
Bachelor of Science (BSc) in Informatik
Modul Advanced Software Engineering 2 (ASE2)

LE 10 – Software Testing

7 Testwerkzeuge

Institut für Angewandte Informationstechnologie (InIT)
Walter Eich (eicw) / Matthias Bachmann (bacn)

<https://www.zhaw.ch/de/engineering/institute-zentren/init/>



Agenda

7 Testwerkzeuge

7.1 Testwerkzeugtypen

7.2 Nutzen und Risiken der Testautomatisierung

7.3 Effektive Nutzung von Werkzeugen

7.4 Wrap-up



Lernziele nach Syllabus ISTQB CTFL

6.1 Überlegungen zu Testwerkzeugen

- FL-6.1.1 (K2) Testwerkzeuge gemäß ihrem Zweck und den Testaktivitäten, die sie unterstützen, klassifizieren können
- FL-6.1.2 (K1) Nutzen und Risiken der Testautomatisierung identifizieren können
- FL-6.1.3 (K1) Sich an besondere Gesichtspunkte von Testdurchführungs- und Testmanagementwerkzeugen erinnern können

6.2 Effektive Nutzung von Werkzeugen

- FL-6.2.1 (K1) Die Hauptprinzipien für die Auswahl eines Werkzeugs identifizieren können
- FL-6.2.2 (K1) Sich an Ziele für die Nutzung von Pilotprojekten zur Einführung von Werkzeugen erinnern können
- FL-6.2.3 (K1) Erfolgsfaktoren für die Evaluierung, Implementierung, Bereitstellung und kontinuierliche Unterstützung von Testwerkzeugen in einem Unternehmen identifizieren können



7 Testwerkzeuge – Ziele und Zweck

- **Steigerung der Effizienz** der Testaktivitäten durch eine Automatisierung sich wiederholender Testaufgaben
- **Verbesserung der Qualität der Tests** durch eine werkzeuggestützte Verwaltung der Testfälle und Handbarkeit einer Vielzahl von Testfällen
- **Verbesserung der Zuverlässigkeit** der Tests durch die Automatisierung manueller Aufgaben
- **Bewerkstelligen der Tests** durch Werkzeuge bei Tests, die manuell nicht zu realisieren sind (z.B. Performanz- und Lasttests)



7.1 Testwerkzeugtypen (1/2)

- Einige Testwerkzeuge unterstützen **lediglich eine Testaktivität**; andere können mehreren Testaktivitäten zugeordnet werden (**Werkzeug-Suiten**).
- Einige Testwerkzeuge werden als **intrusiv** bezeichnet; d.h. sie können das Verhalten des Testobjekts beeinflussen (Instrumentierung beim White-Box-Testing).
- **Testframework** ist ein weiterer Begriff im Zusammenhang mit Testwerkzeugen:
 - Wiederverwendbare und erweiterbare Testbibliotheken, die zum Erstellen von Testwerkzeugen dienen können (auch als Testrahmen bezeichnet)
 - Art des Entwurfs der Testautomatisierung (z.B. datengetrieben, schlüsselwortgetrieben)
 - Der gesamte Prozess der Testdurchführung
- Liste von marktgängiger Werkzeuge mit Bezugsquelle:
<https://www.testtoolreview.de/>



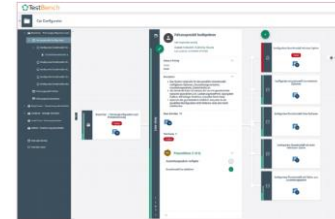
7.1 Typen von Testwerkzeugen (2/2)

- Werkzeuge für Management und Steuerung von Tests
- Werkzeuge zur Testspezifikation
- Werkzeuge für statischen Test
- Werkzeuge für zur Automatisierung dynamischer Test
- Werkzeuge für Last- und Performanztest
- Werkzeugunterstützung für spezielle Testbedürfnisse



7.1.1 Werkzeuge für Management und Steuerung von Tests (1/2)

- **Testmanagementwerkzeuge**
 - Diese Werkzeuge bieten Schnittstellen für die Testdurchführung, das Verfolgen von Fehlerzuständen und das Verwalten von Anforderungen zusammen mit der Unterstützung von quantitativen Analysen und dem Berichten über Testobjekte.
- **Requirements-Managementwerkzeuge**
 - Diese Werkzeuge verwalten Anforderungsbeschreibungen, speichern die Merkmale der Anforderungen (einschliesslich Priorität), liefern eindeutige Bezeichnungen und unterstützen die Nachverfolgung der Anforderungen bis zu einzelnen Tests.
- **Fehler- und Abweichungsmanagementwerkzeuge**
 - Diese Werkzeuge speichern und verwalten Fehler- und Abweichungsberichte, d.h. Fehlerzustände, Änderungsanforderungen (change requests), Fehlerwirkungen oder wahrgenommene Probleme und Anomalien.



