



Abschlusspräsentation

Projekt diva-e | HTW Berlin

Maximilian Berthold, Grace Leton Dodi, Tobias Lindhorst, Leander Piepenbring, Felix Riedel

19. September 2025



Agenda

1	Aufgabenstellung
2	Zeitplan / Planung
3	Finaler Stand
4	Vorstellung der Anwendung
5	Lessons Learned
6	Ausblick
7	Dokumentation



Projektziel:

Entwicklung eines Tools zum automatischen Erstellen von Cypress Tests unter Verwendung von KI-Tools

Cucumber File
Beschreibung des
Tests



Chatbot
(z. B. GPT)



Cypress Test e2e
Sourcecode als .ts

Anforderungen

diva^e

Must

- Automatische Generierung von Cypress-Tests
- Automatisches Einlesen vom Seiten-Quelltext
- Dokumentation zur Nutzung

Should

- Automatische Validierung generierter Dateien
- Chain-of-Thought (Zusammenhang der Informationen erkennen)
-> Agent-Architektur
- VS-Code Plugin

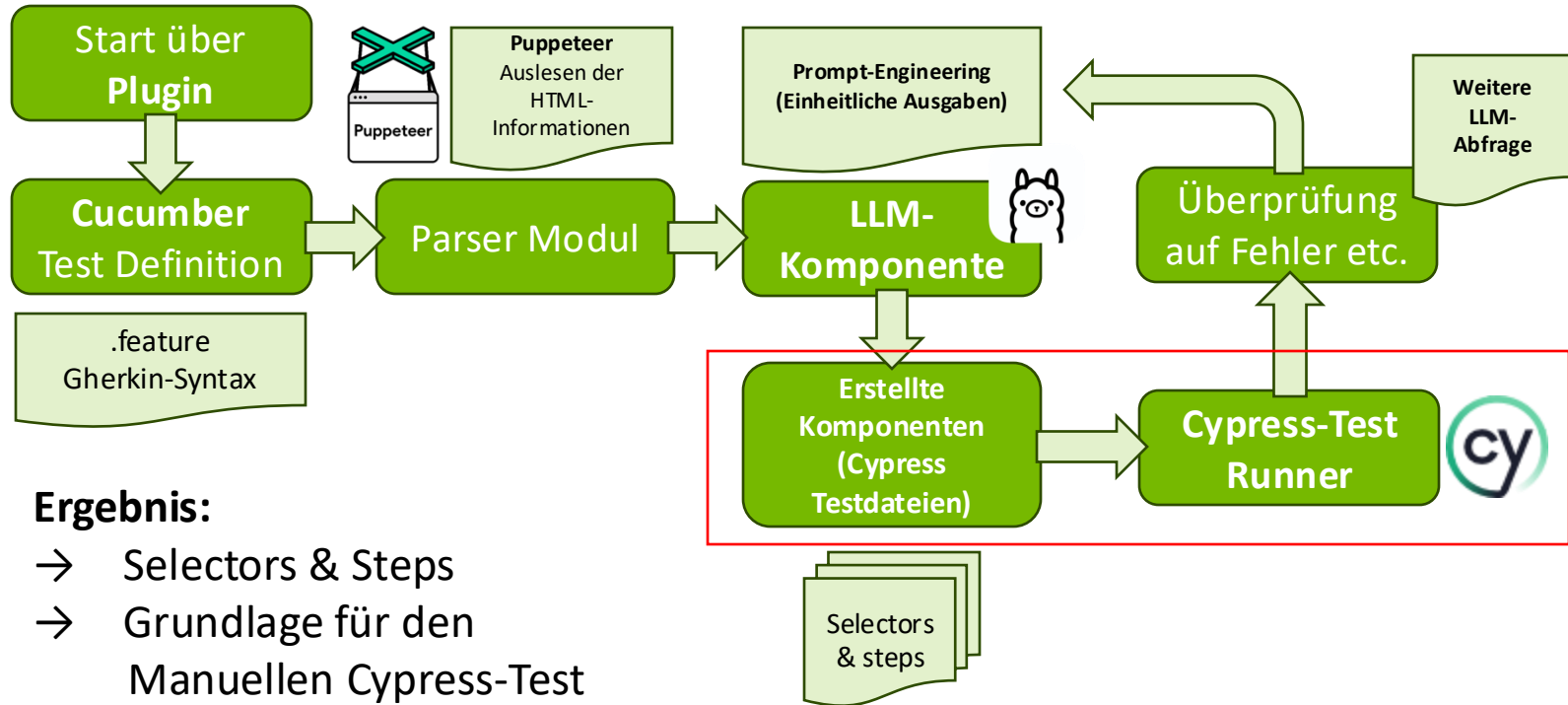
