

# Leitfaden API

## Testing und Debugging



Erstellt am	4.9.2014	Autor	FG API, Rinaldo Lanza
Dokumentenstatus	Freigegeben „at work“	Version	1.0
Verteiler	Fachgruppe API		
Änderungen	Datum	Version	Autor



## Inhaltsverzeichnis

<b>1</b>	<b>Beschreibung / Einleitung</b>	<b>4</b>
<b>2</b>	<b>Testing Fokus aus dem Fokus der Module</b>	<b>5</b>
2.1	Empfehlungen der Fachgruppe .....	5
2.1.1	Aus Datensicht: .....	5
2.1.2	Aus Sicht des Codes:.....	6
2.2	Notizen und Gedanken .....	6
<b>3</b>	<b>Debugging Fokus aus dem Fokus der Module</b>	<b>7</b>
3.1	Empfehlungen der Fachgruppe .....	7
<b>4</b>	<b>Softwareverwaltungssysteme und agile Methoden</b>	<b>8</b>
4.1	Mögliche Vorgehensweise.....	8

## **1 Beschreibung / Einleitung**

Dieser Leitfaden dient

- › den Berufsbildner als Leitfaden für Arbeiten die Lernende im Bereich Testing durchführen könnten.
- › als Leitfaden und Übersicht was während der Ausbildung in den Module für Prinzipien und Konzepte angewendet werden sollen.
- › der Berufsfachschulen als praxisnahe und aktuelle Prinzipien und Konzepte die in der Wirtschaft eingesetzt und in der Ausbildung auf Sekundarstufe 2 gewünscht werden.

## 2 Testing Fokus aus dem Fokus der Module

### 2.1 Empfehlungen der Fachgruppe

Die folgende Tabelle zeigt Prinzipien und Konzepte die in den einzelnen Modulen zur Anwendung kommen könnten. Sie erlaubt die Fokussierung auf die einzelnen Themen im Bereich Testing.

Modul	Fokus
403	Blackbox und Whitebox Testing, Pfadabdeckung. Testprotokolle, Messbarkeit von Tests.
404	Einfache Komponententest, wie auch Blackbox und Whitebox Testing.
411	Fokus auf den Algorithmus, Grenzwerte bestimmen, spezielle Werte, Re-kursionen, etc.
226	xUnit, msTest Tests, Life-Cycle eines Test. Klassen, Klassengruppen und allenfalls Paketttest.
120	Usability- und Ergonomietests. Sehr schwierig zu handhaben.
326	<i>Überprüfung der Anforderungen. „Happy Day Fall“ und andere. Test für funktionale und nicht funktionale Anforderungen.</i>
223	xUnit Tests. Pakete und Komponententests. Überprüfung der Anforderun-gen. Test-Driven Design Ansatz.
101	Einfache Syntax-Checks. Browser Checks. Anforderungen bezüglich Lade-zeiten.
307	Testprotokolle und Grenzwerte.
133	Logging Mechanismen. Was soll protokolliert werden? Load-Tests.
151	Thematik Mockups. Performance- und Load-Tests.
150	Last-Test (nicht funktionaler Bereich), Geschwindigkeit.
152	Codec- und Plattformproblematiken.
183	Sicherheitsanforderungen Sicherheitstest definieren und durchführen (Penetration-Tests)

#### 2.1.1 Aus Datensicht:

Eingabeüberdeckung, Ausgabeüberdeckung, Funktionsüberdeckung (Blackbox)

### **2.1.2 Aus Sicht des Codes:**

Anweisungsüberdeckung, Zweigüberdeckung (von einer Entscheidung abhängig),  
Pfadüberdeckung

## **2.2 Notizen und Gedanken**

### **Qualitätsmerkmale nach ISO 9126**

Funktional, nicht-funktional, strukturell, Änderungs-bezogen

#### **Statisch**

- › Review, Informelles Review, Walkthrough, Technisches Review
- › Inspektion, Phasen eines formalen Reviews

#### **Dynamisch**

- › Unsystematisch, erfahrungsbasiertes Verfahren, Error Guessing, Exploratives Testen

#### **Funktionales Testen**

Äquivalenzklassenbildung, Grenzwertanalyse, Entscheidungsanalyse, Zustandsbezogenes Testen, Use-Case-bezogenes testen

#### **Nicht-funktionale Qualitätsmerkmale**

Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit.

#### **Fehlerbegriffe**

Fehlhandlung, Fehlerzustand, Fehlerwirkung, Fehlermaskierung

#### **Testbegriffe**

Debugging, Festfälle, Testszenario, Teststufen, Testmanagement

„definition of done“ (Scrum)

Copyright bei Bildern (M152)

## 3 Debugging Fokus aus dem Fokus der Module

### 3.1 Empfehlungen der Fachgruppe

Je nach Modul, eingesetzte Technologien und eingesetzte Entwicklungsumgebung ändert sich das Debugging und das damit verbundene Vorgehen bei der Fehlersuche. Die folgende Tabelle zeigt eine mögliche Fokussierung der Fehlersuchstrategien und Vorgehensweisen in den einzelnen Modulen.

Diese Gliederung dient den Berufsfachschulen als Leitfaden für das Vermitteln der Konzepte in den einzelnen Bereichen, wie auch den Ausbildungsbetrieben für die Anwendung an den Firmenspezifischen Werkzeugen.

Modul	Fokus
403	Blackbox, Whitebox, Pfade
404	Komponenten
411	Durchläufe, Abbruchbedingungen,
226	Stacktrace
120	
326	
223	
101	
307	
133	Testing von Objekte mit definierter Lebensdauer (SessionScope, ViewScope, etc.) Debugging auf der Client-Seite im Browser (Inspektor, Netzwerkanalyse etc.)
151	
150	
152	Test von Elementen auf verschiedenen Browsern.
183	Sicherheitstest definieren und durchführen (Penetration-Tests)

## 4 Softwareverwaltungssysteme und agile Methoden

### 4.1 Mögliche Vorgehensweise

- › Starten mit einfachen Übungen am Versionsverwaltungssystem mit einfachen Dokumenten (Word, Excel, Arbeitsjournale, etc.). Beispielsweise könnte der Berufsbildner auf die aktuellen Arbeiten der Lernenden zugreifen.
- › Erste Programmierschritte wie Ablaufsteuerung. Alle Arbeiten an den Programmen werden im Repository versioniert.
- › Einfaches Debugging der programmierten Abläufe. Variableninspektion, Breakpoints, Pfadverfolgung (Step-In, Step-Over, Step-Out, run to cursor, etc.)  
  
White-Box, Black-Box testing
- › Automatisierte Test auf xUnit-Test-Basis programmieren. Systematisches Vorgehen aneignen (Pfadabdeckung, Variablengrenzen, Arrange-Act-Assert-Pattern)  
  
Namenskonventionen für xUnit-Tests umsetzen.  
  
Qualitätssicherung
- › Einsatz von Mockups im Testumfeld.
- › Teststrategien für verschiedene Applikationstypen (Web, Desktop, SOA, etc.)  
  
Debugging-Strategien für Applikationstypen (Debugging im Browser, Multi-Threading, Stacktrace, Logging, etc.)
- › Automatisierte GUI-Test im Web-Umfeld kennen, beispielsweise Selenium.  
  
Allenfalls können an einem Mustersystem Testfälle programmiert werden.

Hinweis: Die Fachgruppe Applikationsentwicklung erarbeitet einen weiteren Leitfaden zu Thema Agile-Softwareentwicklung. Die Mitarbeit durch Vertreter der Ausbildungsbetriebe ist herzlich willkommen.