GoSys-Logo

**Fachhochschule Aachen**

**Campus Jülich**

Fachbereich 9: Medizintechnik und Technomathematik

Studiengang: Angewandte Mathematik und Informatik

Bachelorarbeit

Testautomatisierung im Rahmen von

Test-Driven-Development

Inhalt

Inhaltsverzeichnis

[Einleitung 3](#_Toc41545560)

[Motivation 3](#_Toc41545561)

[Ziel 3](#_Toc41545562)

[Theoretische Grundlagen 4](#_Toc41545563)

[Test-Driven-Development 4](#_Toc41545564)

[Test-Driven-Development und die Test-Ebenen 4](#_Toc41545565)

[Test-Driven-Development Varianten (A-TDD, B-TDD, F-TDD) 4](#_Toc41545566)

[Gherkin 4](#_Toc41545567)

[Gherkin-Cucumber 5](#_Toc41545568)

[Praxisbezug 5](#_Toc41545569)

[Vorwort 5](#_Toc41545570)

[Umsetzung in Altanwendungen 5](#_Toc41545571)

[Umsetzung in Neu-Anwendungen 5](#_Toc41545572)

[Lektüren 6](#_Toc41545573)

[xUnit Test Patterns 6](#_Toc41545574)

[Lean-Agile Acceptance Test-Driven Development 6](#_Toc41545575)

# Einleitung

## Motivation

Motivation der Arbeit sind immer komplexer und unübersichtlichere werdende Prozesse und Entwicklungsansätze in heutigen IT-Unternehmen. Heutzutage wird diverse Software entwickelt, die Unterschiedlichste Anforderungen und Spezifikationen erfüllen muss. Hinzu kommt der digitale Wandel hinsichtlich der Entwicklungs- und Prozessstrukturen in Bezug auf Agilität und moderne Entwicklungsansätze. Diese und viele weitere Faktoren erzwingen eine passende dynamische Qualitätssicherung im Bereich der Software-Tests. Durch den stetigen Wandel ergab sich auch dort im Laufe der letzten Jahre viele Neuheiten und Anpassungen. Software ist heutzutage verflochtener und vielfältiger geworden, die Anforderungen ändern sich immer öfter und genauso steigen die Möglichkeiten der Test-Ansätze sowie die damit verbundene Test-Komplexität. Folglich lässt sich leicht der Überblick verlieren und auch das Testen gerät bei stetig steigenden Softwares unauffällig in den Hintergrund – die Relevanz wird zudem durch mangelnde Ressourcen, wie Zeit oder Kostendeckung oft vernachlässigt.

## Ziel

Folglich ist das Ziel dieser Arbeit eine Übersicht der möglichen Test-Verfahren im Rahmen des Test-Driven-Development (kurz: TDD) darzustellen. Hierbei liegt der Fokus auf TDD, da es sich dabei um die Etablierung in einer agilen Entwicklungsstruktur handelt, welche diesen Stil priorisiert. Dabei soll TDD granularer vorgestellt werden. Dies beinhaltet unter anderem Acceptance-TDD (kurz: A-TDD), Behaviour-TDD (kurz: B-TDD) und Feature-TDD (kurz: F-TDD) hinsichtlich deren Definition, Vor- und Nachteile, das Ziel diverser Test-Ansätze sowie mögliche Umsetzungen. Speziell A-TDD wird genauer Untersucht, wobei auch gezeigt werden soll, welche Technologien (Gherkin in Kombination mit Cucumber) A-TDD Unterstützend für die agile Entwicklung und Test-Automatisierung sind und wie sich diese implementieren lassen. Durch Bespiele aus der Praxis soll ein besseres Verständnis für die oben genannte Theorie gewährleistet werden.

# Theoretische Grundlagen

## Test-Driven-Development

Dieses Kapitel dient als Einführung für die TDD-Entwicklung. Dabei werden über die Definition hinaus Vor- und Nachteile aufgezählt, die Relevanz von TDD mithilfe von möglichen internen Kennzahlen untermauert sowie die Einbindung in einem agilen Umfeld erläutert. Es soll ein fundamentaler Einblick in TDD erboten werden, sodass der Leser im Anschluss grundlegende Kenntnisse für tiefergreifende Themen besitzt.