Could

- Automatische Auswertung der Testergebnisse durch KI

Won't

- Manuelle Nacharbeit der generierten Testfälle
- App oder visuell aufwändiges Overlay

Ablauf der Anwendung – Zwischenpräsentation



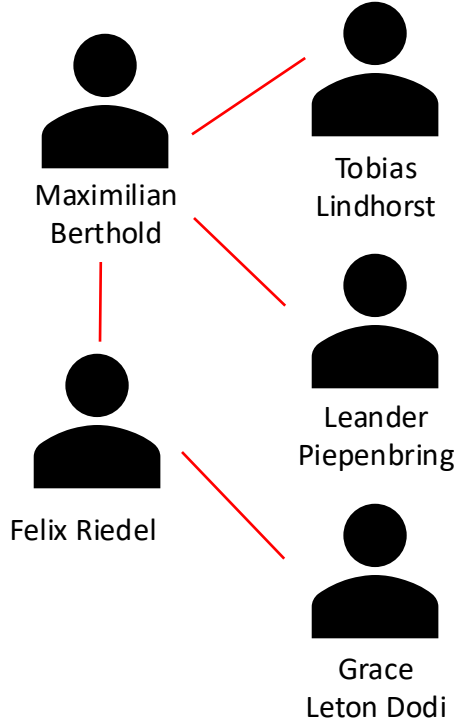
Agenda

1	Aufgabenstellung
2	Zeitplan / Planung
3	Finaler Stand
4	Vorstellung der Anwendung
5	Lessons Learned
6	Ausblick
7	Dokumentation

Aufgabenverteilung

- Projektleitung
- Prompting
- Agent-Logik

- Parser Modul
- Puppeteer
- Optimierung



- Agent-Logik
- Orchestrierung

- LLM-Tests

- VS-Code Plugin

diva^e



Vier-Augen-Prinzip

Genutzte organisatorische Anwendungen

Meetings



- Auftraggeber-Meetings
- Auftraggeber-Kommunikation



- Interne Kommunikation
- Informations-Austausch

Koordination



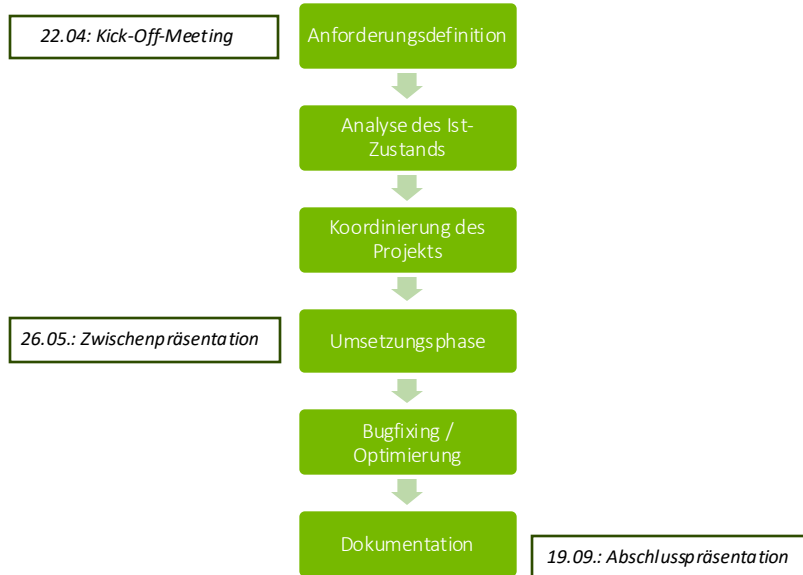
- Anfängliche Ideenfindung
- Meeting-Vorbereitungen



