



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences



Automatisierung und Optimierung von Frontend Unit-Tests mit Hilfe Künstlicher Intelligenz

24.02.2026 • Patrick Lenzen

INHALT

01

Hintergrund

02

Exp. Design

03

Auswertung

04

Anwendung

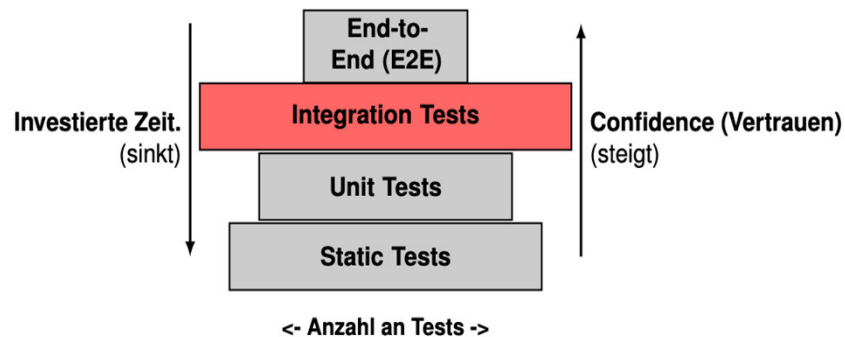


Abbildung 2: Die Testing Trophy – Teststrategie im Frontend (eigene Darstellung in Anlehnung an Dodds, 2021).

Teststrategie im Frontend:

Abweichung von der klassischen Pyramide hin zur Testing Trophy.

Das Problem:

Manuelle Unit-Tests haben ein extrem schlechtes **Zeit-Nutzen-Verhältnis** (unwirtschaftlich).

Das Ziel:

Einsatz von **KI**, um die Testerstellung radikal zu vereinfachen und wieder rentabel zu machen.

Real-World Data: → Echte Artefakte aus einer produktiven Codebasis

ID	Komponente	McCabe	Custom Score
C1	Action Slice	5	1
C2	Service	6	1
C3	Table	6	3
C4	Page	11	4

Technologie-Stack:

- (React, JavaScript, Redux)

McCabe-Metrik:

- Zeigt die Strukturelle Komplexität der Artefakte

Custom-Score:

- Erfasst spezifische Frontend-Hürden
- Eigenentwicklung

72 Testläufe

Objekte	Modelle	Temperatur	Prompting-Strategie
• C1 Actions • C2 Service • C3 Table • C4 Page	• Gemini 3 Pro • Claude Opus 4.5 • GPT-5.1	• 0.3 • 0.6 • 1.0	• Zero-Shot • Few-Shot

Kriterium	Fokus	Max. Punkte
Lauffähigkeit	Technische Lauffähigkeit	3
Korrektheit	Inhaltliche Richtigkeit	3
Abdeckung	Vollständigkeit	2
Qualität	Clean Code	2
Gesamt		10

Beispiel Halluzination



```
describe('ExtractActions exports', () => {  
  it('should export setLoading action creator', () => {  
    expect(ExtractActions.setLoading).toBeDefined();  
    expect(typeof ExtractActions.setLoading).toBe('function');  
  });  
  
  it('should export loadingCompleted action creator', () => {  
    expect(ExtractActions.loadingCompleted).toBeDefined();  
    expect(typeof ExtractActions.loadingCompleted).toBe('function');  
  });  
  
  it('should export reset action creator', () => {  
    expect(ExtractActions.reset).toBeDefined();  
    expect(typeof ExtractActions.reset).toBe('function');  
  });  
  
  it('should export updateData action creator', () => {  
    expect(ExtractActions.updateData).toBeDefined();  
    expect(typeof ExtractActions.updateData).toBe('function');  
  });  
});
```

Herkunft:

Actions_Claude_0.3_Zero

Realität:

Redux-Toolkit garantiert die Funktion

Halluzination:

Claude denkt, Funktionalität müsste getestet werden

Ursache:

