

---

SKILL 2015

# Evaluation der Leistungsfähigkeit ausgewählter Mutationstestwerkzeuge

Lea Kristin Gerling

---

---

## Inhalt

---

### **Gliederung**

1. Einleitung
2. Grundlagen
3. Vorstellung der Mutationstestwerkzeuge
4. Leistungsvergleich der Mutationstestwerkzeuge
5. Fazit

## Gliederung

### 1. Einleitung

- Motivation
- Forschungsfragen

### 2. Grundlagen

### 3. Vorstellung der Mutationstestwerkzeuge

### 4. Leistungsvergleich der Mutationstestwerkzeuge

### 5. Fazit

---

## 1. Einleitung

---

### Motivation

- Qualitätsbestimmung einer Software

---

## 1. Einleitung

---

### Motivation

- Qualitätsbestimmung einer Software
- Qualitätsbestimmung einer Testmenge

---

## 1. Einleitung

---

### **Motivation**

- Qualitätsbestimmung einer Software
- Qualitätsbestimmung einer Testmenge
- Automatisierung der Qualitätsbestimmung

---

## 1. Einleitung

---

### Motivation

- Qualitätsbestimmung einer Software
- Qualitätsbestimmung einer Testmenge
- Automatisierung der Qualitätsbestimmung
- Welches Werkzeug?

---

## 1. Einleitung

---

### Forschungsfragen

- Lassen sich Mutationstests in der Praxis anwenden?
- Welche Mutationstestwerkzeuge eignen sich für diese Anwendung?



---

## Inhalt

---

## Gliederung

1. Einleitung
- 2. Grundlagen**
  - **Mutationstests**
  - **Kernhypothesen**
3. Vorstellung der Mutationstestwerkzeuge
4. Leistungsvergleich der Mutationstestwerkzeuge
5. Fazit

---

## 2. Grundlagen

---

### Mutationstests

- Entdeckungsfähigkeit einer Testmenge bestimmen  
→ Entdecken die Testfälle vorhandene Fehler?

---

## 2. Grundlagen

---

### Mutationstests

- Entdeckungsfähigkeit einer Testmenge bestimmen
- Fehlerbasiert
  - Künstliche Fehler erzeugen (Mutationen)

---

## 2. Grundlagen

---

### Mutationstests

- Entdeckungsfähigkeit einer Testmenge bestimmen
- Fehlerbasiert
- Zwei Kernhypothesen
  - Fehlerentstehung
  - Fehlerentdeckung

---

## 2. Grundlagen - Kernhypothesen

---

### Competent Programmer Hypothesis

*„Programmers have one great advantage that is almost never exploited:  
they create programs that are close to being correct!“ [DLS78, S. 34]*

---

## 2. Grundlagen - Kernhypothesen

---

### **Competent Programmer Hypothesis**

- Basis für die Fehlererzeugung
- Mutations-Operatoren

---

## 2. Grundlagen - Kernhypothesen

---

### Mutationen

Original Code	Erzeugte Mutationen
<code>erg = a + b</code>	<code>erg = a - b</code>
	<code>erg = a * b</code>
	<code>erg = a / b</code>
	<code>erg = a ** b</code>
	<code>erg = a</code>
	<code>erg = b</code>
	<code>erg = a MOD b</code>

## Coupling Effect

*„Test data that distinguishes all programs differing from a correct one by only simple errors is so sensitive that it also implicitly distinguishes more complex errors.“ [DLS78, S. 35]*



---

## 2. Grundlagen - Kernhypothesen

---

### **Coupling Effect**

- Basis für den Mutationstestprozess
- Eine Mutation pro Kopie (Mutant)
- Ausführung der Testfälle über Mutant
- Ausgabe des Mutation Scores

---

## Inhalt

---

### Gliederung

1. Einleitung
2. Grundlagen
- 3. Vorstellung der Mutationstestwerkzeuge**
  - **Übersicht**
  - **Gegenüberstellung**
4. Leistungsvergleich der Mutationstestwerkzeuge
5. Fazit

---

## 3. Vorstellung der Mutationstestwerkzeuge

---

### Übersicht

- MuJava (*Version 4, Juni 2013*)
- Jumble (*Version 1.2.0, April 2013*)
- Pitest (*Version 1.0.0, Mai 2014*)
- Judy (*Version 2.1.0, Januar 2014*)
- Major (*Version 1.1.2, Juni 2014*)

## 3. Vorstellung der Mutationstestwerkzeuge

### Gegenüberstellung

Name	Java Version	JUnit Version	Toolunterstützung
MuJava	6	4	-
Jumble	6	4	Ant, Eclipse, Mock
Pitest	8	4	Ant, Maven, Gradle, TestNG, Eclipse, Mock, Sonar, IntelliJ
Judy	6	4	Ant
Major	7	4	Ant

---

## Inhalt

---

### Gliederung

1. Einleitung
2. Grundlagen
3. Vorstellung der Mutationstestwerkzeuge
- 4. Leistungsvergleich der Mutationstestwerkzeuge**
  - **Testumgebung**
  - **Kriterien**
  - **Ergebnisse**
5. Fazit

