

Frontend-Development mit React.js

Deine 10-Punkte Checkliste zum Start ins automatische Testing

www.qossmic.com



QOSSMIC
PART OF OPEN

Deine 10-Punkte Checkliste zum Start ins Frontend-Testing mit React

Welche Gründe führen Teams hauptsächlich an, wenn sie sich für automatische Frontend-Tests entscheiden?

Automatisches Testing in der Frontend-Entwicklung mit React.js hat viele Vorteile. Ganz vorne stehen sicherlich die erhöhte Entwicklungsgeschwindigkeit, verbesserte Test-Qualität, aber auch ökonomisch ganz einfach ein enormes Kosten-Einsparpotential, das es zu heben gilt.

Kosten einsparen

- stabiler Code führt zu weniger Ausfällen
- weniger Bugs = weniger Bugfixing = Entlastung des Entwicklungsteams führt zu mehr Innovationskraft

Mehr Geschwindigkeit in der Entwicklung

- unerwünschte Seiteneffekte bei Code-Änderungen werden durch Tests erkannt und können sofort behoben werden, anstatt später zu Bugs zu führen

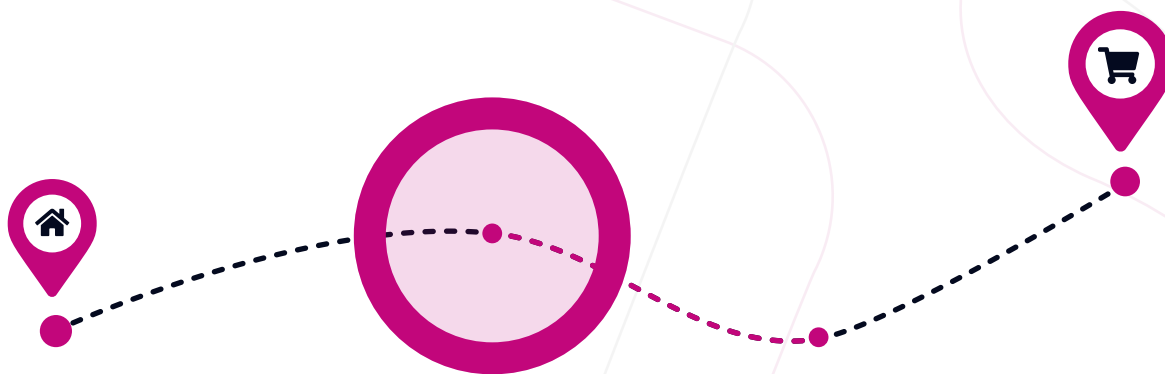
Test-Qualität erhöhen

- menschliche Tester:innen sind deutlich langsamer
 - Bei manuellen Testings werden Funktionen vor dem Deployment von Tester:innen besonders eingehend getestet, später nur noch oberflächlich. Dagegen laufen automatisierte Tests immer genau so detailliert ab, wie sie geschrieben wurden. Dadurch kann eine gleichbleibende Test-Qualität sichergestellt werden.
 - Durch Simulationen (Mocking) können bestimmte Funktionen automatisch getestet werden, die Tester:innen gar nicht testen können. Zum Beispiel können Fehlerfälle bei Drittsystemen (CMS / Paypal) von Testern nicht immer ausgelöst werden, so dass die passende Fehlermeldung nicht geprüft werden kann. Automatische Tests können diverse Antworten eines Drittsystems nach Belieben simulieren und so die Reaktion des Systems auf Korrektheit prüfen.
- dadurch, dass Tests während oder am Ende der Implementierung einer Funktion geschrieben werden, haben die Entwickler:innen noch alle Eigenheiten / Edge Cases dieser Funktion im Kopf und können sie im Test festschreiben.
- insgesamt die Code-Qualität steigern
 - der Code wird granular überwacht
 - bei größeren Eingriffen in den Code (Austausch von Libraries) werden je nach Testabdeckung alle Seiteneffekte erkannt.

Deine 10-Punkte Checkliste zum Start ins Frontend-Testing mit React

Welchen Herausforderungen sehen sich Teams vor der Umsetzung gegenüber?

Die Einführung automatischer Frontend-Tests bedeutet zunächst einen gewissen Arbeitsaufwand, der sich allerdings durch bessere Code-Qualität und erhöhte Geschwindigkeit in der Entwicklung schnell amortisiert. Fehlgeschlagene Tests und deren Korrektur können z.B. ein Deployment verzögern. Die verbesserte User-Experience und Qualitätssteigerung rechtfertigt dies jedoch. Wenn der Code sich ändert, müssen die entsprechenden Tests eventuell auch angepasst werden. Das steigert langfristig die Code-Qualität und führt u.a. zu geringeren Ausfallzeiten.



Hilfreich vor der Teststrategie-Entwicklung:
Den kritischen Pfad der Anwendung identifizieren

Mit unserer **10-Punkte Checkliste** kannst Du prüfen, ob und wie ihr in eurem Unternehmen aufgestellt seid, den Aufwand konkreter kalkulieren und feststellen, wo es ggfs. noch Bedarf zur Unterstützung gibt. Dabei hilft zur Bewertung eures aktuellen Status Quo auch deine persönliche Einschätzung hinsichtlich des Reifegrades eurer Entwicklung:

Haben / machen wir schon

Können wir lernen

Dabei brauchen wir Hilfe

Können wir

Deine 10-Punkte Checkliste zum Start ins Frontend-Testing mit React

1

- a) Hat das Management Verständnis für die Notwendigkeit von Tests?
- b) Hat das Dev-Team Verständnis für die Notwendigkeit von Tests?