Modell reproduziert Legacy-Muster

Beispiel Blender



```
it("ändert activeTab beim Klick auf Tabs (onChange wird aufgerufen)", () => {  
  useAuth.mockReturnValue(true);  
  const data = [];  
  setupState({ isInitialized: true, isLoading: false, data });  
  
  render(<ExtractPage />);  
  
  const extractsTab = screen.getByTestId("tab-extracts");  
  const accountancyTab = screen.getByTestId("tab-accountancy");  
  
  fireEvent.click(accountancyTab);  
  fireEvent.click(extractsTab);  
  
  expect(extractsTab).toBeInTheDocument();  
  expect(accountancyTab).toBeInTheDocument();  
});
```

Herkunft:

Page_GPT_1.0_Zero

Tautologie:

Test prüft Zustand, der schon vor der Aktion wahr war.

Gesamtergebnis

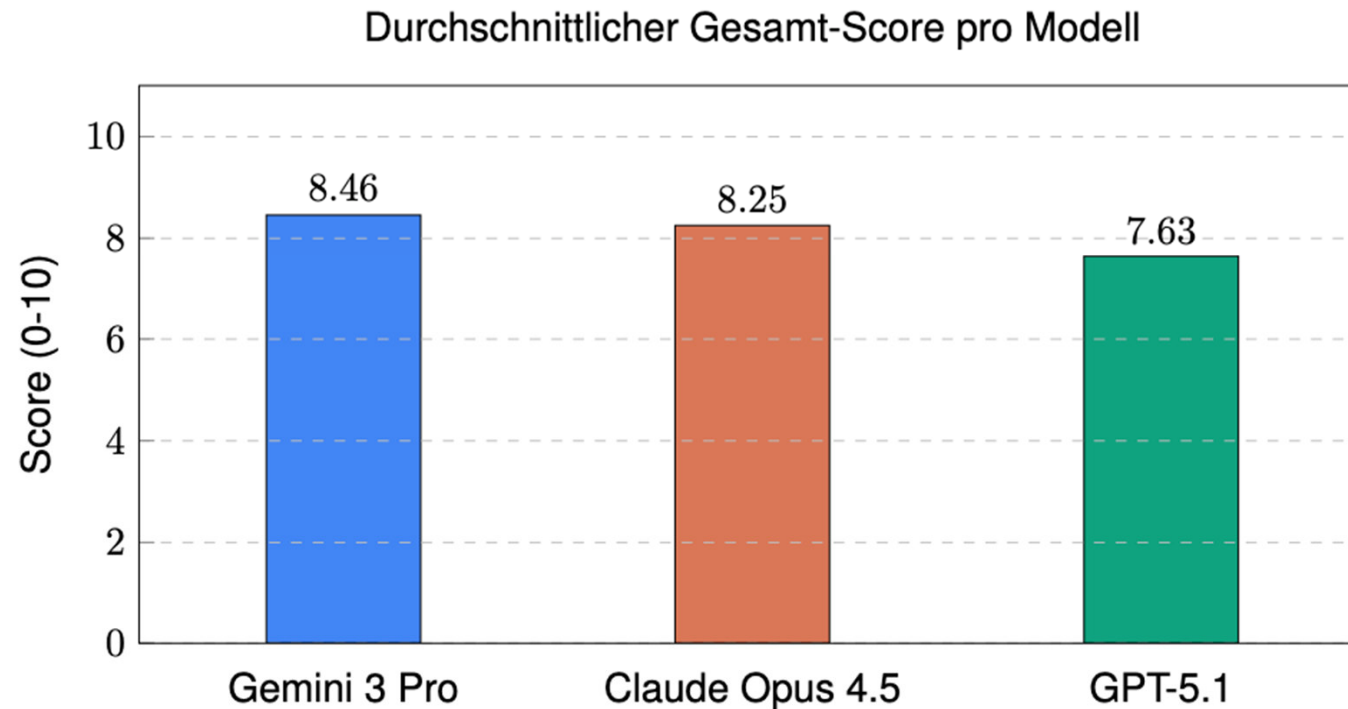


Abbildung 6: Vergleich der Modelle

Einfluss des Prompt-Engineering

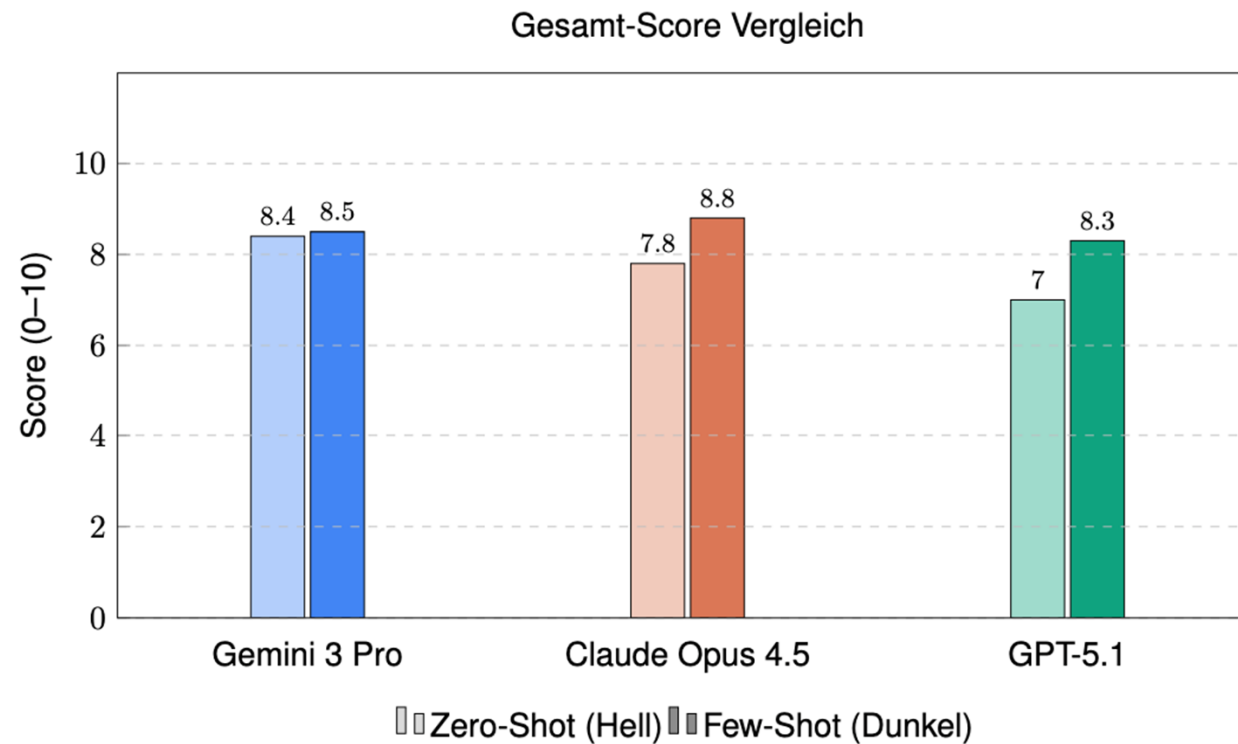


Abbildung 8: Vergleich der Prompting Strategien.

Beispiel Zero vs. Few



Herkunft:

Page_GPT_0.3_Zero

Testet Interna:

KI baut Redux-Middleware nach

Hohe Komplexität:

Logik ist schwer lesbar

```
