

# Anforderungsanalyse und Technologieauswahl für eine automatisierte Testplattform für Vue.js-Anwendungen

Valentin Büttner – 80773  
Betreuer: Prof. Dr.-Ing. habil. Robert Geise

## Problemstellung und Zielsetzung

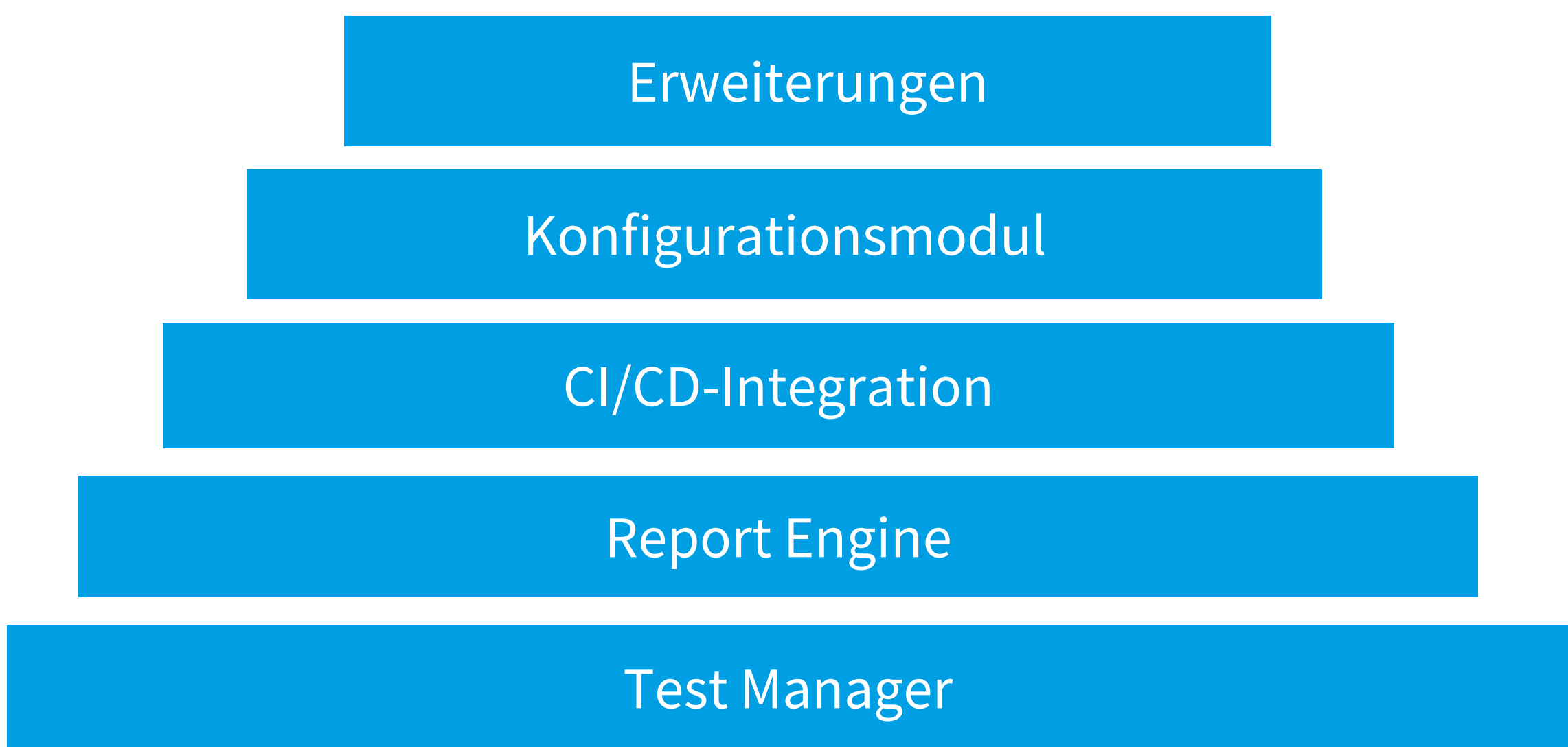
**Problemstellung**

- Steigende Anforderungen an Qualität und Geschwindigkeit in der Webentwicklung.
- Fragmentierte Testlösungen erschweren die Implementierung konsistenter Teststrategien.
- Effiziente und automatisierte Tests reduzieren Fehlerkosten und beschleunigen den Entwicklungsprozess.

