

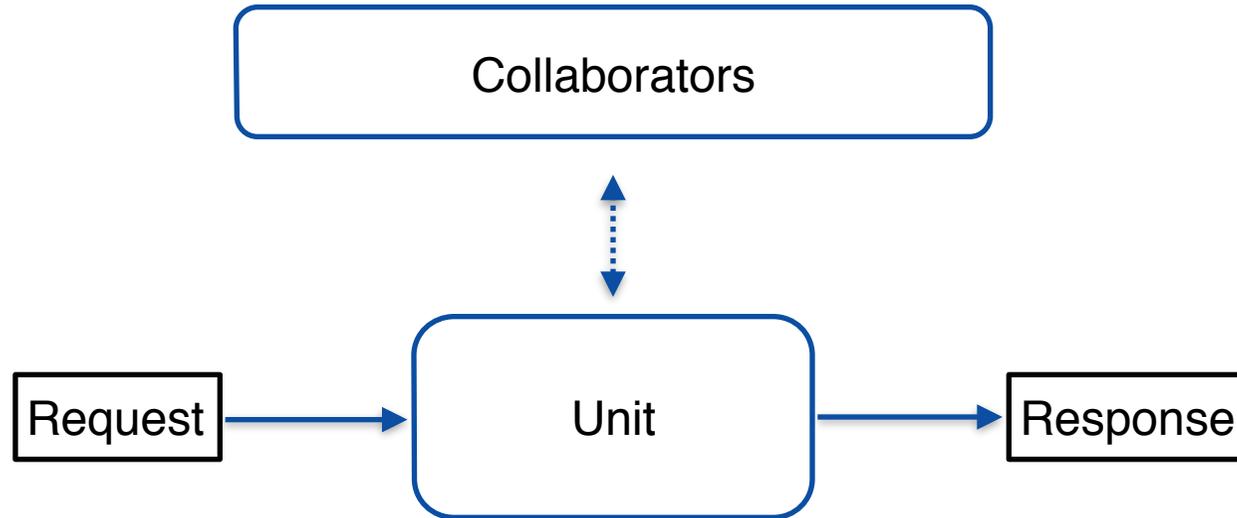
Effektives und effizientes automatisiertes Web-Testing auf Use-Case Niveau - Vorgehen und Anforderungen an ein Tooling

Dr. Thomas Luba

Fragen, Themen, Leitfaden, Diskussion

- Wieso überhaupt noch Tests über die Browser-Oberfläche fahren ?
- Warum es nicht einfach bei Browser-Automation belassen ?
- Wie die Produkt-Anforderungen im Fokus behalten ?
- Wie sicheres Refactoring erreichen ?
- Wie “good enough” festlegen ?
- Wie IT und Produkt-Management einbeziehen ?
- Auf welchen Ebenen lässt sich von Tests profitieren ?
- Diskussion: Wer kann gute Tests schreiben ?

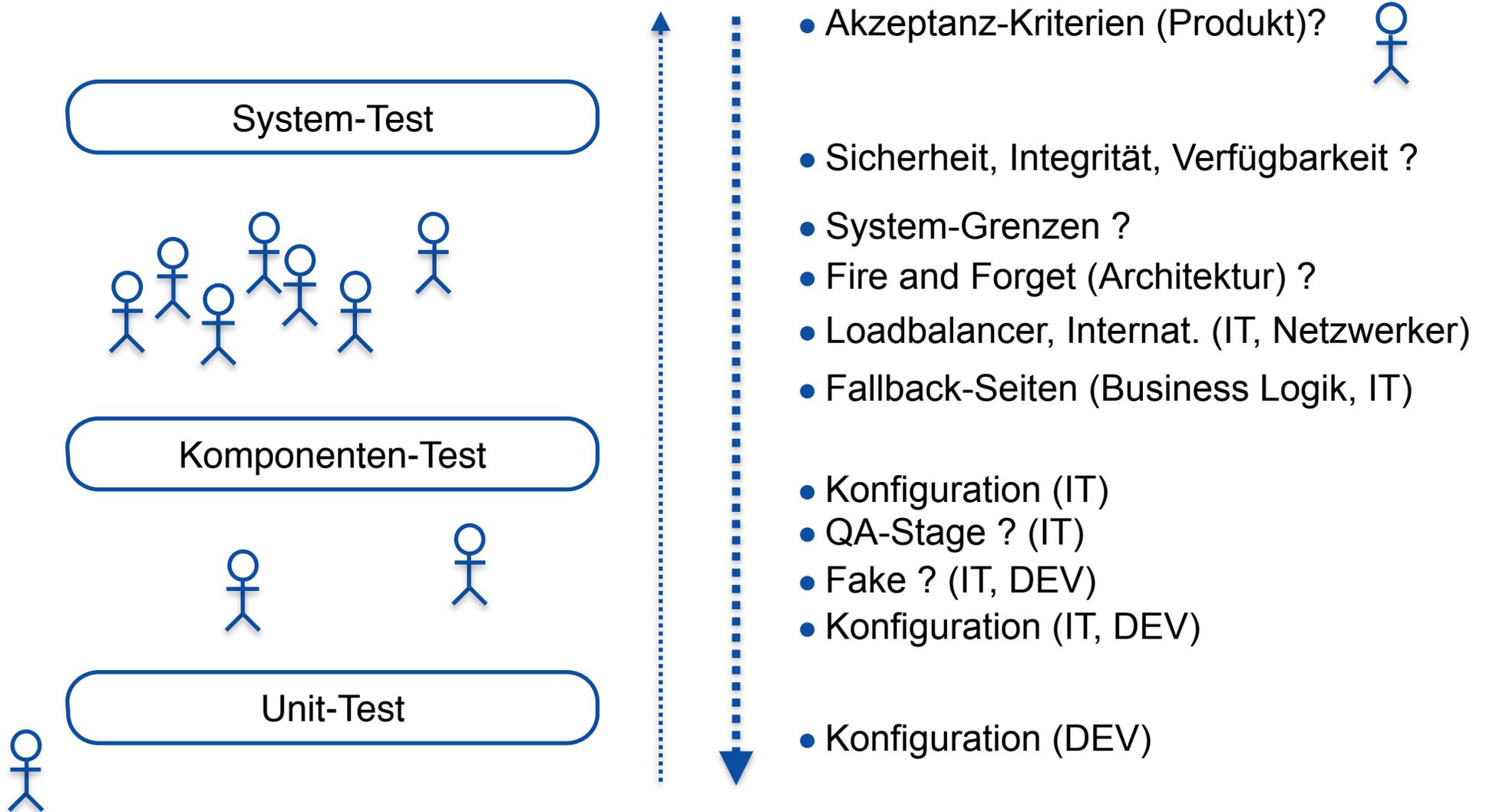
Warum noch Web-Testen: Ausgangspunkt Unit Test



