



PONTON
WE ARE THE 2 IN B2B

GUI Testautomatisierung – 50% Erstellungsaufwände
sparen, wenn man Anforderungen analysiert

Jörg Sievers, PONTON GmbH

43. TAV, 21. Februar 2019

Agenda

- Über PONTON und mich
- > 20 Jahre Testautomatisierung
- Problemstellungen
- Aufgabenstellung
- Kriterienkatalog
- Vorteile der neuen Lösung
- Zusammenfassung

Jörg Sievers – Stationen (Auswahl)



- ❑ 1986 Start ins Berufsleben
- ❑ 1990 Mailbox-Systeme (ZERBERUS®, FIDO®)
- ❑ 1991 Terracom Computersysteme GmbH
 - NeXT® Computer
- ❑ 1991 DX Communication & Systemservice
 - Beratung/Verkauf Kommunikationslösungen
 - Schulung/Verkauf Arztpraxissoftware TurboMed®
- ❑ 1998 Star Division (StarOffice™)
 - Übernahme 1999 durch Sun Microsystems GmbH
 - Testautomatisierer: 15 Sprachen, 5 Plattformen
 - Sun™ WebOffice – browserbasiertes Office mit Dokumentkonverter
- ❑ seit 2010 PONTON GMBH

Über PONTON GmbH

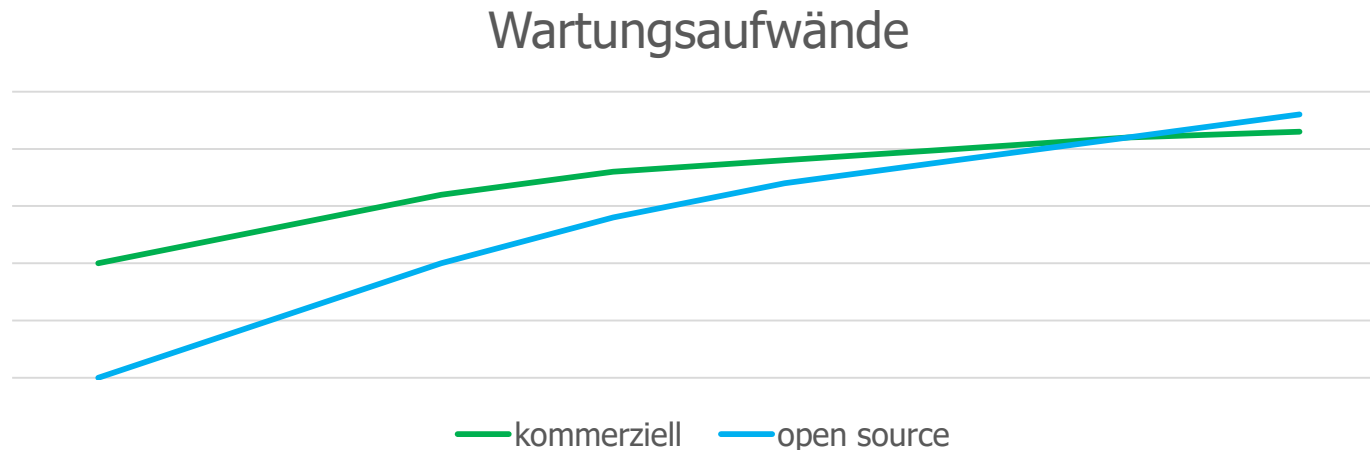
- ❑ In Hamburg (Winterhude) ansässig, gegründet in 2001, >50 Mitarbeiter
- ❑ **Ponton X/P**
 - Nachrichten-basierte B2B Integration mit AS1 / AS2 / AS4 / ebXML
 - Über 350 Installation in Europa und Nord Amerika
- ❑ Energiewirtschaftliche Projektlösungen für den 24/7 Betrieb:
 - **EFETnet** (European Federation of Energy Traders)
 - **eCM** (electronic Confirmation Matching, > 100 Nutzer, seit 2004)
 - **eRR** (Regulatorisches Reporting, > 700 Nutzer, seit 2012
(Dodd-Frank, EMIR, REMIT, MAR, MiFID II, StromVV, FinFrag))
 - **SMSS** – Spot Market Settlement System: Finanzielle und physische Abwicklung von Spot Markt Transaktionen der EEX Börsen Gruppe
 - **Enerchain, Gridchain** – Diverse Blockchain-basierte Proof of Concepts
 - **Project EDA** – Energiewirtschaftliche Marktkommunikation in Österreich

> 20 Jahre (GUI) Testautomatisierung...

- Nach wie vor Probleme mit der systemübergreifenden, deterministischen Ausführung
- Probleme mit Umgebungsbereitstellungen
- Welcher Ansatz ist bei komplexen Systemen bzgl. der Testdaten zu wählen?
- Spezialwissen nicht nur vom Testtool selbst, sondern oftmals auch von den Technologien der Anwendung (wollten wir nicht Black-Box testen?)
- Testautomation Patterns → <https://testautomationpatterns.org> (offenes Wiki-Projekt) ist ein guter Ansatz Testautomatisierung und deren Herausforderungen zu systematisieren (Seretta Gamba, Dorothy Graham)
- Management: RoI, Sinn und Zweck werden nach wie vor hinterfragt bzw. andere Ziele sind wichtiger oder die Zeit fehlt → Der Nutzen nach wie vor nicht klar herausgestellt
- ...