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Testumgebung

- Java Version 7.65
- JUnit Version 4.11
- 2 GB RAM
- 3,3 GHz Prozessor
- Medianwerte bei 20-facher Wiederholung

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Kriterien

- Leistungsfähigkeit
- Bedienbarkeit

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge - Kriterien

---

### Leistungsfähigkeit (Effektivität)

- Mutation Score
- Anzahl erzeugter Mutanten
- Anzahl ausgeführter JUnit Tests



---

## 4. Leistungsvergleich der Mutationstestwerkzeuge - Kriterien

---

### Leistungsfähigkeit (Effizienz)

- Maximaler Speicherverbrauch
- CPU-Auslastung
- Anzahl erzeugter Mutanten pro Sekunde
- Anzahl ausgeführter Tests pro Sekunde

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge - Kriterien

---

### Bedienbarkeit

- Installation
- Mutanten-Erzeugung
- Testausführung
- Dokumentation
- Ergebnisdarstellung
- Konfigurierbarkeit

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %
Erzeugte Mutanten	432	39	44	121	79

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %
Erzeugte Mutanten	432	39	44	121	79
Ausführungsdauer	36 s	1,7 s	4,8 s	3,1 s	3,4 s

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %
Erzeugte Mutanten	432	39	44	121	79
Ausführungsdauer	36 s	1,7 s	4,8 s	3,1 s	3,4 s
Ausgeführte Tests	<i>k.A.</i>	<i>k.A.</i>	116	662	641

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %
Erzeugte Mutanten	432	39	44	121	79
Ausführungsdauer	36 s	1,7 s	4,8 s	3,1 s	3,4 s
Ausgeführte Tests	<i>k.A.</i>	<i>k.A.</i>	116	662	641
Speicherverbrauch	232 MB	5 MB	19 MB	95 MB	<i>k.A.</i>

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %
Erzeugte Mutanten	432	39	44	121	79
Ausführungsdauer	36 s	1,7 s	4,8 s	3,1 s	3,4 s
Ausgeführte Tests	<i>k.A.</i>	<i>k.A.</i>	116	662	641
Speicherverbrauch	232 MB	5 MB	19 MB	95 MB	<i>k.A.</i>
CPU-Auslastung	66 %	2,6 %	27 %	0,01 %	<i>k.A.</i>



## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %
Erzeugte Mutanten	432	39	44	121	79
Ausführungsdauer	36 s	1,7 s	4,8 s	3,1 s	3,4 s
Ausgeführte Tests	<i>k.A.</i>	<i>k.A.</i>	116	662	641
Speicherverbrauch	232 MB	5 MB	19 MB	95 MB	<i>k.A.</i>
CPU-Auslastung	66 %	2,6 %	27 %	0,01 %	<i>k.A.</i>
Mutanten / Sekunde	12	22,9	9,2	39	22,6

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Leistung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Mutation Score	0 %	97 %	82 %	100 %	67 %
Erzeugte Mutanten	432	39	44	121	79
Ausführungsdauer	36 s	1,7 s	4,8 s	3,1 s	3,4 s
Ausgeführte Tests	<i>k.A.</i>	<i>k.A.</i>	116	662	641
Speicherverbrauch	232 MB	5 MB	19 MB	95 MB	<i>k.A.</i>
CPU-Auslastung	66 %	2,6 %	27 %	0,01 %	<i>k.A.</i>
Mutanten / Sekunde	12	22,9	9,2	39	22,6
Tests / Sekunde	<i>k.A.</i>	<i>k.A.</i>	24,2	213,5	183,1

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Ergebnisse (Bedienung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Installation	3	1	1	1	1

---

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Ergebnisse (Bedienung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Installation	3	1	1	1	1
Dokumentation	3	2	3	3	5

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Bedienung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Installation	3	1	1	1	1
Dokumentation	3	2	3	3	5
Mutanten-Erzeugung	2	1	1	2	2

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Bedienung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Installation	3	1	1	1	1
Dokumentation	3	2	3	3	5
Mutanten-Erzeugung	2	1	1	2	2
Testausführung	2	1	1	2	2

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Bedienung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Installation	3	1	1	1	1
Dokumentation	3	2	3	3	5
Mutanten-Erzeugung	2	1	1	2	2
Testausführung	2	1	1	2	2
Ergebnisdarstellung	2	3	2	2	2

## 4. Leistungsvergleich der Mutationstestwerkzeuge

### Ergebnisse (Bedienung)

	<b>MuJava</b>	<b>Jumble</b>	<b>Pitest</b>	<b>Judy</b>	<b>Major</b>
Installation	3	1	1	1	1
Dokumentation	3	2	3	3	5
Mutanten-Erzeugung	2	1	1	2	2
Testausführung	2	1	1	2	2
Ergebnisdarstellung	2	3	2	2	2
Konfigurierbarkeit	5	2	2	3	1



---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Ergebnisse (Zusammenfassung)

- Mutation Score nicht immer nachvollziehbar

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Ergebnisse (Zusammenfassung)

