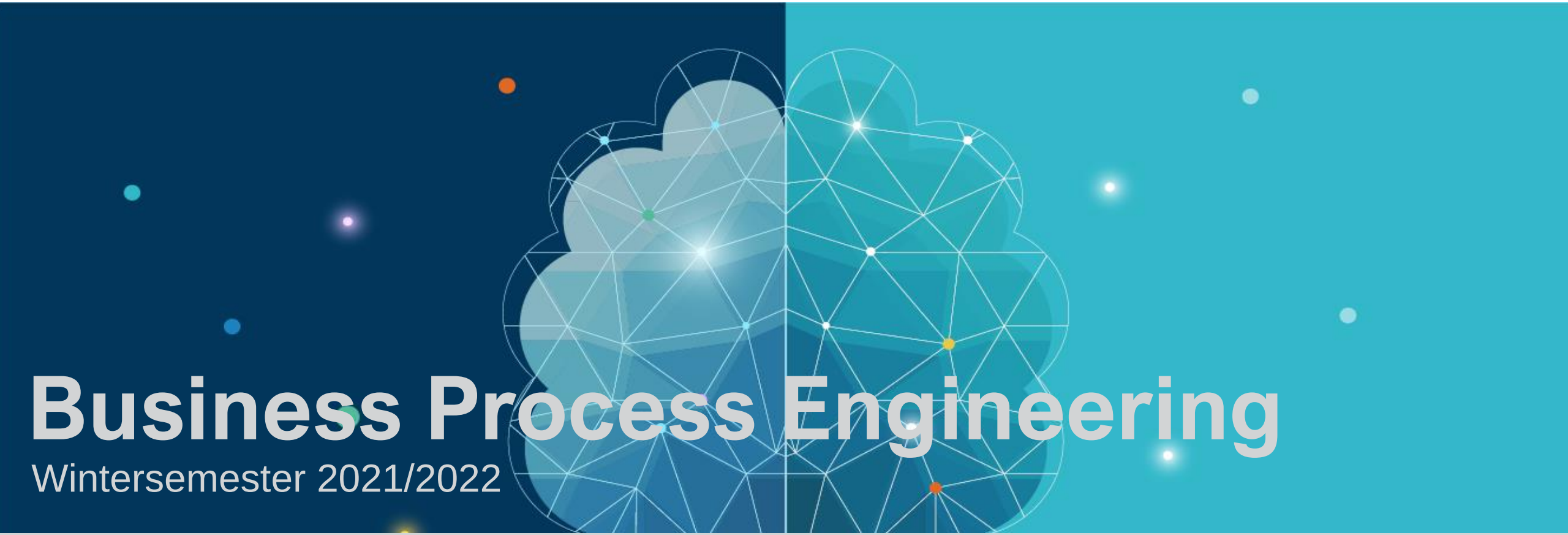
UNIKASSEL VERSITÄT

octaVIA
SAP-Kompetenz und Branchenwissen



Business Process Engineering

Wintersemester 2021/2022

Software Testing & Qualitätssicherung 2

Agenda

- Test Driven Development
- UI Tests
- Gauge + Taiko

Test Driven Development

- Deutsch: Testgesteuerte Entwicklung
- Methode, die häufig bei der agilen Entwicklung eingesetzt wird
- Programmierer erstellt Softwaretests **vor** den zu testenden Komponenten

- Test First
  - Unit Tests werden vor dem eigentlichen Computerprogramm geschrieben
  - Mehrere fehlgeschlagene Unit Tests dürfen gleichzeitig existieren
  - Vorstufe der testgetriebenen Entwicklung

Test Driven Development nach Kent Beck

- Unit-Tests und deren Units werden parallel entwickelt
- Entwicklung in *Mikroiterationen* (wenige Minuten)
  - Drei Hauptteile (Red-Green-Refactor)
    - Red – Test für eine neue Funktionalität schreiben
    - Green – Programmcode mit möglichst niedrigem Aufwand ändern
    - Refactoring – Code „aufräumen“
- Schritte wiederholen, bis:
  - bekannte Fehler bereinigt sind
  - der Code die gewünschte Funktionalität liefert
  - dem Entwickler keine sinnvollen weiteren Tests einfallen

- Nachteil: hoher Aufwand

UI Tests

- Es werden folgende Aspekte geprüft:
  - Visuelles Design
  - Funktionalität
  - Usability
  - Performance
  - Compliance
- Prüfen wie die Applikation Anwenderinteraktionen ausführt
  - Mit Maus- und Tastatureingaben etc.
- Prüfen, ob visuelle Elemente richtig dargestellt werden und funktionieren

UI Tests spielen eine wichtige Rolle für die Produktivsetzung einer Applikation!