7.1.1 Werkzeuge für Management und Steuerung von Tests (2/2)

- **Konfigurationsmanagement**
 - Werkzeuge für das Konfigurationsmanagement sind ebenfalls keine Testwerkzeuge im engeren Sinne. Sie ermöglichen, die unterschiedlichen Versionen und Konfigurationen der zu testenden Software zu verwalten, aber auch die unterschiedlichen Versionen der Testfälle, Testdaten und anderer Produkt- und Testdokumente. Damit wird (leichter) nachvollziehbar, welche Testergebnisse ein Testlauf an einem Testobjekt einer bestimmten Version erbracht hat.
- **Continuous Integration**
 - In Projekten, bei denen agil bzw. iterativ-inkrementell entwickelt wird, werden die Integration und der anschliessende Test neuer oder geänderter Codeteile in der Regel durch einen stark automatisierten Prozess unterstützt.



7.1.2 Werkzeuge zur Testspezifikation

- **Testentwurfswerkzeuge**
 - Diese Werkzeuge werden verwendet, um Testeingaben, ausführbare Tests und/oder Testorakel aus den Anforderungen, der graphischen Benutzungsschnittstelle (GUI), dem Entwurfsmodell (Zustands-, Daten- oder Objektmodell) oder aus dem Code zu generieren.
- **Testdatengeneratoren und –editoren**
 - Mit Hilfe von Testdatengeneratoren und -editoren können aus Datenbanken, Dateien oder Datenströmen zunächst Testdaten ermittelt und dann sämtliche für einen Test benötigte Testdaten bearbeitet werden, um den Datenschutz durch Anonymisierung sicherzustellen.



7.1.3 Werkzeuge für den statischen Test

- **Reviewwerkzeuge**
 - Diese Werkzeuge helfen mit Prozessen, Checklisten und Regeln beim Review und werden zur Ablage und Kommunikation von Reviewkommentaren, Berichten zu Fehlerzuständen und Aufwand verwendet.
- **Statische Analysewerkzeuge**
 - Diese Werkzeuge unterstützen Entwickler und Tester in der Aufdeckung von Fehlerzuständen vor dem dynamischen Testen, indem sie Programmierkonventionen einschliesslich sicherer Programmierung (secure coding) erzwingen und die Analyse von Strukturen und Abhängigkeiten ermöglichen.
- **Modellierungswerkzeuge**
 - Diese Werkzeuge werden verwendet, um Softwaremodelle zu validieren (z.B. fehlende Zustände, fehlende Zustandsübergänge etc.).



7.1.4 Werkzeuge zur Automatisierung dynamischen Test

- **Testausführungswerkzeuge**
 - Diese Werkzeuge ermöglichen eine automatische oder halbautomatische Ausführung von Tests unter Verwendung der gespeicherten Eingaben und der erwarteten Ausgaben mittels einer Skriptsprache (z.B. Testroboter).
- **Testrahmen/Komponententestrahmen**
 - Ein Komponententestrahmen oder Testrahmen erleichtert den Test einer Komponente oder eines Teilsystems durch Simulation der Umgebung des Testobjekts, durch Scheinobjekte (Simulatoren) als Platzhalter und/oder Treiber.
- **Vergleichswerkzeuge/Komparatoren**
 - Vergleichswerkzeuge ermitteln die Unterschiede zwischen Dateien, Datenbanken oder Testergebnissen.
- **Werkzeuge zur Überdeckungsmessung**
 - Diese Werkzeuge messen – mit intrusiven oder nicht-intrusiven Mitteln – den prozentualen Anteil spezifischer Codestrukturtypen



7.1.5 Werkzeugunterstützung für Last- und Performanztest

- Last- und Performanztests sind notwendig, wenn ein Softwaresystem **eine grosse Zahl an parallelen Anfragen oder Transaktionen** (Last) aus führen muss, wobei gewisse maximale Antwortzeiten (Performanz) nicht überschritten werden dürfen.
- Derartige Anforderungen müssen Echtzeitsysteme und in der Regel Client/Server-Systeme sowie web- und Cloud-basierte Anwendungen erfüllen.
- Werkzeuge für den Last- und Performanztest müssen zwei Aufgaben erfüllen:
 - synthetische Last generieren (z.B. Datenbankabfragen, Benutzertransaktionen oder Netzwerkverkehr)
 - und parallel die (abhängig von der eingespeisten Last) resultierende Performanz des Testobjekts messen, protokollieren und visualisieren



