BERNER & MATTNER
AN ASSYSTEM COMPANY

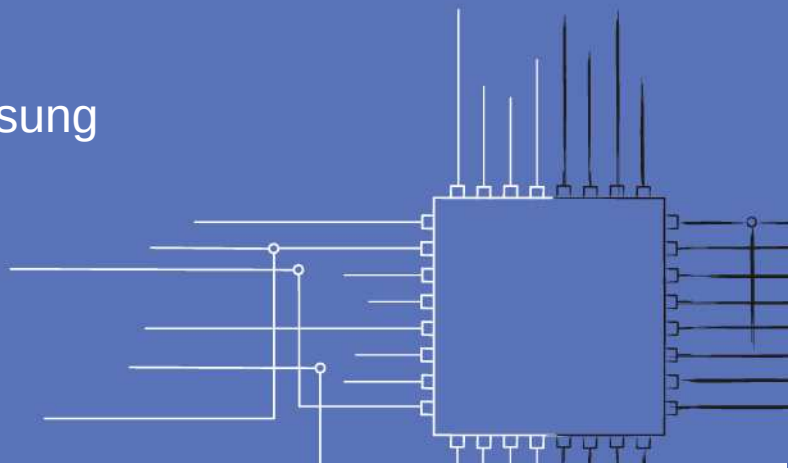
Test graphischer Benutzeroberflächen mit der Klassifikationsbaum-Methode

Am Beispiel von Webanwendungen

21.11.2013 | Jirka Nasarek

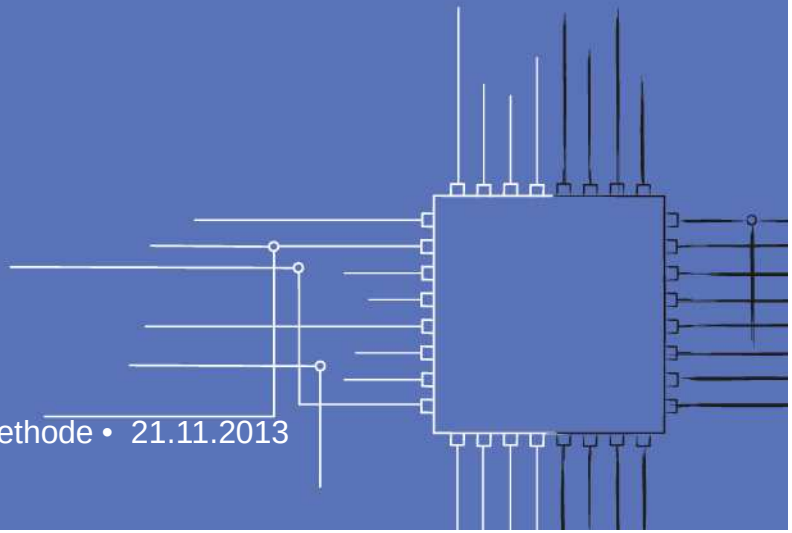
Gliederung

- Motivation: GUI- und Web Testing
- Mögliche Herangehensweisen
- Exkurs: CTE XL, Selenium
- GUI Testing Framework
- Demo
- Zusammenfassung



BERNER & MATTNER
AN ASSYSTEM COMPANY

Motivation: GUI- und Webtesting



Warum Benutzeroberflächen testen?

Software aus Endanwendersicht

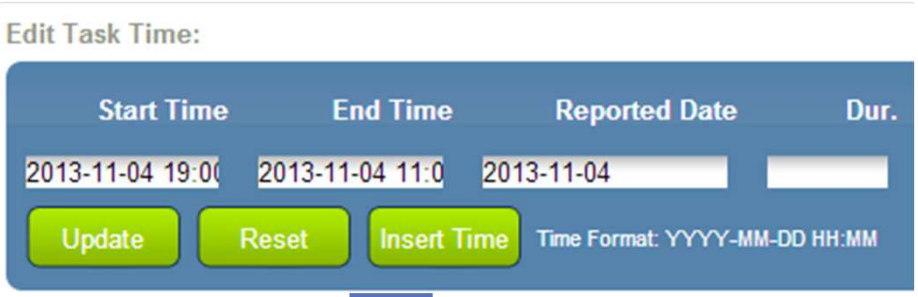
- Produkt wird unter Einsatzbedingungen geprüft
- Ideal: Tester != Programmierer
- Fehlerhaftes Wechselspiel von Eingaben und Funktionen
- Zugriff auf Backend/Quellcode nicht immer möglich

Umfang, der nicht durch andere Tests abgedeckt wird

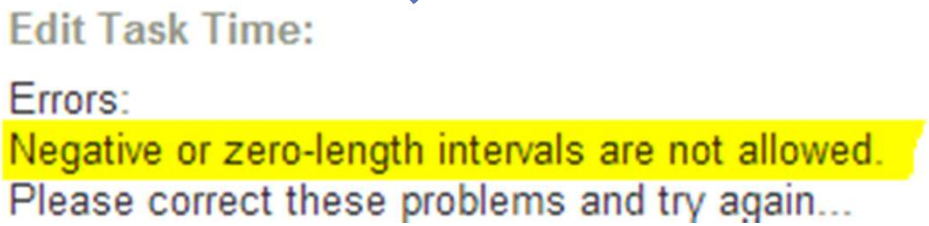
- Unittests und Schnittstellentests eher auf anderer Ebene
- Entkopplung Eingabe Frontend → Verarbeitung Backend
- Validierung von Eingaben im Frontend → eigene Fehlerquelle

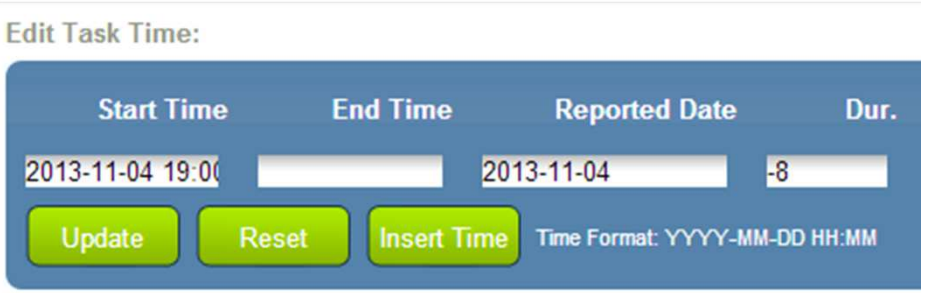
Warum Benutzeroberflächen testen?

