

RESTful Web-Services mit Qualität:

Teil 2: Priorisierung von Best Practices mittels Qualitätsmerkmalen

Aufgrund der zunehmenden Verbreitung von REST zur Gestaltung von Web-Services gilt es heute stärker denn je, RESTful Web-Services mit besonderer Sorgfalt zu entwerfen. Im ersten Teil dieses Artikels haben wir gezeigt, welche Best Practices für RESTful Web-Services eingehalten werden sollten, um eine hohe Qualität des Entwurfs sicherzustellen. Da sich diese gegenseitig ausschließen können und häufig nicht genügend Ressourcen zur Verfügung stehen, um die Einhaltung aller Best Practices zu gewährleisten, ist eine Priorisierung erforderlich. In dem zweiten Teil des Artikels zeigen wir daher, wie durch eine Verknüpfung der Best Practices aus dem ersten Teil mit Qualitätsmerkmalen aus der ISO/IEC 25010:2011 eine Priorisierung erfolgen kann. Als Ergebnis wird ein Bewertungsbogen zum Download bereitgestellt, mit dem RESTful Web-Services in Projekten systematisch geprüft werden können.

In den letzten Jahren hat die Nutzung von REpresentational State Transfer (REST) als Paradigma zur Gestaltung und Umsetzung von Web-Services stark zugenommen (vgl. [Inf11]). Die in diesem Zuge häufig als „RESTful“ bezeichneten Web-Services haben somit eine immer größere Relevanz für die Gestaltung eines gesamten Systems. Architekten und Entwickler sind daher heute stärker denn je gefordert, die Web-Services mit besonderer Sorgfalt zu gestalten, um eine hohe Qualität der Web-Services und somit auch des gesamten Systems sicherzustellen.

In der letzten Ausgabe von OBJEKTspektrum (vgl. [Geb15]) haben wir gezeigt, welche Best Practices für die Gestaltung von RESTful Web-Services derzeit in der Literatur verbreitet sind. Diese Best Practices wurden im ersten Teil des Artikels zusammengefasst und kategorisiert. In Projekten herrscht jedoch häufig hoher Zeit- und Kostendruck, weshalb nicht alle Best Practices berücksichtigt werden können. Zusätzlich können sich Best Practices gegenseitig ausschließen. Aus diesen Gründen ist eine Priorisierung der Best Practices erforderlich, sodass eine gezielte Auswahl an zu berücksichtigenden Best Practices getroffen werden kann.

Hierfür werden im nun im zweiten Teil die bereits zuvor identifizierten Best Practices mit Qualitätsmerkmalen aus der ISO/IEC 25010:2011 (vgl. [ISO25010]) verknüpft. Hierdurch ist es möglich, durch eine Priorisierung der Qualitätsmerkmale eine Priorisierung der Best Practices zu erzielen. Ergebnis dieses zweiten Teils ist eine tabellarische Übersicht der Best Practices mit Bezug zu Qualitätsmerkmalen aus der ISO/IEC 25010:2011 (vgl. [ISO25010]). Mit dieser Tabelle erhalten Architekten und Entwickler die Möglichkeit, Web-Services auf der Basis von REST systematisch

zu analysieren. Die Tabelle steht zusätzlich unter

http://www.sigs.de/publications/download/Analyse_REST_OS_02_2015.pdf in Form eines Bewertungsbogens zum Download bereit, sodass die Ergebnisse direkt in Projekten eingesetzt werden können. Um die Effizienz der Entwicklung sowie die Wiederholbarkeit der Analysen von RESTful Web-Services zu gewährleisten, ist eine Automatisierung der Überprüfungen wünschenswert. Aus diesem Grund zeigen wir neben der Priorisierung von Best Practices im Folgenden einen Ansatz zur teilautomatisierten Überprüfung von Web-Services hinsichtlich der beschriebenen Best Practices.

