

Akzeptanztests bei Thalia

Im Team „Kaufen“ und im Team „Kunde im Mittelpunkt“ setzen wir seit über einem Jahr automatisierte Akzeptanztests zur Sicherung unserer Qualität ein. Beide Teams sind begeistert von diesem Vorgehen. Höchste Zeit für einen Beitrag in unserem Techblog ☐

Was ist ein Akzeptanztest?

Ein Akzeptanztest ist ein funktionaler Test, der das Benutzerverhalten beschreibt, um User Story und Akzeptanzkriterien zu überprüfen.

Dabei steht die Sicht des Benutzers im Vordergrund. Es ist wichtig nicht irgendwas zu testen, sondern genau das, was wir mit einer User Story umsetzen wollen. Durch die Akzeptanztests wird sichergestellt, dass die Software aus Sicht des Benutzers wie gewünscht funktioniert.

Ein Akzeptanztest besteht aus drei Teilen: Vorbedingungen (Angenommen), Aktionen die durchgeführt werden (Wenn) und erwartete Ergebnisse (Dann). Diese sogenannten Szenarien werden später zu automatisierten Tests auf verschiedenen Ebenen der Testpyramide. Zur Definition der Szenarien verwenden wir in unseren Teams das Framework [Cucumber](#).

Beispiel eines einfachen Szenarios:

Szenario: Anzeige der Zahlungsarten unter mein Konto mit Rechnung als präferierte Zahlungsart

Angenommen es existiert ein angemeldeter Kunde bei Thalia.de

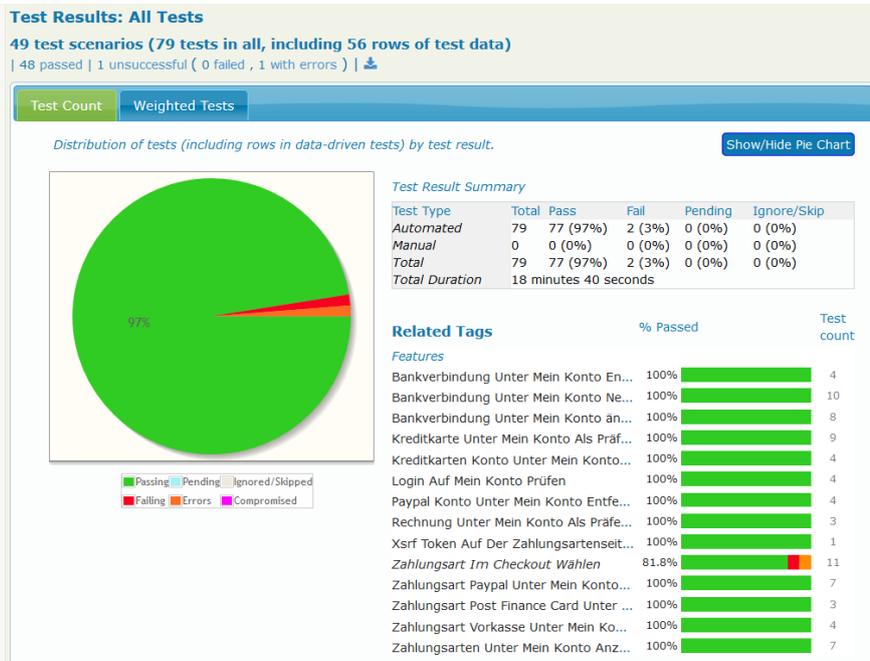
Angenommen der Kunde hat die präferierte Zahlungsart Rechnung