UI Tests

- Manuell
  - Ein User interagiert mit der UI, basierend auf definierten Testfällen.
  - Die Ergebnisse werden dokumentiert.
- Record and Playback Testing
  - Ein User führt einmal die Interaktionen mit der UI aus, welche aufgezeichnet werden.
  - Die Aufzeichnung wird in eine Skript umgewandelt, dieses kann automatisiert ausgeführt werden.
- Model based
  - Ein Modell für das System erstellen
  - Eingaben definieren
  - Erwartete Ergebnisse verifizieren
  - Tests ausführen
  - Tatsächliche Ergebnisse prüfen und abgleichen

UI Testing Checklist:

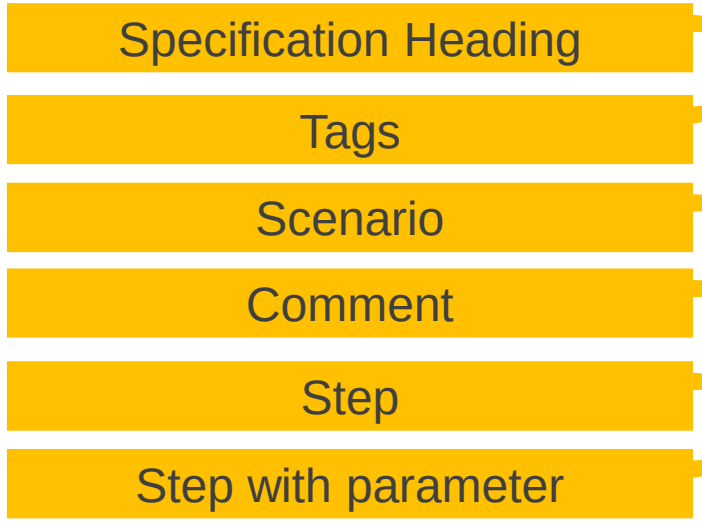
- Data type errors
- Feldlänge
- Navigationselemente
- Fortschrittsleisten
- Type ahead
- Table scrolling
- Error logging
- Menu items
- Working shortcuts
- Confirm action buttons
- …

UI Testing Frameworks: Gauge

- Kostenloses Open-Source Framework zum schreiben und durchführen von Acceptance Tests
- Nutzt Markdown Syntax
- Plattform / Sprachübergreifend
- Tests können prinzipiell auch von “Nicht Entwicklern” geschrieben warden
- Tests können manuell ausgeführt warden → Testprotokoll

https://gauge.org/

Specification




```
# Search specification
Tags: search, admin

The admin user must be able to search for available products on the search

## Successful search
Tags: successful

For an existing product name, the search result will contain the product na

* User must be logged in as "admin"
* Open the product search page
* Search for product "Cup Cakes"
* "Cup Cakes" should show up in the search results


## Unsuccessfull search

On an unknown product name search the search results will be empty

* User must be logged in as "admin"
* Open the product search page
* Search for product "unknown"
* The search results will be empty
```

Specification Heading

- Muss genau ein Mal vorhanden sein

```
# Spec Heading
```

oder

```
Spec Heading
============
```

Scenario

- Repräsentiert einen Workflow in einer Specification
- Es muss mindestens ein Scenario geben

```
## Scenario heading
```

oder

```
Scenario heading
----------------
```

Step

- Ausführbare Komponente einer Specification
- Kann in einem Scenario oder außerhalb existieren

```
* Search for product "Cup Cakes"
* "Cup Cakes" should show up in the search results
```

Parameters

- Simple Parameters
- Dynamic Parameters
  - Werden als Platzhalter benutzt, wenn man sich auf eine Tabellenspalte bezieht
- Table Parameters
  - Werden benutzt, wenn ein Step für mehrere Werte einer Tabelle ausgeführt wird
- Special Parameters
  - So können auch Tabellen und Files zu Parametern werden

