

Automatisches Generieren von Page Objects für GUI Tests

BED-Con 04.04.2014

Andreas Kutschera | Senior Software Developer | StoneOne AG



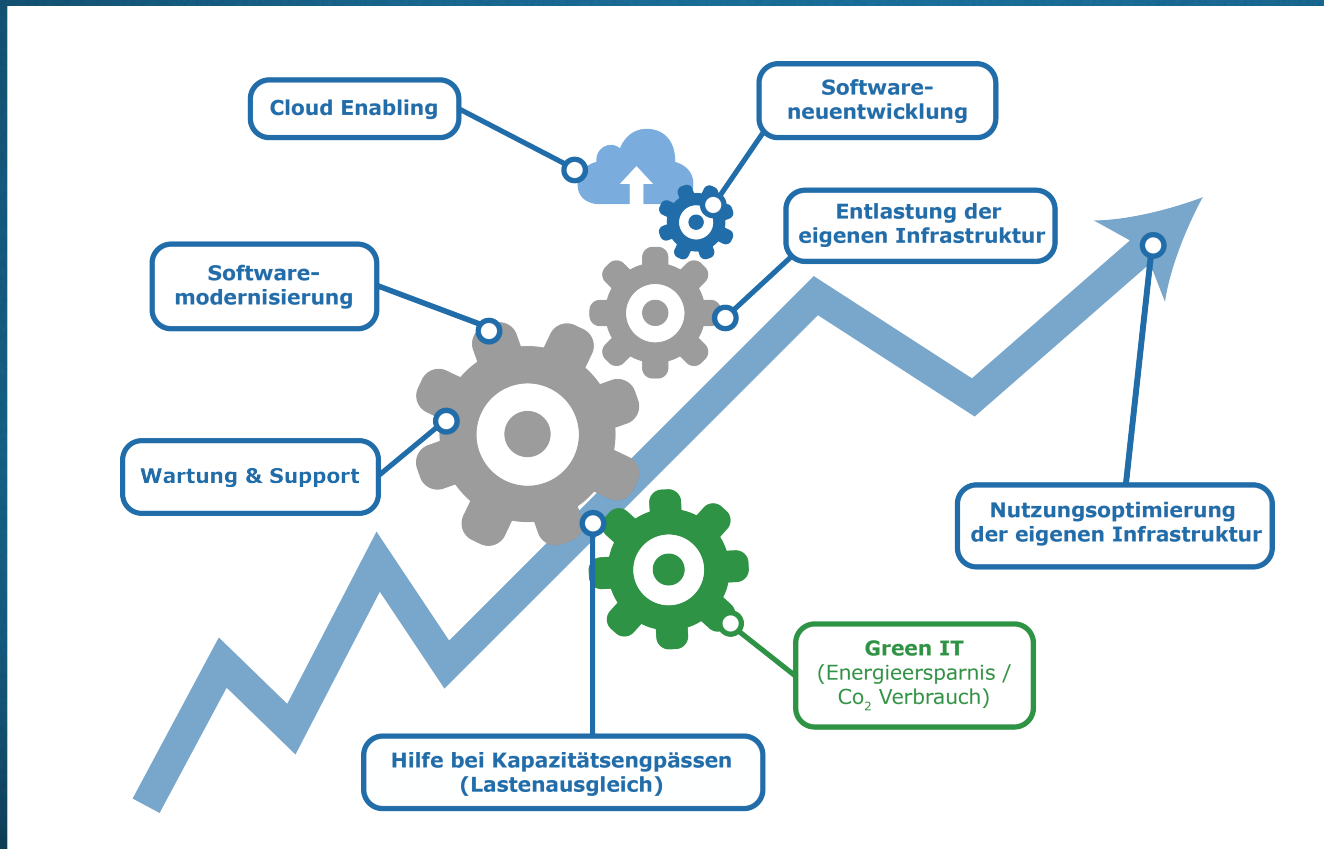
Über uns



- Gründung 2007 in Berlin
- Technologielieferant für Softwaremodernisierung (Legacy), Entwicklungsbeschleunigung und Cloud Enabling
- Technische Beratung und Joint Development
- Fachgremienarbeit u.a. im
 - BITKOM Arbeitskreis Cloud Computing & Outsourcing
 - SaaS-EcoSystem
 - SIBB Forum SaaS & Cloud Computing

Wo unterstützt StoneOne?