7.1.6 Werkzeugunterstützung für spezielle Testbedürfnisse

- **IT-Sicherheitstests**

- Werkzeuge zur Prüfung von Zugriffs- und Datensicherheit überprüfen ein System auf Sicherheitslücken durch deren Ausnutzung sich nicht berechnigte Personen u.U. Zugang zum System verschaffen können (s. zB. das «Open Web Application Security Project» (OWASP), das einen Katalog potenzieller Schwachstellen und Sicherheitslücken publiziert, die im IT-Security-Test von Webanwendungen zu überprüfen sind).

- **Bewertung der Datenqualität**

- In Projekten, in denen ein Altsystem auf ein neues IT-System umgestellt wird, müssen meist umfangreiche Datenkonvertierungen bzw. eine Migration der Datenbestände vom Altsystem ins neue System durchgeführt werden,



7.2 Nutzen und Risiken der Testautomatisierung (1/3)

- Die **Einführung eines neuen Testwerkzeugs** ist mit **Kosten** für Auswahl, Anschaffung und Wartung der Werkzeuge verbunden.
- **Kosten** für **Hardware** und **Mitarbeiterschulung** können hinzukommen.
- Bei **Werkzeugen zur Automatisierung der Testdurchführung** kann **gut abgeschätzt** werden, **wie viel Aufwandsersparnis** ein automatisierter Testlauf gegenüber der manuellen Durchführung bringt.
- Der neue Aufwand für die Programmierung der Tests muss natürlich abgezogen werden, sodass nach nur einem automatisierten Testlauf die **Kosten-Nutzen-Bilanz** meist negativ ausfällt.
- Erst mit **weiteren automatisierten Regressionstestläufen** summiert sich die pro Testlauf erzielte Ersparnis.



7.2 Nutzen und Risiken der Testautomatisierung (2/3)

- Zusammenfassend kann von folgendem **potenziellen Nutzen** aus gegangen werden:
 - Die **kreativen Testarbeiten** können **nicht ersetzt**, aber durch **Werkzeuge unterstützt werden**.
 - Die **mechanische Testdurchführung kann automatisiert** werden, was den Testaufwand reduziert oder bei gleichem Aufwand – in gleicher oder kürzerer Zeit – mehr Tests zulässt.
 - Die Werkzeuge erleichtern die Ermittlung, **Auswertung** und **Informationen über Testfortschritt, erreichte Testüberdeckung** und die **aufgedeckten Fehlerwirkungen** und deren **Häufigkeitsverteilung**.



7.2 Nutzen und Risiken der Testautomatisierung (3/3)

- Mit der Werkzeugeinführung können **auch Risiken** verbunden sein, z.B.:
 - **Erwartungen** an das Werkzeug **unrealistisch hoch**
 - **Zeit, Kosten und Aufwand** der Einführung, aber auch der Nutzung werden **unterschätzt**
 - Werkzeug kann nicht eingesetzt werden oder den **gewünschten Effekt nicht** erzielen, weil **kein ausreichend guter Testprozess etabliert** ist oder dieser nicht praktiziert wird
 - **Verbesserungen und Änderungen im Testprozess**, die erforderlich sind, um das Werkzeugpotenzial auszuschöpfen, **sind umfangreicher als gedacht**
 - **Notwendiger Aufwand zur Wartung und Versionskontrolle** der Testfälle, Testdaten und anderer Testdokumente **wird unterschätzt**
 - Werkzeug auch dort eingesetzt, wo ein **manuelles Vorgehen einfacher** und wirtschaftlicher wäre
 - Werkzeug arbeitet mit anderen vorhandenen Tools **nicht gut genug zusammen**
 - **Werkzeuganbieter nimmt das Werkzeug** vom Markt, wird aufgekauft oder stellt seine Geschäftstätigkeit ein
 - **Neue Versionen des Tools** unterstützen **alte Technologie**, die der Anwender noch einsetzt, nicht mehr
 - **Support für das Werkzeug ist schlecht** oder wird eingestellt



7.3.1 Auswahl und Nutzung von Testwerkzeugen (Wichtig!)

- **Empfohlene Einführungsreihenfolge** für Werkzeuge:
 1. Fehlermanagement
 2. Konfigurationsmanagement
 3. Testplanung
 4. Testdurchführung
 5. Testspezifikation
- Zu beachten sind auch **Zeiträume**, die nötig sind, um den Umgang mit einem neuen Werkzeug zu etablieren!
- Aufgrund der **Lernkurve** kann statt der erhofften Produktivitätssteigerung in einer Übergangsphase sogar eine Produktivitätseinbusse erfolgen.