Beispiel: Gleiches Vorhaben, verschiedene Resultate:

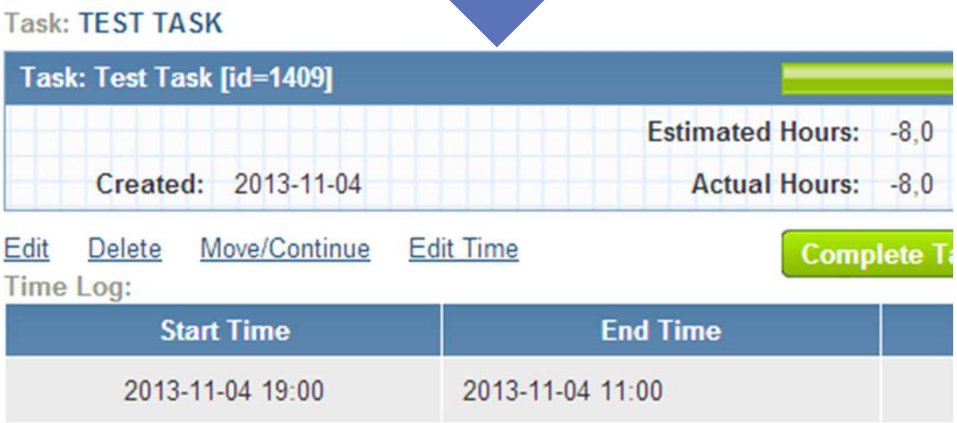


Fehlermeldung





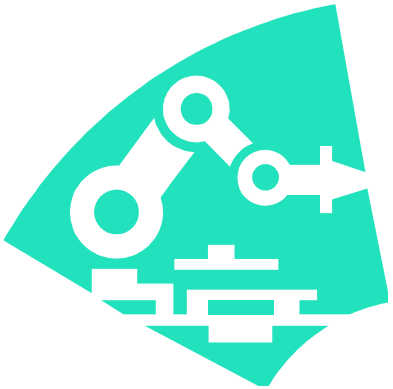
Eingabe akzeptiert



Warum Benutzeroberflächen **automatisiert** testen?

Vorteile Automatisierung

- Kombinatorische Vielfalt
- Geschwindigkeit der Testwiedergabe
- Monotones Testen selbst fehlerträchtig
- Protokollierung
- Wiederholbarkeit bei Regressionstests
- Skripte für ähnliche oder gleiche Vorgänge
- Robustheitstest



Systematischer Test

Was

- Methodisches Testvorgehen
- Geplant, geordnet
- Ergänzung zu Ad Hoc, explorativem oder Zufallstest

Warum

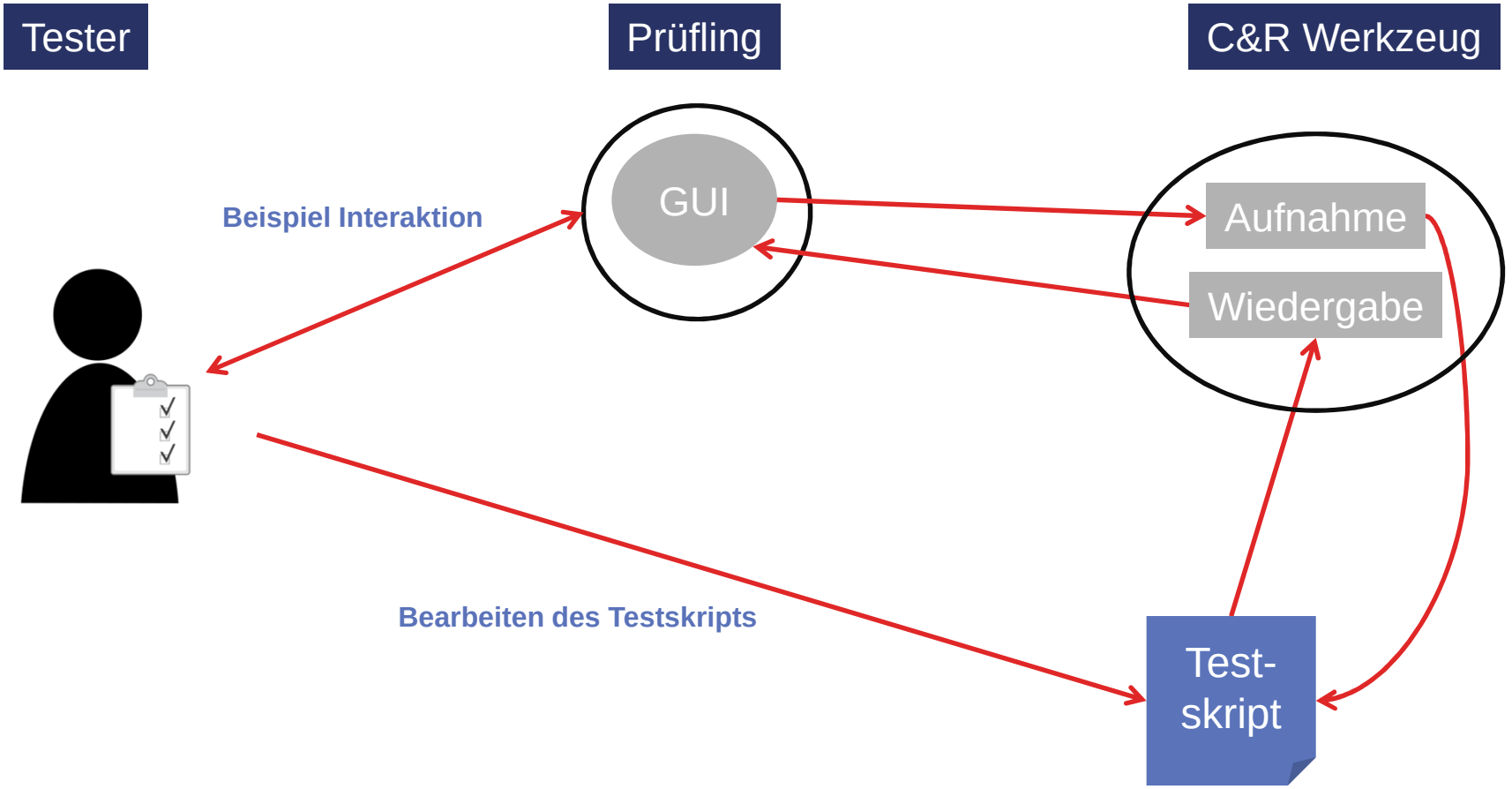
- Formale Beschreibung der Testfälle
- Zeigen, dass der Prüfling der Spezifikation entspricht
- Vielfalt an Eingabemöglichkeit oft groß → Reduktion
- Geschickte Auswahl, um möglichst viel abzudecken