Parameters


```
* Check "product 1" exists
* Check "product 2" exists
```

- Simple Parameters
- Dynamic Parameters
  - Werden als Platzhalter benutzt, wenn man sich auf eine Tabellenspalte bezieht
- Table Parameters
  - Werden benutzt, wenn ein Step für mehrere Werte einer Tabelle ausgeführt wird
- Special Parameters
  - So können auch Tabellen und Files zu Parametern werden

Parameters

```
# Create projects

    |id| name       |
    |--|-----------|
    |1 | Alice      |
    |2 | Bob        |
    |3 | Eve        |

## First scenario
* Say "hello" to <name>.

## Second scenario
* Say "namaste" to <name>.
```

- Simple Parameters
- Dynamic Parameters
  - Werden als Platzhalter benutzt, wenn man sich auf eine Tabellenspalte bezieht
- Table Parameters
  - Werden benutzt, wenn ein Step für mehrere Werte einer Tabelle ausgeführt wird
- Special Parameters
  - So können auch Tabellen und Files zu Parametern werden

Parameters

```
# Create projects

## First scenario

* Create the following projects
     |project name| username |
     |------------|----------|
     | Gauge java | Daredevil|
     | Gauge ruby | Iron Fist|
```

gauge

- Simple Parameters
- Dynamic Parameters
  - Werden als Platzhalter benutzt, wenn man sich auf eine Tabellenspalte bezieht
- Table Parameters
  - Werden benutzt, wenn ein Step für mehrere Werte einer Tabelle ausgeführt wird
- Special Parameters
  - So können auch Tabellen und Files zu Parametern werden

Parameters

```
* Verify email text is <file:email.txt>
```

gauge

- Simple Parameters
- Dynamic Parameters
  - Werden als Platzhalter benutzt, wenn man sich auf eine Tabellenspalte bezieht
- Table Parameters
  - Werden benutzt, wenn ein Step für mehrere Werte einer Tabelle ausgeführt wird
- Special Parameters
  - So können auch Tabellen und Files zu Parametern werden. Syntax: <prefix:value>

Tags

- Helfen beim Filtern und Sortieren von Specifications und Scenarios

```
# Search specification
  Tags: search, admin

## Successful search
   Tags: successful
```

Comments

- Alles ohne Präfix ist ein Comment

```
# Search specification
Tags: search, admin

The admin user must be able to search for available products on the search

## Successful search
Tags: successful

For an existing product name, the search result will contain the product na
```

Step Implementierung

```
* Say "hello" to "gauge"
```

Gauge Step

```
step("Say <greeting> to <name>", function(greeting, name, done) {
    try {
        setTimeout(function () {
            // Code for step
            done();
        }, 1000);
    } catch(e) {
        done(e);
    }
});

// Handling errors in promises with done
step("Say <greeting> to <name>", function(greeting, name, done) {
    // Let promise1 be some promise we need to wait for.
    promise1
        .then(done)
        .catch(function(e) { done(e);});
});
```