- Class Responsibility Collaboration
- Test as Specification
- Gegeben:Wenn,Dann (Behaviour)
- Can / Reject - Goals
- Statische Qualitäts-Metriken

- Äquivalenzklassenbildung ?
- Randfälle ?
- Fake, Mock, ... ?
- Coverage = Prüfung ?
- Contract: produktiv ?
- Sicherheit,Verfügbarkeit,Business?
- Dokumentation ?

Warum noch Web-Testen: Test Stufen und Qualitätsaspekte



User Interface ist ein Interface: Thesen

- stabil - End-Kunden-Sicht
- öffentlich - zugänglich, verfügbar
- akzeptiert - von allen Stakeholdern
- wertschöpfend
- umfassend

Modularer Aufbau: Statische und Dynamische Sicht



WEB.DE Club



Start



Posteingang



Adressbuch



Online-Speicher



Fotoalbum



Club Center



De-Mail



Posteingang



E-Mail schreiben



SMS/MMS



Fax



De-Mail



Grußkarten



Guten Tag Thomas Luba

thomas.luba@web.de

Letzter Login: 19.01.2014 12:38 Uhr

21

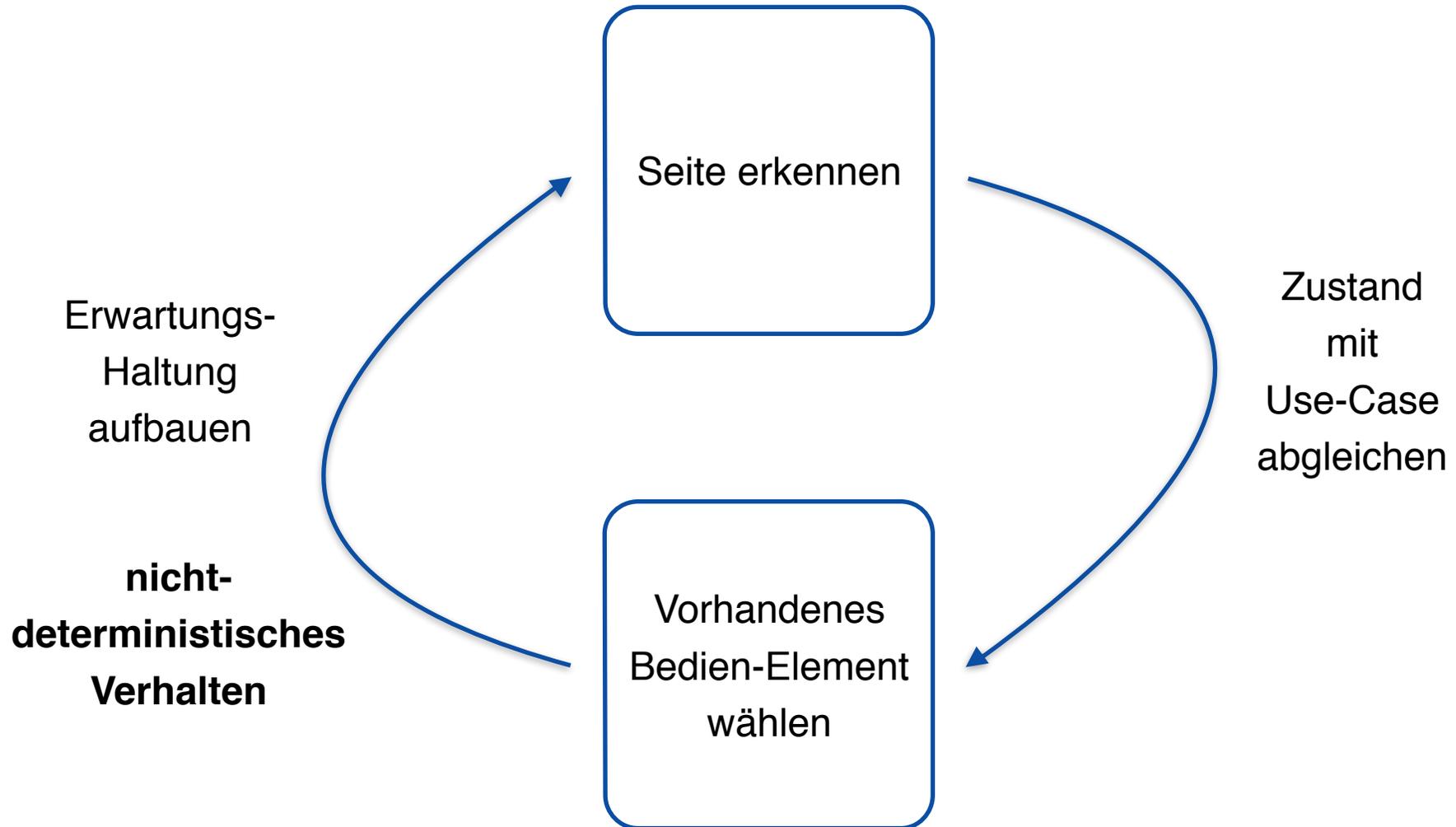
ungelesene E-Mails

2.015 E-Mails

50 WEB.Cent



Interaktions Paradigma: Tester ist User



Modularer Aufbau: Statische und Dynamische Sicht

Team "Start"

