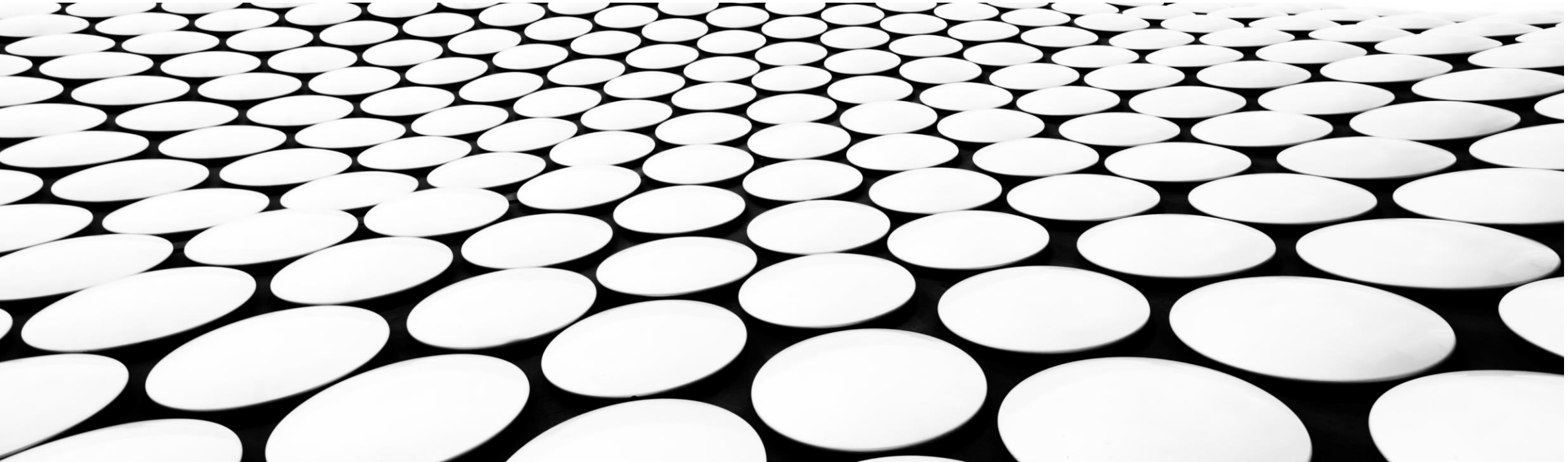

SOFTWARE TESTEN

BLACKBOX, WHITEBOX, UNIT ... KONZEPTE ERSTELLEN UND TESTEN





TESTTECHNIKEN

- Bei den Testtechniken ist grundsätzlich zwischen statischen Testtechniken und dynamischen Testtechniken zu unterscheiden.
- Der dynamische Test beinhaltet die Ausführung des Testelements.
- Der statische Test dagegen testet Arbeitsergebnisse ohne Ausführung von Code.



STATISCHE TESTTECHNIKEN





STATISCHE TESTTECHNIKEN

Statische Testtechniken dienen hauptsächlich dem Prüfen von Arbeitsergebnissen, wie Anforderungen oder Quelltext, ohne diese auf einem Rechner auszuführen. Bei den Prüfobjekten handelt es sich meist um Dokumente oder Quellcode. Prinzipiell kann aber jedes Arbeitsergebnis der Softwareentwicklung durch statische Testtechniken geprüft werden, zum Beispiel Anforderungsspezifikationen, User-Stories, Designspezifikationen, Quellcode, Testkonzepte, Testspezifikationen, Testfälle, Testskripte, Anwenderhandbücher oder Webseiten.



STATISCHE TESTTECHNIKEN

Reviews oder werkzeuggestützte statische Analysen zählen zu den gängigsten statischen Testtechniken.

Statische Analyse bezeichnet einen Prozess der Bewertung eines Testobjekts (Komponente oder System) basierend auf seiner Form, seiner Struktur, seines Inhalts oder seiner Dokumentation, ohne es auszuführen.

VORTEILE STATISCHER TESTTECHNIKEN

- Frühe Fehlererkennung bereits in der Spezifikation
- Fehlerzustände werden direkt aufgedeckt
- Verringert die Anzahl aufwändiger und teurer dynamischer Testtechniken
- Prüft auch auf Programmier-Konventionen, Standards, Design
- Auch nicht-ausgeführte Softwarebestandteile (z.B. Variablendefinitionen) unterliegen der Prüfung
- Prüfung erfasst auch Attribute wie Wartbarkeit
- Prüfung von Dokumenten (z.B. Anforderungsspezifikationen, etc.)
- Wissensaustausch zwischen Teammitgliedern (Review)
- Zwang zu klarer Darstellung von Sachverhalten (Review)
- Team als Ganzes trägt Qualitätsverantwortung (Review)



NACHTEILE STATISCHER TESTTECHNIKEN

- Eher ungeeignet für komplexe Interaktionen zwischen verschiedenen Systemteilen
- Einige Fehlerzustände entstehen erst bei Ausführung. Ein Divisor kann beispielsweise bei der Ausführung den Wert „0“ annehmen und somit einen Laufzeitfehler verursachen. Dieser kann durch reines Sichten des Programmtextes nicht ermittelt werden

NACHTEILE STATISCHER TESTTECHNIKEN

- Eher ungeeignet für komplexe Interaktionen zwischen verschiedenen Systemteilen
- Einige Fehlerzustände entstehen erst bei Ausführung. Ein Divisor kann beispielsweise bei der Ausführung den Wert „0“ annehmen und somit einen Laufzeitfehler verursachen. Dieser kann durch reines Sichten des Programmtextes nicht ermittelt werden



DYNAMISCHE TESTTECHNIKEN





DYNAMISCHE TESTTECHNIKEN

- Dynamische Testtechniken dienen dem Auffinden von Fehlerwirkungen durch Ausführen des Testobjekts (Programm) auf einem Rechner.