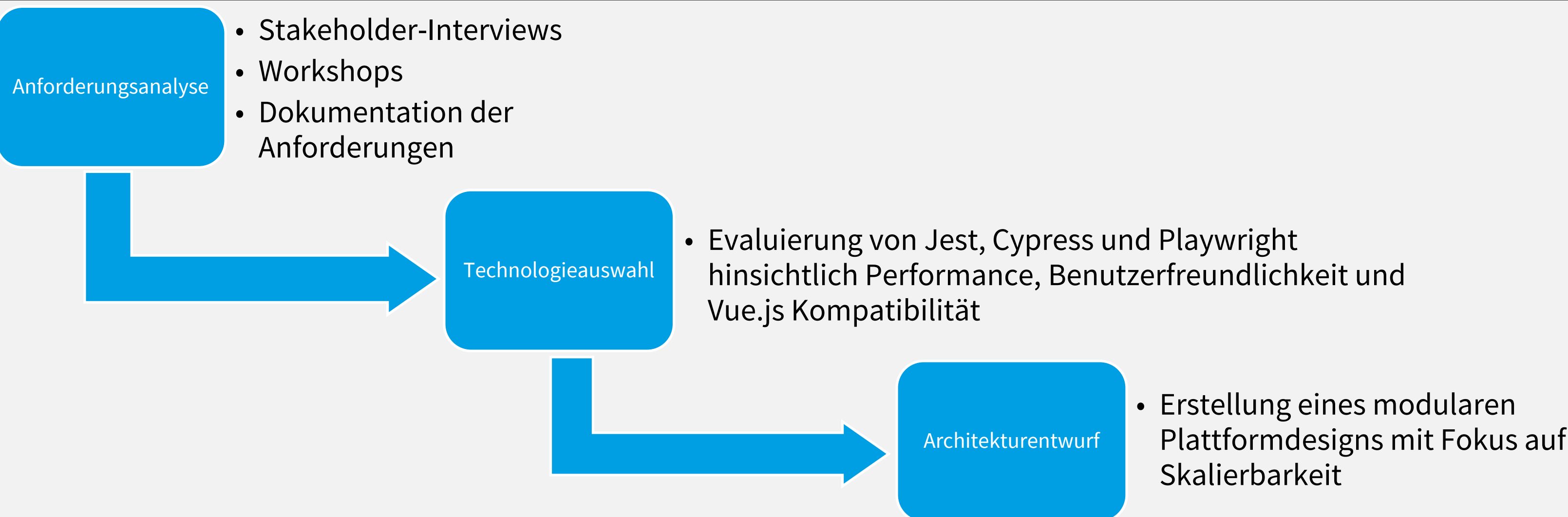
**Zielsetzung**

- Entwicklung einer modularen Plattform zur Unterstützung von Unit-, Integrations- und End-to-End-Tests für Vue.js-Anwendungen.
- Optimierung der Testprozesse durch Auswahl der besten Technologien und einer robusten Architektur.

