



DevOps ILV - Aufgabenstellung 3

Hochschule Burgenland
Studiengang MCCE
Sommersemester 2025

Harald Beier* Susanne Peer[†] Patrick Prugger[‡]

Philipp Palatin[§]

13. Mai 2025

*2410781028@hochschule-burgenland.at

[†]2410781002@hochschule-burgenland.at

[‡]2410781029@hochschule-burgenland.at

[§]2310781027@hochschule-burgenland.at

Inhaltsverzeichnis

1	Aufgabenstellung	3
1.1	Ausgangssituation	3
1.2	Ziel	3
1.3	Bearbeitungsschwerpunkte	3
1.3.1	1. Testumgebungsstabilisierung	3
1.3.2	2. Testarten und Abdeckung	3
1.3.3	3. Testeffizienz und Wartbarkeit	4
1.3.4	4. Reporting & Testtransparenz	4
1.3.5	5. Toolauswahl und Integration	4
1.4	Teamarbeit & Rollenverteilung	4
1.5	Abzugebende Materialien	4
1.6	Vorgesehene Bearbeitungsdauer	4
1.7	Lernziele	5
	Literaturverzeichnis	6

1 Aufgabenstellung 3

1.1 Ausgangssituation

Sie sind Teil eines Qualitätssicherungsteams in einem fiktiven Unternehmen, das eine E-Commerce-Plattform betreibt. Die Plattform bietet Kameraprodukte an, nutzt KI-gestützte Produktsuche, integriert verschiedene Backend-Systeme (z. B. SAP, AWS, NetSuite, HubSpot) und führt den Nutzer durch einen komplexen, serviceorientierten Kaufprozess.

Ein zentraler Bestandteil dieses Systems ist der in der beigefügten Prozessgrafik dargestellte End-to-End-Ablauf, vom Website-Besuch bis zur finalen Mail-Bestätigung. Die Architektur umfasst APIs, externe Services, Cloud-Provisionierung und ERP-Integration.

1.2 Ziel

Erarbeiten Sie als Gruppe eine ganzheitliche Teststrategie für dieses System. Die Strategie soll praktikabel sein, typische Herausforderungen im DevOps-Umfeld adressieren und die Integration verschiedener Testebenen, Tools und Umgebungen berücksichtigen.

1.3 Bearbeitungsschwerpunkte

Bitte erarbeiten Sie eine Ausarbeitung (max. 6 Seiten + Anhang), in der Sie folgende Punkte behandeln:

1.3.1 1. Testumgebungsstabilisierung

- Wie stellen Sie in der Entwicklungs- und Integrationsphase eine stabile Testumgebung sicher, insbesondere angesichts externer Systeme (SAP, AWS, GPT, HubSpot etc.)?
- Wie kann Service-Virtualisierung oder Mocking zum Einsatz kommen?
- Welche Datenanforderungen bestehen (z. B. synthetische Testdaten, Daten-Maskierung)?

1.3.2 2. Testarten und Abdeckung

- Welche Testarten (z. B. Unit-, API-, Integrations-, E2E-, Load-, Performance-, Security-tests) sind erforderlich, um:
 - die funktionalen Anforderungen abzudecken?
 - nicht-funktionale Anforderungen wie Performance, Security, Verfügbarkeit und Datenintegrität zu prüfen?
- Wo und wie werden diese Tests innerhalb der CI/CD-Pipeline ausgeführt?

1.3.3 3. Testeffizienz und Wartbarkeit

- Wie strukturieren Sie Tests, um gezielt auf Systemveränderungen (z. B. SAP-Upgrade) reagieren zu können?
- Wie nutzen Sie z. B. Impact Analysis, modulare Architekturen oder risiko-basiertes Testen, um Wiederverwendbarkeit und Selektivität zu ermöglichen?

1.3.4 4. Reporting & Testtransparenz

- Wo und wie sollen Testergebnisse dokumentiert und ausgewertet werden (z. B. Dashboards, Logs, automatisierte Reports)?
- Wer sind die Stakeholder für das Reporting (Dev, QA, Ops, Management)?

1.3.5 5. Toolauswahl und Integration

- Welche Testtools (open source und/oder kommerziell) schlagen Sie für die Umsetzung vor für z. B.:
 - Testautomatisierung
 - Performance-Testing
 - Service-Virtualisierung
 - Testdatenmanagement
 - Reporting & Testmanagement

1.4 Teamarbeit & Rollenverteilung

Versucht in der Ausarbeitung die folgenden Rollen einzunehmen:

- QA-Architekt: übergreifendes Testkonzept & Architektur
- Testanalyst: Spezifikation von Testfällen und Daten
- Tool-Integrator: Toolauswahl & CI/CD-Verknüpfung

Die Aufteilung ist eine Empfehlung, aber keine Pflicht.

1.5 Abzugebende Materialien

- Schriftliches Konzept (PDF, max. 6 Seiten ohne Anhang)
- Architektur- oder Ablaufdiagramm(e) zur Teststrategie
- Tabelle mit empfohlener Tool-Landschaft

1.6 Vorgesehene Bearbeitungsdauer

Gesamt in etwa 5 Stunden pro Person.

1.7 Lernziele

- Eine effektive Teststrategie im Kontext von DevOps
- Den Einsatz moderner Testtools konzeptionell bewerten
- Risiken in komplexen Integrationslandschaften erkennen und adressieren
- Testgetriebene CI/CD-Prozesse planen und visualisieren

Literaturverzeichnis