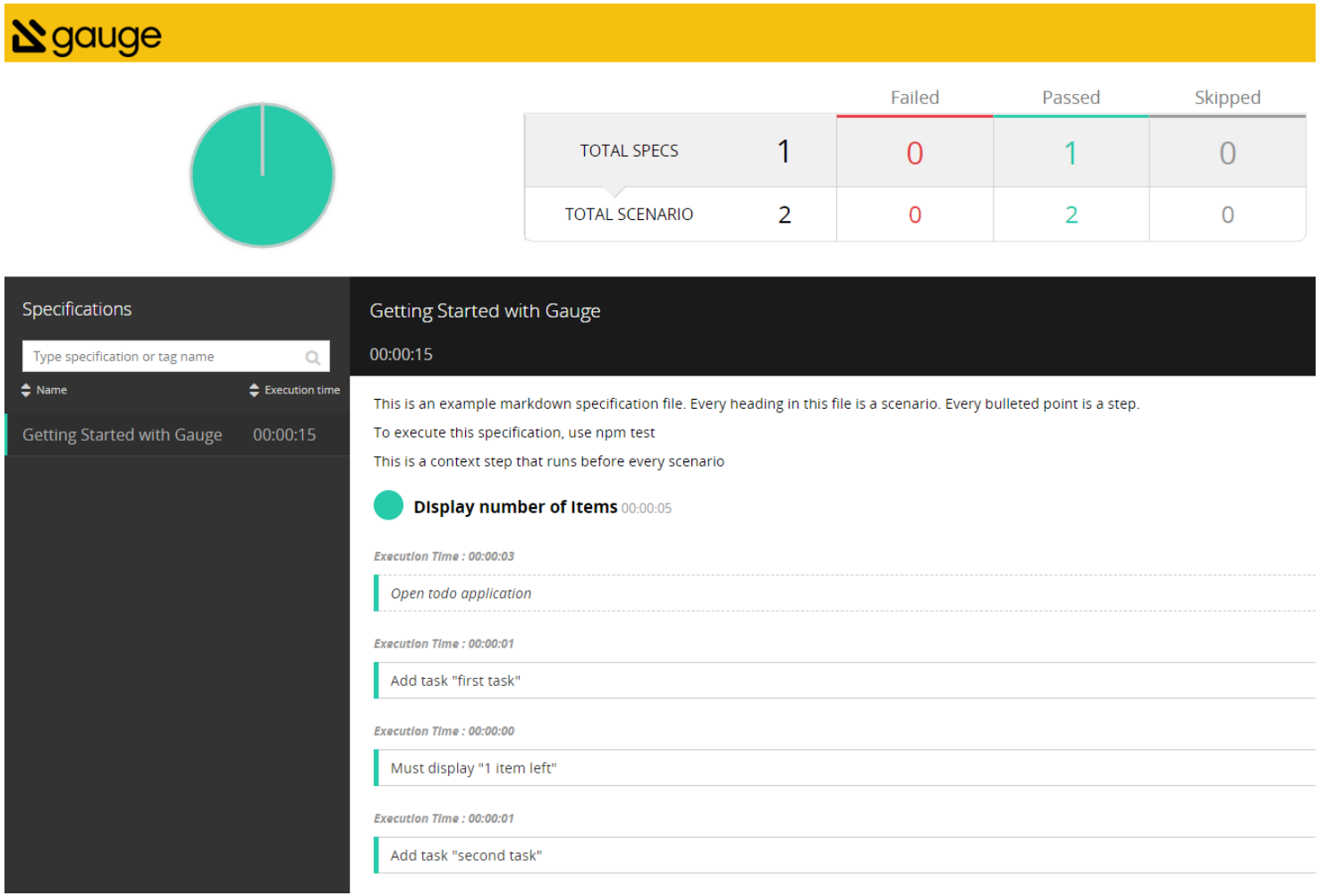
Step Implementation in JavaScript

Gauge Live Demo

specs > example.spec > ...