## Test-Driven-Development und die Test-Ebenen

Anschließend gilt es einen größeren Überblick des TDD-Ansatzes im Bezug der Test-Ebenen zu bekommen. Es gilt zu klären, in welchen Ebenen TDD eine Rolle spielt und welche Auswirkungen dort vorhanden sind (Implementierung / Umsetzung / Komplexität / Kosten).

## Test-Driven-Development Varianten (A-TDD, B-TDD, F-TDD)

Hier wird detaillierter auf diverse TDD Ansätze eingegangen. Es sollen unterschiedliche Varianten verglichen und analysiert werden. Best-Practise-Beispiele zeigen die Anwendung von TDD-Variationen, sodass ein feineres Verständnis vermittelt wird. Hinzu kommen wichtige Personen und Schnittstellen, die bei den Variationen eine wichtige Rolle einnehmen. Es soll gezeigt werden, wer welche Aufgabe in einem agilen TDD-Entwicklungsprozess hat und wo die Grenzen liegen.

## Gherkin

Um Grundlegend sicherzustellen, dass alle Beteiligten Instanzen (technisch und fachlich) sich verstehen, ist eine allgemeine Sprache von hoher Wichtigkeit. Für diese Zwecke wird Gherkin verwendet, was in diesem Kapitel aufgegriffen wird. Gezeigt werden die dadurch resultierende Vorteile sowie Use-Case Beispiele mit verschiedenen Variationen.

## Gherkin-Cucumber

Für eine möglichst optimale Umsetzung wird eine passende Anwendung für Gherkin benötigt. Dabei wird Cucumber im Rahmen von dessen Möglichkeiten und Vorteile vorgestellt. Ziel ist es, einen Überblick über dessen Funktionalitäten in Kombination mit anderen Anwendungen im Allgemeinen zu erhalten.

# Praxisbezug

## Vorwort

Im Rahmen der Praxisbeispiele wird sich auf zwei Anwendungsszenarien bezogen. Beide unterscheiden sich grundlegend vom Aufbau, den Anforderungen sowie der Struktur. Somit ist die Anbindung einer Gherking-Cucumber-Umgebung für einen TDD-Ansatz unterschiedlich. Folglich zeigen beide Szenarien, wie sich dies im Falle eine Altanwendung – und somit älteren eingesetzten Technologien – sowie einer Neuanwendung mit modernen Technologien einbinden lässt.

## Umsetzung in Altanwendungen

Besonders Altanwendungen haben oft keine direkte Schichtentrennung zwischen Frontend und Backend. Folglich stellt sich hierbei die Frage, wie sich Cucumber in Kombination mit TDD sowohl fachlich, als auch technisch in dieser Art von Szenarien etablieren lässt.

## Umsetzung in Neu-Anwendungen

Modernere Anwendungen besitzen oft eine klare Trennung zwischen Frontend und Backend über mehrere Projekte hinweg. Diese besitzen eine klare Trennung zwischen einem Frontend mit dem UI beinhaltend sowie dem Backend für die Ausführung von Server-Logik (zum Beispiel Spring-Boot kombiniert mit Vue.js). Cucumber lässt sich hierbei passend als eigene Zwischenanwendung etablieren.

# Lektüren

## xUnit Test Patterns

Das Buch bezieht sich auf automatisiertes Testen in verschiedensten Konstellationen. Es beschreibt unterschiedliche Strategien, wichtige Begriffe innerhalb des Testens und diverse Patterns und Vorgehensarten. Weiterhin wird gezeigt, worauf beim automatisierten Testen zu achten ist, wie sich diese umsetzen lassen – besonders in agilen Prozessen – und welche Philosophie hinter der Testautomatisierung steckt.

* [Link](https://www.amazon.de/xUnit-Test-Patterns-Refactoring-Signature/dp/0131495054)

## Lean-Agile Acceptance Test-Driven Development

Hier geht es um eine Geschichte über drei fiktive Projekt-Stakeholder, die agile Techniken zum Planen und Ausführen des Projektes nutzen. Es zeigt mithilfe von Szenarien bzw. Beispielen, wie sich A-TDD im Rahmen der agilen Prozesse Anwenden und integrieren, anwenden und umsetzen lässt.

* [Link](https://www.amazon.de/gp/product/0321714083/ref=ppx_od_dt_b_asin_title_s00?ie=UTF8&psc=1)