7.3.2 Werkzeugauswahl – Ablauf (Wichtig!)

1. Anforderungsspezifikation für den Werkzeugeinsatz
2. Marktrecherche und Aufstellen einer Übersichtsliste der Kandidaten
3. Werkzeugdemos und Werkzeugerprobung (Trial)
4. Vorauswahl (Shortlist) anhand der Anforderungsliste erstellen
5. Machbarkeitsstudie (Proof of Concept) durchführen



7.3.2 Werkzeugauswahl - Kriterien

- Güte des Zusammenspiels mit den potenziellen Testobjekten
- Know-how der Tester über Werkzeug oder/und vom Werkzeug unterstützte Methode
- Möglichkeit zur Integration in die vorhandene Entwicklungsumgebung
- Möglichkeit zur Integration mit anderen eingesetzten (Test)Werkzeugen
- Plattform, auf der das Werkzeug eingesetzt werden soll
- Service, Verlässlichkeit und Marktstellung des Herstellers
- Vor- und Nachteile verschiedener Lizenzmodelle (z.B. kommerzielle Software vs. Open Source, Kauf vs. Miete)
- Preis und laufende Wartungskosten



7.3.3 Pilotprojekt zur Werkzeugeinführung

- Ist die **Wahl getroffen**, gilt es, das Werkzeug im eigenen Unternehmen einzuführen.
- Üblicherweise wird hierzu zunächst **ein Pilotprojekt durchgeführt**, um nachzuweisen, dass der erwartete Nutzen in einem realen Projektumfeld auch tatsächlich erreicht wird.
- Das **Pilotprojekt** soll nicht **von denselben Personen durchgeführt** werden, die schon bei der **Evaluation mitgearbeitet** haben.
- Die Evaluationsergebnisse werden **sonst unter Umständen** zu **unkritisch übernommen**.
- Der **Pilotbetrieb** soll **zusätzliches Wissen** über **technische Details des Werkzeugs liefern**, aber auch Erfahrungen mit dessen praktischem Einsatz und über das Einsatzumfeld.



7.3.4 Faktoren für die erfolgreiche Einführung und Nutzung (Wichtig!)

- Wichtige Erfolgsfaktoren bei der Breitereinführung sind:
 - Breitereinführung **schrittweise** vornehmen
 - Die Werkzeugunterstützung **in den Prozessen verankern**
 - **Begleitende Benutzertrainings- und Coachingmassnahmen** vorsehen
 - Richtlinien und Empfehlungen zur Werkzeuganwendung bereitstellen
 - **Einsatz Erfahrungen sammeln** und allen Anwendern zur Verfügung stellen (Tipps & Tricks, FAQ etc.)
 - **Support durch interne Anwendergruppe**, interne Toolexperten u.Ä. anbieten
 - **Akzeptanz und Nutzen** des Werkzeugs **verfolgen und auswerten**



Wrap-up

- Für jede Phase im Testprozess sind Werkzeuge verfügbar, die dem Tester helfen, seine Testarbeiten qualitativ zu verbessern und/oder zu automatisieren.
- Die Nutzung eines Testwerkzeugs bringt nur Vorteile, wenn der Testprozess als solcher beherrscht wird und definiert abläuft.
- Die Einführung eines Testwerkzeugs kann mit hohen Investitionen verbunden sein, weshalb die Werkzeugauswahl sorgfältig und nachvollziehbar erfolgen muss.
- Dem potenziellen Nutzen von Werkzeugen stehen Risiken gegenüber, an denen der Werkzeugeinsatz scheitern kann.
- Die Einführung des ausgewählten Werkzeugs muss begleitet werden. Information und Training der späteren Anwender helfen, deren Akzeptanz und damit den regelmässigen Einsatz des Werkzeugs sicherzustellen.
- Eine systematische Werkzeugeinführung gliedert sich in folgende Schritte: Werkzeugauswahl, Pilotprojekt, Breitereinführung, dauerhafter Anwendersupport.



Ausblick

- Das Thema der nächsten Vorlesung ist:
 - Software Maintenance and Operations

