

entwickler.press  
shortcuts

# Qualitätssicherung mit JavaScript und PHP

Sebastian Springer

*Sebastian Springer*

# **Qualitätssicherung mit JavaScript und PHP**

ISBN: 978-3-86802-437-1

© 2012 entwickler.press

Ein Imprint der Software & Support Media GmbH

# 1 Einleitung

Aus der Entwicklung von Software ist die Qualitätssicherung nicht mehr wegzudenken. Diese Aussage trifft allerdings nicht nur für Enterprise Serverapplikationen und klassische Desktop-Anwendungen zu, sondern gilt in gleichem Maße auch für die Entwicklung von Webapplikationen.

Immer mehr geschäftskritische Anwendungen werden komplett oder zumindest in Teilbereichen ins Web verlagert und auch innerhalb von Firmen wächst die Bedeutung von Intranets und Webapplikationen. Der große Vorteil von Webapplikationen ist, dass sie nur einmal auf dem Server installiert werden müssen und dann von nahezu überall auf sie zugegriffen werden kann. Einzige Voraussetzung ist das Vorhandensein eines Webbrowsers. Der Benutzer muss im Normalfall keine weitere Software installieren. Die zentrale Instanz einer Webapplikation ermöglicht es, Aktualisierungen ohne großen Aufwand an alle Nutzer zu verteilen. Dadurch kann sehr schnell auf geänderte Anforderungen der Benutzer reagiert werden. Durch die Tatsache, dass durch mobile Endgeräte wie Mobiltelefone oder Tablets eigentlich jeder permanenten Zugriff auf das Web hat, steigt die Attraktivität von Webapplikationen. Im Einsatz in einer sehr heterogenen Umgebung mit einer Vielzahl verschiedener Clientsysteme spielt diese Form der Anwendungsarchitektur seine wahre Stärke aus. Die Serverseite ist für sämtliche Systeme die gleiche, was wiederum zur Konsequenz hat, dass sämtliche angebundene Systeme auf der gleichen Datenbasis arbeiten. Auf der Clientseite unterscheiden sich die Anwendungen von Gerät zu Gerät. So steht beispielsweise im normalen Webbrowser der vollständige Funktionsumfang der Anwendung zur Verfügung, wobei die Bedienung auf die Steuerung mit der Maus optimiert ist und die Eingaben meist über die Tastatur getätigt werden. Auf einem mobilen Gerät

wiederum wird mittels Gesten gesteuert, der Funktionsumfang ist meist reduziert und auf die Bedürfnisse im mobilen Umfeld mit seinen beschränkten Ressourcen angepasst.

## **2 Grundlagen**

Der Qualitätssicherung in Webprojekten liegen einige allgemeine Konzepte und Vorgehensweisen zugrunde. Im Gegensatz zu klassischen Softwareentwicklungsprojekten kommt bei der Webentwicklung eine hoch dynamische Komponente hinzu. Durch die kurzen Entwicklungszyklen und die Tatsache, dass eine Webanwendung nur einmal installiert werden muss und dann sämtlichen Nutzern zur Verfügung steht, muss hier nach angepassten und hoch spezialisierten Methoden vorgegangen werden.



## 2.1 Agile Softwareentwicklung

Ein zentraler Aspekt bei der Softwareentwicklung ist die Vorgehensweise. Hier haben sich über die Jahre hinweg viele verschiedene Varianten herausgebildet. Die Entwicklung von Webapplikationen ist vor allem durch sich ständig ändernde Anforderungen gekennzeichnet. Dies liegt zum einen im Erkenntniszugewinn während der Entwicklung begründet und zum anderen im Feedback der Nutzer. Dieses frühe Feedback ist wichtig, um die Anforderungen und Bedürfnisse der Anwender zu erfüllen.

Diese Punkte wurden von einer Gruppe von Softwareentwicklern aufgenommen und im „Agilen Manifest“ festgehalten. Eines der wichtigsten Prinzipien lautet, dass die Kundenzufriedenheit an oberster Stelle steht. Weitere Merkmale sind die häufigen Auslieferungen lauffähiger Produktinkremente und die Förderung offener Kommunikation. Ein weiteres Prinzip der agilen Softwareentwicklung ist, dass Änderung, selbst wenn sie erst spät im Projekt aufkommen, stets willkommen sind und umgesetzt werden.

In der Praxis werden häufig verschiedene agile Methoden miteinander kombiniert, um ein optimales Ergebnis zu erreichen. Ein konkretes Beispiel hierfür ist Scrum, das für das Projektmanagement und die Organisation verwendet wird und auf der technischen Seite durch Extreme Programming unterstützt wird.

Durch diese Kombination erreicht man eine iterative Vorgehensweise mit Feedbackschleifen und somit eine sehr enge Kopplung an den Kunden. Der Vorteil besteht darin, dass die Anforderungen zu Beginn des Projekts noch nicht vollständig und detailliert vorliegen müssen. In Scrum werden die Anforderungen in einer vom Kunden priorisierten Liste, das sogenannte Product Backlog, festgehalten. Die Anforderungen