Verknüpfung mit ISO/IEC 25010:2011

Mit der Verknüpfung der Best Practices mit Qualitätsmerkmalen der ISO/IEC 25010:2011 (vgl. [ISO25010]) erhalten Ar-

chitekten und Entwickler eine Information, auf welche Qualitätsmerkmale sich die einzelnen Best Practices auswirken. Dadurch sind sie in der Lage, eine Priorisierung einzelner Praktiken vorzunehmen, falls ein Projekt beschränkten Ressourcen unterliegt oder aufgrund eines Konflikts nicht alle Best Practices gleichzeitig eingehalten werden können. **Tabelle 1** zeigt die Definition der einzelnen Qualitätsmerkmale gemäß [ISO25010].

Bei der nachfolgenden qualitativen Betrachtung (siehe **Tabelle 2**) wird lediglich ausgesagt, ob die Best Practices eine positive, negative oder keine Auswirkung auf ein Qualitätsmerkmal haben. Der Grad der Beeinflussung wird daher nicht näher beleuchtet. Die Interpretation der qualitativen Betrachtung wird in **Tabelle 3** beschrieben. Die Tabelle zeigt beispielsweise, dass sich die konkrete und domänenspezifische Benennung positiv auf die Benutzbarkeit ei-

Qualitätsmerkmal	Definition
Funktionalität	„Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions“ ([ISO25010], S. 10).
Effizienz	„Performance relative to the amount of resources used under stated conditions“ ([ISO25010], S. 11).
Benutzbarkeit	„Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use“ ([ISO25010], S. 12).
Kompatibilität	„Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment“ ([ISO25010], S. 11).
Zuverlässigkeit	„Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time“ ([ISO25010], S. 13).
Sicherheit	„Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization“ ([ISO25010], S. 13).
Wartbarkeit	„Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers“ ([ISO25010], S. 14).
Übertragbarkeit	„Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another“ ([ISO25010], S. 15).

Tabelle 1: Definition der Qualitätsmerkmale gemäß ISO 25010:2011.

Best Practice	Funktionalität	Effizienz	Benutzbarkeit	Kompatibilität	Zuverlässigkeit	Sicherheit	Wartbarkeit	Übertragbarkeit
Beschreibung von Ressourcen								
Verwendung von Nomen			+					
Konkrete und domänenspezifische Bezeichnung			+				(-)	
Angemessene Anzahl an Ressourcen			(-)				+	
Keine Vermischung des Numerus			+					
Binnenmajuskel-Schreibweise bei Attributen			+					
Beachtung der JavaScript-Namenskonvention			+					
Identifikation von Ressourcen								
Selbsterklärende URI			+					
Maximal zwei URIs pro Ressource			+			+	+	
Nicht vorhersagbarer Identifikator						+		
Keine Verben innerhalb der URI			+				+	
Fehlerbehandlung								
Spezifische HTTP-Statuscodes			+					
Angemessene Anzahl an HTTP-Statuscodes			+					
Applikationsspezifische HTTP-Statuscodes			+					
Fehlerbeschreibung für Entwickler			+					
Fehlerbeschreibung für Nutzer			+					
Hyperlink für weiterführende Fehlerbeschreibung			+					
Dokumentation								
Beispiele zur Interaktion nach HATEOAS			+					
Abstraktes Ressourcenmodell			+				(-)	
Nachschlagewerk für Fehler			+				(-)	
Verwendung von Parametern								
Filterung		+						
Sortierung		+						
Selektion		+						
Paginierung		+						
Interaktion mit Ressourcen								
Korrekte Verwendung der HTTP-Methoden			+					
Unterstützung von HTTP-OPTIONS		+						
Unterstützung von „conditional GET“		+						
Unterstützung von MIME-Typen								
Unterstützung von mindestens zwei Repräsentationsformaten				+				
Festlegung von JSON als Standardformat			+					
Verwendung existierender Hypermedia-Formate			+	+				
Inhaltsvereinbarung mithilfe von HTTP			+	+				

