

- Reproduzieren einer Fehlerwirkung
- Diagnose (Befund der Grundursache)
- Behebung der Ursache

→ Anschließend Fehlernachtests prüfen, ob das Problem durch die Korrekturen behoben wurde

Warum ist Testen notwendig?

→ Qualitätssteuerung, trägt dazu bei, die vereinbarten Ziele innerhalb des festgelegten Umfangs sowie der Zeit-, Qualitäts- und Budgetvorgaben zu erreichen

- kosteneffizientes Mittel zur Erkennung von Fehlerzuständen
- direkten Bewertung der Qualität eines Testobjekts in verschiedenen Phasen des SDLC
- Tester stellen sicher, dass ihr Verständnis für die Bedürfnisse der Benutzer während des gesamten Entwicklungszyklus berücksichtigt wird

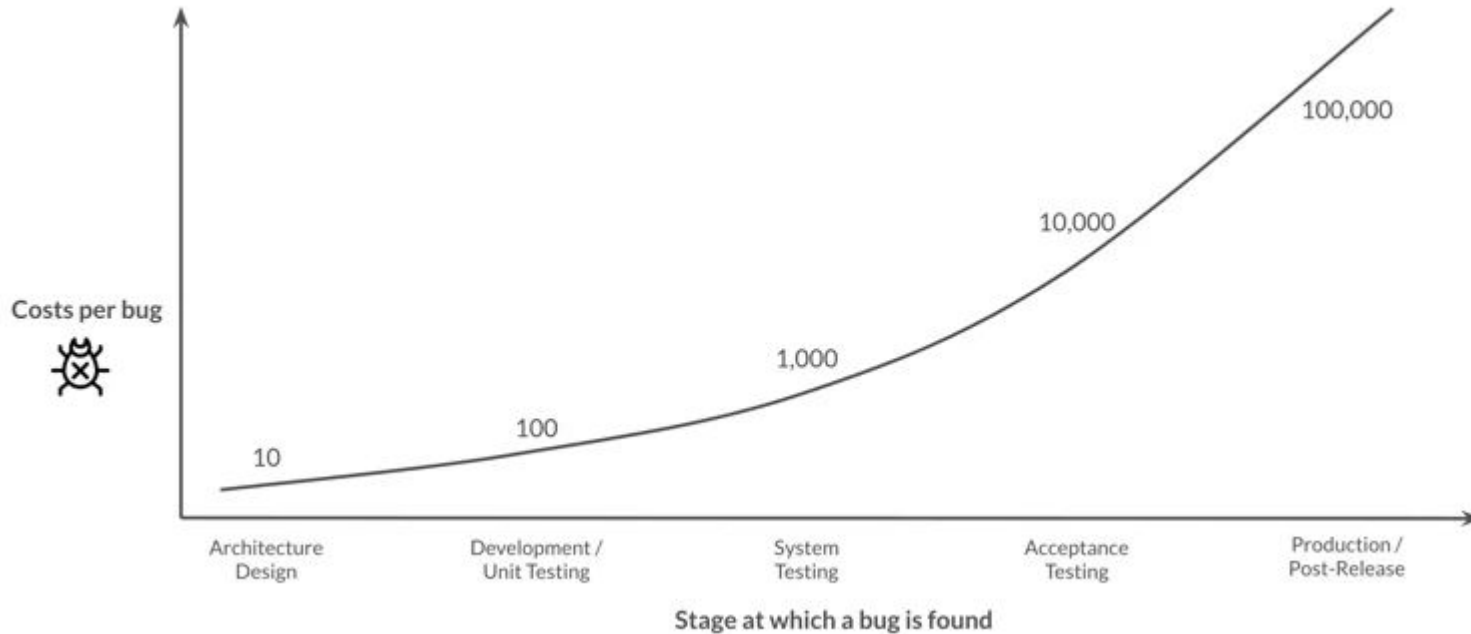
Warum ist Testen notwendig?

- Qualitätssicherung
- Fehlhandlungen (menschlich) → Fehlerzustände (Defekte)
- Vermeidung von technischer Schuld (Technical Debt)
- Einhaltung der Softwareentwicklungsstandards (Linter)

1.3 Grundsätze des Testens

- Testen zeigt das Vorhandensein, nicht die Abwesenheit von Fehlerzuständen
- Vollständiges Testen ist unmöglich
- Fehlerzustände treten gehäuft auf
- Frühes Testen spart Zeit und Geld (The Rule of Ten)

Rule of Ten



1.3 Grundsätze des Testens

- Tests nutzen sich ab
- Testen ist kontextabhängig
- Trugschluss: „Keine Fehler“ bedeutet ein brauchbares System

1.4.1 Testaktivitäten

- Die Testplanung besteht darin, die Testziele zu definieren und dann eine Testvorgehensweise auszuwählen
- Der Testentwurf umfasst die Ausarbeitung der Testbedingungen
- Testüberwachung und -steuerung
- Die Testanalyse umfasst die Analyse der Testbasis

