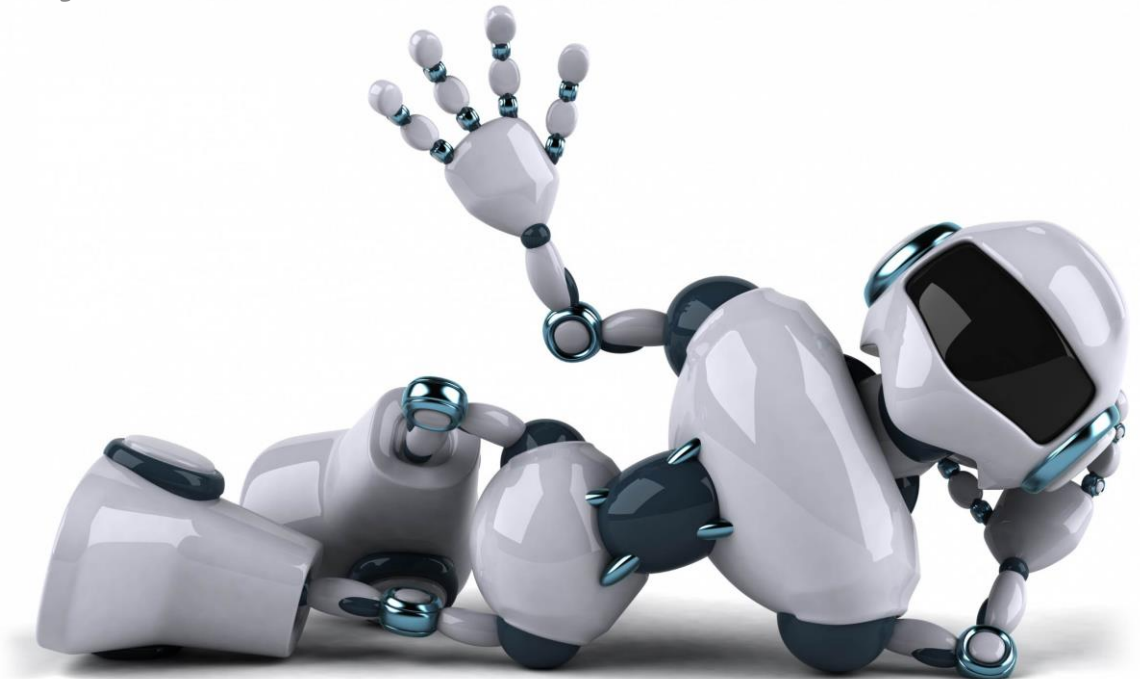


Selenium Session 3

Git, Maven, Page Object
Pattern, Jenkins

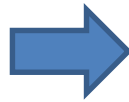
Alexander Henze
Joachim Basler



Inhalte der Sessions

Session 1

- Einführung und grober Überblick zu Selenium
- Hands on Selenium IDE



Session 2

- IntelliJ, JDK
- Java Grundlagen
- WebDriver API
- Selenium-Test in Java



optional

- Appium: „Selenium für Mobile“

optional

- Parallelisierung mit Selenium Grid
- SeleniumBox

Session 3

- das „Drumrum“ eines Software-Projekts für autom. Tests kennenlernen
- „schönere Tests“ schreiben
- Einbindung in CI-System

Was wollen wir also heute konkret tun?