Tabelle 2: Qualitative Betrachtung der Best Practices.

nes Web-Service auswirkt, jedoch negative Auswirkungen auf die Wartbarkeit haben kann, da lediglich 12 bis 24 Ressourcen pro Web-Service verwendet werden sollen. Die Nutzung einer angemessenen Anzahl an Ressourcen hat wiederum positive Auswirkungen auf die Wartbarkeit, kann sich jedoch negativ auf die Benutzbarkeit auswirken, da gegebenenfalls noch einige Ressourcen wünschenswert wären, wodurch aber gegen diese Best Practice verstoßen

würde. Das Beispiel zeigt, dass sich Architekten und Entwickler entscheiden müssen,

welches Qualitätsmerkmal höher zu priorisieren ist. Wird beispielsweise die Wartbarkeit höher gewichtet, so sollte die Anzahl der Ressourcen angemessen gewählt werden, auch wenn die Gefahr besteht, die Benutzbarkeit hierdurch negativ zu beeinflussen.

Ein weiteres Konfliktpotenzial stellt die Dokumentation des Web-Services dar. So wird im optimalen Fall jegliche Änderung am Quellcode nur an einer Stelle durchgeführt, um Fehler zu minimieren und die Wartbarkeit hochzuhalten. Eine Änderung am Quellcode kann allerdings zu einer Veränderung des Ressourcenmodells führen, was schließlich in einem zusätzlichen manuellen Verarbeitungsschritt resultiert, sofern keine automatische Generierung existiert. Gleiches gilt für das Nachschlagewerk bei Fehlern.

Auch wenn die meisten vorgestellten Best Practices in **Tabelle 2** nicht in Konflikt zueinander stehen, kann eine Auswahl wichtiger Qualitätsmerkmale im Zuge der strategischen Ausrichtung des Web-Services erfolgen, wie zum Beispiel das Ziel einer hohen Benutzbarkeit. Das führt dazu, dass die Best Practices mit positiver Auswirkung auf die Effizienz des Web-Services nicht betrachtet werden müssen, was wiederum zu einer Aufwandsminimierung und damit Kosteneinsparung führt.

Automatisierung

Die Identifikation von Best Practices und der Bezug zu [ISO25010] ermöglichen es, RESTful Web-Services systematisch und wiederholbar zu analysieren. Die Best Practices geben hierfür eine klare Auskunft darüber, welche Bestandteile eines Web-Service geprüft werden und wie diese idealerweise gestaltet sein sollten. Dennoch erfordert diese Analyse weiterhin einen gewissen Aufwand, der sich meist aus folgenden Aufgaben zusammensetzt: Zunächst ist es notwendig, die für eine Analyse relevanten Informationen zusammenzutragen. Dies bedeutet, dass zum Beispiel die vorhandenen Web-Services auf Basis von REST innerhalb der Architektur sowie deren Operationen innerhalb des Quellcodes identifiziert werden müssen. In einem zweiten Schritt müssen die zuvor zusammen-

OBJEKTSPEKTRUM ist eine Fachpublikation des Verlags:

SIGS DATACOM GmbH · Lindlaustraße 2c · 53842 Troisdorf

Tel.: 02241/2341-100 · Fax: 02241/2341-199

E-mail: info@sigs-datacom.de

www.objektspektrum.de

www.sigs.de/publications/aboservice.htm

SIGS DATACOM
FACHINFORMATIONEN FÜR IT-PROFESSIONALS

Interpretation	Erläuterung
+	Positive Auswirkung auf das Qualitätsmerkmal.
(-)	Kann eine negative Auswirkung auf das Qualitätsmerkmal haben.
	Keine Auswirkung auf das Qualitätsmerkmal.