Capture Replay und Verwandte

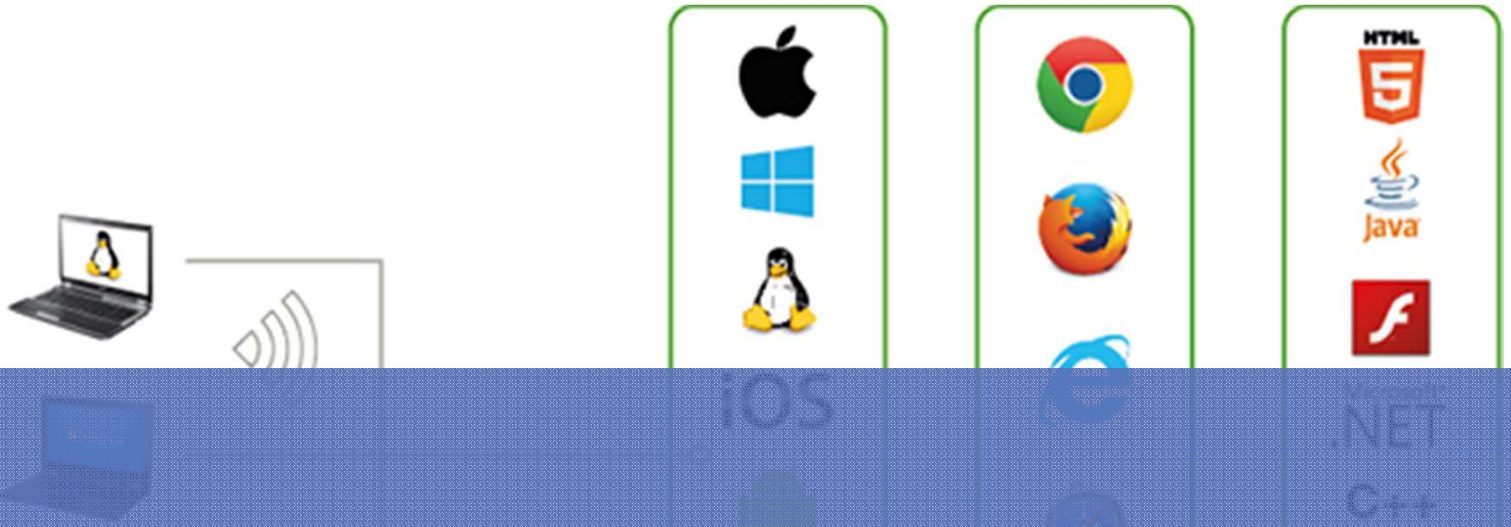
Vorgehen in der Praxis:

- Kein extra GUI Test
- Manuelle Stichproben
- TestRoboter, Page Object Pattern, **einfaches C&R**
  - Beispiele: SWTBot, Selenium WebDriver …
- C&R Test Entwicklungs-Umgebungen
  - Beispiele: QFTest, Test Complete, **EggPlant** …
  - Data Driven Testing
  - Test ObjectMaps