it("dispatches checkStatus for each RUNNING item when initialized", () => {
  const data = [
    { STATUS: "RUNNING", DIM_CALENDAR_ID: 1 },
    { STATUS: "FINISHED", DIM_CALENDAR_ID: 2 },
    { STATUS: "RUNNING", DIM_CALENDAR_ID: 3 }
  ];

  useSelectorMock.mockImplementation(selector =>
    selector({
      extract: {
        isInitialized: true,
        isLoading: false,
        data
      }
    })
  );

  renderComponent();

  // useEffect for checkStatus dispatches for each RUNNING item
  // dispatch is called twice with a function (for 2 RUNNING items)
  const functionDispatchCalls = dispatchFn.mock.calls.filter(
    ([arg]) => typeof arg === "function"
  );
  expect(functionDispatchCalls.length).toBe(2);

  // Execute dispatched functions to assert Extract.checkStatus usage
  functionDispatchCalls.forEach(([fn], index) => {
    const innerDispatch = vi.fn();
    fn(innerDispatch);
    const runningItem = data.filter(d => d.STATUS === "RUNNING")[index];
    expect(ExtractMock.checkStatus).toHaveBeenCalledWith(
      innerDispatch,
      runningItem.DIM_CALENDAR_ID
    );
  });
});
```