- Kanban-Board
- Aufgabenverteilung
- Allgemeiner Überblick

Zeitplan / Planung

- Absprache im zwei-Wochen-Rhythmus
 - Rahmenbedingungen und Prioritätsbestimmung
 - Aktueller Stand, Probleme und Vorgehen



Wichtige Meilensteine / Erkenntnisse



Erste funktionierende Durchläufe

Erste Generierung von "Cypress-artiger".ts-Dateien

Erster funktionierender Durchlauf eines einfachen Tests

Komplexer werdende Szenarien

Optimale LLM-Auswahl

Durchführung multipler Tests

Diverse Open-Source LLMs

Code-basierte LLMs
→ Ergebnis:
llama3.2 / gemma2

Agenten-Logik

Strukturierung des Codes

Zuweisung und Unterteilung von Aufgaben der Agenten

Bessere Prompt-Steuerung

Test-Steps-Erweiterung

Tests mit verschiedenen Websites

Ausgabe-Optimierung durch "Schablonen-Generierung"

Verworfenne Ideen – Verlauf der Entwicklung

OpenAI-Anbindung

OpenAI-API wurde zum
Start des Projekts
kostenpflichtig



Beschränkung auf lokal
laufende LLMs

→ Empfehlung mit
cloudbasierten LLMs zu
testen

LLM-Auswahl-Menü

Verzicht auf LLM-
Auswahl-Menü

Unterschiedliche LLMs
hätten unterschiedliche
Prompts benötigt

Für Kürze des Projekts
war das nicht nötig

→ Mögliche Ergänzung

Verifikations-Agent

Verzicht auf
"Endgültige" LLM-
Verifikation

Hat Dateien teils
"verschlechtert"

Zwischenlösung mit
"Selectors-Verification-
Agent"

→ Mögliche Ergänzung

GUI

Grafische
Benutzeroberfläche zur
Wahl von URLs und
LLMs

Für Prototypen zu
aufwändig

Lösung per „Rechtsklick“
ausreichend

→ Mögliche Ergänzung

Agenda

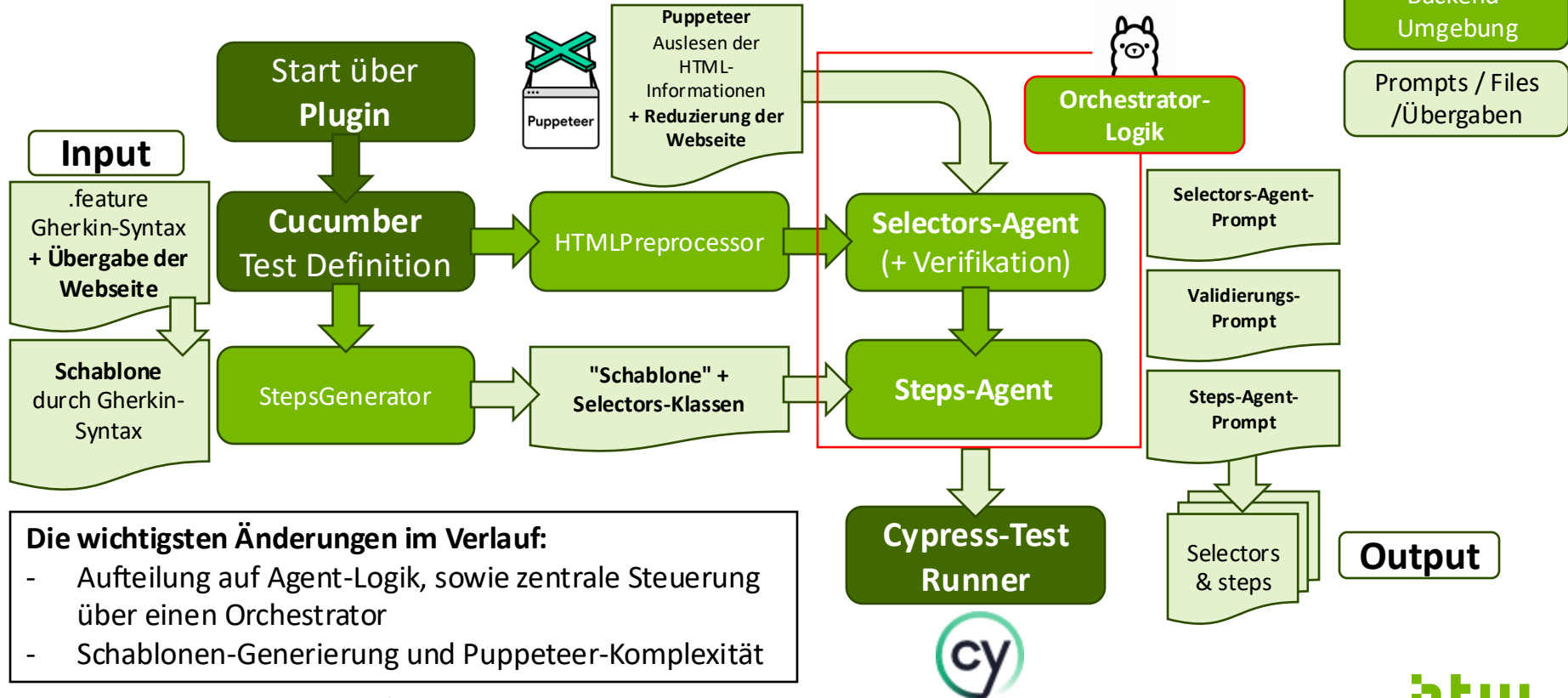
1	Aufgabenstellung
2	Zeitplan / Planung
3	Finaler Stand
4	Vorstellung der Anwendung
5	Lessons Learned
6	Ausblick
7	Dokumentation

