Testkonzept

Moodle Migration auf Docker

|  |  |
| --- | --- |
| Klassifizierung | VERTRAULICH |
| Status | in Arbeit |
| Projektnummer | 1 |
| Projektleiter | Tiziano Corbetti |
| Version | 0.1 |
| Datum | 11. März 2023 |
| Auftraggeber | Thomas Keller |
| Autor/Autoren | Franziska Tobler, Tiziano Corbetti, Tibor Blasko |

Änderungsverzeichnis

| Version | Datum | Änderung | Autor |
| --- | --- | --- | --- |
| 0.1 | 11.03.2023 | Ausfüllen von Informationen | Franziska Tobler,  Tiziano Corbetti |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabelle 1: Änderungskontrolle

Beschreibung

Das Testkonzept beschreibt die Testziele, Testobjekte, Testarten, Testinfrastruktur sowie die Testorganisation. Es umfasst ebenfalls die Testplanung und die Testfallbeschreibungen. Für jeden Testfall wird eine detaillierte Testfallbeschreibung erstellt. Diese stellt die Spezifikation des Tests dar. Die Testplanung legt den logischen und zeitlichen Ablauf der Tests fest. Das Testkonzept bildet die Grundlage, auf der die Testorganisation und die Testinfrastruktur bereitgestellt und die Tests durchgeführt werden. Es wird bei neuen Erkenntnissen stets nachgeführt.

# Testziele

Globale messbare Testziele über alle Testfälle hinweg:

| Nr. | Beschreibung | Messgrösse | Priorität\* |
| --- | --- | --- | --- |
| 1 | Moodle läuft auf Docker | Seite lässt sich ohne Fehler aufrufen | M |
| 2 | SQL-Datenbank alle Daten migriert | Daten sind in der neuen SQL-Datenbank | M |
|  |  |  |  |
|  |  |  |  |
| \* Priorität: M = Muss / 1 = hoch, 2 = mittel, 3 = tief | | | |

Tabelle 2: Übergeordnete Testziele

# Teststrategie und Teststufen

Effiziente Testausführung setzt durchdachte (und evtl. agil einsetzbare) Teststrategie voraus:

* Mit welcher Teststrategie kann die Entwicklung begleitet und ergänzt werden, um die vorgegeben Qualität zu erreichen?
* Welche Teststufen gibt es / ab welcher Teststufe soll getestet werden?
* Wie können Überschneidungen von verschiedenen Teststufen vermieden werden (Verhinderung von Redundanzen)?
* Wie können Defizite der verfügbaren Testumgebung umgangen/gemeistert werden?
* etc.

# Testobjekte

| Nr. | Objekt | Beschreibung |
| --- | --- | --- |
| 1 | Localhost:80 | Webseite mit der Moodle aufrufbar ist auf dem Host |
| 2 | VM mit Docker | Der Host ist eine Virtuelle Maschine auf dem Docker installier ist. |

Tabelle 3: Testobjekte

# Testabdeckung

## Übersicht Testfälle

| Nr. | Testobjekt | Testfälle |
| --- | --- | --- |
| 1 | Localhost:80 | T-005, (T-006) |
| 2 | VM mit Docker | T-001, T-002, T-003, T-004 |

Tabelle 4: Testabdeckung

## Beurteilung Testziele und Testabdeckung

# Testrahmen

## Testvoraussetzungen

Der Tester soll im besten Fall eine Drittperson sein, die das System kennt, aber nicht am Projekt teilgenommen hat. Da dies in unserem Fall nicht möglich ist werden die Testfälle, wenn möglich von einem Teammitglied geprüft, das nicht den entsprechenden Teil bearbeitet hat. Falls das nicht geht, wird dieser Testfall von mehreren Personen überprüft.