Team "Mail"

Team "Navi"

Team D



WEB.DE Club



Start



Posteingang



Adressbuch



Online-Speicher



Fotoalbum



Club Center



De-Mail

Posteingang

E-Mail schreiben

SMS/MMS

Fax

De-Mail

Grußkarten



Guten Tag Thomas Luba

thomas.luba@web.de

Letzter Login: 19.01.2014 12:38 Uhr

21 ungelesene E-Mails

2.015 E-Mails

50 WEB.Cent

Team "Mail"

Modularer Aufbau: Statische und Dynamische Sicht



WEB.DE Club



Start



Posteingang



Adressbuch



Online-Speicher



Fotoalbum



Club Center



De-Mail

“Posteingang ausgefallen”



Komponente: Posteingang
Team: Mail
Einstufung: Ausfall



Resolution: “won’t fix”
“IT, Navi-Team fragen”
Assign: Tester

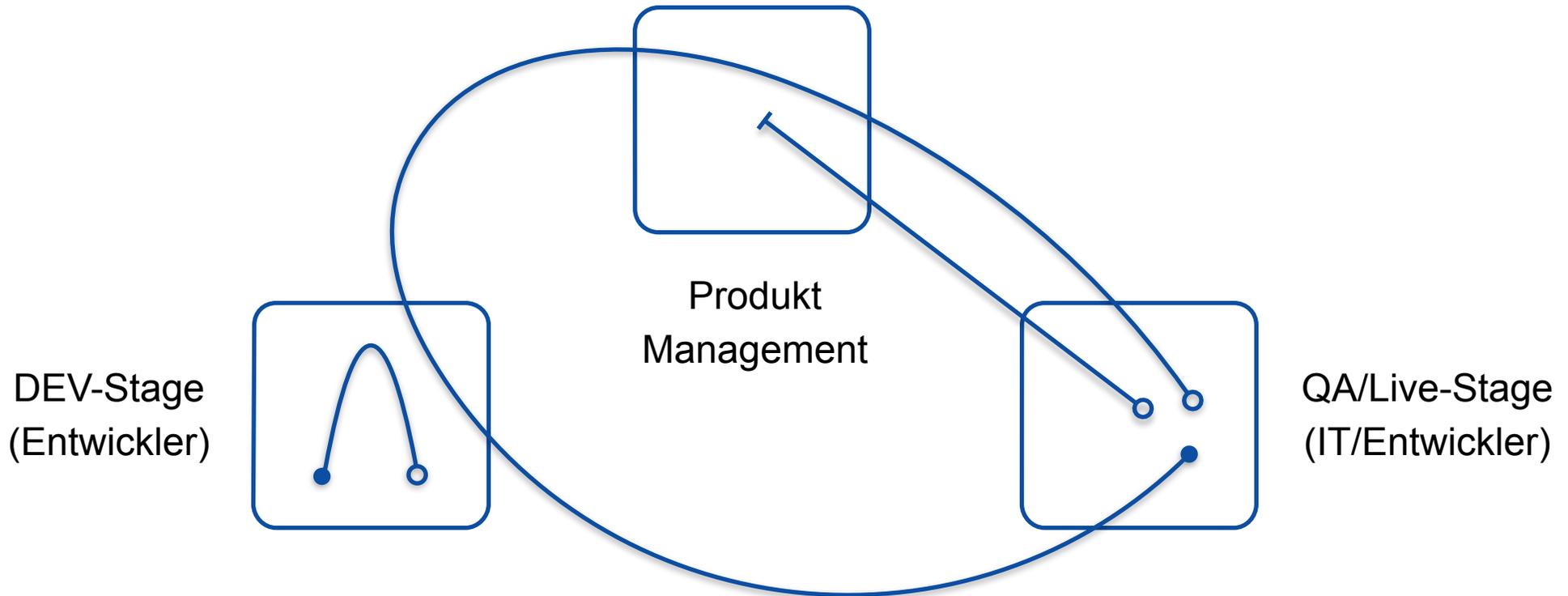
Kunde / Support / Tester

Jira Arbeitsliste

Team “Mail”

Wie vermeiden, diese Diskussion erneut zu führen ?

Gefundene Defekte: Feedback und Lebenserwartung



Entwickler korrigieren Defekte (Test Driven Development)

Produktmanager setzen Prioritäten (Requirement Backlog)

Seiten-Komponenten: Statische und Dynamische Sicht

- können **erscheinen/verschwinden**
- bestehen aus **Komponenten**
- sind **sichtbar**
- sind **vollständig da/nicht da**
- können **optional** oder **vorgeschrieben** sein
- können **eindeutig identifiziert** werden
- sind **relativ** zueinander
- sind **wiederverwendbar**

The screenshot shows a webmail interface for 'WEB.DE Club'. The sidebar on the left contains navigation buttons: 'E-Mail schreiben', 'FAX', 'SMS', 'De', 'Suche', 'Ungelesen' (with a refresh icon and '21'), 'Favoriten', and 'Freunde & Bekannte' (with a wrench icon and '21'). The main content area shows a 'Start' button and a list of folders: 'V', 'H', and 'T'. A settings dialog is open over the 'Freunde & Bekannte' folder, displaying '1.112 E-Mails / 436,33 MB'. The dialog includes options for 'E-Mails löschen nach' (set to 'nie') and 'E-Mails je Seite' (set to '50'). At the bottom of the dialog are buttons for 'Abbrechen' and 'OK'.