Run Spec | Debug Spec

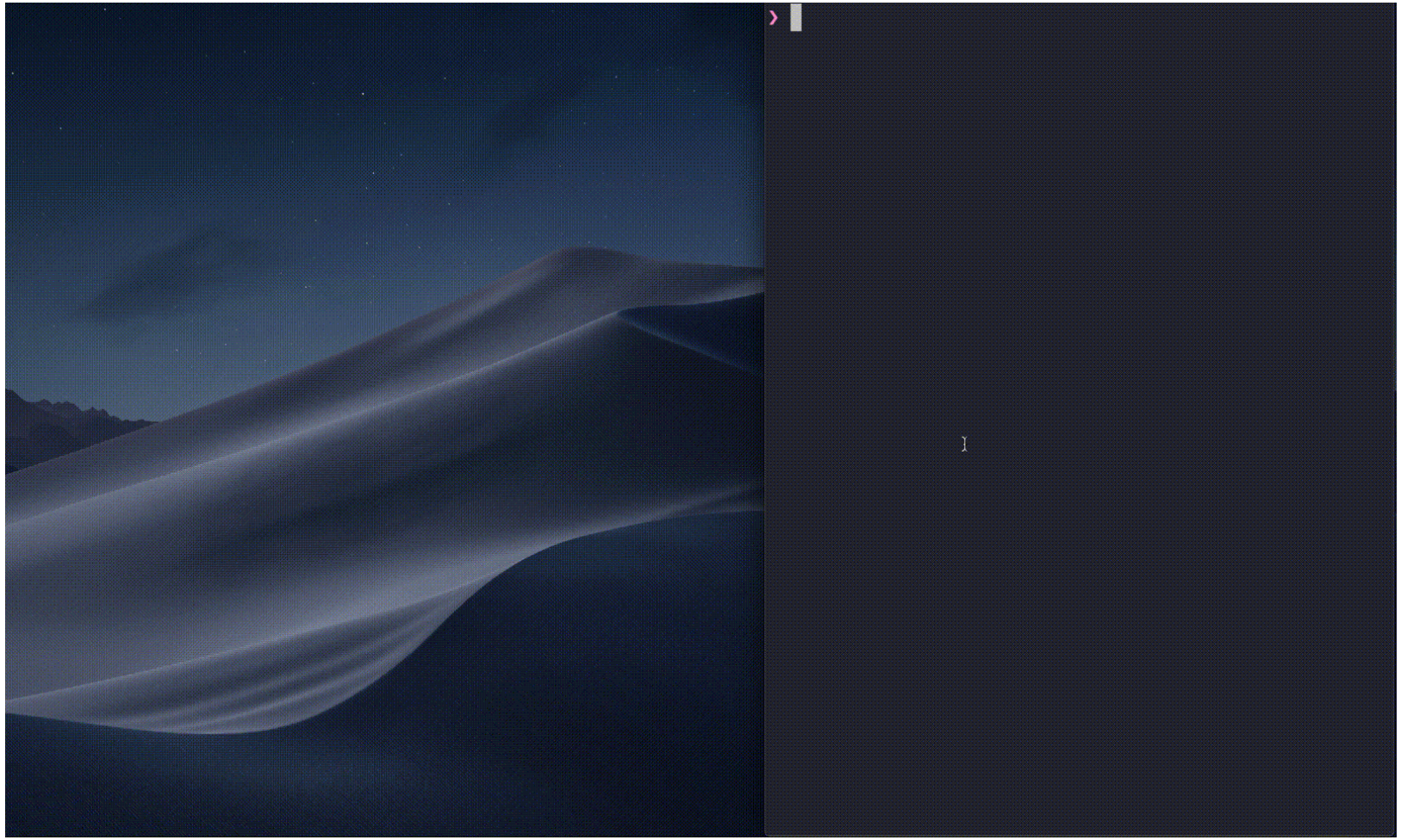
```
# Getting Started with Gauge

This is an example markdown specification file.
Every heading in this file is a scenario.
Every bulleted point is a step.

To execute this specification, use
    npm test

This is a context step that runs before every scenario
* Open todo application

Run Scenario | Debug Scenario
## Display number of items
* Add task "first task"
* Must display "1 item left"
* Add task "second task"
* Must display "2 items left"

Run Scenario | Debug Scenario
## Must list only active tasks
* Add tasks

    |description|
    |-----------|
    |first task |
    |second task|
    |third task |
    |fourth task|
    |fifth task |
```



Taiko

“Taiko is free and open source Node.js library with a simple API to automate Chromium based browsers (Chrome, Microsoft Edge, Opera) and Firefox.”

https://taiko.dev/



Quelle: https://taiko.dev/

Taiko

- API für „Blackbox Testing“
  - Kein Blick in den Seitenquelltext nötig – meistens ☺
- Skripte die direkt auf einer Website ausgeführt werden
  - „Headless“ oder „Observed“

```
1  const { openBrowser, goto, write,
2      click, closeBrowser, textBox } = require('taiko');
3
4  (async () => {
5      await openBrowser();
6      await goto("google.com");
7      await write("taiko test automation");
8      await press("Enter");
9      await closeBrowser();
10 })();
```

```
$ npx taiko tyko-demo.js
[PASS] Browser opened
[PASS] Navigated to URL http://google.com
Error: Element focused is not writable
    at C:\Users\asc\Desktop\gauge-test\uniks-bpe\node_modules\taiko\lib
    at async waitAndGetActionableElement (C:\Users\asc\Desktop\gauge-te
    at async write (C:\Users\asc\Desktop\gauge-test\uniks-bpe\node_modu
    at async module.exports.write (C:\Users\asc\Desktop\gauge-test\unik
    at async module.exports.<computed> (C:\Users\asc\Desktop\gauge-test
    at async global.<computed> (C:\Users\asc\Desktop\gauge-test\uniks-b
    at async C:\Users\asc\Desktop\gauge-test\uniks-bpe\tyko-demo.js:7:5
[PASS] Browser closed
```

Taiko Live Demo

Taiko vs. Selenium

Taiko

- Chrome DevTools Protokoll
- Kein Support für WebKit basierte Browser (z.B. Safari)
- Kein Support für mobile Applikationen
- Ausschließlich JavaScript Unterstützung

Selenium

- WebDriver
- Support beliebiger Browser
- Support für viele Sprachen, z.B. Java, Python, Ruby, PHP etc.
- Support für GUI basiertes Recording

