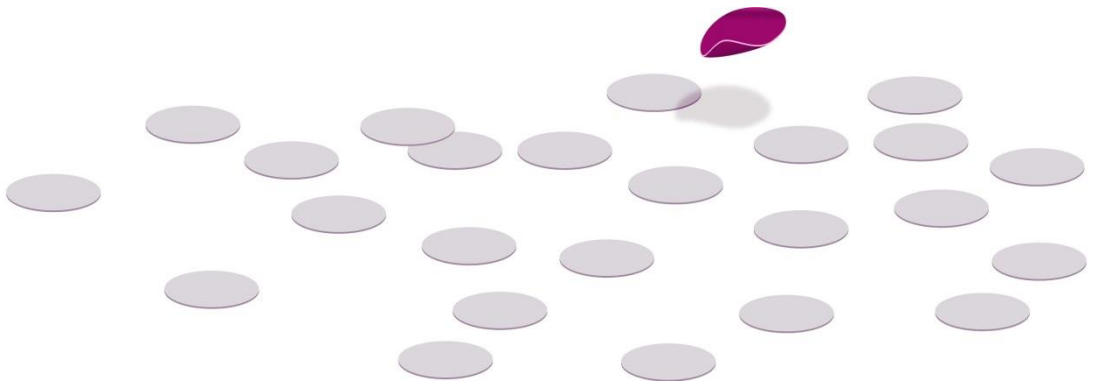
iteratec

KOMPETENZ, DIE ENTLASTET

Einfache automatisierte Akzeptanztests für Web-Anwendungen nach dem KISS-Prinzip mit Geb

Martin Pelzer



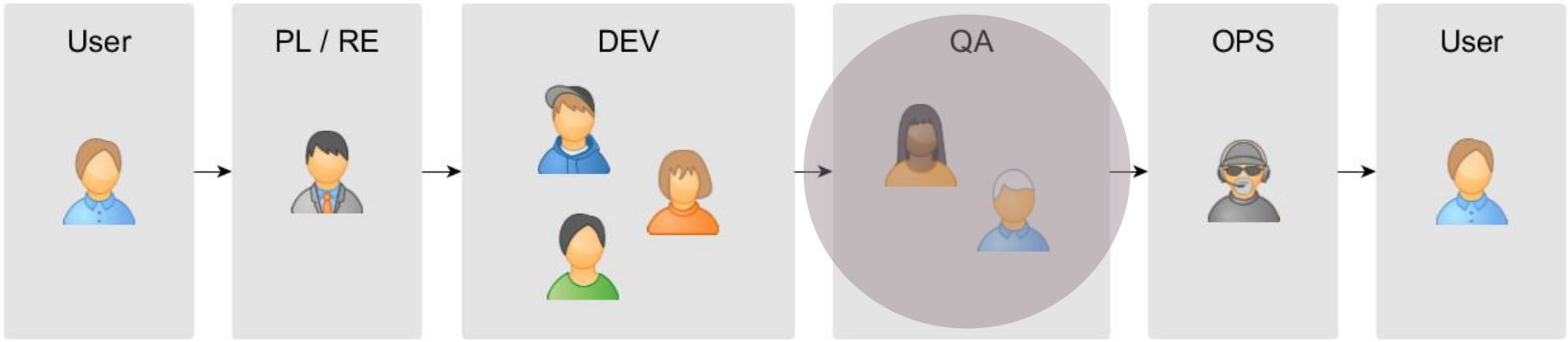
Einleitung

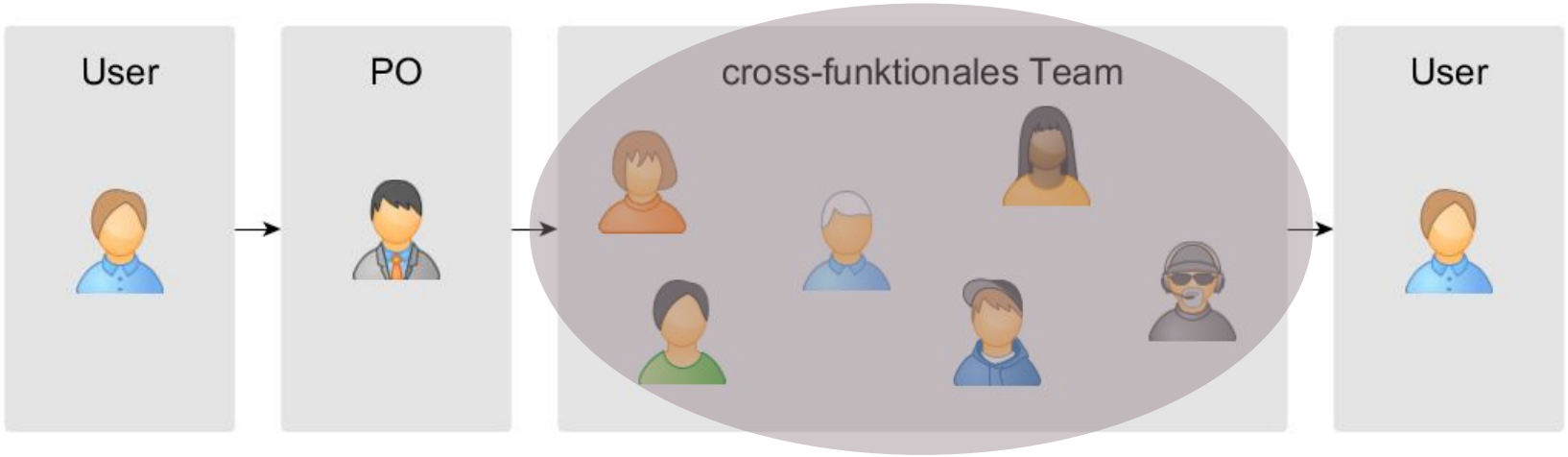
- Kurze Release-Zyklen bedingen Testautomatisierung
- Akzeptanztests oft UI-Tests
- Welches Tool nehme ich für die Automatisierung?