Wenn dieser Kunde die Seite Mein Konto Zahlungsarten aufruft

Dann ist das Akkordeon an der Stelle Rechnung aufgeklappt

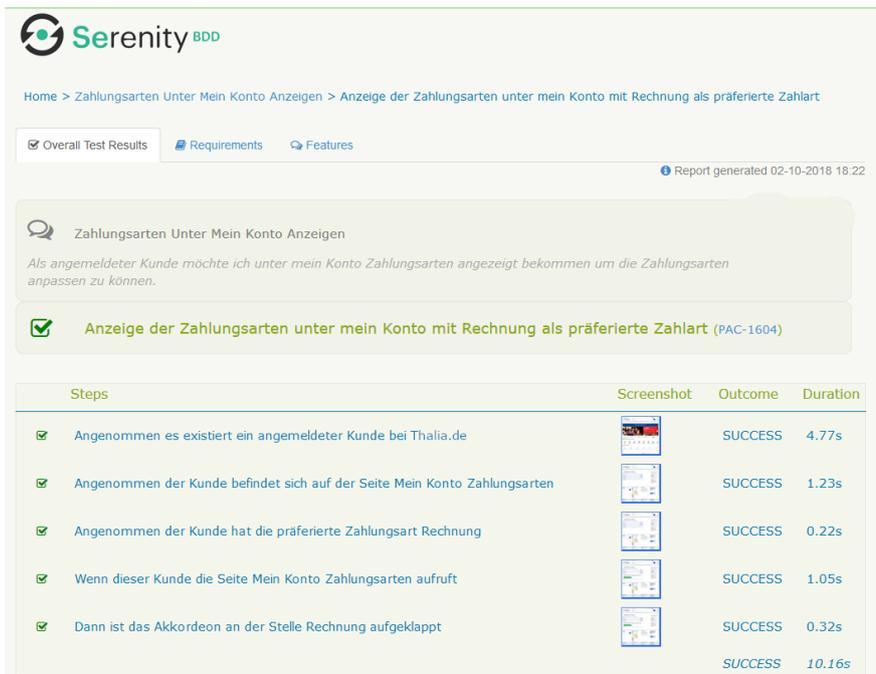
Testreport mittels Serenity

Bei jedem Testdurchlauf wird automatisch mittels [Serenity](#) ein Bericht mit den Testergebnissen erstellt.



Serenity Report - Übersicht

Zu jedem Szenario können die einzelnen Testschritte inklusive Screenshots betrachtet werden. Im Fehlerfall sieht man somit direkt, welcher Testschritt fehlgeschlagen ist.



Serenity Report - Übersicht

Zielbild Testpyramide

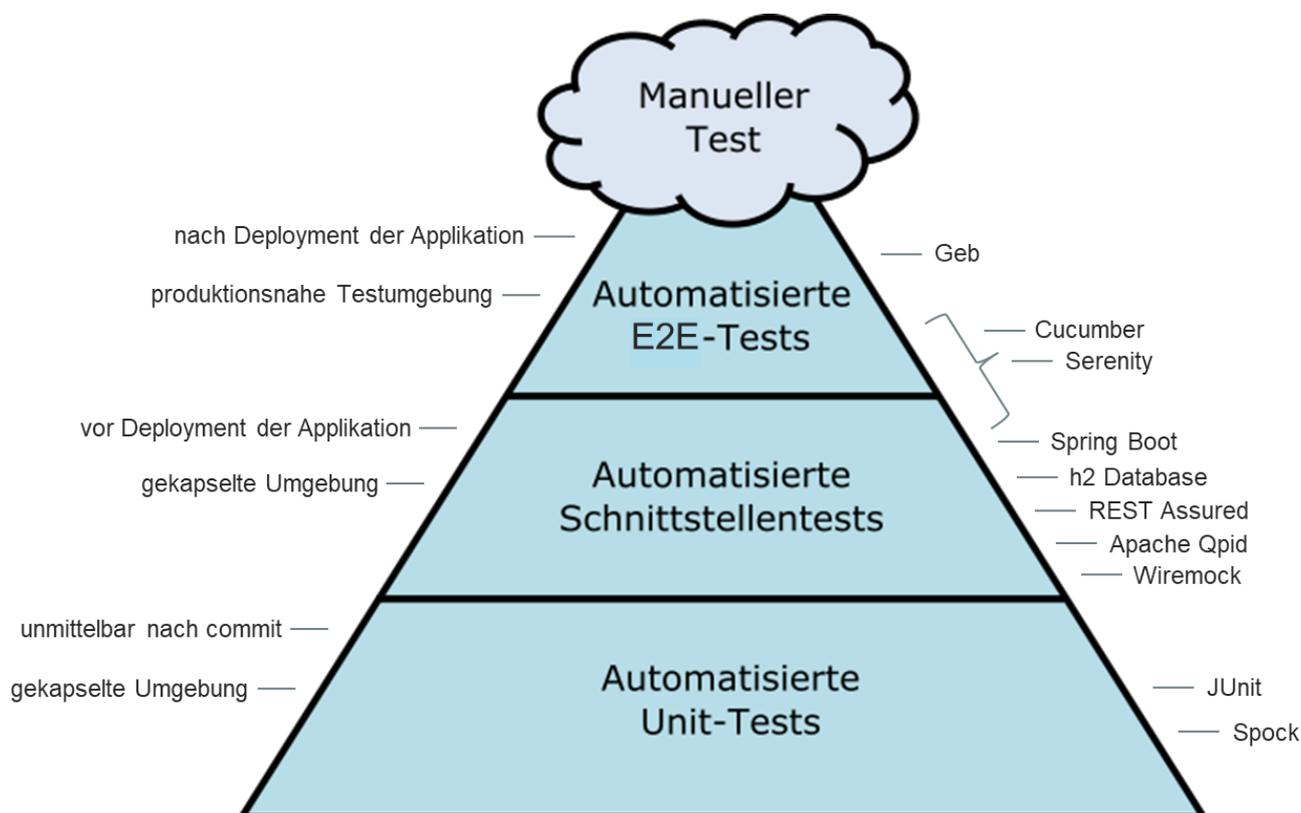
Unser Zielbild bei der Qualitätssicherung im Team „Kaufen“ orientiert sich an der