Einfaches C&R



Beispiel eggPlant

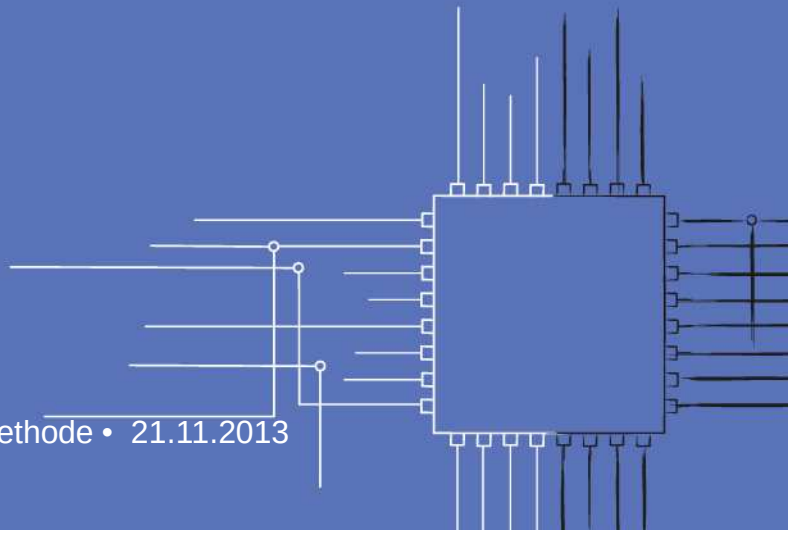


Werkzeuge für die Automatisierung von Oberflächen Tests

- Vorgehen, ähnlich C&R möglich (aber… Besonderheiten)
- Zugriff auf Prüfling per VNC
- Plattformunabhängig (solange VNC Server ex.)
- Schwerpunkt auf Texterkennung und Bildverarbeitung
- Scriptsprache SenseTalk (o. a. über XML-RPC)

BERNER & MATTNER
AN ASSYSTEM COMPANY

Unser Ansatz:
GUI Testing Framework, CTE XL, Selenium



GUI Testing Framework

Ziele

- Identifizieren sich wiederholender Phasen und Tätigkeiten beim GUI Test
- Vereinheitlichung des Testprozesses
- Test-Plattform
- Austauschbarkeit der für die einzelnen Phasen verwendeten Werkzeuge
- Strukturmuster Adapter als Basis
- Weitgehend Technologie-unabhängige Beschreibung der Adapter-Artefakte

Exkurs: Selenium

Hintergrund

- Verschiedene Werkzeuge
- Hier: WebDriver

Aufgaben

- Instrumentierung eines Browsers
- Ansteuern und aktivieren von Elementen einer Webseite
- Adapter A1 und A4 des Frameworks

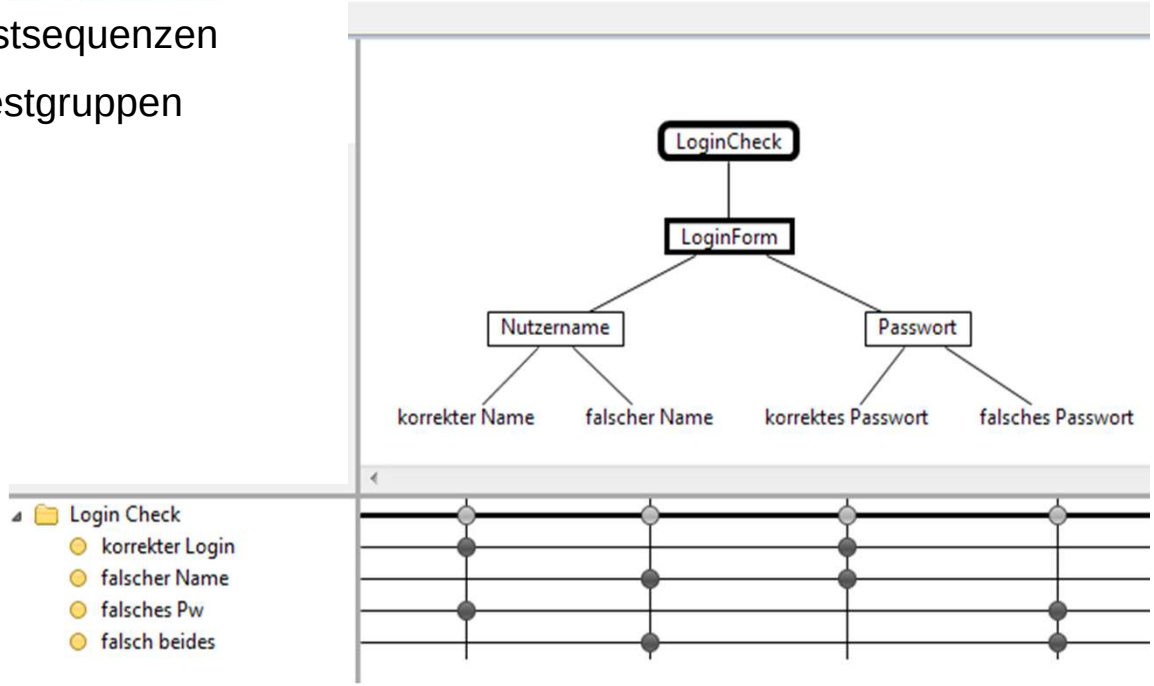
Exkurs: CTE XL Professional

Aufgaben

- Systematische Testfallermittlung
- Zerlegung des Eingabedatenraums
- Bei *abstrakten* Testfällen Bildung von Äquivalenzklassen
- Kombinationsregeln f. Testfallgenerierung
- Für Folgen von Testschritten: Testsequenzen
- Für Sammlung von Testfällen: Testgruppen

Grundlagen Methodik

- Klassifikationsbaummethode
  - Komposition: ”besteht-aus”-Beziehungen
  - Klassifikation: Für den Test relevante Aspekte
  - Klasse: mögliche Eingaben