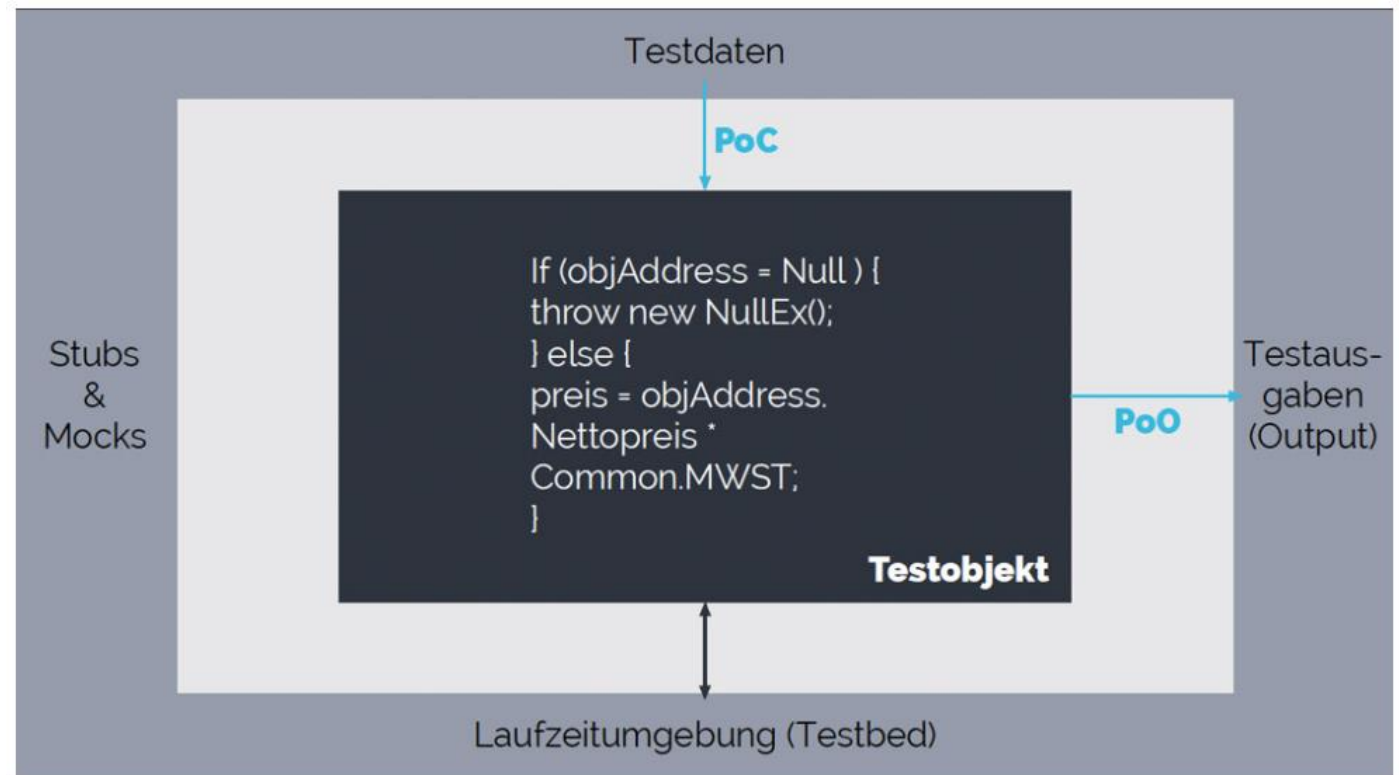
BLACK-BOX-TESTVERFAHREN

Black-Box-Testverfahren sind spezifikationsorientierte Testverfahren. Bei einem Black-Box-Test werden die Testfälle ausschließlich aus der Spezifikation des zu testenden Objekts abgeleitet, ohne dabei dessen innere Struktur, also Architektur und Code, zu berücksichtigen (- diese werden als „Black Box“ behandelt). Es wird also nur das von außen sichtbare Verhalten des Testobjektes beobachtet.

Black-Box-Testverfahren können in allen Teststufen zum Einsatz kommen. Beim Komponententest und Integrationstest ist hierfür jedoch ein Testrahmen (Test Bed) mit Stubs und Treibern notwendig, um ein lauffähiges Testobjekt zur Verfügung zu haben.

BLACK-BOX-TESTVERFAHREN

Beobachtet wird das Verhalten des Testobjektes von außen. Der Point of Observation (PoO) liegt außerhalb des Testobjektes. Es ist keine Steuerung des Ablaufs des Testobjekts außer durch die entsprechende Wahl der Eingabetestdaten möglich. Auch der Point of Control (PoC) liegt außerhalb des Testobjekts.



Ersteller: Lisa Wendland; Quelle: Software Testing Academy



BLACK-BOX-TESTVERFAHREN

WELCHE ZIELE KÖNNEN MIT BLACK-BOX-TESTS ERREICHT WERDEN?

Generell lässt sich die Korrektheit einer Software nicht mit Hilfe eines Tests nachweisen. Anstatt die Abwesenheit von Fehlern nachzuweisen, belegen Tests die mögliche Anwesenheit von Fehlern.

Im speziellen Fall der Black-Box-Tests handelt es sich um Verfahren, mit deren Hilfe Testfälle mit hoher Fehlerwahrscheinlichkeit spezifiziert werden können. Dadurch werden Lücken in der Implementierung aufgedeckt und daneben gleichzeitig auch die Spezifikation auf Vollständigkeit und Verständlichkeit geprüft.

BLACK-BOX-TESTVERFAHREN

WELCHE NACHTEILE HABEN BLACK-BOX-TEST?

Aus einer informellen Spezifikation Testfälle abzuleiten kann aufwändig und je nach Struktur der Spezifikation auch sehr unübersichtlich werden. Zudem werden durch reine Black-Box-Testverfahren nur die Teile des Systems getestet, welche ein Äquivalent in der Spezifikation haben. Wurden während der Implementierung noch weitere Features programmiert – sei es zum Nutzen („goldene Henkel“) oder zum Schaden des Kunden (z.B. Schadsoftware) – werden diese durch Black-Box-Tests höchstwahrscheinlich nicht gefunden.