klassischen Testpyramide. Wir wollen unsere Fehler natürlich möglichst früh und möglichst vor dem End-to-End-Test finden. Da wir allerdings gerade dabei sind unser Frontend abzulösen und unseren Checkout neu zu gestalten, schreiben wir im Moment relativ viele End-to-End-Tests.

Um dem entgegen zu wirken, begeben wir uns als nächstes auf die Suche nach einem geeigneten JavaScript-Test-Framework. Wir erhoffen uns dadurch mögliche Fehler im Frontend bereits auf Unit-Test-Ebene aufzudecken und ggf. sogar einzelne End-to-End-Tests ablösen zu können.

Welche Frameworks verwenden wir und wann werden die Tests ausgeführt?

Die gleichen Szenarien nutzen wir sowohl für die automatisierten End-to-End-Tests, als auch für die Schnittstellentests.



Die automatisierten End-to-End-Tests erfolgen nach dem Deployment in einer produktionsnahen Testumgebung. Dabei verwenden wir das Framework [Geb](#), welches den Selenium WebDriver nutzt und der Browser-Automatisierung dient. Die Tests werden hierbei in der Programmiersprache groovy geschrieben.

Durch die Schnittstellentests soll die jeweilige Applikation inklusive Datenbank- und RabbitMQ-Prüfungen (wo notwendig) zur Laufzeit getestet werden. Um möglichst unabhängig zu sein, erfolgt der Test in einer gekapselten Umgebung. Dazu wird die Applikation mittels h2 Database, Apache Qpid und Spring Boot gestartet, so dass die Schnittstellentests zwar zur Laufzeit, aber noch vor dem Deployment ausgeführt werden können. Fremdsysteme werden dabei mit Wiremock simuliert. REST Assured wird zur Durchführung von HTTP Requests und Validierung der HTTP Response verwendet.

Welche Vorteile bringen uns die Akzeptanztests?

Die Akzeptanztests helfen uns offene Punkte und Ungereimtheiten aufzudecken und mit dem Product Owner zu besprechen.

Aktuell werden im Team „Kaufen“ die Szenarien meistens durch eine Person aus dem Dev-Team kurz vor oder zeitgleich zur Implementierung geschrieben. Dies kann, muss aber nicht, die QAlerin des Teams sein. Wie beim Code-Review erfolgt auch hier ein Review durch eine zweite Person des Dev-Teams.

Noch besser wäre es die Szenarien bereits vor dem Start der Software-Implementierung zu schreiben. Optimalerweise sogar in der „3 Amigos“ Konstellation, welche aus Product Owner, QAlerin/Analyst und Entwickler besteht, um die Anforderung aus unterschiedlichen Blickwinkeln zu betrachten. Dadurch können Ungereimtheiten und Komplexitätstreiber bereits während der Anforderungsanalyse aufgedeckt und besprochen werden. Bisher fehlte uns im Team „Kaufen“ leider die Zeit / der Vorlauf diese Variante auszuprobieren. Wir sind aber zuversichtlich, dass es dazu eine Gelegenheit geben wird ☐

Im Team „Kunde im Mittelpunkt“ wird bei der Erstellung der Szenarien bereits mit der Product Ownerin und der Fachseite zusammen gearbeitet. Dabei wurden bisher sehr gute Erfahrungen gemacht, da anschließend alle relevanten Personen den gleichen Wissenstand auf einer formalen Ebene haben. Siehe auch [hybride Testkonzeption](#).

Es gibt weitere Vorteile, die uns als Dev-Team zugutekommen.

Wir haben es geschafft die Teamsicht auf unsere Tests zu verbessern. Alle im Dev-Team schreiben Szenarien und implementieren die daraus resultierenden automatisierten Tests. Die Akzeptanztests sind Teil des Repositories der jeweiligen Applikation, werden in der Entwicklungsumgebung geschrieben und

bei Bedarf lokal ausgeführt. Durch die Verknüpfung zwischen Fachlichkeit und automatisierten Tests in Form der Szenarien, sind die Auswirkungen bei fehlgeschlagenen Testfällen außerdem direkt sichtbar und helfen uns so bei GoLive-Entscheidungen.