GitHub
Repository
klonen und
verändern



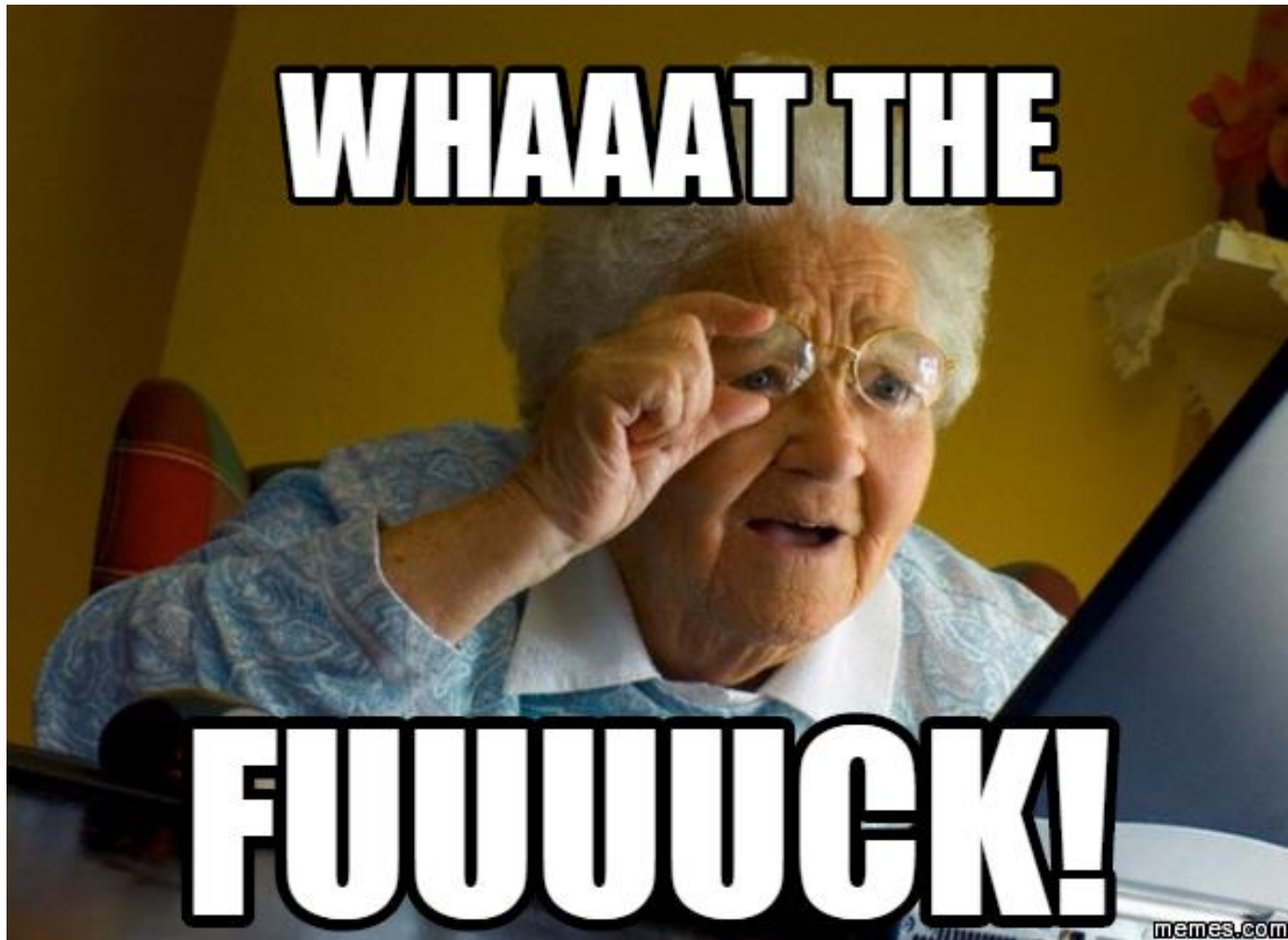
Tests über Maven
ausführen



Job in Jenkins
anlegen und
Selenium-Tests
ausführen



Page Object Pattern
anwenden



Grundlagen zu Git

Was kann ich mit Git tun?

- kollaborativ an einer Codebasis arbeiten
- Dateien unter Versionskontrolle setzen



Was bringen mir GitHub, GitLab, o.ä.?

- bietet Hosting von Entwicklungsprojekten
 - selbst etwas entwickeln und teilen
 - andere Projekte einsehen und downloaden
- webbasierte Benutzeroberfläche

....und noch deutlich mehr.

(Ist uns für den Moment aber egal!)

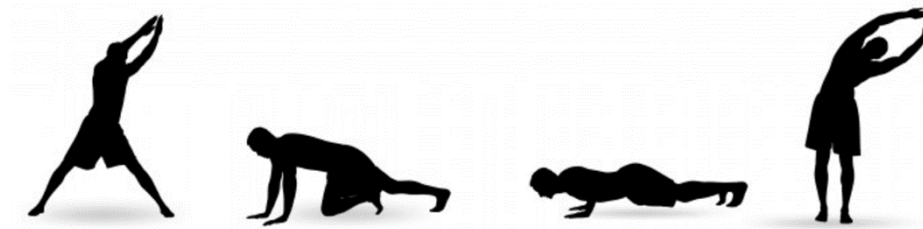
Git super basic Grundbegriffe

- **Repository:** unter Versionskontrolle stehendes Dateien-Verzeichnis/Projekt
- **Clone:** eine lokale Kopie eines Repositories erstellen
- **Commit:** eine lokale Änderung an einem Repository einchecken und anderen verfügbar machen

Demo: Git



Übung 1: GitHub-Repository



Quelle: [2]