In anfänglichen Phasen des Softwareentwurfs erstellte Spezifikationen decken spätere Detailentscheidungen und Entscheidungen hinsichtlich der Implementation nicht ab. Folglich ist ein erfolgreicher Black-Box-Test keine Garantie für die Fehlerfreiheit der Software.

BLACK-BOX-TESTVERFAHREN

BLACK-BOX-TEST – WELCHE TESTVERFAHREN GIBT ES?

Es gibt viele verschiedene Black-Box-Testverfahren, im folgenden werden die fünf verbreitetsten Verfahren aufgezählt:

- Äquivalenzklassenbildung
- Grenzwertanalyse
- Entscheidungstabellen
- Zustandsübergangstest
- Anwendungsfallbasierte Verfahren

WHITE-BOX-TESTVERFAHREN

White-Box-Testverfahren sind in der Qualitätssicherung von Software unverzichtbar. White-Box-Testverfahren werden auch als strukturbasierte oder strukturelle Testverfahren bezeichnet. Im Unterschied zu den Black-Box-Verfahren wird bei den White-Box-Verfahren die interne Struktur des Testobjekts als Basis verwendet – daher die Namensgebung. Die Ableitung der Testbedingungen, Testfälle oder Testdaten erfolgt basierend auf der Analyse der Struktur des zu testenden Systems.

Zu den wichtigsten White-Box-Testverfahren zählen:

- Anweisungsüberdeckung
- Entscheidungsüberdeckung
- Pfadüberdeckung



WHITE-BOX-TESTVERFAHREN

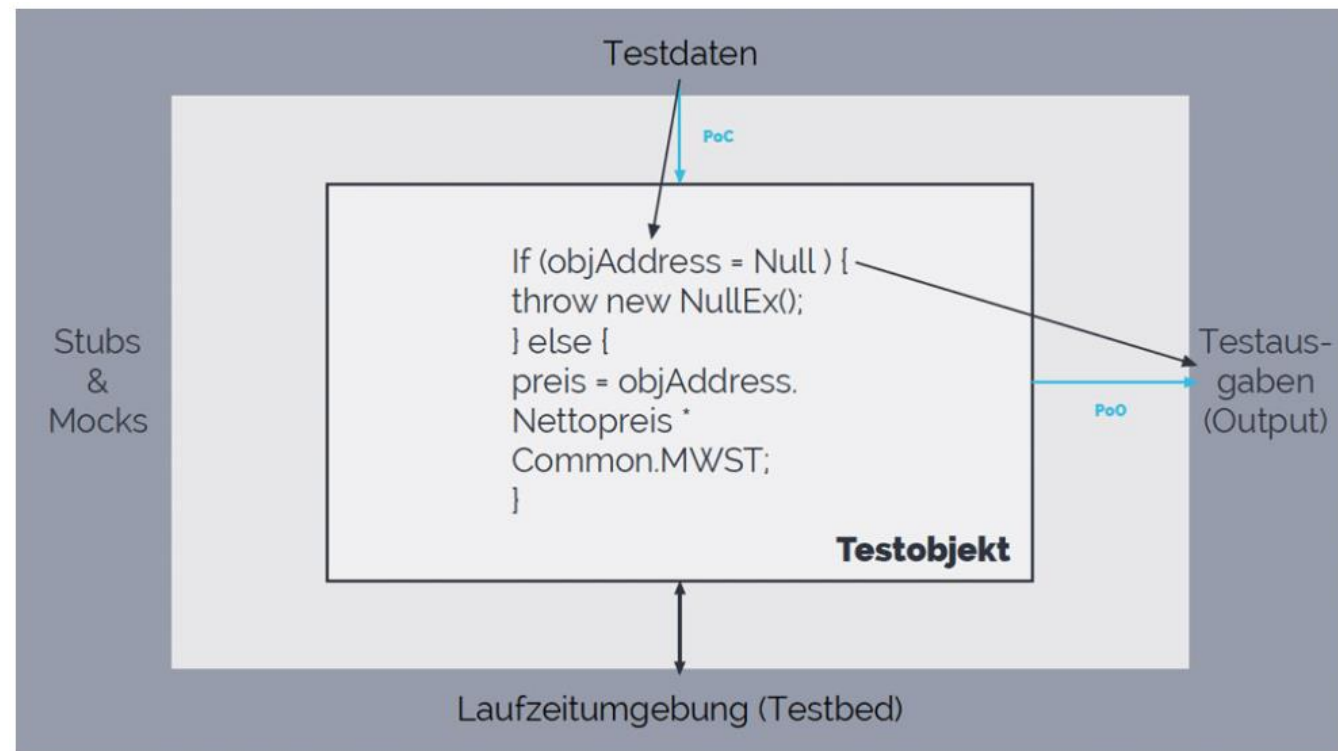
Im Gegensatz zu Black-Box-Tests wird bei White-Box-Tests auch das Innenleben des Testobjektes, d.h. der Code, berücksichtigt. Idee hinter der Namensgebung ist also das Hineinsehen in die Box. Genaugenommen müssten White-Box-Tests damit als Glass-Box-Tests bezeichnet werden. Tatsächlich hat sich in der Praxis jedoch „White-Box“ als Bezeichnung durchgesetzt.