Genutzte Anwendungen / Libraries



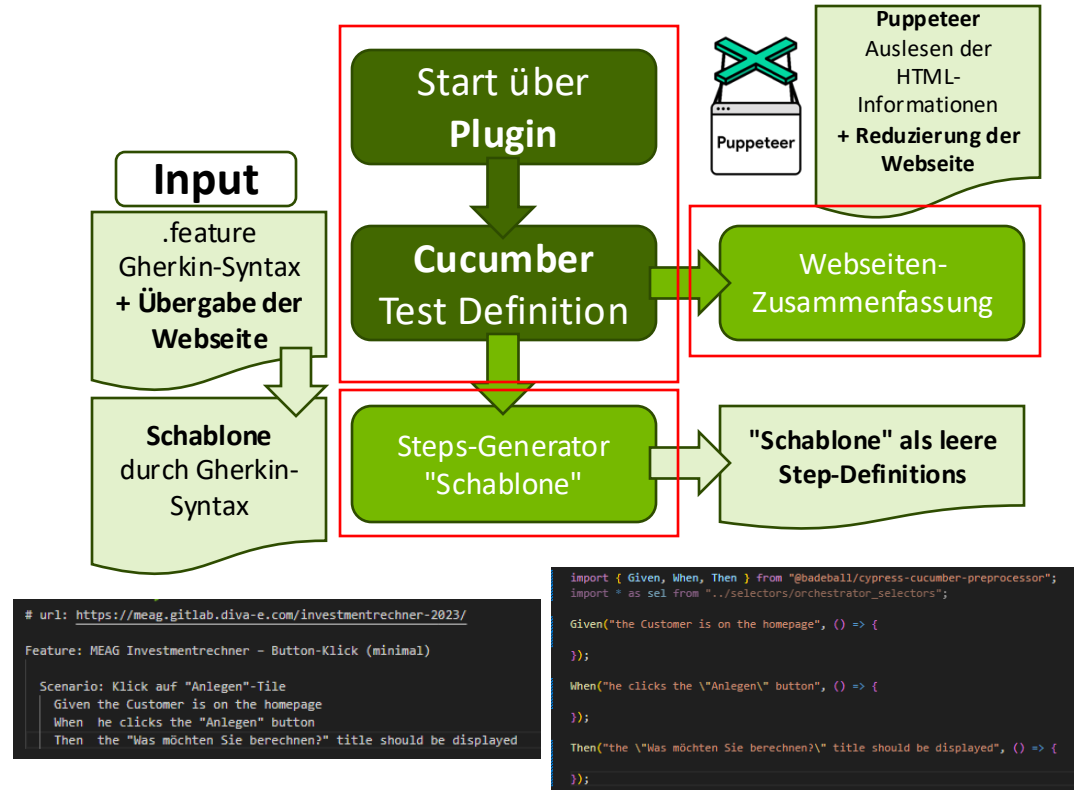
Tool	Zweck
VSCode	Allgemeine Entwicklungsumgebung (IDE)
TypeScript / Node.js	Sprache & Laufzeit / Paketverwaltung
GitHub	Versionsverwaltung, Zusammenarbeit
Cypress / Cucumber	Test-Framework für Webanwendungen + Gherkin-Logik
Ollama	Lokales Ausführen von LLMs
Llama3.2	Aktuell verwendetes Modell
Puppeteer	Auslesen der HTML Informationen
LangChain	Orchestrierung & LLM-Anbindung (Core + @langchain/ollama)
TSDoc / TypeDoc	Dokumentation - API-Doku (TypeDoc) & Kommentar-Standard (TSDoc)