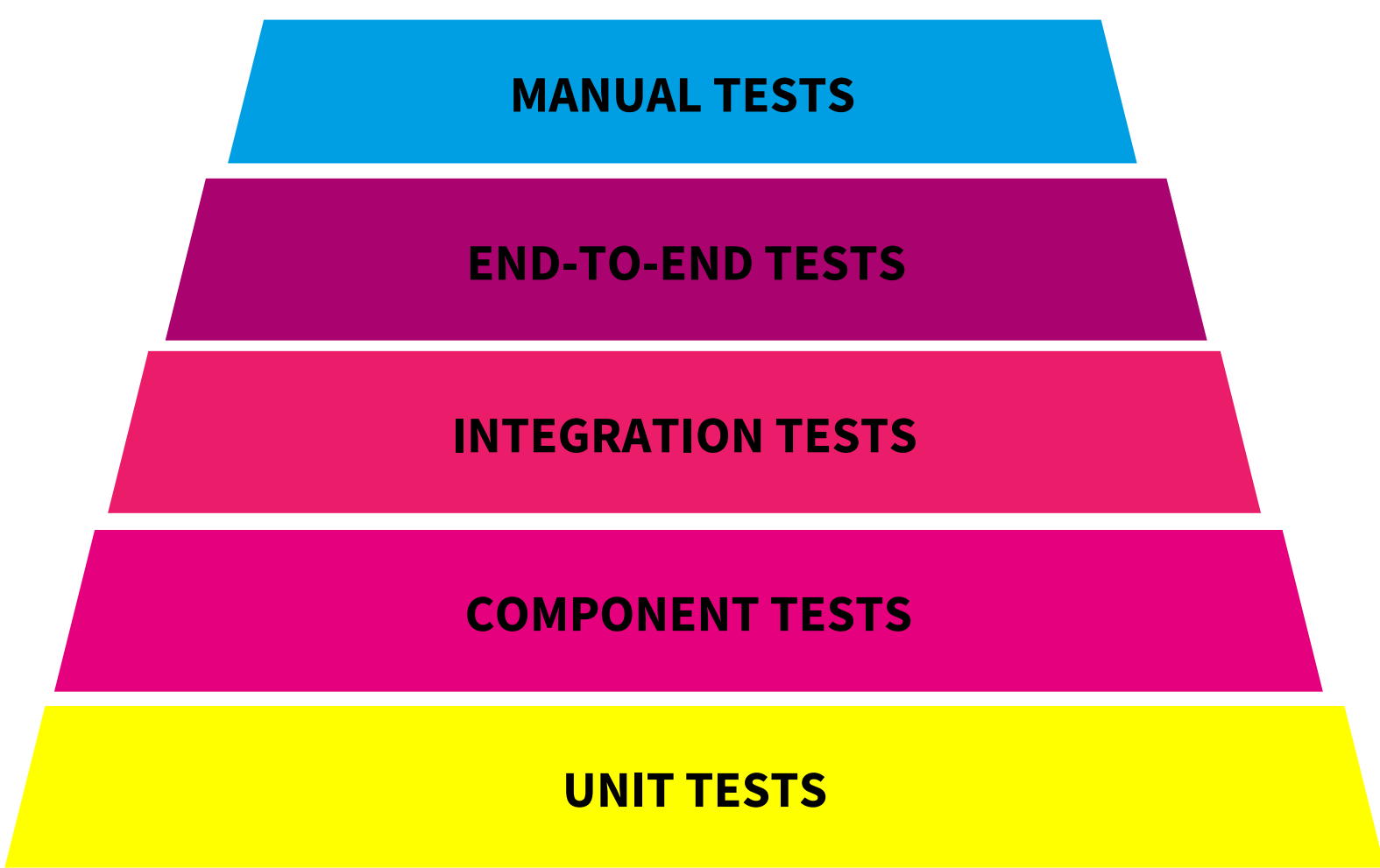
## Modularer Aufbau der Testplattform



## Durchführung des Entwicklungsprozesses



## Testpyramide



## End-to-End Test mit Cypress

```
describe('User Login Flow', () => {
  it('allows a user to log in and navigate to the dashboard', () => {
    // Besuche die Login-Seite
    cy.visit('/login');
    // Fülle das Login-Formular aus
    cy.get('input[name="username"]').type('testuser');
    cy.get('input[name="password"]').type('password123');
    cy.get('button[type="submit"]').click();
    // Überprüfe, ob der Benutzer auf das Dashboard weitergeleitet wurde
    cy.url().should('include', '/dashboard');
    // Überprüfe, ob das Dashboard korrekt geladen wurde
    cy.contains('Welcome, testuser').should('be.visible');
  });
});
```

## Unit-Test mit Jest

```
describe('capitalize', () => {
  const capitalize = (str) => {
    str.charAt(0).toUpperCase() + str.slice(1);
  };
  test('capitalizes the first letter of a word', () => {
    expect(capitalize('hello')).toBe('Hello');
  });
  test('returns an empty string if input is empty', () => {
    expect(capitalize('')).toBe('');
  });
});
```

## Ergebnisse der Anforderungsanalyse und Technologieauswahl

Aspekt	Tool/Framework	Alternative	Begründung
Unit-Tests	Jest	Vitest	<b>Jest</b> ist breiter etabliert, verfügt über eine umfassende Dokumentation und funktioniert auch außerhalb von Vue.js.
Integrationstests	Testing Library	Vue Test Utils	<b>Testing Library</b> ist benutzerzentriert und einfacher für das Testen ganzer Komponenten oder Workflows geeignet.
End-to-End-Tests	Cypress	Playwright	<b>Cypress</b> bietet eine einfache Einrichtung, eine intuitive API und eine starke Community-Unterstützung.
CI/CD-Integration	Gitlab CI	Jenkins	<b>GitLab CI</b> ist nahtlos in GitLab integriert, bietet weniger Konfigurationsaufwand und ist Cloud-nativ.

## Diskussion

- Plattform bietet klare Vorteile in der Vue.js-Entwicklung
- Potenzial zur Reduzierung der Entwicklungszeit und Erhöhung der Testabdeckung

## Ausblick

- Implementierung der Plattform als Teil der Bachelorarbeit
- Testabdeckung: Sicherstellen, dass mindestens 80% der Codebasis abgedeckt sind
- Leistungsbewertung: Laufzeit und Ressourcennutzung der Tests analysieren

## Quellen

- Vue.js „The Progressive JavaScript Framework“ Zugriff am 08.01.2025, <https://vuejs.org/>
- Jest. „Jest – Delightful JavaScript Testing.“ Zugriff am 08.01.2025, <https://jestjs.io/>
- Vitest. „Vitest – A blazing fast unit test framework powered by Vite.“ Zugriff am 08.01.2025, <https://vitest.dev>
- Vue Testing Library. „Vue Testing Library Documentation.“ Zugriff am 08.01.2025, <https://testing-library.com/docs/vue-testing-library/intro/>
- Cypress. „Cypress – Fast, Easy and Reliable Testing for Anything that Runs in a Browser.“ Zugriff am 08.01.2025, <https://www.cypress.io/>
- Playwright „Playwright – Reliable End-to-End Testing for Modern Web Apps.“ Zugriff am 08.01.2025, <https://playwright.dev>
- GitLab. „GitLab CI/CD – Automate Your Development Workflow with Ease.“ Zugriff am 08.01.2025, <https://docs.gitlab.com/ee/ci/>
- Jenkins. „Jenkins – The Leading Open Source Automation Server.“ Zugriff am 08.01.2025, <https://www.jenkins.io>