WHITE-BOX-TESTVERFAHREN

Strukturbasierte Tests bauen auf der tatsächlichen Struktur des Testobjekts auf. Bei der Entwicklung der Testfälle kommen dabei strukturbasierte Testverfahren (auch genannt “White-Box-Testverfahren”) zum Einsatz. Die Entwicklung von Testfällen auf Basis der inneren Struktur kann auf verschiedenen Ebenen ganz unterschiedlich aussehen, zum Beispiel:

- Komponentenebene: Als Basis dient die Struktur der Softwarekomponente, d.h. Anweisungen, Entscheidungen, Zweige oder sogar einzelne Pfade
- Integrationsebene: Die Struktur kann ein Aufrufgraph sein (ein Diagramm, das zeigt, welche Module andere Module aufrufen)
- Systemebene: Als zugrundeliegende Struktur können die Menüstruktur, Ablaufdiagramme zu einzelnen Use-Cases, Geschäftsprozesse oder die Struktur einer Webseite dienen

WHITE-BOX-TESTVERFAHREN





WHITE-BOX-TESTVERFAHREN

ANWENDUNG VON WHITE-BOX-TESTS

White-Box-Testverfahren finden vorwiegend, aber nicht ausschließlich, im Komponententest Anwendung. Der jeweilige Überdeckungsgrad kann unter Zuhilfenahme von Werkzeugen gemessen werden. Während der Ausführung der Testfälle wird der innere Ablauf im Testobjekt analysiert. Der Point of Observation (PoO) liegt in diesem Fall also innerhalb des Testobjektes.

Auch Eingriffe in den Ablauf des Testobjekts sind in Ausnahmefällen möglich, z.B. wenn Negativtests durchzuführen sind und über die Komponentenschnittstelle die zu provozierende Fehlbedienung nicht ausgelöst werden kann. Der Point of Control (PoC) kann also bei White-Box-Tests auch innerhalb des Testobjekts liegen.

ERFAHRUNGSBASIERTE TESTVERFAHREN

Bei erfahrungsbasierten Testverfahren geht es weniger um eine systematische Ableitung von Testfällen, sondern vielmehr darum, das Können, die Intuition und die Erfahrung der Entwickler, Tester und Anwender wirksam einzusetzen. Zum Beispiel:

- Die Erfahrung mit ähnlichen Applikationen und Technologien
- Fehlersammlungen/Fehlermodelle zu den verwendeten Technologien bzw. Programmiersprachen
- Kenntnis von Vorgängersystemen und deren Schwächen
- Kenntnis über die Rahmenbedingungen der Softwareentwicklung z.B. Komponenten, die unter starkem Zeitdruck erstellt werden

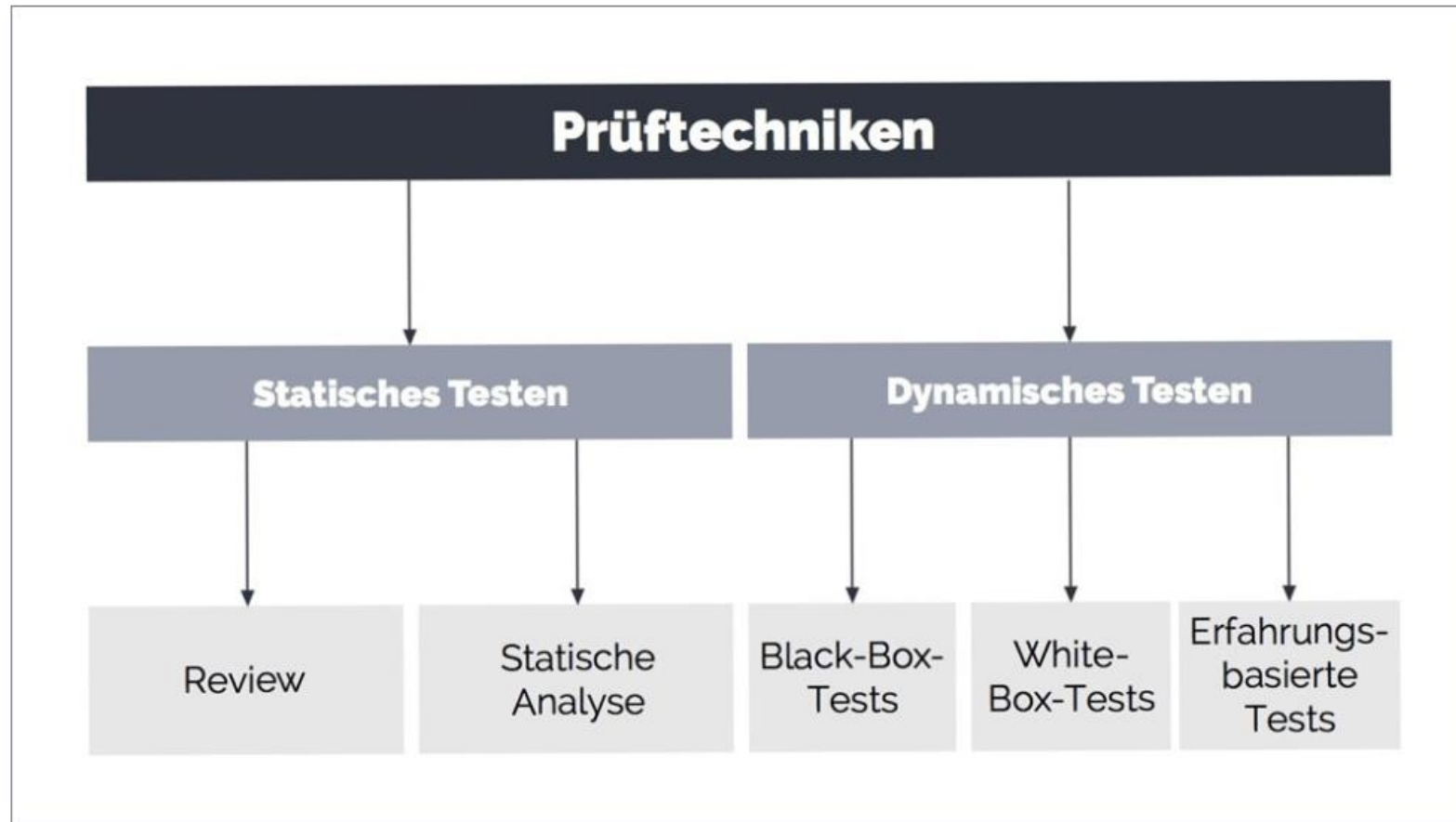
ERFAHRUNGSBASIERTE TESTVERFAHREN

Da die Wirksamkeit erfahrungsbasierter Testverfahren stark mit der Erfahrung des Testers variiert ist der Einsatz dieser Testverfahren insbesondere als Ergänzung zu systematischen Testverfahren sinnvoll.