Modellierung der Anwendung

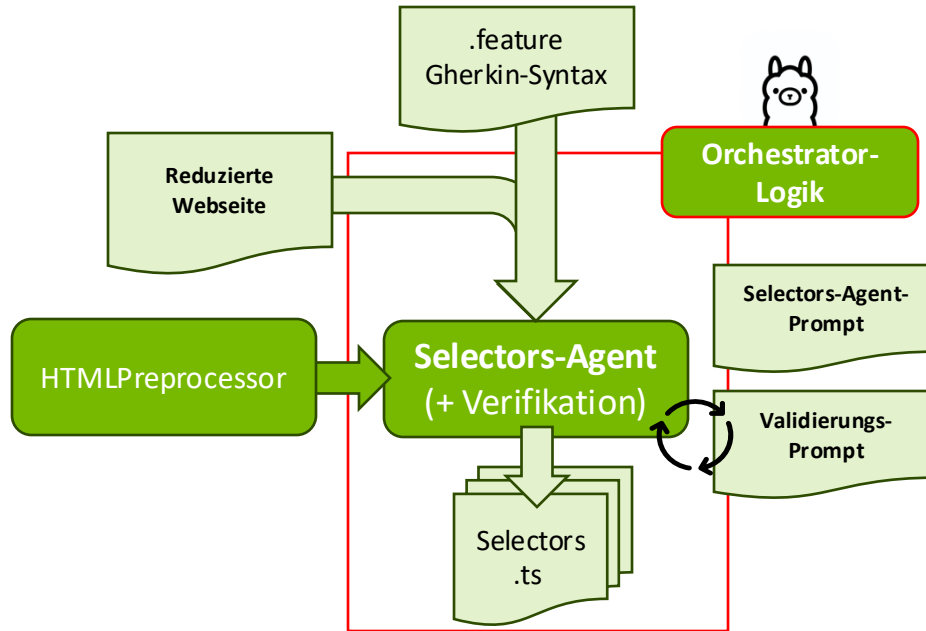


1. Einlesen der Input-Daten

- **Start** der Anwendung über integrierte **Extention.ts**
 - Rechtsklick auf die gewünschte Feature-Datei
 - Erweiterung durch eingebauten URL-Switch
- **Voraussetzung:**
 - Ollama läuft
- **Anstoß:**
 - HtmlPreprocessor.ts
 - StepsGenerator.ts



