

Smoke Tests mit dem Magento Test Automation Framework

Fabian Schmengler



TAF – was ist das eigentlich?

Test-Framework

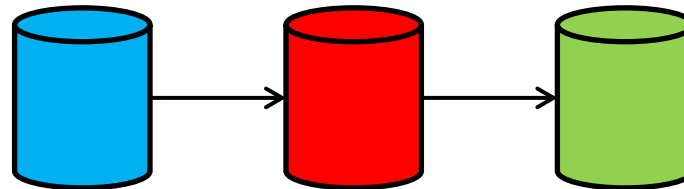
- Erweiterung der PHPUnit_Selenium Extension für Magento
- Abstraktion des User Interfaces

Test Suite

- Tests für Magento-Standard-Funktionalität
- 15-20 Stunden Gesamtlaufzeit
- Regressionstest
- Standard-Fixtures
- Helper für Standard-Aktionen

■ Bestandteile:


- **Testumgebung:** PHPUnit, Test Framework
- **Tests:** Test Cases, Form Data, UI Maps
- **Test-Server:** Selenium, Browser
- **System under Test (SUT):** Magento



■ Installations-Hilfe:

- <http://www.matthias-zeis.com/archiv/magento-test-automation-framework-taf>
- <http://www.dhmedia.com.au/blog/how-install-and-run-magento-test-automation-framework>

- **Abstraktion des User Interface: UI Maps**
 - YAML-Dateien
 - UI Map bildet interne (in den Tests genutzte) Namen von Seiten und Elementen auf URLs und Identifier ab
- **Form-basierte Testdaten**
 - YAML-Dateien
 - Werden in Formulare übertragen

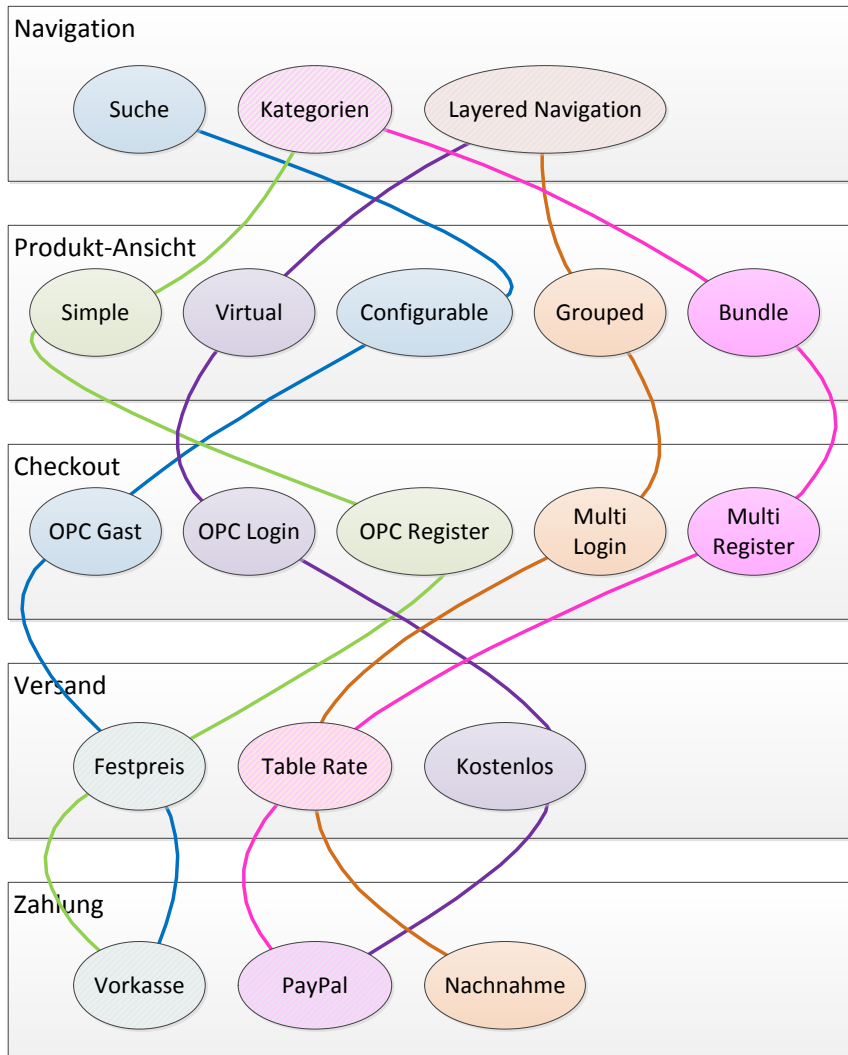
- 1567 Tests
- 15 Stunden Gesamtlaufzeit
- UI Maps für Standard Installation
- Test der PayPal-Integration
- Jeder Test legt benötigte Test-Daten zu Fuß im Backend an
-  **Failures** mit Standard-Installation CE 1.7.0.2