Tabelle 3: Interpretation der qualitativen Betrachtung.

getragenen Informationen systematisch auf die Einhaltung von Best Practices hin überprüft werden. Im Vergleich zu vollautomatisierten statischen Quellcode-Analysen erfordern Analysen zur Einhaltung von Best Practices häufig Wissen, wie z.B. Domänenwissen, das über Informationen im Quellcode hinausgeht. Des Weiteren lässt sich aus dem Quellcode nicht immer automatisiert ableiten, ob die gewünschten Best Practices eingehalten wurden, da sich diese in verschiedenster Form im Quellcode abbilden lassen.

Aus diesem Grund schlagen wir einen Ansatz zur hybriden Analyse vor, wie wir ihn bereits für Web-Services in serviceorientierten Architekturen auf der Basis von SOAP (vgl. [Geb11], [Geb14-a]) und zuletzt auch zur Analyse von Geschäftsprozessen auf der Basis von BPMN (vgl. [Geb14-b], [Geb14-c]) vorgestellt haben. Dieser Ansatz sieht vor, dass die Analyse soweit wie möglich automatisiert durchgeführt wird,

jedoch den Experten mit einbezieht, sobald keine weitere Automatisierung mehr möglich ist. Ob eine Automatisierung in vertretbarem Aufwand umsetzbar ist oder nicht, hängt dabei stark von den verwendeten Technologien ab.

Das folgende Beispiel veranschaulicht die hybride Analyse: Um Web-Services, die in Java mittels JAX-RS entwickelt wurden, auf die Einhaltung der beschriebenen Best Practices hin zu untersuchen, müssten im Falle einer manuellen Analyse zunächst der Quellcode des Projekts durchsucht und alle in Java mittels JAX-RS entwickelten Web-Services ermittelt werden. Hierfür können gegebenenfalls Werkzeuge wie Entwicklungsumgebungen zur Unterstützung herangezogen werden. Anschließend müssten die Best Practices zunächst im Detail verstanden und für jeden der identifizierten Web-Services Schritt für Schritt geprüft werden. Am Beispiel der „selbsterklärenden URI“ müssten IT-Architekt und

Entwickler noch einmal genau nachvollziehen, wie dieses Best Practice zu verstehen ist. Anschließend müssten sie für jeden gefundenen Web-Service die Annotation „@Path“ herausuchen und die URI müsste dahingehend bewerten, ob sie selbsterklärend ist oder nicht. Dies kann bei einer großen Anzahl an Web-Services und auch einer größeren Menge an relevanten Best Practices sehr aufwändig werden.

Im Fall der hybriden Analyse würde ein Großteil der Analyse automatisiert erfolgen: Das System versteht, aus welchen Informationsbedürfnissen sich eine Anfrage zusammensetzt, und versucht, diese soweit wie möglich automatisiert zu beantworten. Das obige Beispiel besteht aus den folgenden Informationsbedürfnissen:

- Suche nach allen Web-Services, die in Java mit JAX-RS entwickelt wurden.
- Identifiziere für jeden Web-Service die Annotation „@Path“.
- Extrahiere für jede gefundene Annotation die URI.
- Bestimme, ob die URI selbsterklärend ist oder nicht.

In der hybriden Analyse und im Fall der „selbsterklärenden URI“ würde die Analyse zunächst automatisiert nach allen in Java mit JAX-RS entwickelten Web-Services suchen. Anschließend würde innerhalb dieser Web-Services automatisiert die Annotation