Im ISTQB Certified Tester Foundation Level wird auf folgende erfahrungsbasierte Testverfahren näher eingegangen:

- **Intuitive Testfallermittlung**
Häufig können Tester aufgrund ihrer Erfahrung bestimmte Fehler erahnen. Bei der intuitiven Testfallermittlung erstellen die Tester im ersten Schritt eine Liste intuitiv möglicher Fehlerzustände. Darauf basierend werden dann Testfälle entworfen, die auf diese Fehlerzustände abzielen.
- **Exploratives Testen**
Beim explorativen Testen startet der Tester fast ohne vorherige Vorbereitung mit der Testdurchführung. Getestet wird dann typischerweise in einer auf 1-2 Stunden begrenzten Testsitzung, in der Unterbrechungen so wie möglich vermieden werden.
- **Checklistenbasiertes Testen**
Checklistenbasiertes Testen ist ein Testverfahren, bei dem entweder eine Liste von Kontrollpunkten genutzt wird, welche beachtet, überprüft oder beim Testen in Erinnerung gerufen werden müssen, oder eine Menge von Regeln und Kriterien, gegen welche das zu testende Produkt verifiziert werden muss.

ERFAHRUNGSBASIERTE TESTVERFAHREN





TESTVERFAHREN

VORTEILE DYNAMISCHER TESTTECHNIKEN:

- Testobjekt wird tatsächlich ausgeführt, was der Bestimmung von Software näher kommt
- Das Zusammenspiel von Systemteilen wird getestet
- Umfasst auch die Durchführung wichtiger nicht-funktionaler Tests (Performanztest, Lasttest, etc.)

NACHTEILE DYNAMISCHER TESTTECHNIKEN:

- Setzen lauffähiges Testobjekt und eine entsprechende Testumgebung voraus
- Nur Programmteile, die zur Ausführung gelangen, werden getestet
- Je nach Umfang und Anzahl verknüpfter Teilsysteme sind dynamische Testtechniken mit hohem Aufwand verbunden
- Es werden zunächst nur Fehlerwirkungen aufgedeckt. Die zugrundeliegenden Fehlerzustände müssen in einem separaten Arbeitsgang (Debugging) lokalisiert werden



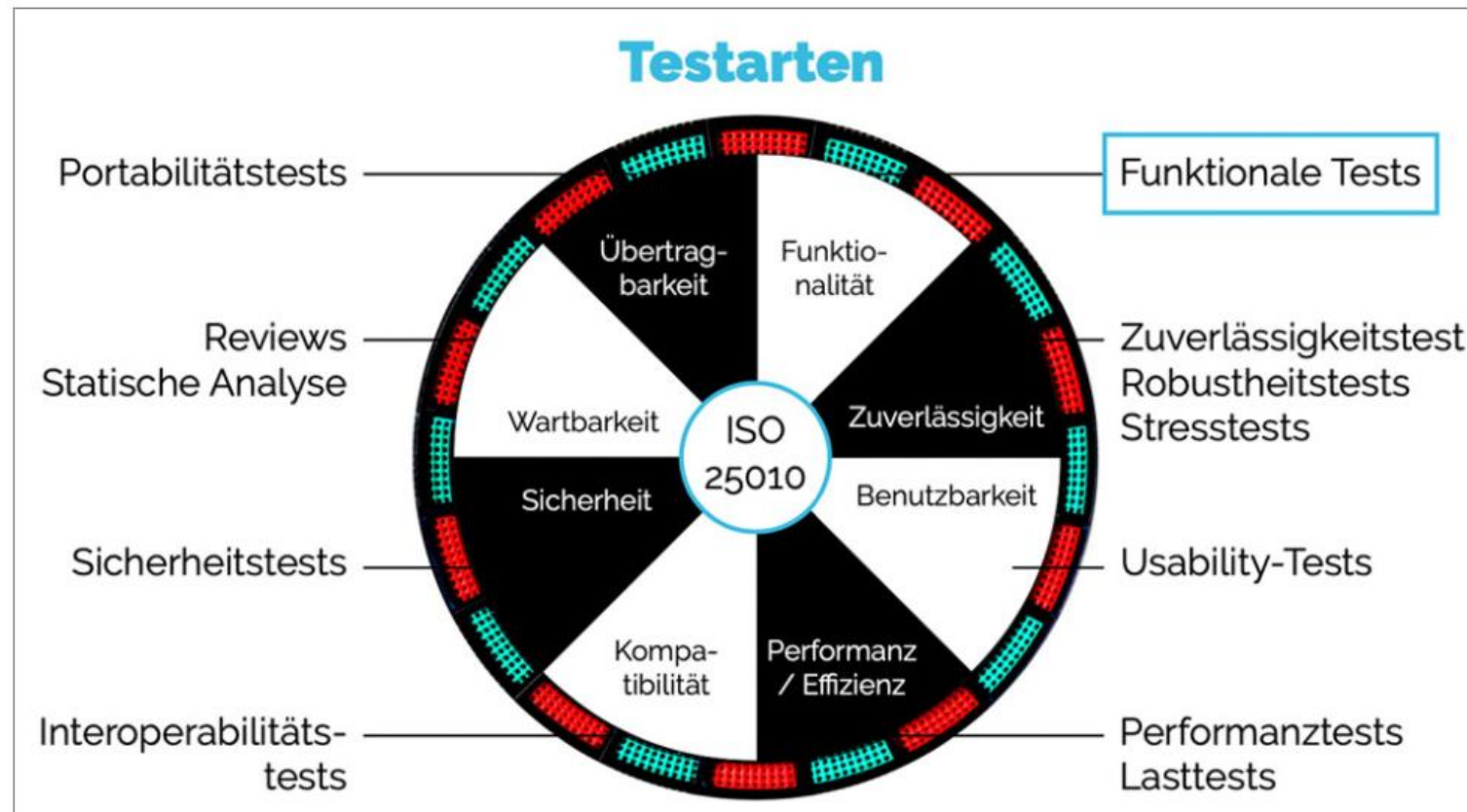
TESTARTEN

Neben den oben genannten Testtechniken können Tests noch bezüglich Ihres Zieles differenziert werden. Nach dieser Unterscheidung ergeben sich verschiedene Testarten. Dahingehend muss im Testmanagement differenziert werden, welche funktionalen und nicht-funktionalen Qualitätsmerkmale im Fokus stehen.