DEMO

- Ziel: Aufdecken von schwerwiegenden Bugs
- Kurze Laufzeit
- Lange Lebensdauer
- Hohe Pfadabdeckung von *Support Code*
- Kein Test von *Feature Code*
=> Unit Tests und Acceptance Tests

“ An automated test’s value is mostly unrelated to the specific purpose for which it was written. It’s the accidental things that count: the untargeted bugs that it finds. [Marick98] ”

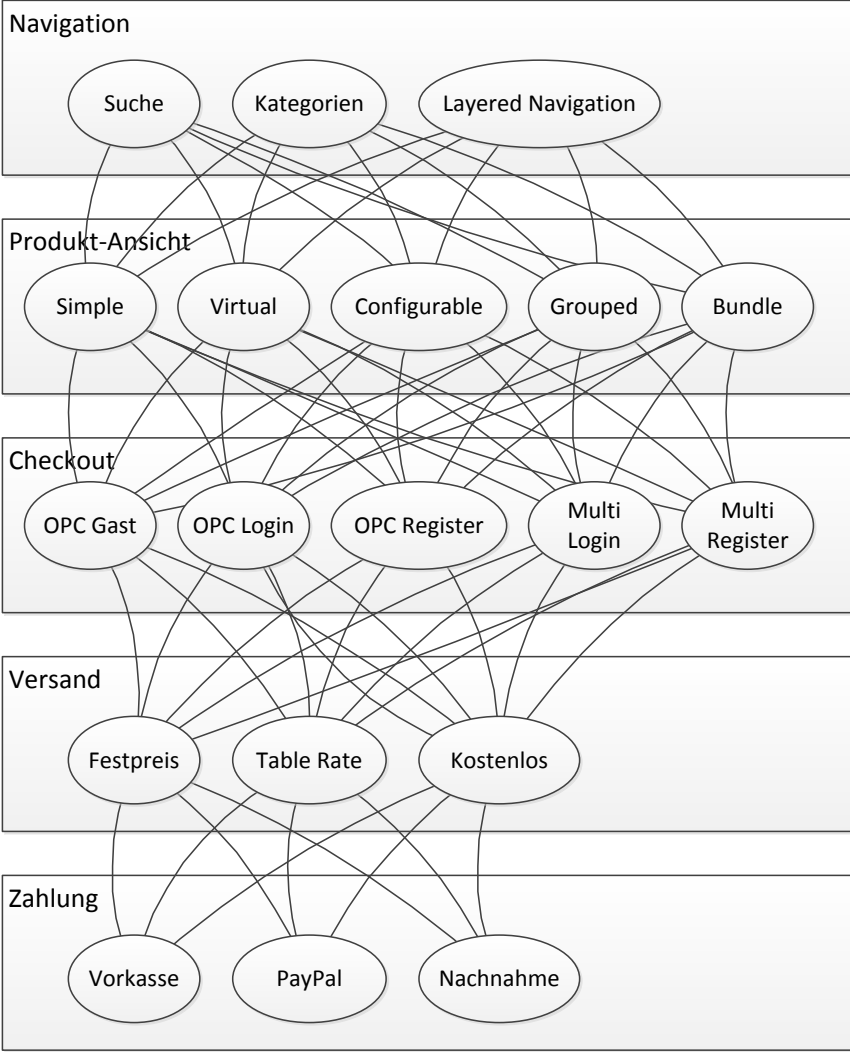
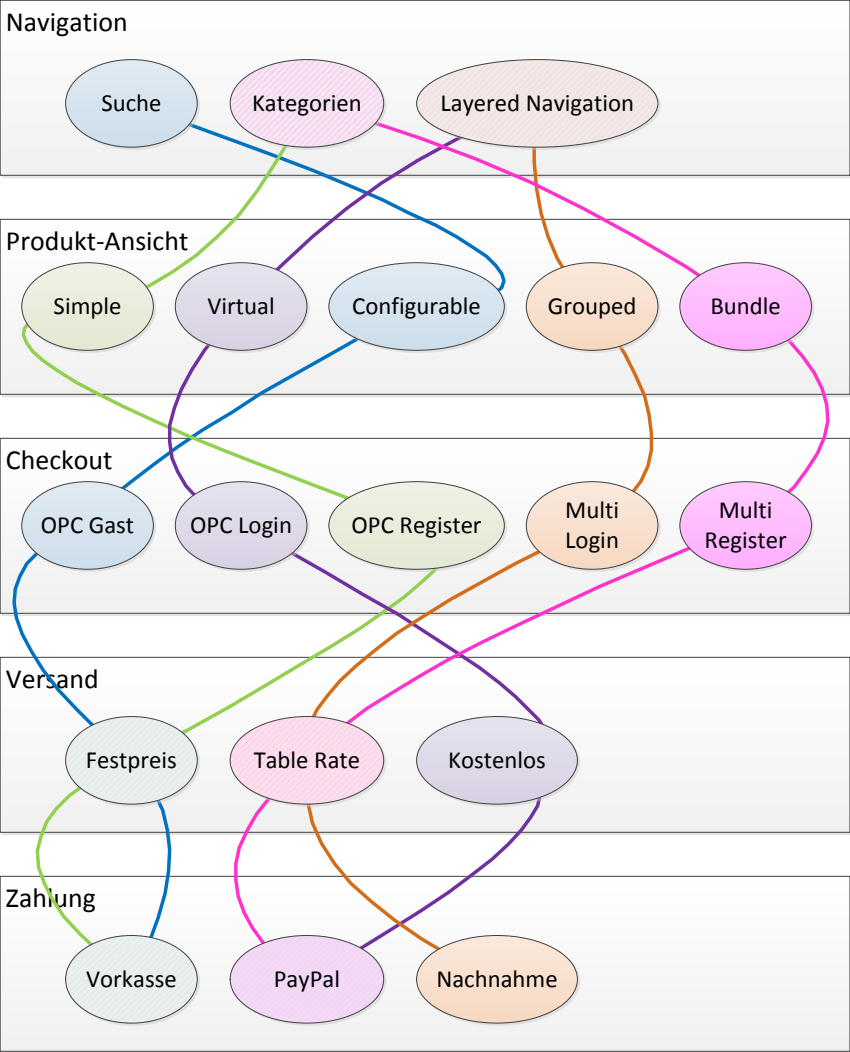
Smoke Test Suite: Testpfade (Beispiel)



- Systemkritische Funktionalität:
 - Finde Produkt(e)
 - Bestelle Produkt(e)
- Totalausfall einer Komponente wird schnell erkannt

← wenige Minuten!

Smoke Test Suite: Testpfade im Vergleich zur Standard Test Suite



- Aus Standard Test Suite:
 - UI Maps
 - Fixtures
 - Helpers
- Je Pfad ein Test Case mit voneinander abhängigen Tests

DEMO

- Test-Daten je Suite programmatisch anlegen
 - Daten via Selenium im Backend anlegen ist langsam
 - Backend hat für Smoke Tests niedrige Priorität
 - Daten müssen nicht für jeden Test einzeln angelegt werden

- Smoke Test Suite projektübergreifend einsetzen
 - Ggf. einzelne UI Maps überschreiben
 - Automatische Ausführung mit Continuous Integration

Danke für die Aufmerksamkeit

Fragen?