Für alle Testfälle kommen deshalb im jetzigen Stadium des Projekts die folgenden Personen in Frage:

Tiziano Corbetti

Franziska Tobler

Tibor Blasko

## Mängelklassifizierung

Die festgestellten Mängel, bzw. die nicht erfüllten Anforderungen (Erwartungen), werden in Klassen von 1 bis 4 eingestuft. Die Klasse 0 findet nur dann Verwendung, wenn ein einwandfreies Ergebnis gesondert ausgewiesen werden soll:

| Nr. | Mängelklassen | Beschreibung |
| --- | --- | --- |
| 0 | mängelfrei | Einwandfrei und anforderungsgerecht |
| 1 | belangloser Mangel | Verwendung möglich, Brauchbarkeit ist vorhanden, Mängel sollte dennoch nicht vorkommen |
| 2 | leichter Mangel | Verwendung möglich, Brauchbarkeit ist nur wenig beeinträchtigt |
| 3 | schwerer Mangel | Verwendung ist noch möglich, Brauchbarkeit ist stark verringert |
| 4 | kritischer Mangel | Unbrauchbar;  Wesentliche Funktionalität ist nicht gegeben;  Betrieb ist nicht verantwortbar (z.B. sicherheitsspezifisch) |

Tabelle 5: Mängelklassen

Die Klassifizierung spiegelt die Folgenschwere und den Aufwand zur Behebung der möglich feststellbaren Mängel. Die Zuordnung der festgestellten Mängel zu einer Mängelklasse gibt grob auch die Priorität vor, in welcher Reihenfolge die Behebung der Mängel angegangen werden soll.

Wird eine Mängelklasse zwischen 1-3 erreicht, kann das System/Produkt unter Vorbehalt abgenommen werden. Zur Behebung der Mängel sind jedoch Massnahmen zu definieren. Eine Nachprüfung ist zwingend.

Werden hingegen Mängel der Klasse 4 festgestellt, kann das System/Produkt nicht abgenommen werden und der Auftragnehmer muss umgehend Massnahmen treffen, um diese Mängel zu beheben. Der Auftragnehmer hat zudem die erneute Abnahme zu veranlassen.

## Start- und Abbruchbedingungen

Vorbedingungen für Teststart:

Das System muss die aktuellsten Änderungen haben, d. h. auf dem Stand sein der von allen Projektmitgliedern als aktuell angesehen wird.

Abgebrochen wird, wenn beim Ausführen des Tests grobe Vorgehens Fehler passieren.

# Testumgebung

Die Testumgebung ist eine Virtuelle Maschine, diese wird so viel wie möglich unter den Projektmitarbeitern (inkl. Projektleiter) ausgetauscht, um auf dem gleichen Stand zu sein.

Diese VM hat das Betriebssystem Ubuntu 22.04 und Docker und Portainer sind vorinstalliert.

Diese VM wurde vom Auftraggeber zur Verfügung gestellt.

# Testinfrastruktur

## Testsystem

VM mit Ubuntu 22.04, Docker und Portainer sind schon vorinstalliert.

## Testdaten

Moodle-Daten und SQL-Datenbank.

# Testorganisation

# Testfallbeschreibungen

| ID / Bezeichnung | T-001 | Verwendung von Docker |
| --- | --- | --- |
| Beschreibung | Moodle läuft neu in Docker und ist als Microservice verfügbar. | |
| Testvoraussetzung | Moodle ist auf Docker migriert mit einem eigenen Docker Image | |
| Testschritte | Eigen geschriebenes Image funktioniert  Moodle ist auf Port 80 erreichbar | |
| Erwartetes Ergebnis | Die Container kommunizieren miteinander und Moodle ist erreichbar | |

Tabelle 6: Testfallbeschreibung

| ID / Bezeichnung | T-002 | Moodle Version |
| --- | --- | --- |
| Beschreibung | Moodle muss auf eine neue Version upgedatet werden. | |
| Testvoraussetzung | Die neue Moodle Version ist vorhanden | |
| Testschritte | Funktioniert das neue Moodle -> responsive | |
| Erwartetes Ergebnis | Neue Moodle-Version funktioniert | |

Tabelle 7: Testfallbeschreibung

| ID / Bezeichnung | T-003 | Übernahme SQL Datenbank |
| --- | --- | --- |
| Beschreibung | Die SQL-Tabelle, die für Moodle verwendet wurde, ist auf die SQL-Datenbank auf Docker verschoben und ist als Microservices vorhanden | |
| Testvoraussetzung | Datenbank ist auf Docker migriert | |
| Testschritte | Datenbank auf einem eigenen Container  Daten übernommen | |
| Erwartetes Ergebnis | Die Daten wurden auf den Datenbank Container übernommen | |