Entwickler – Produkt-Management Gap

WebElement.click()
WebElement.isDisplayed()
Driver.switchTo().frame()
By.xpath()
WebElement.findElement(By by)
StaleElementReferenceException
WebElement.getLocation()
WebElement.getText()
WebElement.sendKeys()
WebElement.getAttribute()
waitForElementPresent()
waitForVisible()

webdriver

“first”



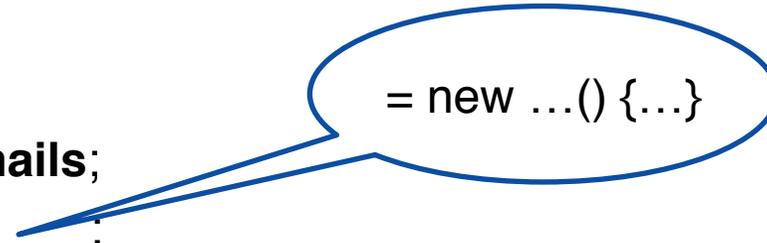
Click Save-Button
Open File-Menu
Look at Page
There is always ...
If and only If ...
Just one of ...
Given ... when ... then ...
Maybe an interception ...
The ... is optional
Looks like ...
Is part of ...
... in a row
... is a bug

scenarios

Seiten-Komponenten: Statische Struktur-Sicht

```
class Postfach extends Komponente {  
    @At(...) NavigatorLeiste navigator;  
    @At(...) Button emailSchreiben;  
    @At(...) Ordner freundeUndBekannte;  
}
```

```
class Ordner extends Komponente {  
    @At(...) Text name;  
    @At(...) Optional<Int> anzahlUngelesenerEmails;  
    @At(...) Menue<OrdnerSettings> settings ;  
}
```



= new ...() {...}

```
class OrdnerSettings extends Komponente {  
    @At(...) Int emailAnzahl;  
    @At(...) Button leeren;  
    @At(...) Button ok;  
}
```

Seiten-Komponenten: Dynamische Struktur-Sicht

```
class OrdnerFreundeUndBekannteLeeren extends Postfach {  
    public void call() {  
        freundeUndBekannte.settings.open().leeren.click();  
        lookup(new PruefeDassFreundeUndBekannteGeleert());  
    }  
}
```

```
class PruefeDassFreundeUndBekannteGeleert extends Postfach {  
    public void call() {  
        verifyThat(freundeUndBekannte.get().anzahlUngelesenerEmails.isZero());  
        verifyThat(freundeUndBekannte.settings.open().emailAnzahl.isZero());  
    }  
}
```

Matcher-Information: Beispiel

```
verifyThat(freundeUndBekannte.get().anzahlUngelesenerEmails.isZero());
```

~~{ Postfach:freundeUndBekannte: "missing" }~~

{ Postfach:freundeUndBekannte.anzahlUngelesenerEmails: "missing" }

{ Postfach:freundeUndBekannte.anzahlUngelesenerEmails: { is: 1, should-be: 0 } }

{ Postfach:freundeUndBekannte.anzahlUngelesenerEmails: { not-a-number: "x" } }

{ Postfach:freundeUndBekannte.anzahlUngelesenerEmails: "multiple finding" }

Matcher-Informationen: Produktmanagements Lieblinge

frei

genau dann,
wenn

gebunden

`webdelogo.isPresent() iff(brand.isEqualTo(WEB_DE))`

`name.isEqualTo(_("save").or(_("sichern")))`

freier Bezugspunkt

gebunden

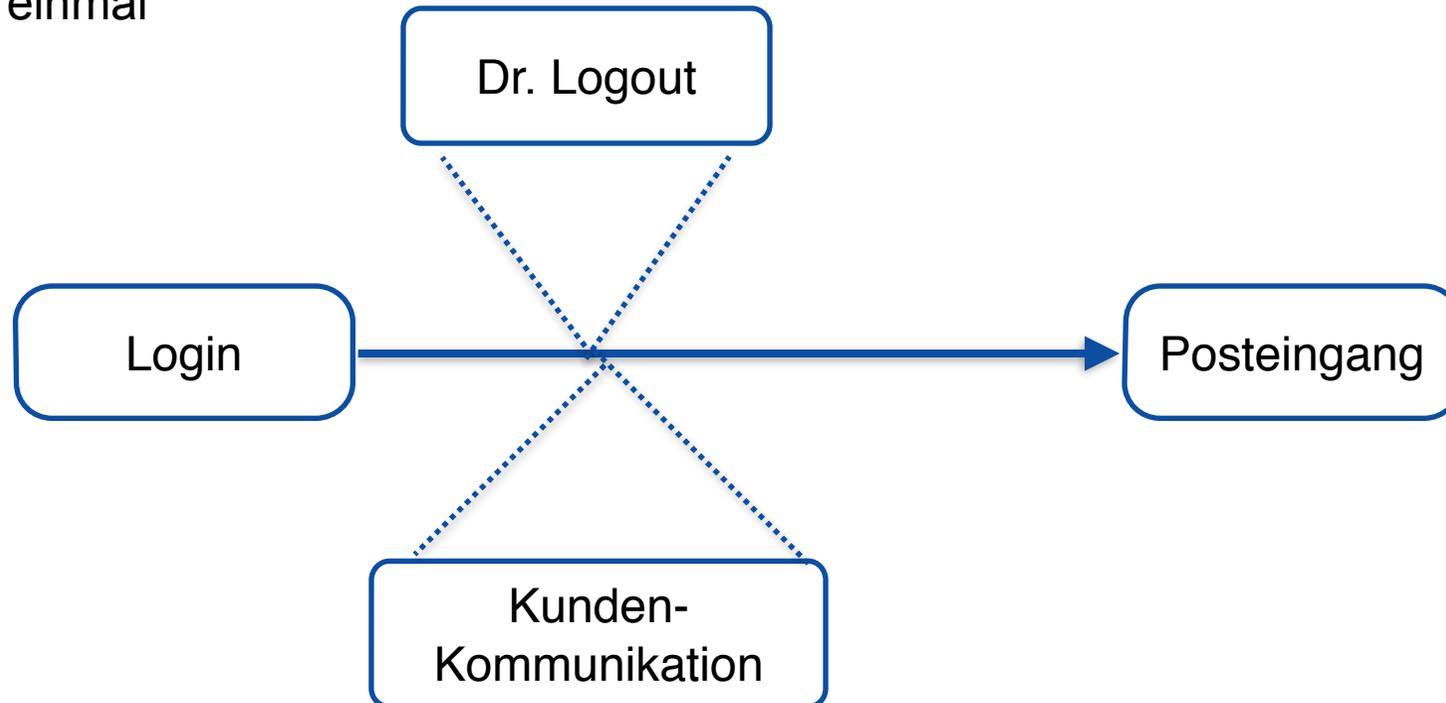
Vorbedingungen / Invarianten an Seitenbeschreibung

```
class Postfach extends Komponente {  
    ...  
    public void checkOnEntry(){  
        testThat(  
            webdelogo.isPresent().iff(brand.isEqualTo(WEB_DE))  
        );  
    }  
}
```

