

# CASE-STUDY

## Standardisierung und Automatisierung bei der E-Post Development

Veröffentlichung: 2016



## Unternehmen

# E-POST

Deutsche Post E-POST Development GmbH  
Berlin, Deutschland

Die E-Post Development GmbH mit Sitz in Berlin entwickelt und betreibt die auf dem Microservice-Ansatz beruhende E-Post-Plattform zum digitalen Versand von Briefen für Groß- und Privatkunden. Dabei steht sie und jedes einzelne Entwicklungsteam stetig vor der Herausforderung, die Umsetzung der gemeinsamen Standards kontinuierlich sicherzustellen. Das Open-Source-Framework jQAssistant hilft dabei, dieses Ziel zu erreichen.

Die E-Post Development GmbH ist eine hundertprozentige Tochter der Deutsche Post AG und im Rahmen des Konzerns für die Digitalisierung des Briefgeschäfts verantwortlich. Dementsprechend übernimmt die E-Post Development eine Schlüsselposition für den Aufbau von digitalen Dienstleistungen rund um den Markt für Briefdienstleistungen für Geschäfts- und Privatkunden der Deutsche Post AG.

Im Rahmen der Digitalisierungsstrategie der Deutsche Post AG entwickelt die E-Post Development Softwaresysteme zur Optimierung bestehender Prozesse rund um die postalische Kommunikation. Um diese Strategie technologisch zu unterstützen, verfolgt die E-Post Development stetig das Ziel, eine flexible und nachhaltige Plattform bereitzustellen. So wird sichergestellt, dass Geschäftsprozesse schnell angepasst und erweitert werden können. Daraus ergeben sich für die mit der Umsetzung der Geschäftsprozesse betrauten Entwicklungsteams hohe Herausforderungen an die von ihnen betreuten Softwaresysteme, insbesondere, da jedes Team parallel eine Vielzahl von Software-Systemen zu verantworten hat.

## Eingesetztes Open-Source-Framework

# jQAssistant

jQAssistant powered by buschmais GbR

[www.buschmais.de](http://www.buschmais.de)  
[www.github.com/jqassistant](https://www.github.com/jqassistant)

Die E-Post Development setzt dabei durchgehend auf Java-basierende Technologien. Architektonisch wird der Ansatz verfolgt, die für den Betrieb der Gesamtplattform notwendige Fachlichkeit auf einzelne, fachlich geschnittene Services zu verteilen. Dieser Ansatz wird auch als Microservice-Architektur bezeichnet und erfährt eine immer höhere Aufmerksamkeit bei der Entwicklung und dem Betrieb von größeren Plattformen.

Den daraus erwachsenden Vorteilen, wie unter anderem eine verbesserte Wartbarkeit und eine bessere Skalierbarkeit, stehen aber auch neue Herausforderungen gegenüber. Die erhöhte Anzahl von Softwaresystemen rückt die Bedeutung der Einhaltung von Standards für eine schnelle Entwicklung und eine nachhaltige Weiterentwicklung wieder verstärkt in den Vordergrund.

Das für die Stammdatenhaltung zuständige Team suchte daher für seine Services nach einem Werkzeug, mit dem automatisch verschiedenste individuelle Aspekte eines Softwareprojektes - von der Konfiguration über die Projektstruktur bis hin zum Dependency-Management - analysierbar und prüfbar sind. Dabei fiel die Wahl auf das Werkzeug jQAssistant, da es nicht nur auf die Analyse von Source-Code beschränkt ist, sondern es ermöglicht, Regeln übergreifend über verschiedenste Artefakte hinweg flexibel zu formulieren.



Abb. 1: Mit jQAssistant gescannte Softwareprojekte können sowohl manuell als auch automatisiert ausgewertet werden.

## Einsatz von jQAssistant im Rahmen der E-Post Development

Das Team wollte den erwähnten Herausforderungen durch mehrere Maßnahmen begegnen: Zum einen durch einen vereinheitlichten Build-Prozess und zum anderen durch die für alle selbstverantworteten Softwaresysteme geltenden Architekturrichtlinien. Der von jQAssistant verfolgte Ansatz, beliebige Artefakte - von Java-Klassen, Bibliotheken, Konfigurationsdateien bis hin zu Verzeichnisstrukturen - einzulesen, in eine Graph-Datenbank abzulegen und über die so gesammelten Informationen anschließend Abfragen mit einer für alle erfassten Artefakte einheitlichen Abfragesprache stellen zu können, erwies sich als besonders flexibel und leicht erweiterbar (Abbildung 1).

Wesentlich für die Entscheidung für jQAssistant war jedoch, die sich daraus ergebende Möglichkeit, unterschiedliche Aspekte mittels eines einzigen Tools überprüfen zu können, anstatt für die gleiche Aufgabe mehrere unterschiedliche Tools oder selbstgeschriebene Skripte zu verwenden.

## Vereinheitlichung des Build-Prozesses durch jQAssistant

Als Erstes ging das Team die Aufgabe der Vereinheitlichung des Build-Prozesses an. Bis dahin unterschieden sich die Build-Prozesse der einzelnen Softwaresysteme hinsichtlich ihres Ablaufs und ihrer Konfigurationsmöglichkeiten voneinander. Bei der Anzahl der von dem Entwicklungsteam zu verantwortenden Softwaresysteme bedeutete dies einen hohen Aufwand für Erweiterung und Pflege der genutzten Konfigurationen im Conti-

nuous-Integration-System (CI-System). Mit der Zeit verschob sich ein Großteil des Wissens über die Build-Prozesse auch von dem genutzten Build-Tool Apache Maven in das verwendete CI-System. Versehentlich gelöschte oder fehlerhaft geänderte Build-Prozesse waren so ohne erheblichen Mehraufwand nicht mehr wiederherstellbar.