- Mutation Score nicht immer nachvollziehbar
- Performante automatisierte Ausführung möglich

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Ergebnisse (Zusammenfassung)

- Mutation Score nicht immer nachvollziehbar
- Performante automatisierte Ausführung möglich
- Verbesserungspotenzial bei Dokumentation

---

## 4. Leistungsvergleich der Mutationstestwerkzeuge

---

### Ergebnisse (Zusammenfassung)

- Mutation Score nicht immer nachvollziehbar
- Performante automatisierte Ausführung möglich
- Verbesserungspotenzial bei Dokumentation
- → Empfehlung von Pitest und Jumble

---

## Inhalt

---

## Gliederung

1. Einleitung
2. Grundlagen
3. Vorstellung der Mutationstestwerkzeuge
4. Leistungsvergleich der Mutationstestwerkzeuge
- 5. Fazit**
  - **Zusammenfassung**
  - **Zukünftige Forschungsfragen**

---

## 5. Fazit

---

### Zusammenfassung

- Mutationstests lassen sich in der Praxis anwenden
- Pitest und Jumble eignen sich für die Anwendung

---

## 5. Fazit

---

### Zukünftige Forschungsfragen

- Bedienbarkeit verbessern
- Integration in Industrieprojekt

---

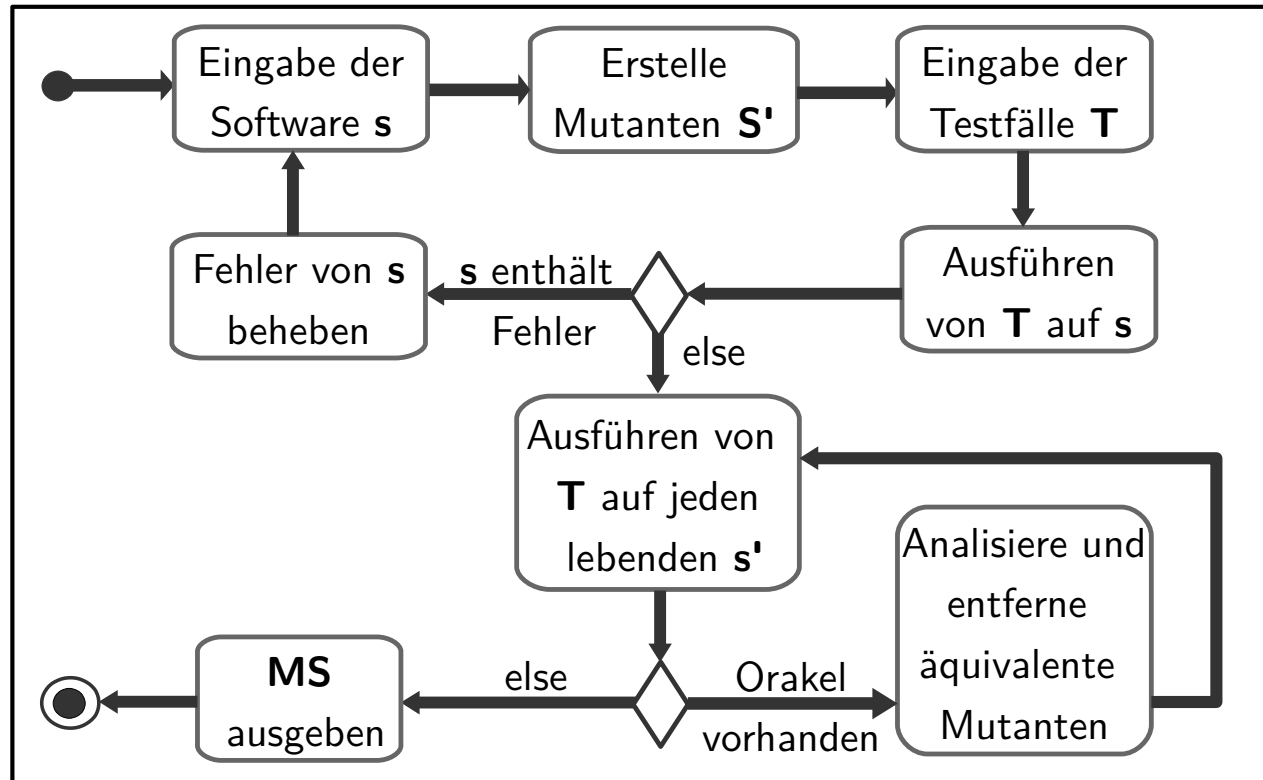
Diskussion

---

**Vielen Dank für die Aufmerksamkeit**



## Mutationstestprozess nach [OU01, S. 3, Fig. 1]



---

## Quellen

---

### Quellen

**[DLS78]** R. A. DeMillo, R. J. Lipton, F. G. Sayward: „Hints on test data selection: Help for the practicing programmer“, IEEE Computer 11 (4), S. 34–41, 1978.

**[OU01]** J. Offutt, R. H. Untch: „Mutation 2000: Uniting the orthogonal“, Mutation testing for the new century, Springer US, S. 34-44, 2001.