Tools für Akzeptanztests

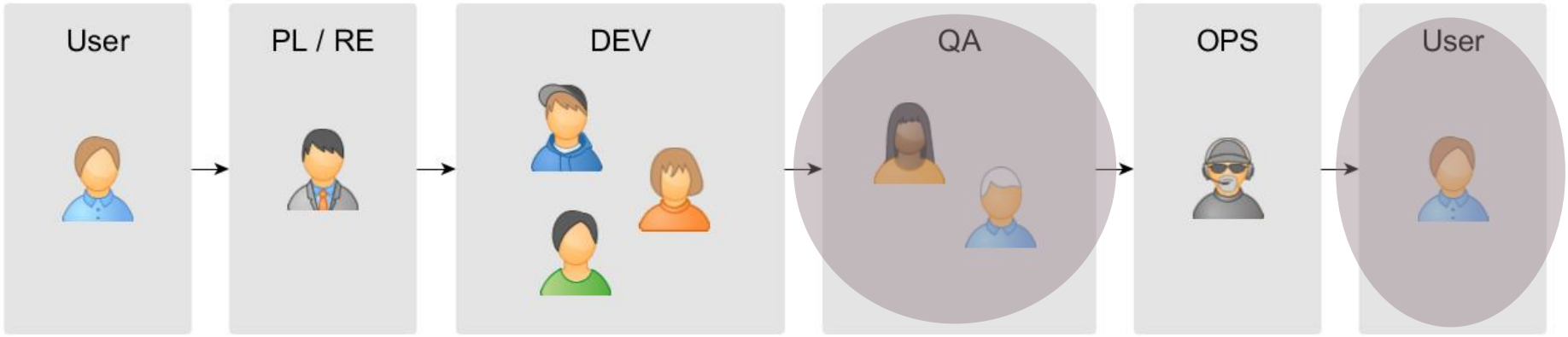
- Ansatz natürlichsprachlicher Testfälle
  - z.B. Cucumber, Fitnesse, …
  - Umsetzung in Quellcode erforderlich
- Ansatz Testfälle direkt als Quellcode
  - z.B. Junit, Spock, Selenium, Geb, …
  - Grundlegende Coding-Kenntnisse erforderlich

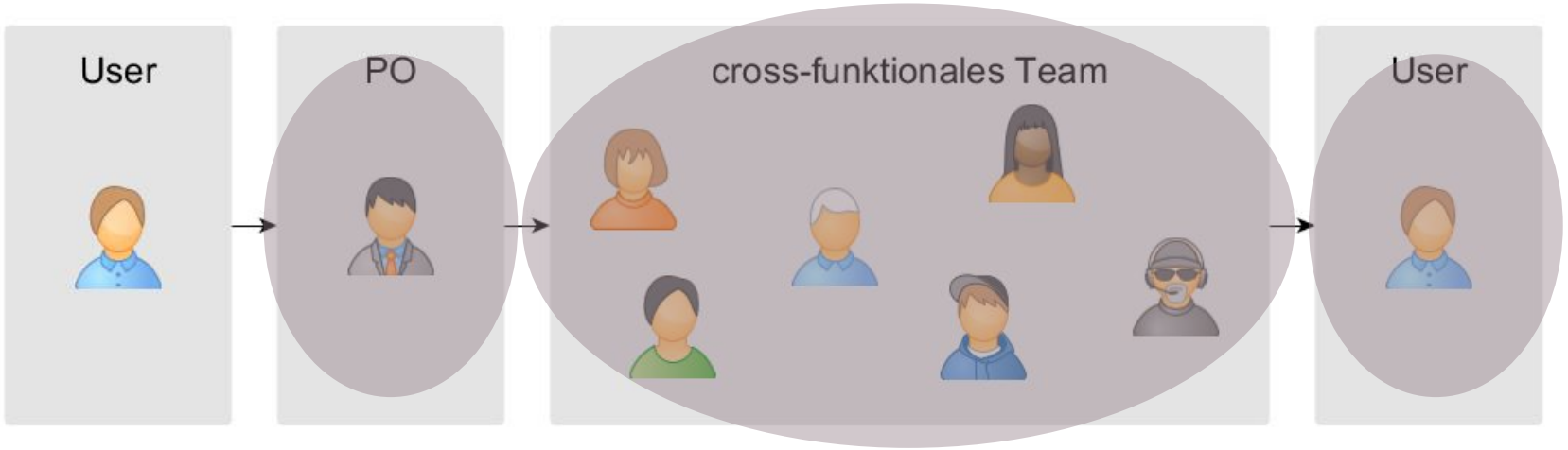
Wer schreibt Akzeptanztestfälle?





