

Allergeck

# Projekt Edible

Testkonzept

23.03.2015

## Testumgebungen

### PHPUnit

PHPUnit hat eine vernünftige Doku auf der eigenen Webseite. Für einfache Einbindung in Eclipse sorgt diese Anleitung: <http://www.thewebhatesme.com/entwicklung/phpunit-mittels-pti-in-eclipse-einbinden/>

### JUnit

Auch JUnit besitzt eine vernünftige Doku auf der eigenen Webseite und eine kleine Einführung sowie Anweisungen zur Einbindungen gibt es auf dieser Seite:

<http://www.vogella.com/tutorials/JUnit/article.html>

## Vorbereitung der Tests

### Testuser

Im Rahmen der Vorbereitung für die Tests müssen verschiedene Testuser angelegt werden. Diese Testuser müssen möglichst unterschiedliche Eigenschaften haben (z.B. leere Blackliste, alle vorhandenen Zutaten in der Blackliste).

### Testdaten

Zur Vorbereitung der Tests zählen ebenfalls das Anlegen von Testdaten / Testprodukten. Die Testdaten müssen unterschiedliche Inhaltsstoffe haben (z.B. keine, alle vorhandenen). Die Testdaten werden in eine, vom Produktivsystem getrennte, Testdatenbank geschrieben und dort verwaltet. Die Testdatenbank unterscheidet sich in der Struktur nicht vom Produktivsystem.

## Verantwortlichkeiten, Aufgaben des Testers

### Verantwortlichkeiten

Der Entwickler muss für alle seine Funktionen ausreichende Unit-Tests schreiben.

### Aufgaben

Der Tester muss das GitHub-Issue nach dem Test in die entsprechende Pipe verschieben, sollte der Test Fehlerfrei durchlaufen sein kann der Tester das Issue aus der Pipe "Test" in die Pipe "Done" verschieben. Sollte es aber noch Fehler geben muss der Tester das Issue in die "Backlog" Pipe verschieben und den entsprechenden Entwickler informieren.

## Zu testende Eigenschaften

Bei den Tests wird die Funktionalität der Funktionen getestet nicht der Inhalt der Daten.

## Definition der Testobjekte

- Vuzix-App
- Web-Oberfläche
- API (Serverseitige API)
- Smartphone-App

## Testfallkatalog

- automatische Anmeldung in der Vuzix-App
- Sprachbefehle, die erkannt werden
- korrekte (Fehler-)Meldungen nach Produktscan
- Profilerstellung funktioniert
- Bearbeitung der Blacklist
- Bearbeitung der persönlichen Daten
- Geräte koppeln und entkoppeln können
- Richtige Zusatzinformationen zu Produkten erhalten
- Funktionierendes Errorhandling

## Risikobetrachtung, Priorisierung

Um eine Risikobetrachtung durchzuführen bzw. eine Priorisierung ansetzen zu können wird der Testfallkatalog priorisiert:

1. Profilerstellung funktioniert
2. Geräte koppeln und entkoppeln können
3. Automatische Anmeldung in der Vuzix-App
4. korrekte (Fehler-)Meldungen nach Produktscan
5. Richtige Zusatzinformationen zu Produkten erhalten
6. Funktionierendes Errorhandling
7. Sprachbefehle, die erkannt werden
8. Bearbeitung der Blacklist
9. Bearbeitung der persönlichen Daten

## Zeit- und Ressourcenmanagement

Alles wird innerhalb desselben Sprints von einer anderen Person getestet. Es entsteht ein Pairing durch Entwickler und Tester.