Beispiel Zero vs. Few



Herkunft:

Page_GPT_0.3_Few

Testet Business-Logik:

KI prüft nur den Aufruf

Urteil:

Clean Code

```
it('startet Polling (checkStatus) nur für Items mit STATUS "RUNNING"', () => {
  useAuth.mockReturnValue(true);

  const runningId = 1;
  const finishedId = 2;

  useSelector.mockImplementation((selector) =>
    selector({
      extract: {
        isInitialized: true,
        isLoading: false,
        data: [
          { DIM_CALENDAR_ID: runningId, STATUS: 'RUNNING' },
          { DIM_CALENDAR_ID: finishedId, STATUS: 'FINISHED' },
        ],
      },
    })
  );

  render(<ExtractPage />);

  expect(Extract.checkStatus).toHaveBeenCalledTimes(1);
  expect(Extract.checkStatus).toHaveBeenCalledWith(dispatchMock, runningId);
  expect(Extract.checkStatus).not.toHaveBeenCalledWith(dispatchMock, finishedId);
});
```

Einfluss der Temperatur

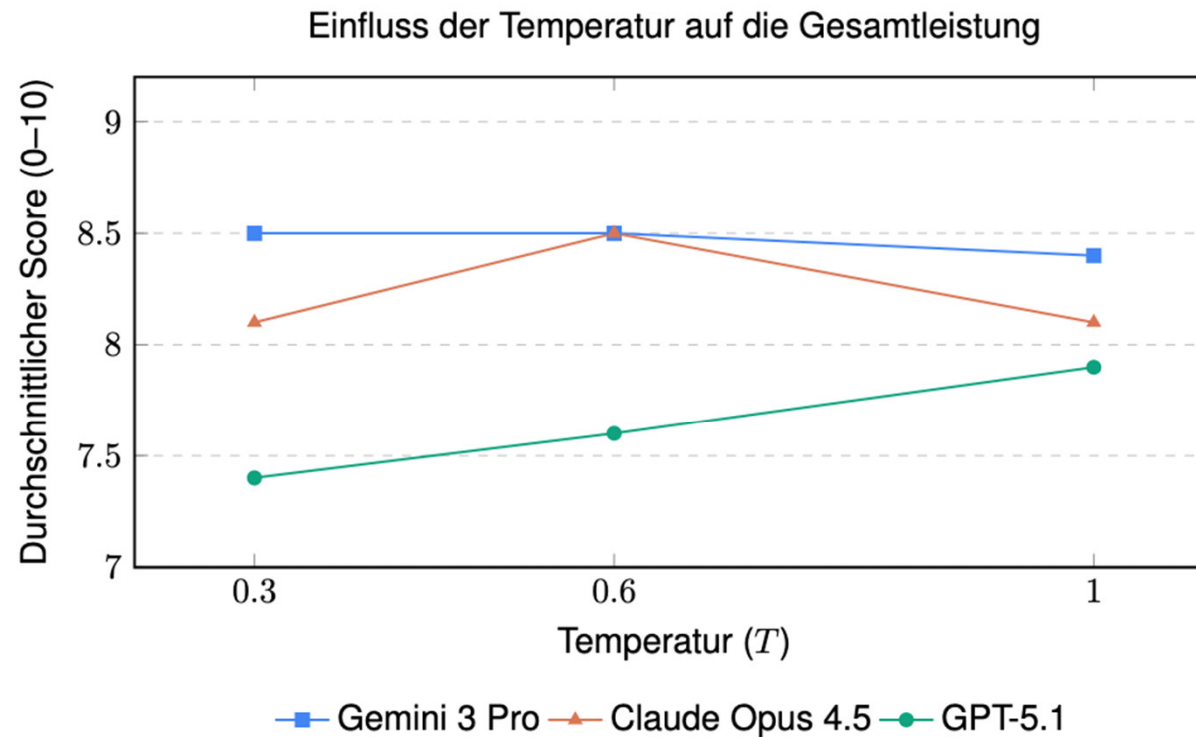


Abbildung 10: Performance-Verlauf in Abhängigkeit von der Temperatur

Gemini 3 Pro

Rolle: Primäres Werkzeug

Begründung:

- RWEST-Konform: Fokussiert präzise auf den kritischen Pfad.
- Robustheit: Liefert valide Ergebnisse im Zero-Shot verfahren.
- Wirtschaftlichkeit: Kein Overengineering → Minimierte Token-Kosten und Review Zeit.



Claude Opus 4.5

Rolle: Ergänzender Einsatz als Zweitmeinung

Begründung:

- Stärke: Deckt komplexe Edge Cases ab, die Gemini ggf. übersieht.
- Schwäche: Neigt signifikant zu Overengineering.



System-Governance:

- Nutzung von KI-Agenten zur Erstellung von Boilerplate-Code.
- Zentrale Hinterlegung von System Instructions in der Entwicklungsumgebung.

Rollenwandel:

- Vom Coder zum Reviewer
- Fokus-Shift: Validierung von Logik und Semantik statt manueller Syntaxerstellung.

1. Grundkurse:

- **Empfehlung:** Bewusster Verzicht auf KI-Tools