FUNKTIONALE TESTS

Funktionale Tests zielen auf die Überprüfung der reinen Funktionalität des Testobjekts ab. Die zugrundeliegende Frage könnte daher lauten: „Macht das System das, was von ihm erwartet wird?“

TESTARTEN



TESTARTEN

- NICHT-FUNKTIONALE TESTS

Nicht-funktionale Testarten zielen auf die Überprüfung weiterer allgemeiner Qualitätsmerkmale von Software, außer der reinen Funktionalität. Diese sind entlang der Qualitätsmerkmale in der DIN-ISO 25010 definiert:

- PERFORMANZ / EFFIZIENZ

Wird durch Performanztests (betreffend Antwortzeiten und Verarbeitungsgeschwindigkeit) und Lasttests (betreffend Systemlast z.B. in Form von parallel zugreifender Benutzer, Anzahl Transaktionen oder Verarbeitungsvolumen) geprüft.

TESTARTEN

- ZUVERLÄSSIGKEIT

Zuverlässigkeitstests prüfen, ob das Testobjekt ein bestimmtes Leistungsniveau unter bestimmten Bedingungen über einen bestimmten Zeitraum aufrechterhalten kann. Hierzu gehören Robustheitstests, sowie Tests, die prüfen, wie schnell das System im Fehlerfall wieder einsatzfähig ist („Recovery“-Tests). Auch Stresstests fallen in diesen Bereich: Sie beobachten das Systemverhalten bei Überlastung.

- BENUTZBARKEIT / GEBRAUCHSTAUGLICHKEIT

Benutzbarkeitstests (auch: Gebrauchstauglichkeits-Tests oder Usability-Tests) prüfen typischerweise, wie gut und wie einfach ein System für die Benutzer zu bedienen ist.



TESTARTEN

- BENUTZBARKEIT / GEBRAUCHSTAUGLICHKEIT

Benutzbarkeitstests (auch: Gebrauchstauglichkeits-Tests oder Usability-Tests) prüfen typischerweise, wie gut und wie einfach ein System für die Benutzer zu bedienen ist.

- SICHERHEITSTESTS

Diese prüfen, ob das System und die damit zusammenhängenden Daten ausreichend vor unerlaubten Zugriffen und externen Bedrohungen, wie z.B. Viren, geschützt sind.

TESTARTEN

- KOMPATIBILITÄT

Interoperabilitätstests bewerten die Fähigkeit des Softwareprodukts mit ein oder mehreren spezifizierten Komponenten oder Systemen zu interagieren.

- WARTBARKEIT

Wird typischerweise durch Reviews und werkzeuggestützte statische Analyse geprüft.

- ÜBERTRAGBARKEIT

Wird durch Portabilitätstests geprüft. Fokus dabei ist, wie leicht sich die Software in eine andere Umgebung übertragen lässt. Ein inzwischen weit verbreitetes Beispiel für Portabilitätstests sind Tests von Applikationen auf mobilen Endgeräten. Hierbei spielt die Übertragbarkeit auf verschiedene Hardware (z.B. mobile Geräte) ebenso eine Rolle wie die Übertragbarkeit in verschiedene Software-Umgebungen (z.B. Windows oder Apple iOS).

TESTARTEN

STRUKTURELLE UND ÄNDERUNGSBEZOGENE TESTS

Nach ISTQB gibt es neben den funktionalen und nicht-funktionalen Tests noch zwei weitere Testarten: nämlich strukturelle und änderungsbezogene Tests.

Testarten

Funktionales Testen

Nicht-funktionales Testen

White-Box-Tests

Änderungsbezogenes Testen



TESTARTEN

Strukturelle Tests – auch: White-Box-Tests – können in allen Teststufen angewandt werden. Strukturelle Tests setzen immer auch auf Modellen der Software auf. Ein großes Plus der strukturellen Tests ist, dass es bei ihrer Anwendung oft einfach möglich ist, die Testintensität messbar zu machen – als sogenannte Testüberdeckung.

Als änderungsbezogenes Testen wird das Testen bezeichnet, das durch Änderung einer Komponente oder eines Systems veranlasst wird. Konkret gibt es hierbei nochmals zwei Testarten zu unterscheiden: Den Fehlernachtest, was den Test bezeichnet, der die erfolgreiche Beseitigung eines Fehlers validiert sowie den Regressionstest, durch den geprüft wird, ob durch eine Änderung noch zusätzliche, ungewollte Änderungen oder Seiteneffekte entstanden sind.



TESTMANAGEMENT





TESTMANAGEMENT

Die Testplanung ist eine wesentliche Aufgabe des Testmanagements und ist, zumindest in leitender Funktion, vom Testmanager wahrzunehmen. Die Testplanung ist eine sehr frühe Aktivität innerhalb des Testprozesses mit dem Ziel der Erstellung oder Fortschreibung des Testkonzepts für das Softwareprojekt. Die Testplanung sollte in enger Abstimmung mit der Softwareentwicklung und den Mitarbeitern des Testteams erfolgen. Die Testplanung erfolgt dabei auf unterschiedlichen Abstraktionsebenen.

TESTMANAGEMENT LEVEL

■ HIGH-LEVEL

Auf dieser höchsten Ebene der Testplanung ist zu klären, wie viel Testaufwand für das aktuelle Projekt überhaupt angemessen ist. Für jedes Testobjekt ist die Intensität der durchzuführenden Tests und die Auswahl der zu überprüfenden Qualitätsmerkmale individuell festzulegen. (Beispiel: Brauchen wir einen Performancetest? Wenn ja, für welche Komponenten?).