### i jQAssistant im Web

#### Beratung

[www.buschmais.de](http://www.buschmais.de)

#### Blog

[www.jqassistant.org](http://www.jqassistant.org)

#### GitHub

[github.com/jqassistant](https://github.com/jqassistant)

#### Twitter

[www.twitter.com/jqassistant](https://www.twitter.com/jqassistant)

Als Verbesserungsmaßnahme wurde von dem Entwicklungsteam daher eine Vereinheitlichung der Build-Prozesse in Hinsicht auf Ablauf und Möglichkeiten zur Parametrisierung und des Build-Ablaufs aller Softwaresysteme beschlossen und eine für alle Softwaresysteme anwendbare, auf Apache Maven beruhende Build-Struktur entworfen. Anschließend wurden mit jQAssistant prüfbare Regeln aufgestellt, die die Einhaltung der Maven-Build-Struktur absichern. Nach jedem Check-In in das Source-Control-Management-System (SCM-System) wird die Einhaltung dieser Regeln sofort überprüft. Verletzungen der Regeln, die die Build-Struktur absichern, führen damit sofort zu einem Fehlschlag des Builds und geben dem Entwicklungsteam die Möglichkeit, aufgedeckte Verletzungen sofort zu beheben.

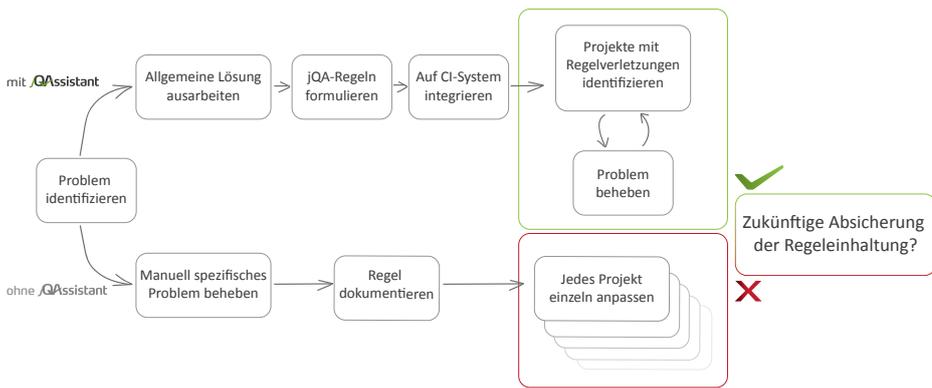


Abb. 2: Durch jQAssistant kann jetzt das erneute Auftreten von bekannten Problemen nachhaltig ausgeschlossen werden.



Abb. 3: Neue Services können jetzt durch die getroffenen Maßnahmen wesentlich schneller bereitgestellt werden.

## Standardisierung der Softwaresysteme durch jQAssistant

Um für die Weiterentwicklung bestehender und die Entwicklung neuer Softwaresysteme sicherzustellen, dass die geltenden und vom Team selbst aufgestellten Architekturrichtlinien eingehalten werden, wurden auch für diese Vorgaben entsprechende, prüfbare Regeln mit jQAssistant ausgearbeitet. Die Einhaltung dieser wird ebenfalls bei jedem Check-In in das SCM-System durch den Anstoß eines Build-Prozesses in der CI-Umgebung sichergestellt. Das Spektrum der nach der Einführung von jQAssistant automatisch überprüften Architekturregeln erstreckt sich über unterschiedliche Bereiche, beginnend bei Logging, Fehlerbehandlung, Bereitstellung von Metriken und Verwendung von Bibliotheken. Neue Regeln werden bei Bedarf oder nach Identifizierung von Fehlern dem bestehenden Regelsatz hinzugefügt. So wird sichergestellt, dass diese Fehler nicht erneut in die Softwaresysteme des Teams eingebracht werden (Abbildung 2).

## Erhöhte Softwarequalität und Regeleinhaltung durch jQAssistant

Durch den Einsatz von jQAssistant konnte nach einem halben Jahr im Bereich der Stammdatenverwaltung eine Erhöhung der Softwarequalität und eine Verbesserung der Einhaltung geltender Architekturregeln erreicht werden. Die darüber hinaus

gewonnene Harmonisierung des Aufbaus der im Team verantworteten Services resultierte in einer schnelleren Bereitstellung neuer Services. Wurden ursprünglich sieben Arbeitstage für das Aufsetzen eines neuen Service benötigt, kann die selbe Aufgabe jetzt innerhalb eines Tages umgesetzt werden (Abbildung 3). Die neue Einheitlichkeit der Services nutzte das Team zudem, um Tools für eine höhere Automatisierung seiner Arbeit zu entwickeln.



### Ihr Ansprechpartner

Für weitere Informationen und Consulting-Angebote wenden Sie sich bitte an:  
 Herrn Dirk Mahler (buschmais GbR)  
 Tel. +49 351 3209230  
 dirk.mahler@buschmais.com

Ein weiterer Vorteil besteht darin, dass mehrfach auftretende Fehlermuster leichter über unterschiedliche Services hinweg mit jQAssistant identifiziert, behoben und dauerhaft ausgeschlossen werden können.

Aufbauend auf den Möglichkeiten, die sich durch den Einsatz von jQAssistant ergeben, ist das Team jetzt in der Lage, neue Geschäftsprozesse schneller und nachhaltiger umzusetzen sowie neue Dienste zu entwickeln.