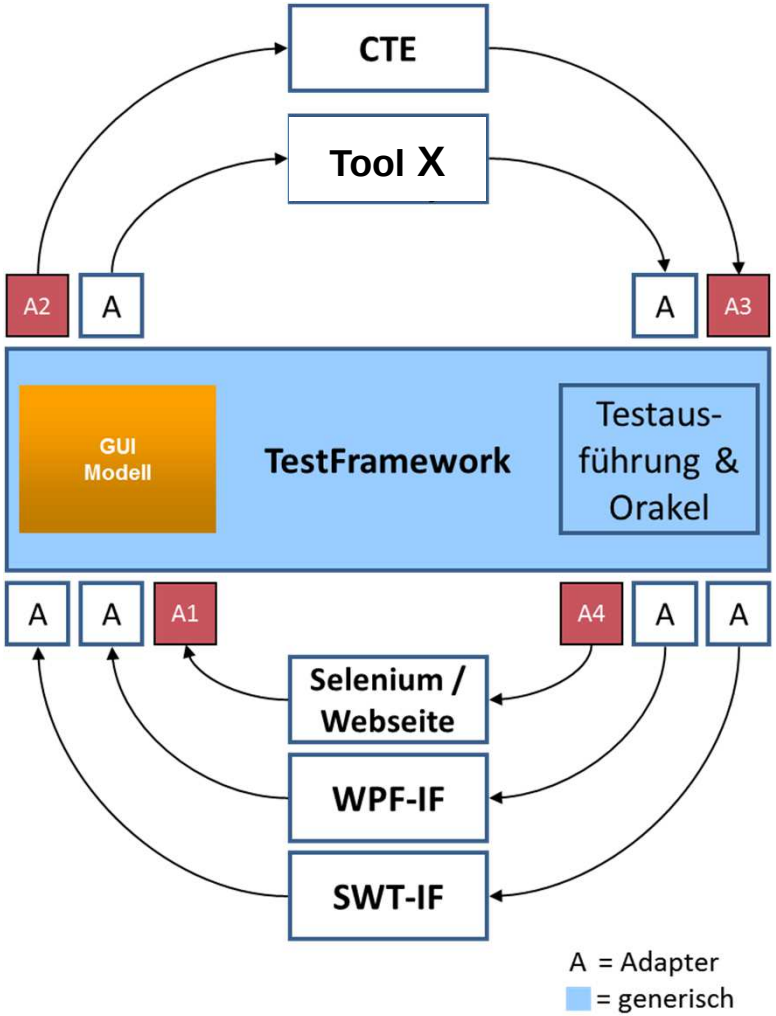


BERNER & MATTNER
AN ASSYSTEM COMPANY

GUI Testing Framework

2. Import für den Testfallentwurf

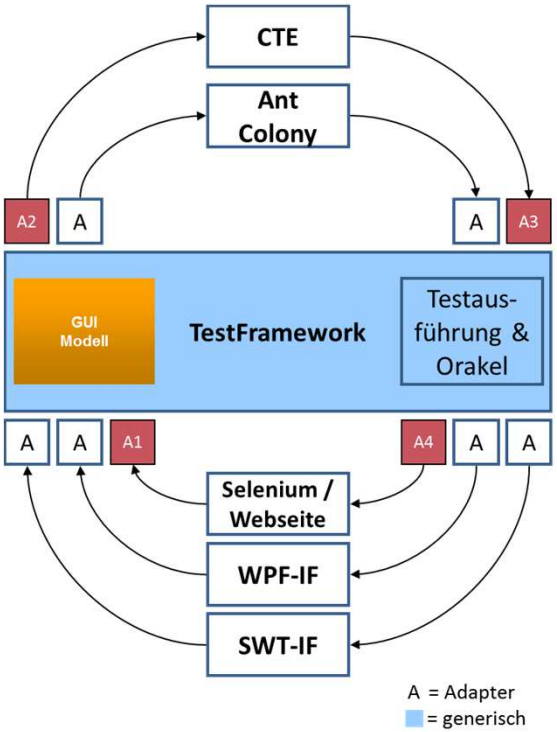
3. Export Testfälle und Orakelanweisungen



1. Auslesen des GUI

4. Ausführung und Reporting

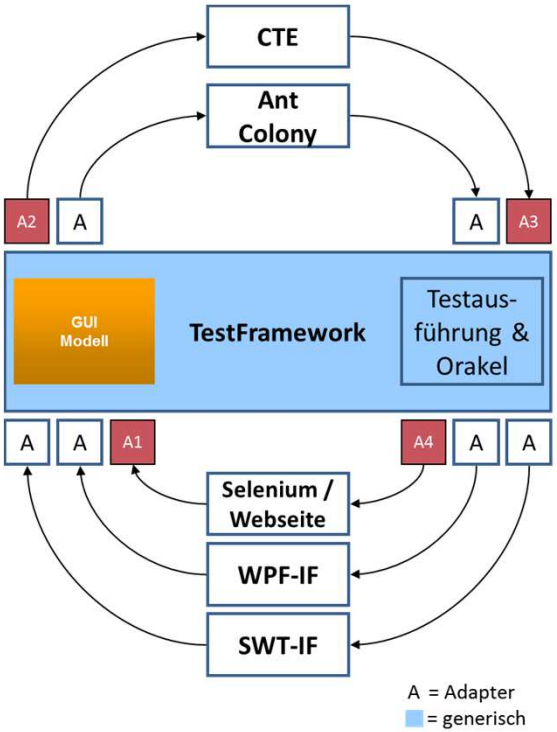
A1: Auslesen des GUI



Aufgaben A1 (speziell Selenium)

- Vorbereiten und Steuern des Prüflings
- Umwandeln der Zustände der Benutzerschnittstelle in GUI Modell
- Parsen der Seite, wie sie im Browser dargestellt wird
- Identifizieren und Kategorisieren von Elementen
  - Eingabe-Elemente (mit mögliche Wertebereiche)
  - Ausgabe-Elemente (für Orakel)
  - Aktionselemente (registrierte Eventhandler)
- Erzeugen der XML Beschreibung des Modells
- Default-Aktionen die Plattform bereitstellt