1. Öffne das Repository im Browser:
<https://github.com/n-jooy/SeleniumDojoBasic>
2. Kclone das Repository mit unseren „unschön“ geschriebenen Tests:
<https://github.com/n-jooy/SeleniumDojoBasic.git>
3. Öffne das Projekt in IntelliJ
4. Schau dir die Commit-History an
5. Erstelle einen Commit mit einer neuen Klasse, die deinen Namen trägt

GitHub-Nutzer

Name: seleniumHorst

Mail: im_wind@web.de

PW: seleniumHorst123

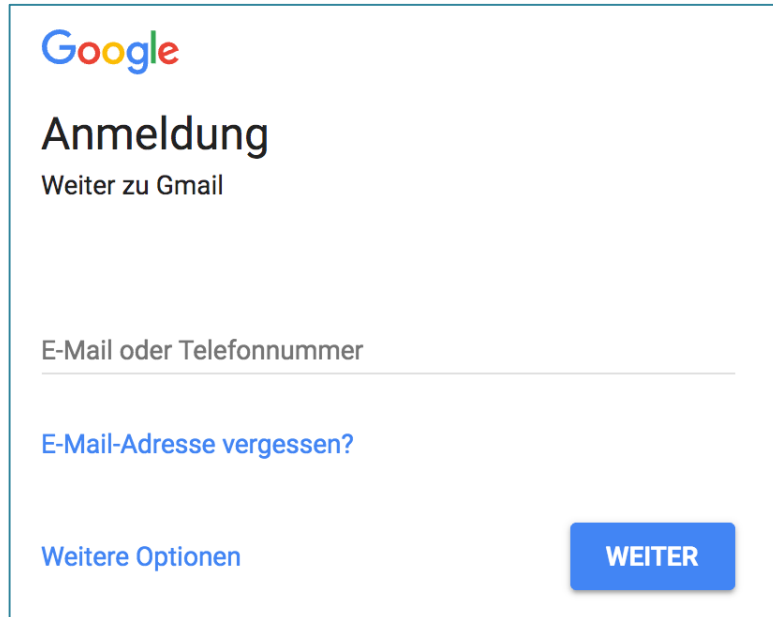
Page Object Pattern

A "pattern" has been defined as "an idea that has been useful in one practical context and will probably be useful in others."

Martin Fowler

- Abbildung der UI der zu testenden Anwendung in Page-Klassen durch:
 - Elemente
 - Funktionen
- Zugriff über Objekt(e) der Page-Klasse
- mögliche Vorteile:
 - Reduzierung von Redundanz
 - bessere Les- und Wartbarkeit

Bsp. Page Object



The screenshot shows the Google login interface. At the top is the Google logo. Below it is the heading 'Anmeldung' followed by the text 'Weiter zu Gmail'. There is a text input field with the placeholder 'E-Mail oder Telefonnummer'. Below the input field is a link 'E-Mail-Adresse vergessen?'. At the bottom left is a link 'Weitere Optionen' and at the bottom right is a blue button labeled 'WEITER'.

LoginPage-Klasse

Elemente:

- Inputfeld für Mailadresse
- Continue-Button
- ...

Methode:

Befülle Inputfeld und klicke auf Continue-Button

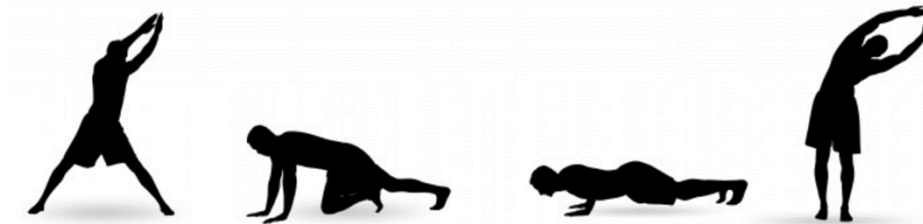
Testklasse

- neues Objekt der LoginPage-Klasse instanziiieren
- über dieses Objekt auf die Methode zugreifen

Demo: Page Object Pattern



Übung 2: Page Object Pattern



Quelle: [2]

1. Öffne das geklonte Repository aus Übung 1
2. Wende das Page Object Pattern auf die vorhandenen Tests an

Maven

- Build Management Tool von Apache
- Dependency Management
 - lokal/intern
 - Download von bspw. Bibliotheken über das Internet
- bauen und ausführen eines (Maven) Projekts auch außerhalb der IDE via Command Line
- setzt auf standardisierte Verzeichnisstruktur
- Konfigurationsdatei: pom.xml