2. Erstellen der Selectors-Datei

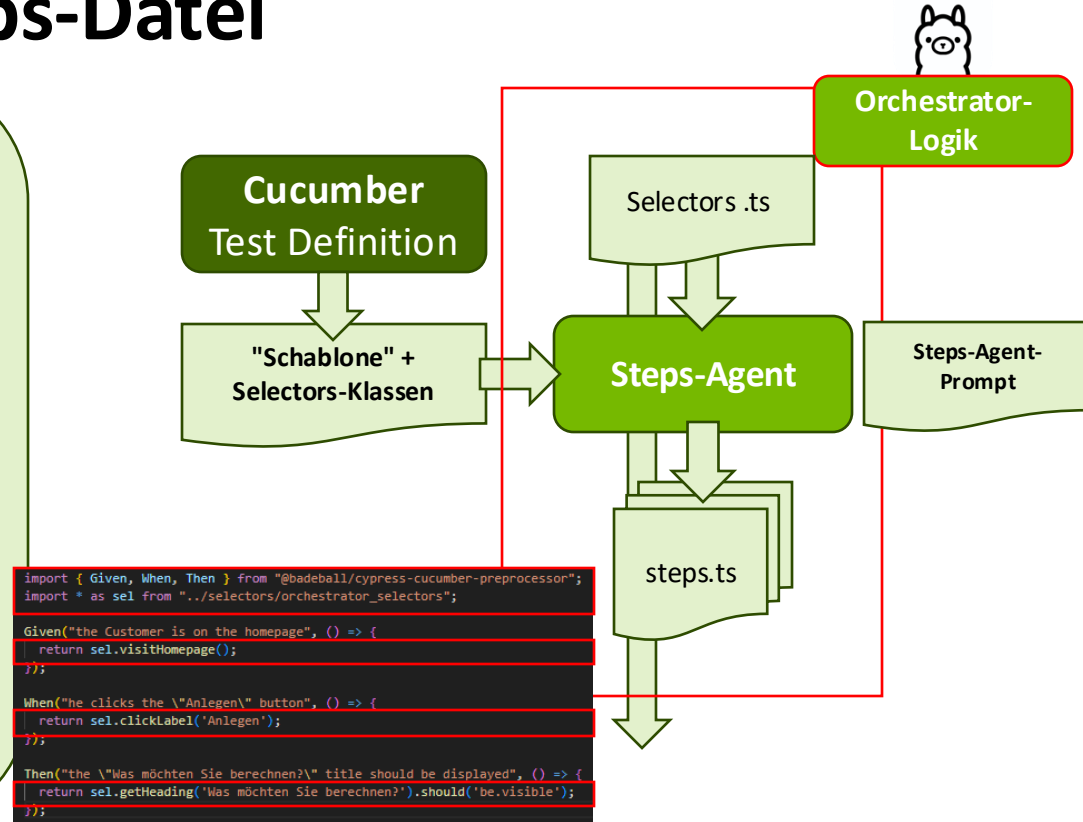


```
export function getHeading(label: string) {  
  return cy.contains('h1:visible, h2:visible, [role="heading"]:visible', String(label));  
}  
  
export function getLabel(label: string) {  
  return cy.contains(':visible', String(label));  
}
```

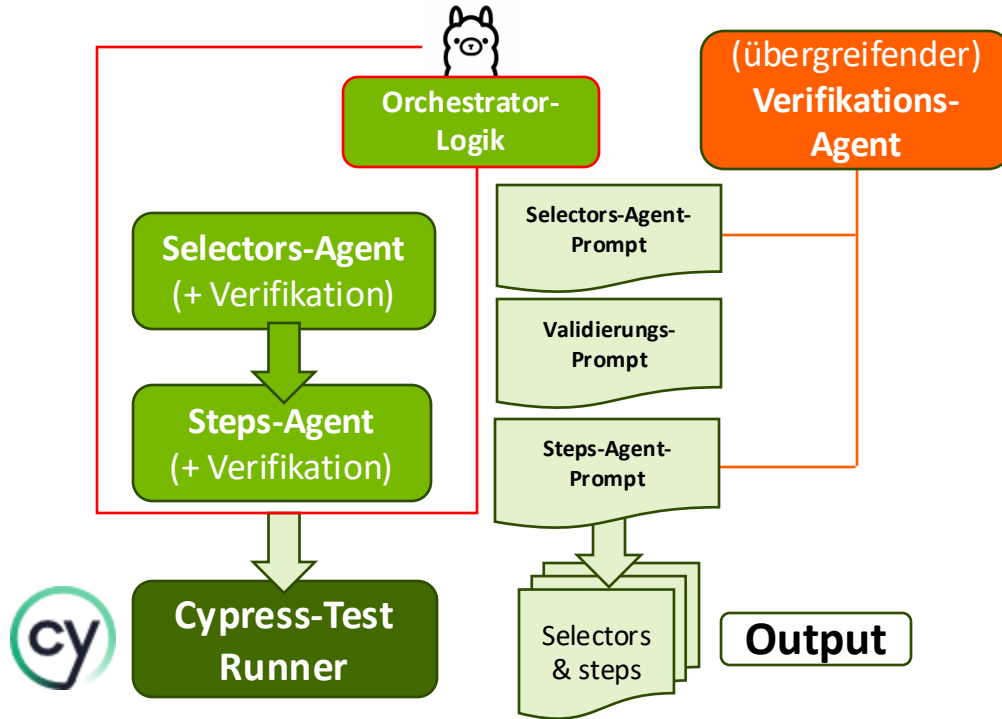
- **Ziel:** TS-Helferfunktionen für Cypress generieren
- **Input:**
 - Webseiten-Zusammenfassung
 - Feature-Text
- **Agent-Pipeline:**
 - Prompt bauen
 - Modellaufruf
 - Bereinigung (Sanitizing)
 - Heuristik-Check
 - Hardening / Normalisierung
 - Retry (strenger)
 - Deterministische Auto-Fixes
 - RefactorAgent