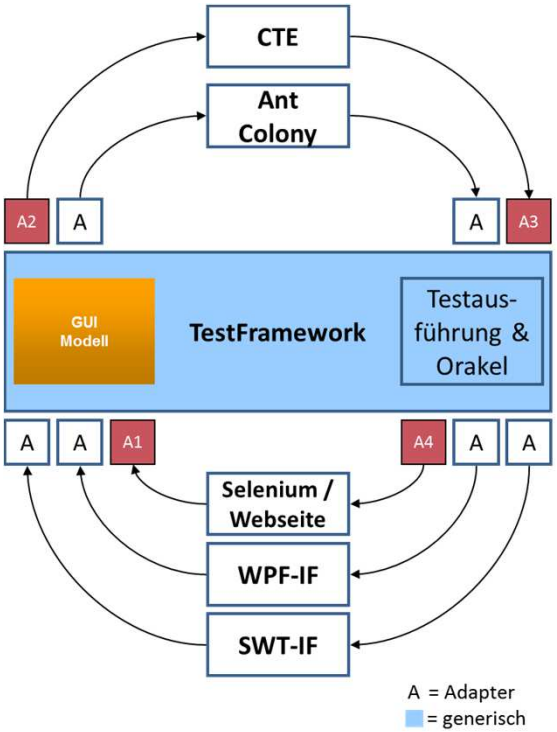
A2: Integrieren des GUI Modells für den Testfallentwurf



Aufgaben A2 (speziell CTE)

- Darstellung des GUI Modells in View
- Metainformation zu Element als Attribute
- Kennzeichnen von besonderen Elementen
- Auswahl für die Verwendung im Klassifikationsbaum
- Darstellung der Wertebereiche, registrierten Aktionen
- Einfügen manueller Aktionen
- Erzeugen von Setup- und Teardown Anweisungen
- Festlegen der Reihenfolge durch Attribute oder Testsequenzen

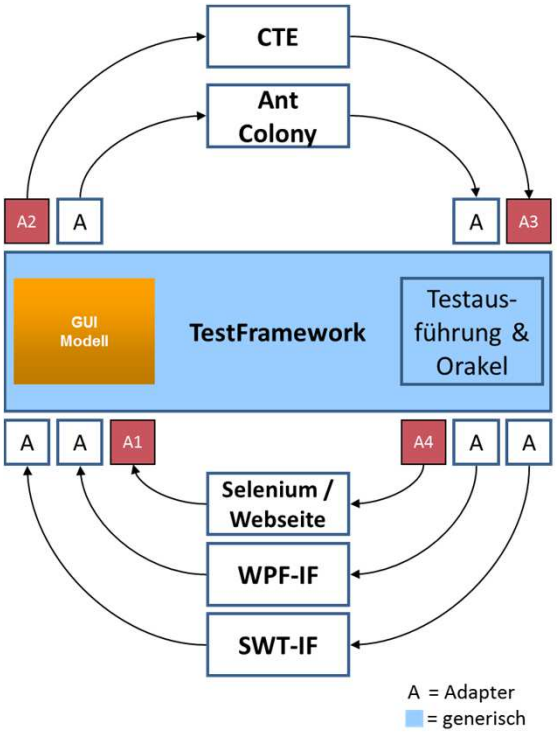
A3:Erzeugen des Testfallmodells



Aufgaben A3 (speziell CTE)

- Konvertierung Testfälle und Testsequenzen der Testmatrix
- Gesetzte Markierung:
  - Wert für Eingabe-Element (InsertCmd)
  - Aktion für Aktions-Element (PerformCmd)
  - Prüfender Wert für Ausgabe-Element (OracleCmd)
- Erzeugen einer technologie-unabhängigen Beschreibung der Testfälle
- Testfall hier: Sequenz von Kommandos

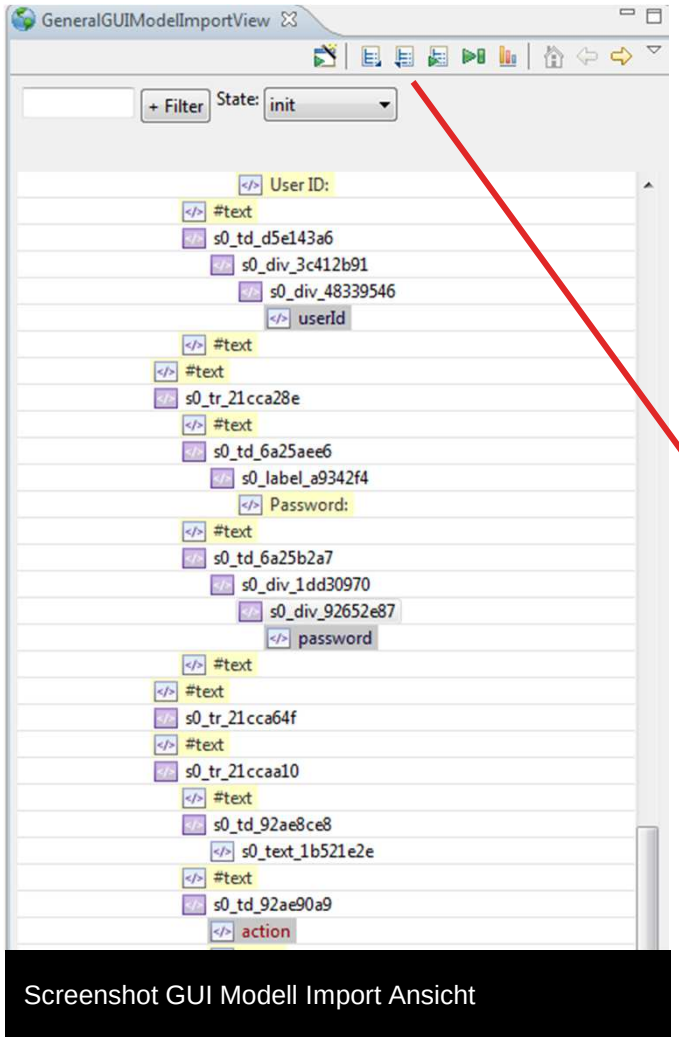
A4: Ausführen der Testfälle



Aufgaben A4 (speziell Selenium)