Warum ist das wichtig?

- a) Das Management muss die zusätzliche Arbeit / Verzögerungen absegnen. Rückhalt im Management macht vieles einfacher.
- b) Ein Team, das Tests eher halbherzig macht, weil es die Wichtigkeit von Tests nicht erkennt oder unterschätzt, wird keine guten Tests / keine gute Testabdeckung herstellen.

2

- Ist geklärt, ob Tests für Legacy-Code erstellt werden sollen, oder für ein ganz neues Projekt?

Warum ist das wichtig?

Bei Legacy-Code muss geklärt werden, in welcher Reihenfolge Tests für die bestehenden Komponenten geschrieben werden sollen. In einem neuen Projekt werden Tests für jede neue Komponente direkt mitgeschrieben. Das beeinflusst die Arbeitsweise.

3

- Sind die Funktionen identifiziert, die für die erfolgreiche Durchführung von Prozessen einer Anwendung unerlässlich sind, ggfs. anhand des "Kritischen Pfades"?

Warum ist das wichtig?

Insbesondere bei Legacy-Code sollte priorisiert werden, wo angefangen wird, damit zunächst alle Prozesse abgesichert werden, die zum Betrieb zwingend erforderlich sind.

4

- Ist schon identifiziert, ob / wo e2e-Tests nötig sind, oder ob Tests mit Mocks möglich sind?

Warum ist das wichtig?

End-to-End-Tests nutzen "echte" Daten z.B. aus dem Backend. Simulierte Tests benutzen Testdaten. Es ist wichtig, die Vor- und Nachteile von Mock-Tests und E2E-Tests abzuwägen und eine Teststrategie zu entwickeln, die auf den spezifischen Anforderungen und Zielen des Projekts basiert. Ein Ansatz könnte sein, Mock-Tests als Ergänzung zu E2E-Tests zu verwenden, um die Testabdeckung zu erhöhen und gleichzeitig die Testeffizienz und -geschwindigkeit zu verbessern.

Deine 10-Punkte Checkliste zum Start ins Frontend-Testing mit React

5

Sind Tests auf verschiedenen Bildschirmgrößen / Geräten / Browsern erforderlich?

Warum ist das wichtig?

Die Anforderungen beeinflussen die Auswahl des passenden Frameworks.

6

Ist geklärt, wann die Testsuite automatisch gestartet werden soll?

Warum ist das wichtig?

Die Testsuite ist die Gesamtheit aller Tests eines Projekts. Der beste Zeitpunkt zum Starten einer automatischen Testsuite hängt von verschiedenen Faktoren ab, die es zu berücksichtigen gilt. Ein guter Moment, um alle Tests durchlaufen zu lassen, kann zum Beispiel das Erstellen eines Pull-Requests sein. Fehlerfreie Tests sollten eine Voraussetzung für einen Merge sein.

7

Ist bereits ein Test-Framework ausgewählt und in das Projekt eingebunden?

Warum ist das wichtig?

Test-Frameworks sorgen dafür, dass komplizierte Vorgänge einheitlich und automatisiert ausgeführt werden. Sie müssen möglichst alle Funktionen enthalten, die eine optimale Testabdeckung im Rahmen deines Projektes gewährleisten.

8

Gibt es ein Test-Know How im Team?

Warum ist das wichtig?

Test-Knowhow im Team ist wichtig, insbesondere auf das genutzte Framework bezogen. Wenn diese Kompetenz (noch) nicht vorhanden ist, sollte sie aufgebaut bzw. vertieft werden.

9

Ist das Schreiben von Tests in interne Prozesse aufgenommen worden, z.B. in die "Definition of Done"?

Warum ist das wichtig?

Tests zu schreiben sollte verpflichtend sein. Am einfachsten ist es daher, dies verbindlich in die internen Prozesse aufzunehmen und mit einzuplanen.

Deine 10-Punkte Checkliste zum Start ins Frontend-Testing mit React

10

Sind eventuelle Hindernisse beim Schreiben oder Ausführen von Tests vorhanden?

Warum ist das wichtig?

Tests zu schreiben muss Teil des normalen Implementierungsprozesses sein. Die häufigsten Probleme wie Zeitmangel oder fehlerhafte Testautomatisierung müssen identifiziert und gelöst werden.

Du und dein Team benötigen Hilfe? Wir können dich unterstützen!

Unsere QOSSMIC Frontend Entwickler:innen sind Expert:innen darin, dich und dein Team ganzheitlich zu unterstützen:

Mit Workshops - Coachings - Trainings - Pair-Programming - Architektur-Beratung

Von der Planung deiner Teststrategie, über die Identifikation des Kritischen Pfades bis hin zur Dokumentation und Prozessmanagement: Mit QOSSMIC entwickelst du zukunftssichere Lösungen mit maximalem Nutzen für deinen Unternehmenserfolg.

Reserviere jetzt deinen kostenlosen Beratungstermin.



Gero Duppel



Stephan Schmidtke

So erreichst du uns:

+49 221 165 35 40

contact@qossmic.com



QOSSMIC
PART OF OPEN