Problem Wartbarkeit

- ❑ Browserupdate → Test Anpassungen
- ❑ Inhalte „anders darstellen“, keine funktionale Änderung → Test Anpassungen
- ❑ Neue Funktionen → tlw. Anpassungen pro Browser (IE, Edge, Chrome, Firefox, ...)
- ❑ Open Source Software != „umsonst“ und kann u.U. teu[r]er werden



Weitere Problemstellungen

- ❑ Ausführung lokal und auf CIS mit sehr unterschiedlichen Ergebnissen (nicht deterministisch)
- ❑ AJAX-Behandlung (dynamische Seiten) → fluent-wait()-Funktionen
- ❑ GUI-Testtool nutzt CSS oder XPATH → Änderungen von optischen oder nicht-funktionalen Teilen bedeutet Test-Anpassung, da vom Code abhängig
- ❑ Anwenderszenarien automatisieren (über mehrere Seiten und Medien hinweg) sehr kompliziert und zeitaufwendig

Aufwandsschätzungen schlecht!
Product Owner unzufrieden!

Aufgabenstellung

Finde ein Tool / Framework, das funktionale Blackbox Tests auf der Web Oberfläche durchführt und leistungsfähiger und zeitsparender als das Open Source Framework Selenium ist.

Kriterienkatalog

- ❑ [Test-] Entwickler fragen, wo die Knackpunkte bei der derzeitigen Lösung sind
- ❑ Erfüllungsgrad und Gewichtung von Kriterien in eine Matrix

Anforderungskriterium	Anmerkung	Erfüllungsgrad				Gewichtung
		Tool A	Tool B	Tool C	Tool D	
Möglichkeit Verw. Testfälle (Test Suiten)		7	1	8	8	5
Unterstützung AJAX		8	1	10	10	7
Robustheit / Stabilität der Tests		10	1	5	8	8
Unabhängigkeit Code-Struktur der Anw.	Black-Box-Ansatz	10	1	10	10	5
...						
Browserkompatibilität	Firefox, IE, Chrome	10	7	10	10	5
Summe (Erfüllungsgrad * Gewichtung)		1750	1043	1387	1676	

Vorteile der neuen Lösung I

- ❑ Abstraktionsschicht ist nicht dieselbe wie die Sprache der Webanwendung (kein CSS/XPATH, sondern eigene Zugriffsschicht (Accessor)) → Test anwenderorientiert, nicht source-code-orientiert
- ❑ Keine *fluent wait()*-Funktionen mehr notwendig → Framework bietet das selbst
- ❑ Anwender-Szenarien mit Up- & Download-, DB-, Microsoft Office Dokument-Operationen samt Überprüfungen möglich → Komplexe Anwenderszenarien mit vertretbarem Aufwand
- ❑ Zentrale Deklaration und Nutzung der Accesoren anstatt Pflege via *PageObject*-Pattern in einzelnen „Objekten“ (Testansteuerungs-Repräsentation der Seiten) → reduziert Wartung

Vorteile der neuen Lösung II

- ❑ Szenarien können mit Asserts schon bei der Aufzeichnung versehen werden
→ Fachanwender können Test-Szenarien im Rohformat selber erstellen
- ❑ Neue Web-Technologien (Shadow DOM, Custom HTML Elements, Polymer) werden unterstützt
→ Zukunftsfähigkeit
- ❑ Test-Skripte müssen nicht an unterschiedliche Browser angepasst werden
→ Wartbarkeit
- ❑ Testausführung ist lokal (macOS, Windows) identisch mit CIS
→ Zeitreduktion
- ❑ Komplexe, nachladbare Elemente ohne ID's können mittels RegEx auf der Anwendungsseite nun getestet werden
- ❑ Akzeptanz durch die [Test-] Entwickler deutlich höher

Zusammenfassung

- ❑ Aufgabenstellung wurde erfüllt
- ❑ Anforderungen der [Test-] Entwickler wurden sehr gut berücksichtigt
- ❑ Lt. Product Owner des PoC-Projektes werden **50% weniger Aufwände für die Testerstellung** aufgewendet und zudem erfreut sich die neue Lösung wesentlich breiterer Akzeptanz
- ❑ Drei Jahre nach dem PoC benutzen drei weitere Projekte die gewählte Tool-Lösung, da deren Anforderungen sehr ähnlich sind und tlw. sogar noch erweitert werden konnten. Bspw. „alte“ GUI und REST-basierte GUI gleichzeitig zur Verifikation benutzen

Erstellt:

PONTON GmbH

Dorotheenstraße 64

22301 Hamburg

Deutschland

Kontakt:

Jörg Sievers

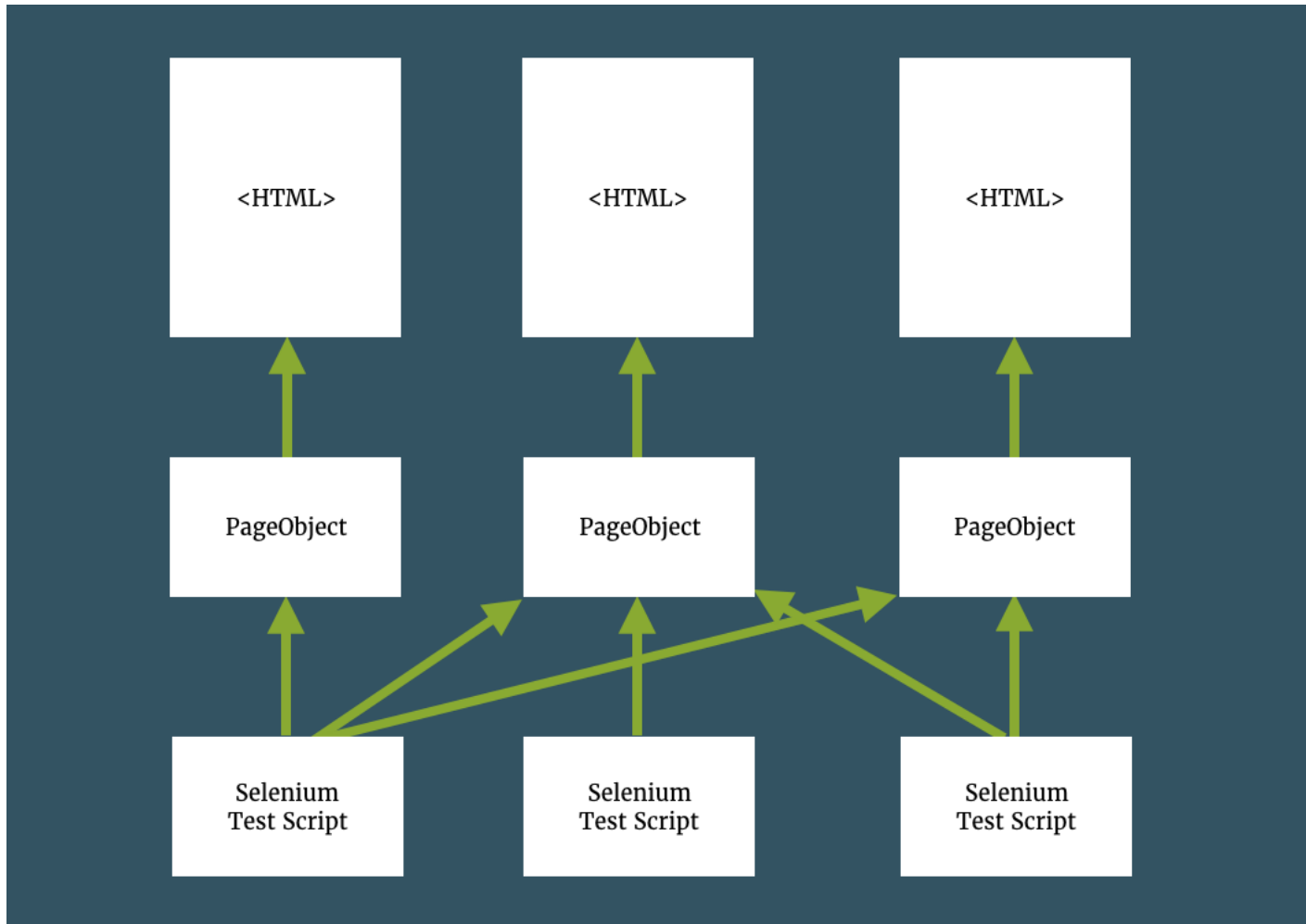
Tel.: +49 40 866 275 259

Mobil: +49 170 63 82 81 0

Email: sievers@ponton.de

Zusatzfolien

PageObject Pattern



<https://www.pluralsight.com/guides/getting-started-with-page-object-pattern-for-your-selenium-tests>

Zentrales Accessor Repository

```
❑ _include("../declaration/declaration.sah");
```

```
(...)  
var $_CHECKBOX_EXECUTION_CONTAINER_EXECUTION = _checkbox('executionContainer:execution');  
var $_CHECKBOX_FCA = _checkbox('FCAPositionReport');  
var $_CHECKBOX_FCA_PLUS = _checkbox('FCAExtendedPositionReport');  
var $_CHECKBOX_MIFID_POSITION_REPORTING = _checkbox('mifidCheckboxDiv:mifidPositionReporting');  
var $_CHECKBOX_MIFID_II_CHECKBOX_POSITION_REPORTING =  
_checkbox('mifidCheckboxDiv:mifidPositionReporting');  
var $_CHECKBOX_REPORT_BOTH_SIDES_CONTAINER_REPORT_BOTH_SIDES =  
_checkbox('reportBothSidesContainer:reportBothSides');  
var $_DIV_FEATURE_STATE_ICON_FEATURE_ACTIVE = _div('featureStateIcon featureActive');  
var $_FILE_FILE_INPUT = _file('fileInput');  
(...)
```