1.4.1 Testaktivitäten

- Die Testrealisierung umfasst die Erstellung oder Beschaffung der für die Testdurchführung erforderlichen Testmittel
- Testdurchführung
- Der Testabschluss findet in der Regel zu Projektmeilensteinen statt (z. B. Freigabe, Ende der Iteration, Abschluss der Teststufe)

1.4.2 Testprozess im Kontext

- Stakeholder (Bedürfnisse, Erwartungen, Anforderungen, Bereitschaft zur Zusammenarbeit, usw.)
- Teammitglieder (Kompetenz, Wissen, Erfahrungsstand, Verfügbarkeit, Schulungsbedarf, usw.)
- Unternehmensbereich (Kritikalität des Testobjekts, identifizierte Risiken, Marktbedürfnisse, spezifische gesetzliche Vorschriften, usw.)
- Technische Faktoren (Art der Software, Produktarchitektur, verwendete Technologie, usw.)
- Projektbedingte Randbedingungen (Umfang, Zeit, Budget, Ressourcen, usw.)
- Organisatorische Faktoren (Organisationsstruktur, bestehende Richtlinien, angewandte Praktiken, usw.)
- Softwareentwicklungslebenszyklus (technologische Praktiken, Entwicklungsmethoden, usw.)
- Werkzeuge (Verfügbarkeit, Gebrauchstauglichkeit, Konformität, usw.)

1.4.3 Testmittel

- Zu den Arbeitsergebnissen der Testplanung gehören: Testkonzept, Testzeitplan
- Zu den Arbeitsergebnissen der Testüberwachung und -steuerung gehören: Testfortschrittsberichte, Dokumentation
- Zu den Arbeitsergebnissen der Testanalyse gehören: (priorisierte) Testbedingungen (z. B. Abnahmekriterien, siehe Abschnitt 4.5.2) und Fehlerberichte
- Zu den Arbeitsergebnissen des Testentwurfs gehören: (priorisierte) Testfälle, TestChartas, Überdeckungselemente,

1.4.3 Testmittel

- Zu den Arbeitsergebnissen der Testrealisierung gehören: Testabläufe, automatisierte Testskripte, Testsuiten, Testdaten, Testausführungspläne
- Zu den Arbeitsergebnissen der Testdurchführung gehören: Testprotokolle und Fehlerberichte
- Zu den Arbeitsergebnissen des Testabschlusses gehören: Testabschlussberichte, Maßnahmen zur Verbesserung nachfolgender Projekte oder Iterationen,

1.4.4 Verfolgbarkeit zwischen Testbasis und Mitteln

- Durch die Verfolgbarkeit von Testfällen zu Anforderungen kann überprüft werden, ob die Anforderungen durch Testfälle überdeckt werden.
- Durch die Verfolgbarkeit von Testergebnissen zu Risiken kann das Ausmaß des Restrisikos eines Testobjekts bewertet werden.

1.4.5 Rolle des Testmanagement vs. des Testens

- Die Rolle des Testmanagements übernimmt die Gesamtverantwortung für den Testprozess → konzentriert sich hauptsächlich auf die Aktivitäten der Testplanung
- Die Rolle des Testens übernimmt die Gesamtverantwortung für den operativen Aspekt des Testens → konzentriert sich hauptsächlich auf die Aktivitäten der Testanalyse, des Testentwurfs, der Testrealisierung und der Testdurchführung

1.5.1 Erforderliche Kompetenzen für das Testen

- Testwissen (zur Steigerung der Effektivität des Testens, z. B. durch den Einsatz von Testverfahren)
- Gründlichkeit, Sorgfalt, Neugier, Detailgenauigkeit, methodisches Vorgehen (um Fehlerzustände zu erkennen, insbesondere solche, die schwer zu finden sind)
- Gute Kommunikationsfähigkeit, aktives Zuhören, Teamfähigkeit (um mit allen Stakeholdern effektiv zu interagieren, Informationen an andere weiterzugeben, verstanden zu werden und Fehlerzustände zu berichten und zu diskutieren)

1.5.1 Erforderliche Kompetenzen für das Testen

- Analytisches Denken, kritisches Denken, Kreativität (zur Steigerung der Effektivität des Testens)
- Technische Kenntnisse (um die Effizienz des Testens zu steigern, z. B. durch den Einsatz geeigneter Testwerkzeuge)
- Wissen in der Anwendungsdomäne (um Endanwender/Fachbereichsvertreter verstehen und mit ihnen kommunizieren zu können)

1.5.2 Whole-Team-Ansatz

- Beim Whole-Team-Ansatz kann jedes Teammitglied, das über die erforderlichen Kompetenzen verfügt, jede Aufgabe ausführen
- verbessert die Teamdynamik, fördert die Kommunikation und Zusammenarbeit innerhalb des Teams und schafft Synergien

1.5.3 Unabhängigkeit des Testens

- Arbeitsergebnisse können von ihrem Autor (keine Unabhängigkeit)
- von den Kollegen des Autors aus demselben Team (etwas Unabhängigkeit)
- von Testern außerhalb des Teams des Autors, aber innerhalb der Organisation (hohe Unabhängigkeit)

→ Jede Unabhängigkeitsstufe hat Vor- und Nachteile