- Bereitstellen des Testkontext
- Starten des Browsers zum Zugriff auf Prüfling
- Identifizierung der konkreten Elemente und Aktionen (d. A1)
- Ausführen der Testfälle
  - Vorbedingung
  - Nachbedingungen herstellen
- Logging und Auswertung
- Screenshot nach Ausführung des Testfalls

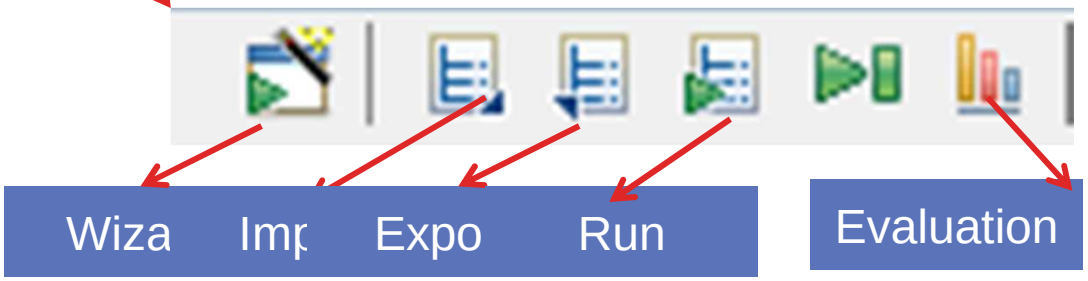
GUI Model Plugin für CTE XL Professional



Screenshot GUI Modell Import Ansicht

Aufgaben

- Schema für GUI Modell und GUI Testfallmodell
- Darstellung des GUI Modells in View
- Kennzeichnung I/O u. Aktions-Elemente
- Integration Elemente GUI Modell in CTE Modell
- Darstellen von Eingabe-Wertebereichen
- Export von CTE ins GUI Testfallmodell