- **Warum:** Wer Syntax und Algorithmen nicht selbst durchdrungen hat, kann später keine *Halluzinationen* erkennen.
- **Ziel:** Aufbau von tiefem Verständnis ("Deep Dive") als Basis für spätere Validierung

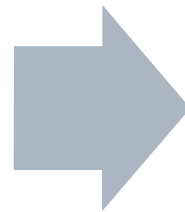
2. Vertiefung Software-Engineering

- **Empfehlung:** Verbindliche Integration & Lizenzen
- **Maßnahme:** Bereitstellung von Industriestandard-Tools (z.B. Copilot-Lizenzen) durch die Hochschule.
- **Ziel:** Realistische Vorbereitung auf den Arbeitsmarkt (Industrialisierung), wo diese Tools Standard sind.

Ökonomische Effizienzanalyse

Status Quo:

Fokus auf technischer
Machbarkeit & Qualität



Future Work:

Wirtschaftlichkeits-
Analyse nach GDPval –
Methodik



Vielen Dank!

Patrick Lenzen

24.02.2026

Literatur

- Arora, C., Sayeed, A. I., Licorish, S., Wang, F., & Treude, C. (2024). *Optimizing large language model hyperparameters for code generation*. arXiv. <https://doi.org/10.48550/arXiv.2408.10577>
- Garousi, V., Joy, N., Keleş, A. B., Değirmenci, S., Özdemir, E., Jafarov, Z., & Zarringhalami, R. (2024). *AI-powered software testing tools: A systematic review and empirical assessment of their features and limitations*. arXiv. <https://doi.org/10.48550/arXiv.2409.00411>
- Li, K., & Yuan, Y. (2024). *Large language models as test case generators: Performance evaluation and enhancement*. arXiv. <https://doi.org/10.48550/arXiv.2404.13340>
- Patwardhan, T., Dias, R., Proehl, E., Baker, B., Chng, P., O’Keefe, C., Wainwright, C., Lowe, R., Menick, J., Radford, A., & Neelakantan, A. (2025). *GDPval: Evaluating AI model performance on real-world economically valuable tasks*. arXiv. <https://doi.org/10.48550/arXiv.2510.04374>
- Tricentis. (2025). *2025 quality transformation report: Closing the gap between speed, quality, and trust in the AI era*. <https://www.tricentis.com/resources/quality-transformation-report>