Quality Attribute	Element	Value	Assignees	Persons	Yes	No	Other
Managed Entities	construct ID: AutomotiveManufacturer(public_html)\WEB-INF\wsdl\Construction.wsdl#WSDL1X#OPERATION#/*[local-name()='operation' and namespace-uri()='http://schemas.xmlsoap.org/wsdl/' and @name='construct']	Operation [operation]	Assignees	Persons	(No Entities)	Edit List	No Entities
Business-Relation	manufacture ID: AutomotiveManufacturer(Manufacturing.wsdl#WSDL1X#OPERATION#/*[local-name()='operation' and namespace-uri()='http://schemas.xmlsoap.org/wsdl/' and @name='manufacture']	Operation [wsdl:operation]	Assignees	Persons	Yes	No	x
Long Running	manufacture ID: AutomotiveManufacturer(Manufacturing.wsdl#WSDL1X#OPERATION#/*[local-name()='operation' and namespace-uri()='http://schemas.xmlsoap.org/wsdl/' and @name='manufacture']	Operation [wsdl:operation]	Assignees	Persons	Yes	No	x
Abstraction	manufacture ID: AutomotiveManufacturer(Manufacturing.wsdl#WSDL1X#OPERATION#/*[local-name()='operation' and namespace-uri()='http://schemas.xmlsoap.org/wsdl/' and @name='manufacture']	Operation [wsdl:operation]	Assignees	Persons	Yes	No	x
Compensating	manufacture ID: AutomotiveManufacturer(Manufacturing.wsdl#WSDL1X#OPERATION#/*[local-name()='operation' and namespace-uri()='http://schemas.xmlsoap.org/wsdl/' and @name='manufacture']	Operation [wsdl:operation]	Assignees	Persons	Yes	No	x
Functional Naming	Manufacturing.wsdl ID: AutomotiveManufacturer(Manufacturing.wsdl#WSDL1X#SERVICE_INTERFACE#/*[local-name()='definitions' and namespace-uri()='http://schemas.xmlsoap.org/wsdl/' and not(@name)])	Service Interface [wsdl:definitions]	Assignees	Persons	No	✓	x
Functional Naming	ManufacturingPortTypeProvider ID: AutomotiveManufacturer(Manufacturing.wsdl#WSDL1X#ROLE#/*[local-name()='role' and namespace-uri()='http://schemas.xmlsoap.org/ws/2003/05/partner-link/' and @name='ManufacturingPortTypeProvider']	Role [plink:role]	Assignees	Persons	Yes	No	x

Abb. 1: Beantwortung von Fragen durch Architekten und Entwickler im QA82 Analyzer.

„@Path“ ermittelt und die entsprechende URI extrahiert werden. Das letzte Informationsbedürfnis – die Bestimmung, ob die URI selbsterklärend ist oder nicht – erfordert jedoch Domänenwissen, weshalb eine automatisierte Überprüfung nicht möglich ist. Der hybride Ansatz sieht in diesem Fall vor, dass nun Experten – in diesem Kontext Architekten oder Entwickler – hinzugezogen werden können. Diesen wird die URI präsentiert und eine Frage gestellt, die das Informationsbedürfnis repräsentiert. In diesem Fall würde gefragt werden, ob die dargestellte URI selbsterklärend ist. Die Frage müsste mit Ja oder Nein beantwortet werden. Im Anschluss wird die Antwort gespeichert und bei der nächsten Analyse mit einbezogen.

Der hybride Ansatz hat somit drei zentrale Vorteile:

- Architekten und Entwickler bekommen die manuell zu analysierenden Informationen automatisiert präsentiert.
- Sie müssen diese nicht zuerst mit hohem Aufwand eigenständig, beispielsweise aus dem Quellcode, zusammentragen.
- Des Weiteren werden die für eine vollautomatisierte Analyse erforderlichen Informationen durch die Beantwortung der Fragen schrittweise ergänzt, sodass der Grad der Automatisierung sukzessive erhöht wird.

Abbildung 1 zeigt in der ersten Spalte eine beispielhafte Auflistung von Fragen für die Analyse von Web-Services in serviceorientierten Architekturen auf der Basis von SOAP.