mit der höchsten Priorität werden als Erstes umgesetzt. Diese Tatsache macht es erforderlich, dass diese Anforderungen einen so hohen Detailgrad aufweisen, dass sie vom Entwicklerteam auch umgesetzt werden können. Zu diesem Zweck werden in einem Meeting, an dem sowohl der Kunde als auch das Projektmanagement und das Entwicklerteam teilnehmen, die Anforderung mit der höchsten Priorität in eine separate Liste, das sogenannte Sprint Backlog, überführt. Diese Liste enthält die Aufgaben, die während der nächsten Iteration erledigt werden müssen. Hier kommt es bereits zu ersten Berührungspunkten mit der Qualitätssicherung, denn die formulierten Anforderungen und Aufgaben dienen als Grundlage für Softwaretests, die automatisiert sicherstellen, dass die Anforderungen durch die Software erfüllt werden. Nach Abschluss dieses Planungsmeetings hat der Kunde keinen Einfluss mehr auf die Iteration und die Entwickler können sich für die Dauer der Iteration, die normalerweise von einer bis vier Wochen reicht, ungestört der Umsetzung der Aufgaben widmen. Zu beachten ist allerdings, dass die Iteration in kleinere Abschnitte von je 24 Stunden aufgebrochen wird, innerhalb derer ein kurzes, teaminternes Standup-Meeting stattfindet, in dem die Kommunikation gefördert und konkrete Probleme adressiert werden sollen. Im nächsten Schritt, der konkreten Implementierung der Anforderungen durch die Entwickler in einer geschlossenen Iteration, kommt erneut die Qualitätssicherung ins Spiel, denn hier wird der eigentliche Quellcode erstellt, an den bestimmte Qualitätsanforderungen gestellt werden. Während der Umsetzung findet ein regelmäßiger, im Idealfall automatisierter, Soll-Ist-Vergleich zwischen Qualitätsanforderungen und Quellcode statt. Am Ende einer jeden Iteration stehen zwei weitere Meetings, die ebenfalls wieder den Kunden direkt in die Entwicklung einbinden. Zum einen findet hier das Review-Meeting statt, in dem die umgesetzten Features von den Entwicklern vorgestellt und demonstriert werden. Zum anderen wird eine Retrospektive abgehalten, bei der nach der Methode „Inspect

and Adapt“ vorgegangen wird. Die vorangegangene Iteration wird besprochen und die verschiedenen Ereignisse und Vorgehensweisen werden hinterfragt, um herauszufinden, welche Methode erfolgreich war und weiter ausgebaut und gefördert werden soll. Außerdem werden Strategien geplant, um die vorhanden Probleme zu lösen und ihr erneutes Auftreten zu verhindern. So lässt sich sowohl die Software als auch das Entwicklerteam in kleinen Iterationen immer weiter optimieren.

Was Scrum auf der Ebene des Projektmanagements bedeutet, sind Methoden wie beispielsweise Extreme Programming für die konkrete Softwareentwicklung. Wie auch die übrigen agilen Methoden geht Extreme Programming in Iterationen vor. Den Kern von Extreme Programming bilden verschiedene Praktiken, die das Entwicklerteam in ihrer Arbeit, qualitativ hochwertige Software zu erstellen, unterstützen. Hierbei reicht die Bandbreite von offensichtlich qualitätsrelevanten Themen wie Testdriven Development bis hin zu nicht ganz so offensichtlichen Techniken wie Pair-Programming. Beim Testdriven Development entsteht vor der eigentlichen Umsetzung einer Programmeinheit ein Modultest. Anhand dieses Tests wird dann der eigentliche Quellcode implementiert. Auf diese Weise kann eine sehr hohe Abdeckung durch automatisierte Tests erreicht werden. Weitere Elemente von Extreme Programming sind die Forderung nach einheitlichen Coding-Standards, regelmäßigen Refactorings und permanenter Integration. Sämtliche dieser Elemente finden sich auch direkt oder indirekt im Bereich der Qualitätssicherung wieder.



## 2.2 Das Team

Die agilen Methoden stellen die Menschen in den Mittelpunkt der Softwareentwicklung. Das Team steht also über Prozessen und standardisierten Vorgehensweisen. Der Begriff Team schließt hierbei nicht nur die Entwickler ein, sondern alle direkt am Projekt Beteiligten. Im Team sind also neben den Entwicklern auch Architekten, Tester, Designer und vor allem auch das Projektmanagement enthalten. Hintergrund für diese Definition ist, dass dieser Personenkreis direkt zusammenarbeiten muss, um das Projekt zum Erfolg zu führen. Jeder Stakeholder hat so die Möglichkeit seine eigenen Interessen zu vertreten und durch einen gemeinsamen Aushandlungsprozess eine optimale Lösung zu erarbeiten. Der größtmögliche Nutzen für den Kunden kann nur erreicht werden, wenn sämtliche Interessen und Anforderungen berücksichtigt werden und die Lösung aus einem Ausgleich besteht, wobei nicht individuelle Interessen im Vordergrund stehen, sondern der Erfolg des Projekts und die Zufriedenheit des Kunden. Dies kann nur durch eine sehr enge Zusammenarbeit und ein gemeinsames Verständnis erzielt werden.

Die Organisation in Teams hat weitere Vorteile, die vor allem in der Verteilung von Wissen liegt. Obwohl im Normalfall die wichtigsten Eckdaten Einzug in die Dokumentation der Software halten, ist es nie möglich, sämtliche Zusammenhänge vollständig zu dokumentieren. Beispiele hierfür sind die Beweggründe, die zur Entstehung einer Software geführt haben und die Grundlage für verschiedene Architektur- und Design-Entscheidungen innerhalb einer Software sind.

Ein weiterer Aspekt ist die Ausfallsicherheit. Fällt ein Teammitglied krankheits- oder urlaubsbedingt aus, besteht die Möglichkeit, diesen Ausfall durch das gemeinsame Knowhow zu kompensieren.