Bedingungen auf Seitenbeschreibung werden **immer eingefordert**

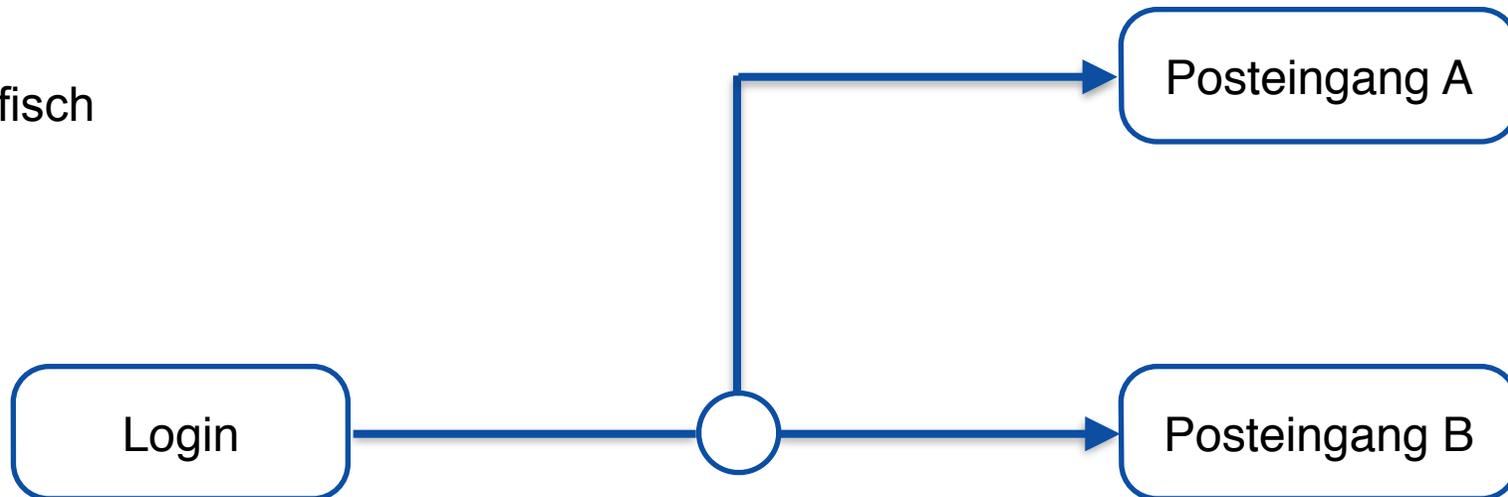
Standard Workflow Design Patterns: Interceptions

- optional
- nicht deterministisch
- eindeutig
- jeweils einmal



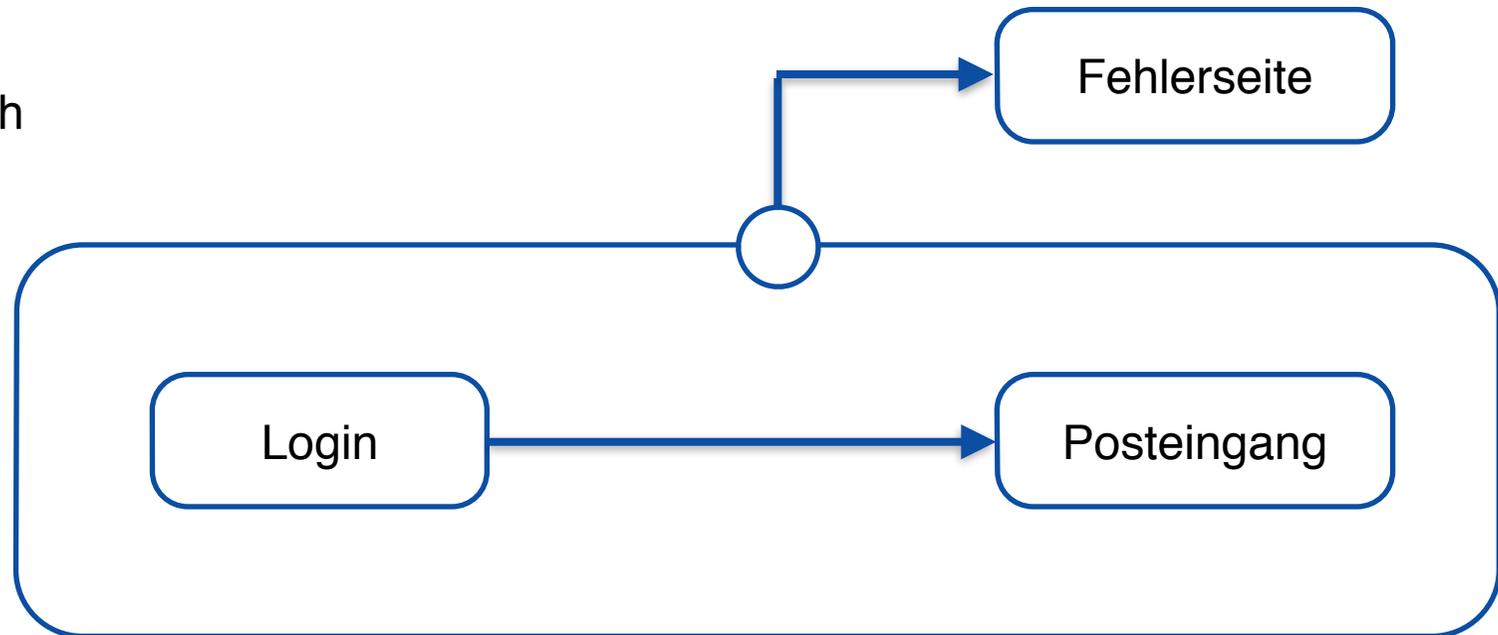
Standard Workflow Design Patterns: Spezifische Extension