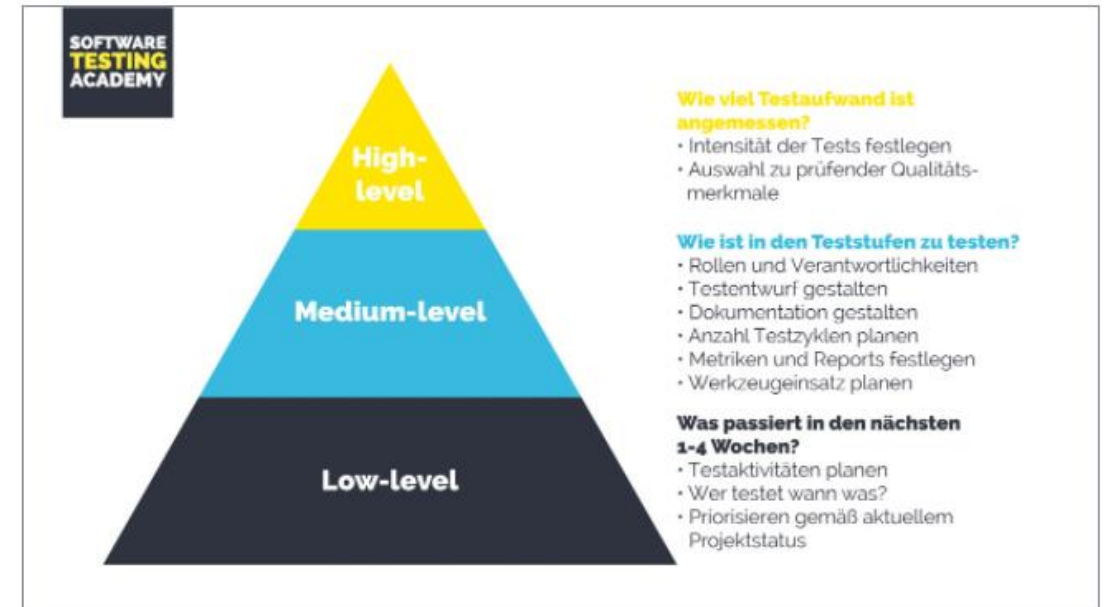
■ MEDIUM-LEVEL

Hier geht es hauptsächlich um die Frage, wie in den einzelnen Teststufen zu testen ist und wer die jeweiligen Tests durchführt. Konkret ist zu klären, wie der Testentwurf zu gestalten ist, wie detailliert die Dokumentationsführung sein soll, wie viele Testzyklen vorzusehen sind, welche Metriken und Reports für die Teststeuerung benötigt werden, welche unterstützenden Tools einzusetzen sind, etc.

TESTMANAGEMENT LEVEL

LOW-LEVEL

Auf der untersten Ebene erfolgt eine detaillierte Planung, was in den nächsten 1-4 Wochen zu tun ist. Insbesondere werden Testaktivitäten gemäß des aktuellen Projektstatus priorisiert. Die Testplanungen im High und Medium Level sollten möglichst früh im Projektzeitplan erfolgen. Die Planungsergebnisse werden mit jedem neuen Zyklus entsprechend dem aktuellen Entwicklungsstand angepasst. Darauf basierend erfolgt auch die Aufwandsschätzung.



AKTIVITÄTEN DER TESTPLANUNG

Im Testmanagement spielt die korrekte Testplanung eine entscheidende Rolle für den späteren Erfolg des Tests. Sie umfasst eine Reihe von Aktivitäten, deren Abarbeitung vom Testmanager zu kontrollieren ist:

- Identifizierung und Kategorisierung der Risikobereiche der zu testenden Software
- Definition der Testziele und der Abnahmekriterien
- Entwicklung einer Teststrategie und einer passenden Testvorgehensweise
- Erstellung eines Zeitplans für die Testaktivitäten
- Festlegung der Testobjekte, des Testfokus (Qualitätsmerkmale), des Testumfangs und der Testintensität
- Festlegung, was explizit nicht zu testen ist
- Integration der Testaktivitäten in den Softwarelebenszyklus

AKTIVITÄTEN DER TESTPLANUNG

Im Testmanagement spielt die korrekte Testplanung eine entscheidende Rolle für den späteren Erfolg des Tests. Sie umfasst eine Reihe von Aktivitäten, deren Abarbeitung vom Testmanager zu kontrollieren ist:

- Planung der weiteren Stufen des Testprozesses (Testentwurfs und Testanalyse, Testimplementierung, -Ausführung und -Bewertung)
- Organisation der für die Tests notwendigen Testumgebung und Ressourcen
- Definition des Umfangs, des Detaillierungsgrads, der Struktur und ggf. der Vorlagen für die Testdokumentation
- Identifizierung von Art und Umfang der Testautomatisierung sowie Organisation der dafür notwendigen Tools
- Bestimmen des Detaillierungsgrads für Testablaufspezifikationen (um die Reproduzierbarkeit sicherzustellen)
- Festlegung der Eingangs- und Endkriterien für die einzelnen Teststufen.
- Definition von Kriterien für einen Testabbruch sowie für die Testwiederaufnahme

AKTIVITÄTEN DER TESTPLANUNG



AKTIVITÄTEN DER TESTPLANUNG

EINGANGSKRITERIEN

Eingangskriterien sind Bedingungen, die erfüllt sein müssen, um den jeweiligen Test zu starten. Ein einfaches Beispiel wäre, dass die Teststufe des Systemtests erst gestartet werden darf, wenn alle Integrationstests erfolgreich abgeschlossen sind.

Weitere Beispiele:

- Die Testumgebung ist einsatzbereit
- Alle Testwerkzeuge stehen in der Testumgebung bereit.
- Die zu testende Version ist erfolgreich installiert.
- Alle notwendigen Testdaten sind eingespielt.

