Ein **Testumgebungs-Leitfaden** beschreibt die technischen und organisatorischen Schritte zur Einrichtung einer stabilen Testumgebung. Er ist zentral für die Konsistenz und Reproduzierbarkeit der Testergebnisse und beschleunigt die Einarbeitung neuer Teammitglieder.

**1. Überblick**

Kurze Erläuterung der Umgebung und deren Zweck:

* Welche Arten von Tests finden statt?
* Handelt es sich um eine Staging-, QA- oder dedizierte Testumgebung?

**Beispiel:**

Dieses Dokument beschreibt die Einrichtung der Staging-Umgebung zur Durchführung von Regressionstests und Smoke-Tests im Modul „Login & Benutzerverwaltung“.

**2. Systemarchitektur**

* Auflistung der beteiligten Komponenten: Frontend, Backend, API, Datenbank
* Optional: Architekturdiagramm oder Beschreibung

**3. Voraussetzungen**

* Hardware: RAM, CPU, Betriebssystem
* Software: Java-Version, Browser, Test-Tools
* Zugangsdaten / Testbenutzer
* Netzwerkfreigaben, Proxy- oder Firewall-Regeln

**4. Schritt-für-Schritt Anleitung**

Klare Anweisungen zur Einrichtung:

* VM-Start oder Klonen
* Installation notwendiger Software (z. B. Java, Browser-Treiber)
* Setzen von Umgebungsvariablen
* Deployment der Anwendung
* Testdaten vorbereiten oder importieren
* Aktivieren von Testmodi oder Flags

**5. Testdatenbereitstellung**

* Woher kommen die Testdaten?
* Welche Datentypen: gültige/anonyme Benutzer, Sonderfälle, Rollen, Tokens

**6. Validierung der Umgebung**

* Ist die Anwendung erreichbar (UI/API)?
* Datenbankverbindung erfolgreich?
* Logging aktiv?
* CI/CD-Builds grün?

**7. Zurücksetzen / Rollback**

* Wiederherstellung nach dem Test
* Skripte zum Rücksetzen der Datenbank, Cache-Leeren, temporäre Dateien löschen

**8. Bekannte Einschränkungen**

* Instabile Komponenten
* Deaktivierte oder simulierte Drittanbieter-Integrationen

**9. Kontakte / Support**

* DevOps-Kontaktperson
* Testumgebungs-Admin
* Ansprechpartner bei Build-/Deployment-Problemen

A **Test Environment Setup Guide** documents all technical and procedural steps required to prepare the environment for manual or automated testing. It ensures consistency, repeatability, and faster onboarding of QA engineers.

**1. Overview**

Briefly explain the purpose of this environment:

* What kind of tests will be executed (e.g. functional, regression, automation)?
* Is this a staging, QA, or dedicated test environment?

**Example:**

This guide outlines the setup of a staging environment used for end-to-end regression and smoke testing of the Login and User Management modules.

**2. System Architecture**

* List involved components: frontend, backend, database, APIs, services
* Diagram (optional) or text-based overview

**Example:**

Components: Web Frontend (React), REST API (Node.js), MySQL DB, Auth Service

**3. Pre-requisites**

* Hardware specs (RAM, CPU, OS)
* Software required (Java version, browser versions, automation tools)
* Access rights or test accounts needed
* Required network access or firewall rules

**4. Step-by-Step Setup Instructions**

Provide clear, ordered steps for setting up the environment. Include:

* VM/server provisioning or cloning
* Software installation steps (e.g. Java SDK, browser drivers, test tools)
* Configuration of system variables
* Deploying application build
* Loading test data or dummy records
* Setting environment flags or test modes

**5. Test Data Setup**

* Location or method for inserting test data (manual import, scripts)
* Types of data: valid credentials, edge cases, admin users, expired tokens

**6. Environment Validation Checklist**

* Application reachable via browser/API?
* Database connected and seeded?
* Logs enabled?
* CI/CD pipeline status?

**7. Rollback / Reset Instructions**

* How to reset the environment after testing
* Scripts for DB restore, file cleanup, cache clear

**8. Known Issues / Limitations**

* Unstable modules
* Third-party integrations that are mocked or skipped

**9. Contacts / Support**

* DevOps engineer
* QA environment admin
* Emergency contact for build/deploy issues