3. Erstellen der Steps-Datei

- **Input:**
 - Feature-Text
 - Steps-Schablone
 - Selectors-Helper (Phase 2)
- **Agent-Pipeline:**
 - Prompt bauen
 - TypeScript Header Imports
 - Aktionen / Assets über **sel.**
 - Exakte Step-Logik
 - LLM-Aufruf
 - Bereinigung
 - Validierung (Imports, Schablonen-Match)
- **Output:**
 - Ausführbare Step-Definitionen (TS)
- **Typische Fehler:**
 - Multiple / Undefined Steps verhindern
 - Importpfade und fehlende Elemente



4. Durchführung der Cypress-Tests

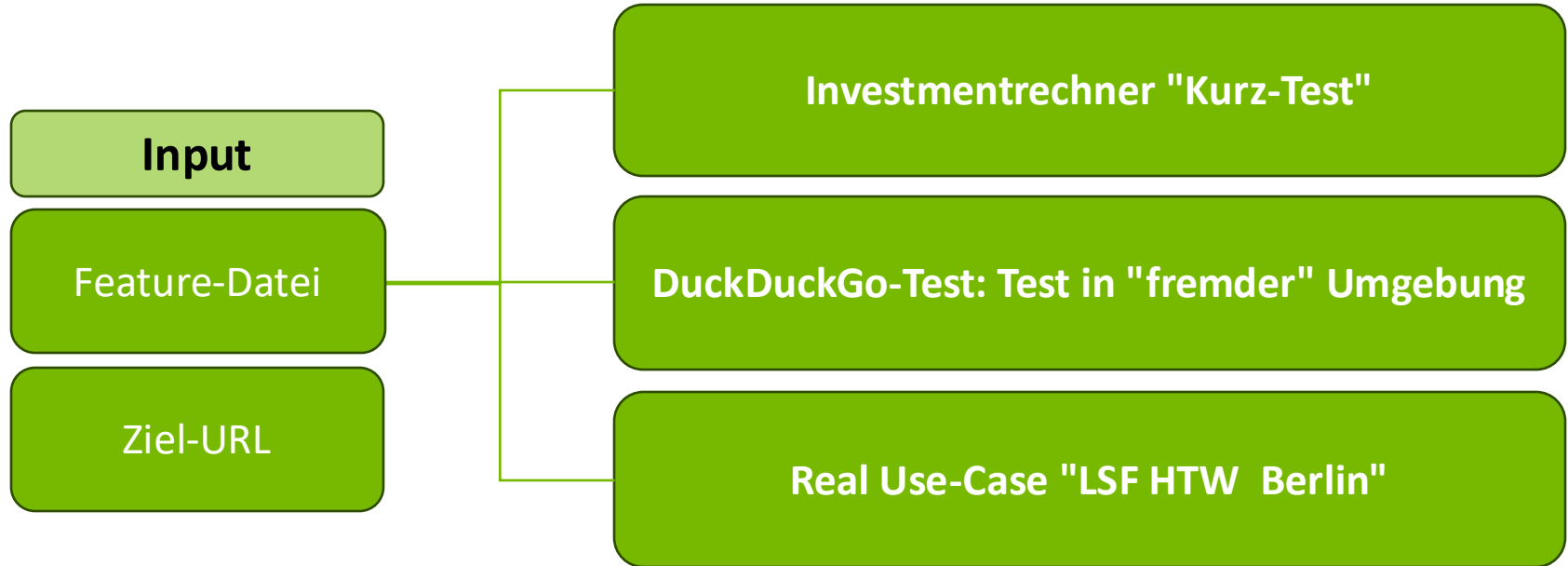


- **Voraussetzungen:**
 - `orchestrator_selectors.ts`
 - `orchestrator_steps.ts`
 - Feature-Datei (Szenario) + URL
 - `Cypress.config.ts` (Cucumber-Preprocessor)
 - Cypress-Ordnerstruktur
- **Start:**
 - **GUI:** `npx cypress open`
 - **Headless:** `npx cypress run -- browser...`
- **Ablauf im Runner:**
 - Lädt **Feature** → mappt auf **Steps**
 - Steps rufen **Selectors-Helfer** auf
 - Automatisches **Waiting** durch Cypress-Chains (bis Bedingung erfüllt oder Timeout)
→ **Output:** Test - Pass / Fail
- **Future-Work:**
 - Verifikations-Agent

Agenda

1	Aufgabenstellung
2	Zeitplan / Planung
3	Aktueller Stand
4	Vorstellung der Anwendung
5	Lessons Learned
6	Ausblick
7	Dokumentation

Use-Cases (Test-Szenarien)



Live-Demo

Erstellung eines Cypress-Tests durch KI



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Agenda

1	Aufgabenstellung
2	Zeitplan / Planung
3	Finaler Stand
4	Vorstellung der Anwendung
5	Lessons Learned
6	Ausblick
7	Dokumentation

Lessons Learned

Anwendung

- Intensive Kenntnisse über Cypress
- Nutzung von lokalen LLMs
- Promptengineering
- Eigenständige Test-Umgebung
- VSCode-Plugin-Entwicklung
- Limitierung von LLMs
- Interne Dokumentation
- Optimierung durch vorherige "Steps-Schablonen"

Organisatorisch

- Sinnvolle Aufgabenverteilung
- Intensive Nutzung von Kanban