Zeit- und Kostenreduktion bei:



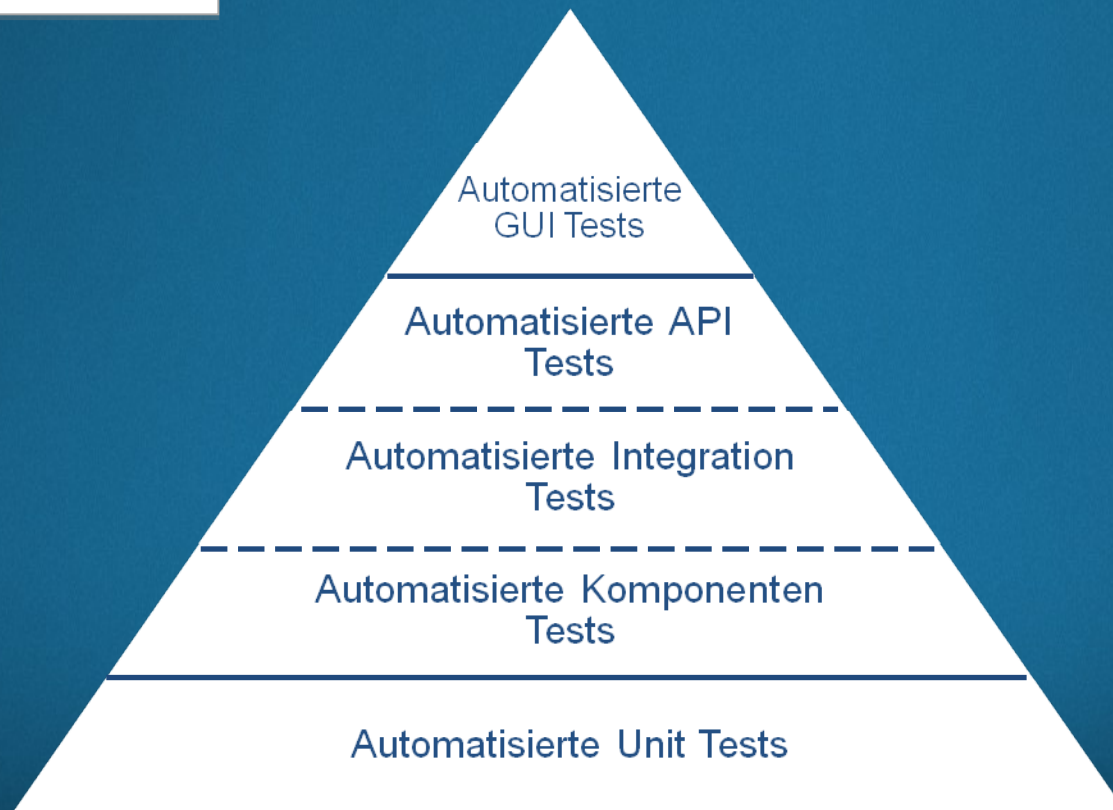
Zielsetzung

- Wozu überhaupt GUI-Tests?
- Aufbau von GUI-Tests
- Page Objects
- Grenzen der Generierung
- Vorteile der Generierung
- Positive Nebeneffekte

Testkategorien

(<http://www.watirmelon.com/tag/software-testing-pyramid/>)