Wer liest Akzeptanztestfälle?





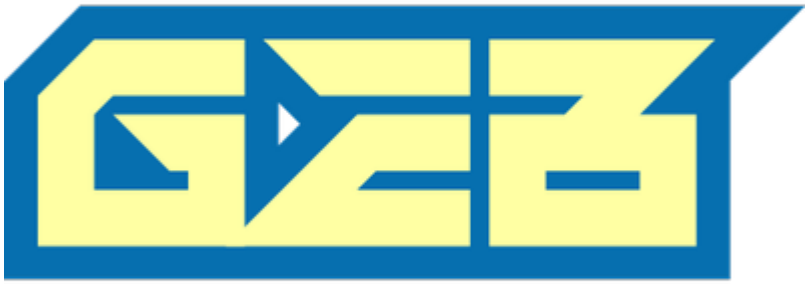
Tools für Akzeptanztests

Die Wahl des richtigen Tools

hängt auch vom Vorgehensmodell ab.

In cross-funktionalen Teams kann man die Komplexität reduzieren, in dem man Testfälle direkt als Quellcode schreibt.

Allerdings muss auch dies Lesbarkeit gewährleistet bleiben.



Logo-Bild von twitter.com/GebFramework

Geb

- Test-Tool für Web-UIs
- Basiert auf Selenium WebDriver
- Groovy-basierte DSL
- Page Objects
- Apache-2-Lizenz

Geb + X

- Verwendung in Kombination mit (Unit-)Test-Framework
  - JUnit
  - TestNG
  - Cucumber
  - **Spock**
- „While Geb works great with all of these frameworks, it really shines with Spock.”

Geb – Show me the code!

```
def "login to admin section works"() {
    given: "I am at the login page"
    to LoginPage

    when: "I enter the admin credentials"
    loginForm.username = "admin"
    loginForm.password = "password"

    and: "I click the login button"
    loginButton.click()

    then: "I am forwarded to the admin page."
    at AdminPage
}
```

Geb – Show me the code!

```
def "login to admin section works"() {
    given: "I am at the login page"
    Page page = to(LoginPage);

    when: "I enter the admin credentials"
    page.loginForm.username = "admin";
    page.loginForm.password = "password";

    and: "I click the login button"
    page.loginButton.click();

    then: "I am forwarded to the admin page."
    at(AdminPage);
}
```

Geb – Show me the page object!

```
class LoginPage extends Page {

    static url = "index.html"

    static at = { $("h1").text() == "Blubb" }

    static content = {
        inputUserName { $("input", id: "user") }
        inputPassword { $("input", id: "user") }
        loginButton { $("button", class: "myButton") }
    }
}
```

Geb - Selektoren

- $("h1", 2, class: "heading")
- $("p", text: "p1")
- $("div").$("p")
- $("div").has("input", type: "text")
- $("p").next(class: "c")
- $("p", id: "3").closest("a")
- …

Geb – Beispiel Umgang mit dynamischen Inhalten

```
class GoogleResultsPage extends Page {
    static at = { waitFor { title.endsWith("Google Search") } }
    static content = {
        results(wait: true) { $("li.g") }
        result { index -> results[index] }
        resultLink { index -> result(index).find("a.l") }
    }
}
```

Geb – gib mir mehr davon!

- www.gebish.org
- The Book of Geb: www.gebish.org/manual/current/
- github.com/geb
- www.spockframework.org
- Übung heute Nachmittag

Worum ging‘s jetzt?

- **Beachten Sie bei der Toolauswahl auch ihre Projektorganisation.**
- **Bei crossfunktionalen Teams kann man evtl. auf Komplexität durch Trennung von Testfalldefinition und Testfallimplementierung verzichten.**
- **Ein Tool zum Erstellen lesbarer UI-Tests ist Geb.**
- **Geb gucken wir uns heute Nachmittag in der Übung noch genauer an.**

iteratec

KOMPETENZ, DIE ENTLASTET

Vielen Dank!

Martin Pelzer

martin.pelzer@iteratec.de

github.com/mpelze/geb_workshop

Iteratec GmbH