 - Optimalere Planung der letztendlichen Umsetzung
- Arbeit mit Git

Agenda

1	Aufgabenstellung
2	Zeitplan / Planung
3	Aktueller Stand
4	Vorstellung der Anwendung
5	Lessons Learned
6	Ausblick
7	Dokumentation

Ausblick

Ziel-Erreichung



- Automatische Generierung von e2e Cypress-Tests
- Automatisches Einlesen vom Seiten-Quelltext
- Validierung der .ts-Dateien (nur selectors)
- VS-Code Plugin
- Dokumentation

Optimierungspotential



- Optimalere Code-Struktur (professionelles Aufräumen)
- VSC-Extension erweitern / veröffentlichen
- Cloud-basierte LLMs
- Standardisierte Selektoren
- Fehlerquote der Tests minimieren

Schlussfolgerung ("lohnt es sich weiterzumachen?"):

- **Ja – wenn** Zugang zu **leistungsfähigeren LLMs/Providers** möglich ist (GPT-Modelle sehr interessant)

Agenda

1	Aufgabenstellung
2	Zeitplan / Planung
3	Aktueller Stand
4	Vorstellung der Anwendung
5	Lessons Learned
6	Ausblick
7	Dokumentation

Dokumentation



- Wahl einer Projektordner-internen Dokumentation
 - **TypeScript Dokumentation**
 - Im Repo vorhanden und aktualisierbar
- **Vorteile:**
 - Interaktive Einbindung von Klassen und Beziehungen
 - Optimiert für TypeScript
- **Nachteile:**
 - Komplett neue Art der Dokumentation (für uns)
 - Ungeeigneter für "Nicht-TypeScript-Dateien"

Dokumentation



Gliederung:

1. ReadMe

- Zentrale Anleitung (Befehle, Vorgehen, Bibliotheken,...)
- Gliederung (einzelne Dateien)
- Erklärung von "nicht-TypeScript-Dateien"

2. TypeScript-Code erklärt pro Datei

3. Klassen und Beziehungen innerhalb dieser Dateien



Vielen Dank.

Gruppe diva-e

Maximilian Berthold	573037
Grace Leton Dodi	557312
Tobias Lindhorst	590867
Leander Piepenbring	571633
Felix Riegel	590945

www.htw-berlin.de



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences



**Hochschule für Technik
und Wirtschaft Berlin**

University of Applied Sciences

www.htw-berlin.de