Tabelle 8: Testfallbeschreibung

| ID / Bezeichnung | T-004 | Übernahme SQL User |
| --- | --- | --- |
| Beschreibung | Die User der SQL-Datenbank müssen in die neue migriert werden. | |
| Testvoraussetzung | Die Datenbank ist auf Docker migriert | |
| Testschritte | Die User können in der User-Datenbank abgerufen werden.  Mit den Usern kann man sich bei der Datenbank einloggen. | |
| Erwartetes Ergebnis | User sind wieder gleich verfügbar wie vorhin. | |

Tabelle 9: Testfallbeschreibung

| ID / Bezeichnung | T-005 | Übernahme von Moodle Daten |
| --- | --- | --- |
| Beschreibung | Die Daten des alten Moodles sollen im neuen Moodle vorhanden sein. | |
| Testvoraussetzung | Volume ist erstellt und mit Moodle Container verbunden. | |
| Testschritte | Das Verzeichnis kann inspiziert werden.  Die Daten sind in Moodle abzurufen (Themes, Plugins, Kürse). | |
| Erwartetes Ergebnis | Das neue Moodle hat die gleichen Daten des alten Moodles. | |

Tabelle 10: Testfallbeschreibung

| ID / Bezeichnung | T-006 | Altes System lauffähig |
| --- | --- | --- |
| Beschreibung | Das alte System muss auf dem Port 8080 erreichbar sein | |
| Testvoraussetzung | Beim alten System muss die Konfiguration abgeändert werden | |
| Testschritte | Altes System auf Port 8080 erreichbar | |
| Erwartetes Ergebnis | Das alte System ist lokal auf dem Port 8080 erreichbar | |

Tabelle 11: Testfallbeschreibung

# Testplan

| Nr. | Aktivität | Verantwortlich | Mitarbeit | Termin |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Tabelle : Testplan

Abkürzungen und Glossar

| Abkürzung / Fachwort | Erläuterung |
| --- | --- |
| HERMES | Vorgehensmethodik für Projekte und Programme  HERMES 5 ist ein eCH Standard |
| VM | Virtuelle Maschine |
| SQL | Structured Query Language |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabelle 13: Abkürzungen und Glossar

Inhaltsverzeichnis

Änderungsverzeichnis 1

Beschreibung 1

1 Testziele 2

2 Teststrategie und Teststufen 2

3 Testobjekte 2

4 Testarten 2

5 Testabdeckung 3

5.1 Übersicht Testfälle 3

5.2 Beurteilung Testziele und Testabdeckung 3

6 Testrahmen 3

6.1 Testvoraussetzungen 3

6.2 Mängelklassifizierung 3

6.3 Start- und Abbruchbedingungen 3

7 Testumgebung 3

8 Testinfrastruktur 4

8.1 Testsystem 4

8.2 Testdaten 4

8.3 Testhilfsmittel 4

9 Testorganisation 4

10 Testfallbeschreibungen 4

11 Testplan 4

Abkürzungen und Glossar 5

Inhaltsverzeichnis 6

Tabellenverzeichnis 7

Tabellenverzeichnis

[Tabelle 1: Änderungskontrolle 1](#_Toc129440245)

[Tabelle 2: Übergeordnete Testziele 2](#_Toc129440246)

[Tabelle 3: Testobjekte 2](#_Toc129440247)

[Tabelle 4: Testabdeckung 2](#_Toc129440248)

[Tabelle 5: Mängelklassen 3](#_Toc129440249)

[Tabelle 6: Testfallbeschreibung 4](#_Toc129440250)

[Tabelle 7: Testfallbeschreibung 5](#_Toc129440251)

[Tabelle 8: Testfallbeschreibung 5](#_Toc129440252)

[Tabelle 9: Testfallbeschreibung 5](#_Toc129440253)

[Tabelle 10: Testfallbeschreibung 5](#_Toc129440254)

[Tabelle 11: Testfallbeschreibung 6](#_Toc129440255)

[Tabelle 12: Testplan 6](#_Toc129440256)

[Tabelle 13: Abkürzungen und Glossar 7](#_Toc129440257)