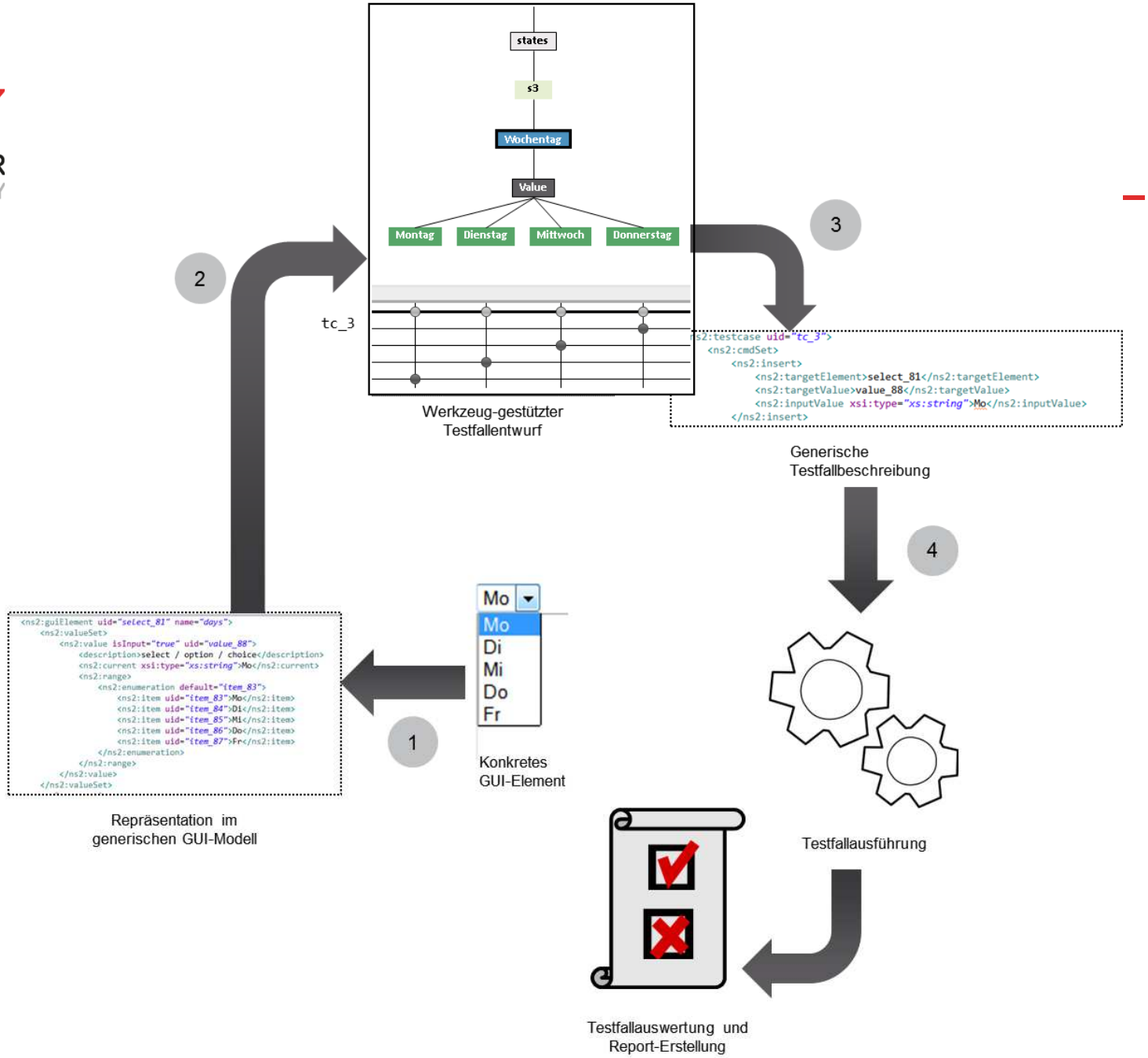


BERNER & MATTNER
AN ASSYSTEM COMPANY

Demo

tav_mid.wmv

BERNER & MATTNER
AN ASSYSTEM COMPANY



Bonus Folien

Nicht in der Demo…

Zustände

- Ein GUI Modellenthält i.d.R. mehrere Zustände
- Alle Elemente eines Zustands als Baum unter Zustand dargestellt

Setup-, Teardown-Mechanismen

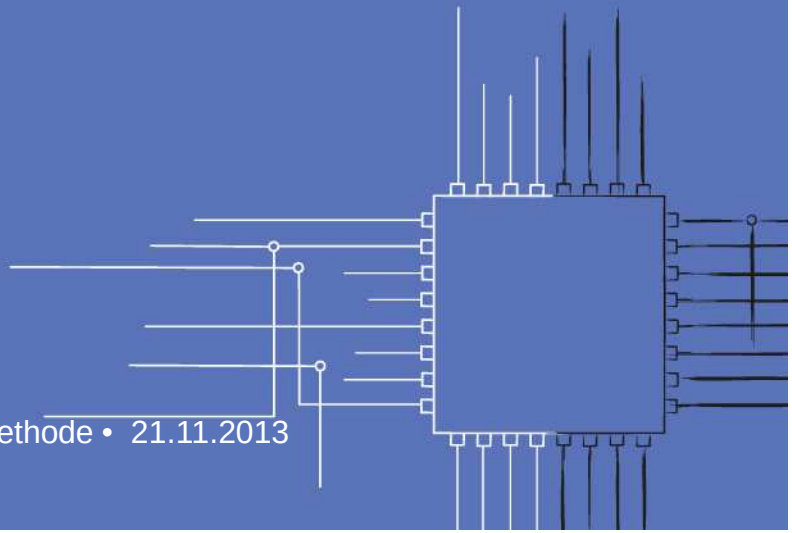
- Durch Pseudotestsequenzen
- Vor- und nach jedem Testfall ausgeführt

Testsequenzen

- Menge von Aktionen, die in Testschritten ausgeführt werden sollen
- Legt Reihenfolge der Aktionen fest

BERNER & MATTNER
AN ASSYSTEM COMPANY

Zusammenfassung & Ausblick



BERNER & MATTNER
AN ASSYSTEM COMPANY

Zusammenfassung & Ausblick

Allgemeines Framework mit Konzentration auf Abläufen

- Spezielle Adapter mit Fokus auf Plattform-spezifische Eigenheiten
- Verschiedene GUIs ein vereinheitlichter Ablauf
- Fertige Schemata und Formatbeschreibungen
- Referenzimplementation
- Strukturierung des Testprozesses
- Steigerung der Flexibilität durch Austauschbarkeit der Adapter
- Wartbarkeit verbessern

In Zukunft

- Adapter für weitere Plattformen und Testverfahren
- Evaluierung einer Kombination aus Klassifikationsbäumen und Zustandsautomaten

BERNER & MATTNER
AN ASSYSTEM COMPANY

Quellen

- Ostrand, Thomas; Anodide, Aaron; Foster, Herbert / Goradia, Tarak: A visual test development environment for GUI systems. In: *SIGSOFT Softw. Eng. Notes* 23 (1998), S. 82-92.
- Memon, Atif M. / Xie, Qing: Studying the Fault-Detection Effectiveness of GUI Test Cases for Rapidly Evolving Software. In: IEEE Trans. Softw. Eng. 31 (2005), S. 884-896
- Selenium: http://docs.seleniumhq.org/projects/
- eggPlant: http://www.testplant.com/eggplant/testing-tools/eggplant-developer/

BERNER & MATTNER
AN ASSYSTEM COMPANY

Beispiel Page Object Pattern

```
public class LoginPageToPJOverview {
    private final WebDriver driver;

    public LoginPageToPJOverview(WebDriver driver) {
        this.driver = driver;

        // Check that we're on the right page.
        if (!(driver.getTitle().contains("XPlanner Login"))) {
            // Alternatively, we could navigate to the login page, perhaps logging out first
            throw new IllegalStateException("This is not the login page");
        }

        // These are the HTML elements that will be represented as WebElements.
        // The locators should only be defined once.
        By usernameLocator = By.name("userId");
        By passwordLocator = By.name("password");
        By loginButtonLocator = By.name("action"); // submit button

    // The login page allows the user to type their username into the username field
    public LoginPageToPJOverview typeUsername(String username) {..
    // The login page allows the user to type their password into the password field
    public LoginPageToPJOverview typePassword(String password) {..

    // The login page allows the user to submit the login form
    public ProjectPage submitLoginToPJOverview() {..
```

Webseite entspricht Java Klasse

Plausibilitätscheck im Konstruktor

Ansprechen der Elemente über Locator

Dienste der Seite als Methoden:

Bei Seiten-Wechsel wird ein neues Page-Objekt zurückgegeben