- user content getrieben
- application state getrieben
- nicht deterministisch
- eindeutig
- spezifisch

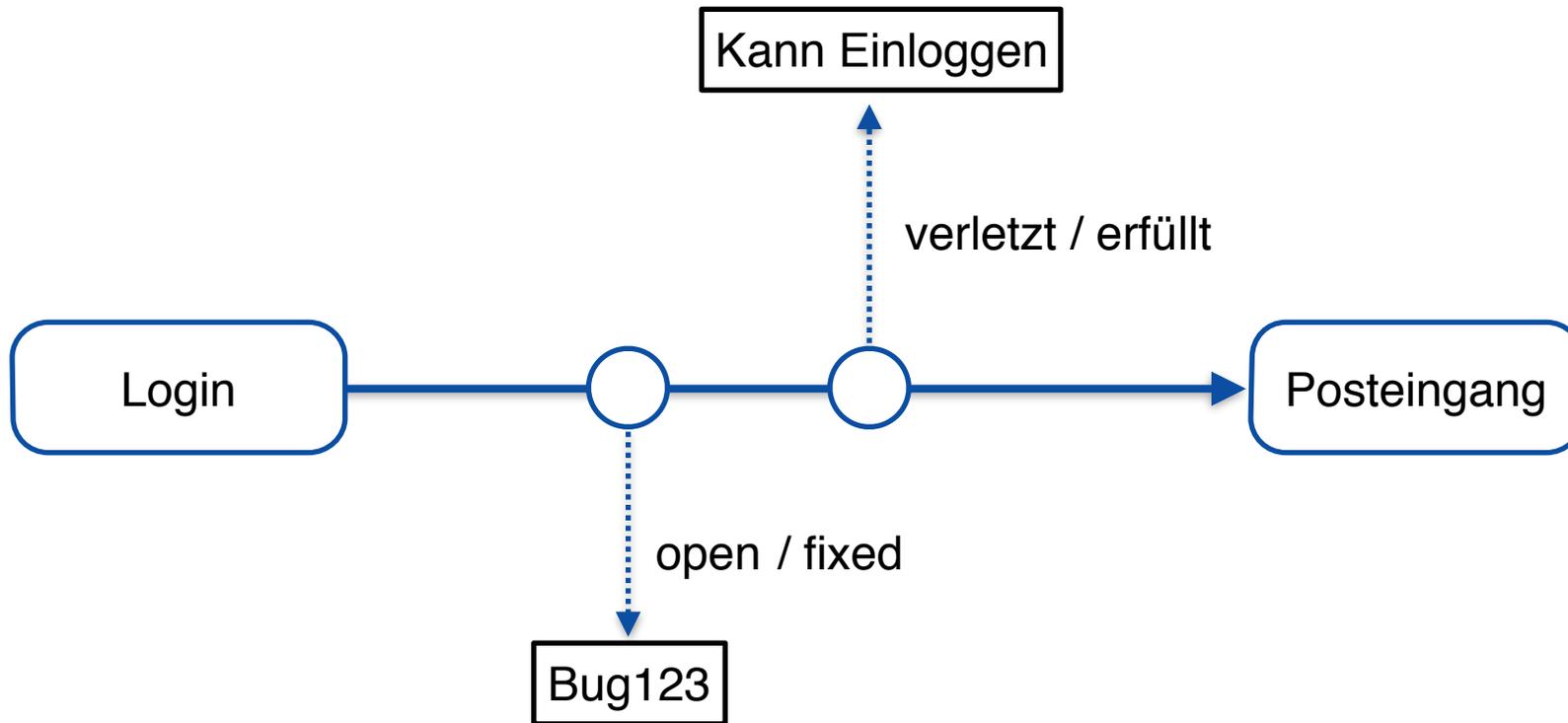


Standard Workflow Design Patterns: Allgemeine Extension

- user content getrieben
- application state getrieben
- nicht deterministisch
- eindeutig
- unspezifisch



Test-Cases und Capabilities (Requirements)



- Test-Cases überdecken Requirements: many-to-many

Dynamische Entscheidung mit Report

```
class ZumPostfachÜberNavigatorGehen extends Startseite {  
    public void call() {  
        navigator.postfach.click();  
        lookup( new OrdnerFreundeUndBekannteLeeren(),  
                new PostfachAusgefallen());  
    }  
}
```

```
@Issue("IT oder Team 'Navi' fragen")  
class PostfachAusgefallen extends Komponente {  
    ...  
    public void call() {  
        reportAsOpen(PostfachAusgefallen.class);  
    }  
}
```