AKTIVITÄTEN DER TESTPLANUNG

ENDEKRITERIEN

Endekriterien sind Bedingungen, die erfüllt sein müssen, um den jeweiligen Test abzuschließen. Grundlage für Endekriterien können Intensitätsmaße (Codeüberdeckung, etc.), Maxima nicht behobener Fehlerzustände, Kosten oder auch Zeitpläne sein.

Beispiele für Endekriterien:

- Überdeckungsgrad (codebasiert oder anforderungsbasiert)
- Testlauf ohne Fehler der Klasse-1 + Klasse 2
- Fehlerfindungsrate nach investierter Testzeit
- Zeit oder Kosten

TESTKONZEPT

Die Ergebnisse der Testplanung finden ihren Niederschlag im Testkonzept. Es kann in Form eines für alle Teststufen gültigen Mastertestkonzepts oder in Form mehrerer Stufentestkonzepte, jeweils gültig für eine einzelne Teststufe, erstellt werden. Eine mögliche Gliederung für ein Testkonzept kann dem Standard IEEE 29119-3: "Test Documentation" entnommen werden.

1. Testkonzeptbezeichnung
2. Einführung
3. Testobjekte
4. Zu testende Leistungsmerkmale
5. Leistungsmerkmale, die nicht getestet werden
6. Teststrategie
7. Abnahmekriterien
8. Kriterien für Testabbruch und Testfortsetzung

TESTKONZEPT

Die Ergebnisse der Testplanung finden ihren Niederschlag im Testkonzept. Es kann in Form eines für alle Teststufen gültigen Mastertestkonzepts oder in Form mehrerer Stufentestkonzepte, jeweils gültig für eine einzelne Teststufe, erstellt werden. Eine mögliche Gliederung für ein Testkonzept kann dem Standard IEEE 29119-3: "Test Documentation" entnommen werden.

- 9. Testdokumentation
- 10. Testaufgaben
- 11. Testumgebung
- 12. Verantwortlichkeiten/Zuständigkeiten
- 13. Personal, Einarbeitung, Ausbildung
- 14. Zeitplan, Arbeitsplan
- 15. Planungsrisiken und Unvorhergesehenes
- 16. Genehmigung, Freigabe



TESTKONZEPT

Sollte eine Testrichtlinie der zu testenden Organisation existieren, so sollte diese während der Testplanung berücksichtigt werden und im Testkonzept Niederschlag finden. Testplanung ist eine kontinuierliche Aktivität, das Testkonzept ein lebendes Dokument. Erfahrungen aus den einzelnen Testaktivitäten sind zu nutzen, um ändernde Risiken und Randbedingungen zu erkennen. Es ist Aufgabe des Testmanagements diese durch Fortschreiben des Testkonzepts angemessen zu berücksichtigen.



TESTSTRATEGIE, TESTVORGEHENSWEISE

Bei der Teststrategie handelt es sich um eine abstrakte Beschreibung der vorgesehenen Teststufen und der Art und Weise, wie innerhalb dieser Teststufen vorzugehen ist.

Im Testmanagement geht die konkrete Umsetzung einer Teststrategie in einem spezifischen Projekt aus der Testvorgehensweise hervor. Die Testvorgehensweise enthält die getroffenen Entscheidungen zur Erreichung der (Test-)Projektziele, die Ergebnisse der Risikoanalyse, die Testentwurfsverfahren, die Endekriterien und die geplanten durchzuführenden Tests (Testarten).



TESTSTRATEGIE, TESTVORGEHENSWEISE

Die Wahl einer geeigneten Testvorgehensweise ist abhängig vom Kontext. Die Auswahl wird beeinflusst durch Faktoren wie Risiken, verfügbare Ressourcen und Kompetenzen, die Technologie, Art des Systems (z.B. maßgeschneidert oder kommerzielle COTS-Software), Testziele und Vorschriften. Die Teststrategie und Testvorgehensweise werden in den Testkonzepten definiert. Im Testentwurf werden diese dann weiter konkretisiert und verfeinert.

Zu den gängigen Testvorgehensweisen zählen:

ANALYTISCHE VORGEHENSWEISEN

Wie das risikoorientierte Testen, in dem das Testen auf die Bereiche der größten Risiken ausgerichtet ist

TESTSTRATEGIE, TESTVORGEHENSWEISE

- **MODELLBASIERTE VORGEHENSWEISEN**

Wie das stochastische Testen, das statistische Informationen über Ausfallraten (beispielsweise Zuverlässigkeitswachstumsmodelle) oder Systembenutzung (beispielsweise Benutzungsprofile) nutzt

- **METHODISCHE VORGEHENSWEISEN**

Wie das ausfallbasierte (einschließlich intuitiver Testfallermittlung und Fehlerangriff), erfahrungsbasierte, checklistenbasierte und qualitätsmerkmalbasierte Testen

- **PROZESS- ODER STANDARDKONFORME VORGEHENSWEISEN**

Spezifiziert durch Industriestandards oder die verschiedenen agilen Methoden



TESTSTRATEGIE, TESTVORGEHENSWEISE

■ DYNAMISCHE UND HEURISTISCHE VORGEHENSWEISEN

Wie das explorative Testen, bei dem das Testen weniger vorgeplant ist und stärker auf Ereignisse reagiert und Durchführung und Auswertung parallel laufen. Zu den gängigsten Methoden zählen Black Box Test und White Box Test.

■ BERATENDE VORGEHENSWEISEN

In denen die Testüberdeckung primär durch Hinweise und Beratung von Technologie- und/ oder Geschäftsbereichsexperten außerhalb des Testteams getrieben wird

TESTSTRATEGIE, TESTVORGEHENSWEISE

■ WIEDERVERWENDUNGSORIENTIERTE VORGEHENSWEISEN

Bei denen man vorhandene Tests und Testumgebungen (aus früheren Projekten), umfangreiche Automatisierung von funktionalen Regressionstests und Standardtestsuiten als Ausgangsbasis übernimmt. Ziel ist, die Tests schnell und pragmatisch aufzusetzen

Unterschiedliche Ansätze können kombiniert werden, beispielsweise zu einer risikoorientierten dynamischen Vorgehensweise.