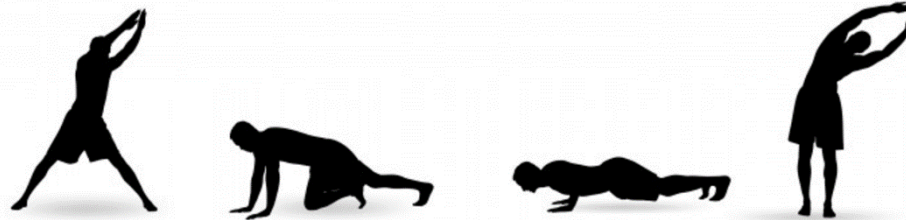
Jenkins

- Jenkins ist ein Continuous Integration (CI) System
 - CI (sehr grob!): Prozess um eine Software kontinuierlich und automatisiert zu bauen, testen und zur Verfügung zu stellen
- Trigger ist bspw. Änderung am Repository
- aufrufbar über Webbrowser
- viele Plugins installierbar, z.B. für Test-Reports
- kann lokal installiert werden, oder (intern) auf einem Server gehostet werden

Demo: Jenkins



Übung 3: Jenkins



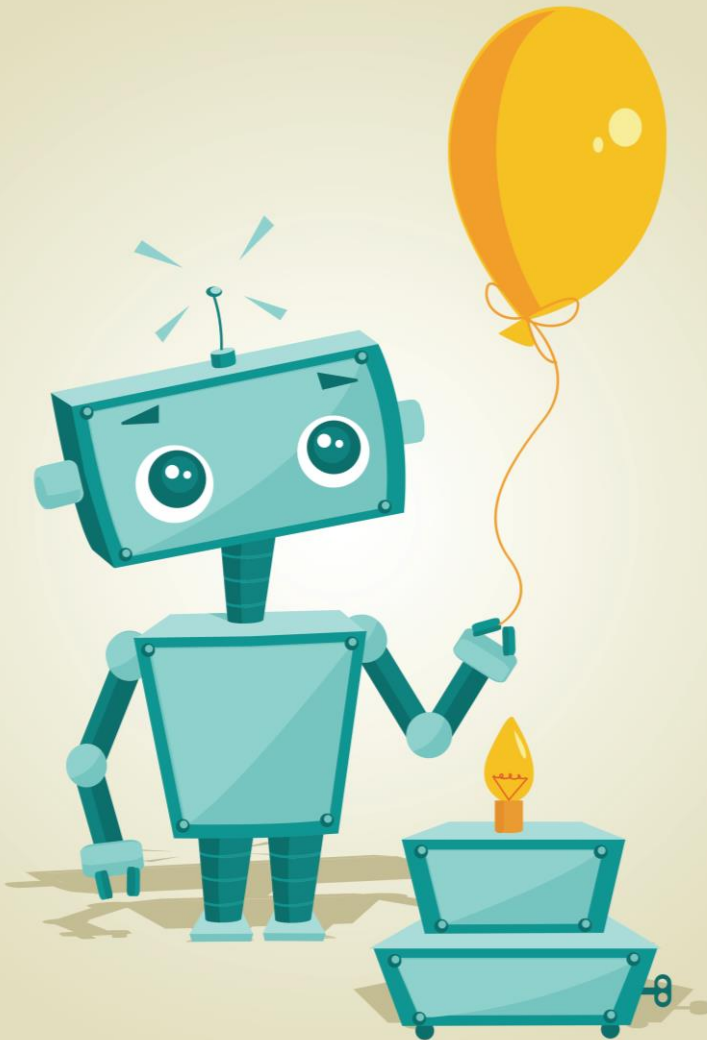
Quelle: [2]

1. Jenkins lokal installieren und über <http://localhost:8080/> im Browser aufrufen
2. Maven-Installation in Jenkins definieren
3. Selenium-Test-Job anlegen, der folgendes tun soll:
 - a. GitHub Repository klonen
 - b. Tests per Maven Command ausführen
4. Job ausführen

Ausblick

Weitere Sessions?

- Übungs-Session zu den Inhalten aus Session 2+3
- „Mobile Selenium“ mit Appium
- Parallelisierung mit Selenium Grid



Quelle: [3]

Anhang: Quellen der Abbildungen

- [1] <http://weknowyourdreams.com/images/robot/robot-04.jpg>
- [2] <https://blog.23andme.com/23andme-research/exercise-can-modify-dna-in-fat-cells/>
- [3] <http://www.freegreatpicture.com/other/robot-picture-12455>
- [4] <https://s-media-cache-ak0.pinimg.com/originals/f3/bd/eb/f3bdebf5a62337b5705cae7ea2199c1c.jpg>