Qualifikation

Integrierte Entwicklungsumgebung - Assistenz

- Katalog
 - Komponenten und Aktivitäten **übergreifend** wiederverwendbar
 - Strukturübersicht **Seitenbeschreibung**
 - Strukturübersicht Workflow-**Schritt-Abfolge**
- Schreiben und Refaktorisieren
 - **"fluent"** mit dem "." schreiben und auslösen
 - **Wechsel** Optional/Pflicht-Felder IDE geführt
 - **Terminologie-Pflege** sicher und einfach
- Suchen und Finden
 - **Bedingungen** an **Komponenten** in Workflows
 - **Testpläne** zusammenstellen
 - Testplan-**Coverage** ermitteln
 - Workflows zum Überdecken von **Issues**

Behaviour Driven Development - jbehave

Narrative: One of the main business concerns is the accessibility of email

Scenario: Entering the email-account via navigator

Given I am looking at **Startseite**

When I do **click** on **navigator.postfach**

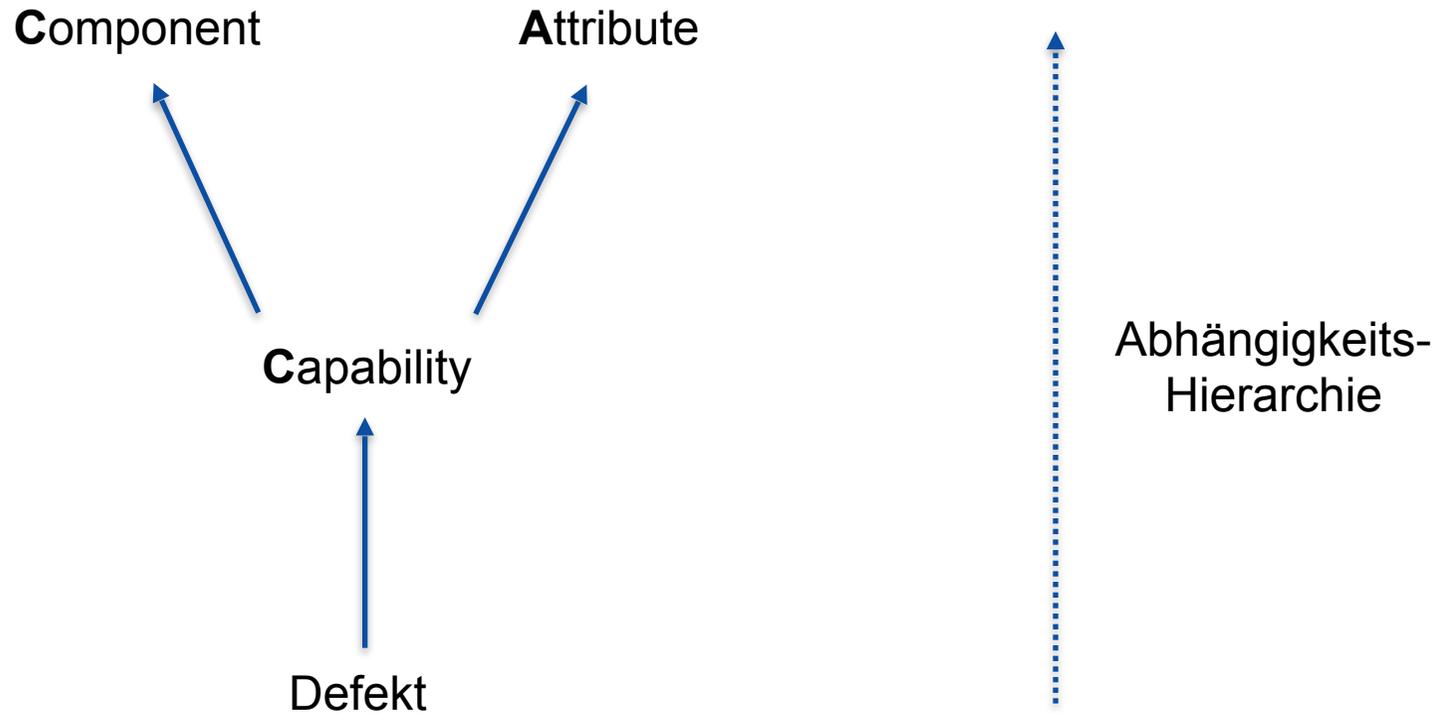
Then I am looking at **Postfach**

And **freundeUndBekannte.anzahlUngelesenerEmails** has value 21

Prozess: Produkt-Management getriebener Aufbau

- Google ACC
 - **Capabilities** von
 - **Components** erfüllen
 - **Attributes**
- Use-Cases
 - Workflows die **Kunden** wirklich brauchen und nutzen
 - **Nachweis** von wirklich relevanten Capabilities
- Defekte
 - deren Auftreten über die **Zeit** verfolgt werden soll
 - die **Capabilities** verletzen

Capability und Defekte als Issues



Attribute von Bedeutung: Key Process Indicators

- Business
 - zahlungspflichtiger Content, Werbe-Flächen
- Verfügbarkeit
 - Ausfall, Fehler-Seite, Störungen, hängende Links
- Kunden-Daten Integrität
 - Verarbeitung, Haltung
 - Feedback: Info-Meldung “gespeichert” gdw. Speicherung erfolgt
- Kunden-Daten Sicherheit
 - Zugang
 - Session

Components: Verantwortlichkeiten und Teams

- z. B.
 - Infrastruktur
 - Prozesse
 - Services
- mit
 - Verortung in der Organisation
 - funktionalen und nicht funktionalen Verantwortlichkeiten
- so dass sich
 - Defekte zielgerichtet Teams zuweisen lassen
 - ein kontinuierlicher Verbesserungsprozess ergibt
 - sich geeignete Ansprechpartner finden lassen
 - Aufwand und Kosten (Features) zuweisen lassen

Unqualifizierte blockierende Defekte

Test bleibt “**stecken**” ohne möglichen Workaround

- **Umfang** und Wert von Tests eingeschränkt
- Tester melden “**blockiert**” an das Projekt-Management
- weitergehende “**manuelle**” Analyse erforderlich
- notwendige **Maßnahmen** verzögert
- **Vertrauen** in Produkt, Prozesse und Tests beeinträchtigt
- Tester werden **Flaschenhals**