Aufgabe 10

- Gauge + Taiko (Testen der Applikation aus Woche 8/9)

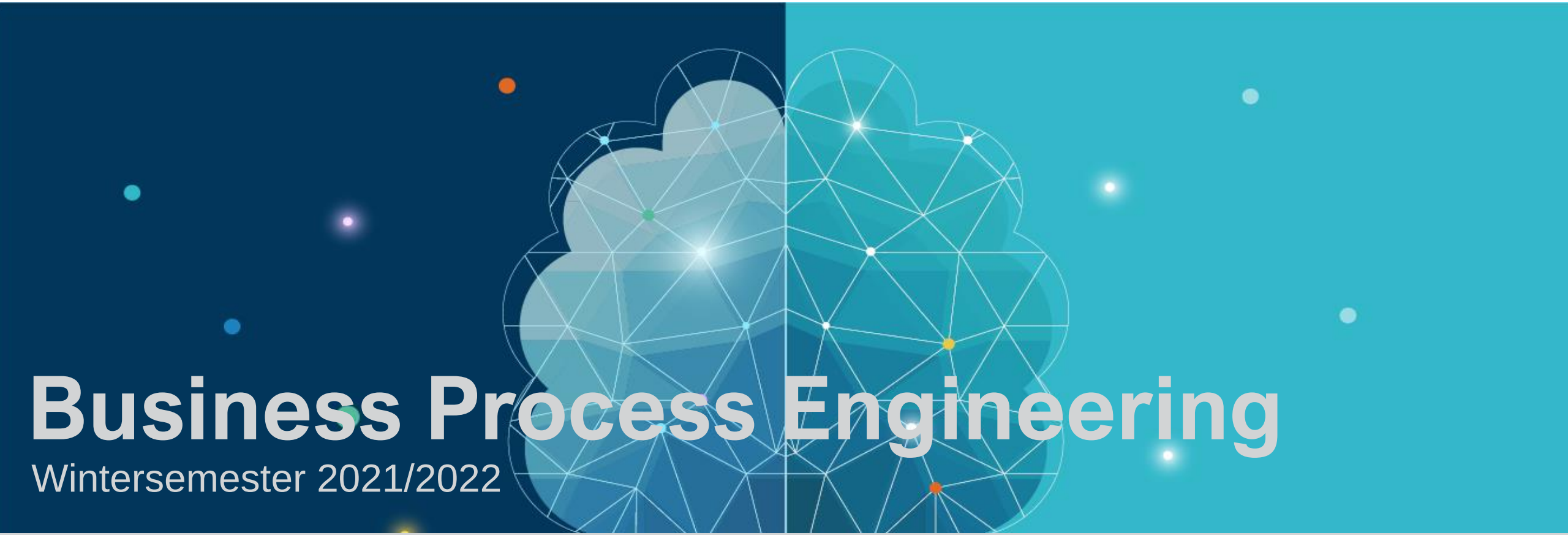
- Deadline:
  - Montag, 31.01.2022, 14:00 Uhr (1. Abgabe)
  - Donnerstag, 03.02.2022, 14:00 Uhr (2. Abgabe)

Quellen

- https://deepsource.io/blog/exponential-cost-of-fixing-bugs/
- https://blog.seibert-media.net/blog/2011/05/16/software-tests-notwendigkeit-arten/
- https://www.oliver-lampert.at/glossar/testarten/
- https://www.testingexcellence.com/seven-principles-of-software-testing/
- https://comquent.de/de/die-sieben-grundsaetze-des-softwaretestens/
- https://www.youtube.com/watch?v=3bJcvBLJViQ
- https://www.youtube.com/watch?v=Wi75S5TTfQ0
- https://blog.christianposta.com/deploy/blue-green-deployments-a-b-testing-and-canary-releases/
- https://de.wikipedia.org/wiki/Datei:Kde-bugtracking-via-bugzilla-firefox-1.0.6-kde-3.4.2-de.png
- https://www.redmine.org/projects/redmine/wiki/RedmineIssueList
- https://trac-hacks.org/wiki/ChildTicketsPlugin/Examples
- https://taiko.dev
- https://gauge.org

UNIKASSEL VERSITÄT

octaVIA
SAP-Kompetenz und Branchenwissen



Business Process Engineering

Wintersemester 2021/2022

Dr. Andreas Scharf