Literatur & Links

[Geb11] M. Gebhart, Qualitätsorientierter Entwurf von Anwendungsdiensten, KIT Scientific Publishing 2011

[Geb14-a] M. Gebhart, Query-Based Static Analysis of Web Services in Service-Oriented Architectures, in: International Journal on Advances in Software Vol. 7, 2014

[Geb14-b] M. Gebhart, M. Mevius, P. Wiedmann, Schlank mit Qualität, in: OBJEKTSpektrum 02/2014

[Geb14-c] M. Gebhart, M. Mevius, P. Wiedmann, F. Kurz, Ausgesprochen hochwertig: Hybride Qualitätskontrolle in agilem BPM, in: OBJEKTSpektrum 05/2014

[Geb15] M. Gebhart, P. Giessler, S. Abeck, RESTful Web-Services mit Qualität – Teil 1: Mit Best Practices zu einem qualitätsorientierten Entwurf, in: OBJEKTSpektrum 01/2015

[Inf11] InfoQ, How REST replaced SOAP on the Web: What it means to you, 2011, siehe:

<http://www.infoq.com/articles/rest-soap>

[ISO25010] ISO und IEC, ISO/IEC 25010:2011, siehe:

<http://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

[QA82] QA82 Analyzer, siehe: www.qa82.com

Ein Beispiel für ein Werkzeug, das den Ansatz der hybriden Analyse unterstützt, ist der „QA82 Analyzer“. Dieser wurde in der Vergangenheit als proprietäres Werkzeug entwickelt. Aktuell erfolgt jedoch die Entwicklung eines Nachfolgers als Open-Source (vgl. [QA82]), sodass der hier vorgestellte Ansatz zur Automatisierung von jedem Interessenten in Projekten verwendet und um eigene Überprüfungen erweitert werden kann. Ebenso wurde der Aufbau des QA82 Analyzer von Michael Gebhart in [Geb14-a] dokumentiert, sodass die Umsetzung einer hybriden Analyse öffentlich einsehbar ist.

Fazit

In diesem zweiten Teil des Artikels haben wir die Best Practices aus dem ersten Teil mit Qualitätsmerkmalen der ISO/IEC 25010:2011

verknüpft. Da aus Kostengründen oder wegen sich gegenseitig ausschließenden Eigenschaften selten alle Best Practices gleichzeitig berücksichtigt werden können, ermöglicht der Bezug zur ISO/IEC 25010:2011 eine Priorisierung der Best Practices. Um unsere Ergebnisse direkt in Projekten anwenden zu können, haben wir diese in Form eines Bewertungsbogens zusammengefasst. Zusätzlich haben wir einen Ansatz zur Automatisierung einer Analyse von Best Practices vorgestellt. Der hierbei genutzte hybride Ansatz kombiniert automatisierte Analysen mit manuellem Wissen. Unser Ziel ist es, durch die Zusammenfassung der Best Practices im ersten Teil sowie die Verknüpfung mit der ISO/IEC 25010:2011 und die Methode zur Automatisierung in diesem zweiten Teil eine hohe Qualität von Web-Services systematisch sicherzustellen. ||

Die Autoren



|| Dr. Michael Gebhart
(michael.gebhart@iteratec.de)

ist IT-Management-Berater bei der iteratec GmbH. Als Berater unterstützt er Kunden bei der Konzeption und Durchführung von IT-Projekten sowie der strategischen Beratung im EAM-Umfeld. Seine Forschungsschwerpunkte sind die Qualitätsanalyse von Web-Services und SOAs.



|| Pascal Giessler
(pascal.giessler@iteratec.de)

ist Software-Ingenieur bei der iteratec GmbH in Stuttgart. Neben dieser Tätigkeit strebt er eine externe Promotion im Bereich der Analyse von Softwaresystemen am Karlsruher Institut für Technologie (KIT) innerhalb der Forschungsgruppe Cooperation & Management (C&M) an.



|| Prof. Dr. Sebastian Abeck
(abeck@kit.edu)

leitet an der Fakultät für Informatik des KIT die Forschungsgruppe Cooperation & Management (C&M). Seine Forschungsinteressen betreffen die Service-orientierten Architekturen, das Internet der Dinge und das Identitäts- und Zugriffsmanagement.