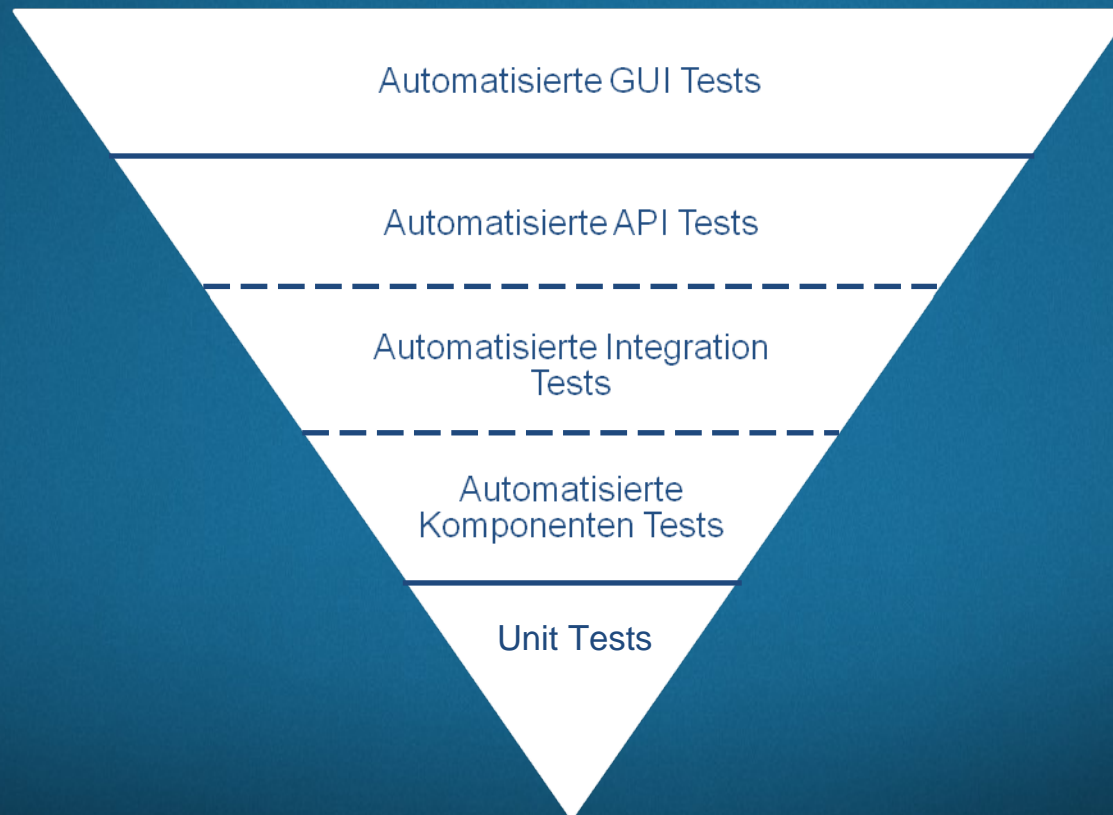
Anzahl Tests



Die andere Seite der Testkategorien

(<http://www.watirmelon.com/tag/software-testing-pyramid/>)

Aufwand pro Test



Webseiten aus Frameworks

- Framework mit eigener Beschreibungssprache
 - HTML kommt aus JSF, XML, JSP, GSP, Play usw.
- Framework (XML, JSF)

```
<h:commandButton value="save" action="saveData" />  
<g:submitButton name="save" value="save" />
```

- HTML

```
<input type="submit" value="save" />
```

GUI Tests mit Selenium

➤ HTML

```
<input type="submit" value="save" />
```

➤ Test (Java)

```
WebElement saveButton = driver.findElement(By.name  
                                             ("save"));
```

```
saveButton.click();
```


Page Objects als Abstraktion

- Schicht zwischen Test und GUI

```
<h:commandButton value="save" action="saveData" />
```

- Gekapselte Methodenaufrufe

```
public void clickSaveButton() {  
    WebElement saveButton = driver.findElement  
        (By.name("save"));  
    saveButton.click();  
}
```

- Jede Webseite entspricht einem Page Object

Page Controller als neue Abstraktion

- Neue Schicht zwischen Test und Page Object

 - Test → Logik

 - Page Controller → Seitenaufbau

 - Page Object → Interaktion mit der Seite

PageObjects werden generiert

- Aus Framework wird Code
- Methoden für jeden beliebigen Anwendungsfall
- Konventionen helfen bei der Generierung

Konvention über Konfiguration

- Namen zur Lesbarkeit nutzen
- Erhöhte Nutzung “derselben” Elemente ist erwünscht
- Einheitliche Dateistruktur der Webseiten

```
<h:commandButton id="save" value="save"  
  action="saveData" />
```


Grenzen der Generierung

- Seltene Fälle
- “Nicht-HTML”
- Wechselnde Frameworks
- Enger Zeithorizont

Vorteile der Generierung

- ✓ Code Completion ist dein Freund
- ✓ Iterativ erweiterbar
- ✓ Clean Code für Webseiten
- ✓ TDD geeignet

Positive Nebeneffekte

- Compile-Fehler sind gewollt
- i18n-Tests
- “Wartefehler” beim Seitenaufbau gehen gegen Null.
- Der Test ist der erste Nutzer der Webseiten

Fazit

- Generieren spart viel Zeit
- GUI-Tests schreiben wird einfach
- Die Tests werden stabiler
- Wir würden es wieder so machen

Danke
für Ihre
Aufmerksamkeit.

Fragen?

Ein Vortrag von:

Andreas Kutschera | Senior Software Developer | StoneOne AG
Andreas.Kutschera@stoneone.de