Qualifizierte Defekte

Ein **Zustand** oder eine **Bedingung** wird qualifiziert, indem er/sie

- einer **capability** zugeordnet wird
- einer abgesprochenen **component** zugewiesen wird
- mit einer abgesprochenen **Gegen-Maßnahme** dokumentiert wird
- mit einem **Ticket** kommuniziert wird
- falls möglich mit einem **Workaround** versehen wird
- **zuverlässig** “positiv” erkannt wird
- Hinweise auf eine **root cause** liefert

Schnelles Feedback über Qualifikation

Bedarfsgerechte **Verbesserung** der automatisierten Tests

- **Coverage** der Tests kann “gerettet” werden
- **keine manuellen** Eingriffe nötig
- klare Aussage zum **Zustand** des Produkts
- **Maßnahmen** werden unmittelbar ergriffen
- **wiederholtes** Auftreten wird erkannt
- **definition** von “good enough” wird möglich
- “neu” auftretende Defekte lassen sich **bedarfsgerecht** behandeln
- es muss nicht “auf Verdacht” qualifiziert werden

Evidenz für Issues



reportAs...()	<i>blockierend</i>	<i>erwartet</i>
<i>closed</i>	-	<i>open</i>
<i>open</i>	<i>nein</i>	<i>closed</i>
<i>open</i>	<i>nein</i>	<i>open</i>
<i>open</i>	*	"unqualifiziert"
<i>open</i>	<i>ja</i>	*

<i>Bewertung</i>
"Verbesserung"
"Verschlechterung"
"Keine besonderen Vorkommnisse"
"Analyse notwendig"
"Test blockiert"

QUALIFIED - ISSUE - REPORT

The Qualified Issues Report **monitors issues** like **defects** and **quality features over time** - use it to **address problems** to IT and SE **directly**.

Blocked tests may **obstruct tracking**.

- . No Run
- _ Consider Issue as Closed
- Affirmed Issue as Closed
- x Open Issues Witnessed
- X Closed Issue Reappeared

TEST - CASE - REPORT

The Use-Case-Report **monitors** the **capability** of the system to **execute** the use cases **until the end** and **shows blocked tests**.

Used to **assess availability** or **testability**.

- . No Run
- _ Tracking through
- Use-Case witnessed unknown issues on the way
- x Use-Case blocked in known defective state
- X Use-Case blocked unexpectedly

NON - BLOCKING - UNQUALIFIED - ISSUE - REPORT

The Non-Blocking Unqualified-Issues-Report shows **non-blocking unqualified issues by location**, that is unknown issues corresponding to [-] in the Test-Case-Report.

No Data

Witnessed

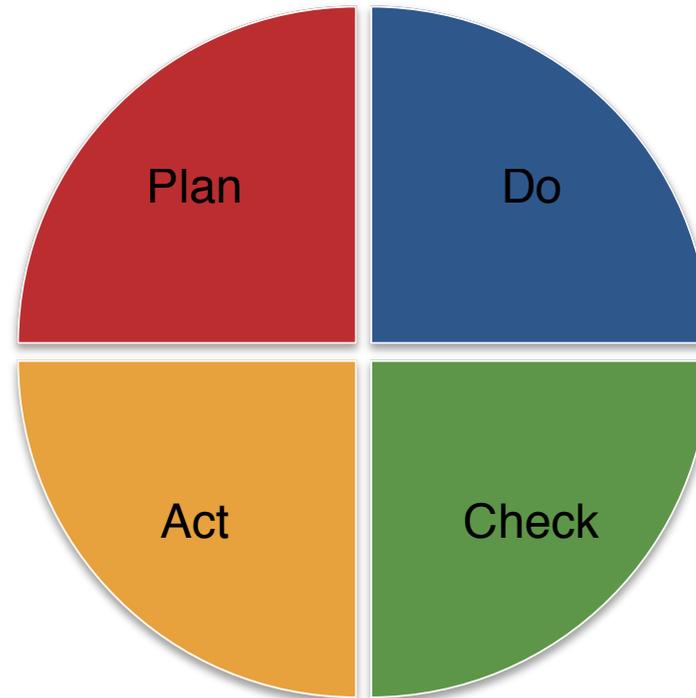
Test-Automatisierung als Verbesserungs-Prozess

Planen

- Monitoring
- Tests
- Gateways

Ausführen

- kontinuierlich
- umfassend
- bedarfsgerecht



Verbessern

- Defekte qualifizieren
- Defekte kommunizieren

Prüfen

- qualifizierte Fähigkeiten
- unqualifizierte Zusicherungen

Praktischer Einsatz im agilen Umfeld

- Akzeptanz-Gateway
 - als **Smoke**-Test nach **Rollout** (continuous delivery) und vor dem Einchecken
 - als untere Schranke für die Produkt-**Qualität**
- Feature/Defect-**Tracking**/Monitoring
 - **kontinuierlich** für Betrieb und Entwicklung (Standup)
 - als **Input** für den Produkt-Manager und Projekt-Leiter
- Test-Daten Erzeugung
 - für manuelle Test-Durchführung
 - unter Benutzung des Produktiv-Systems

Arbeitsteilung im agilen Umfeld

- Seitenbeschreibung
 - scharfe **Trennung** in **Deklaration** und **Definition** (Mapping auf **html**)
 - Rolle: Entwickler/Designer, Tester
 - Workflows
 - **deklarative** Seitenbeschreibung genügt
 - Rolle: Tester, Test-Manager
 - Text-Bausteine (“Fitnessse”)
 - auf Basis **vorgefertigter** workflows
 - Rolle: Produkt-Manager, Test-Manager
- + ???

Entwicklertag Karlsruhe 22.5.2014

Fragen + Diskussion