

# **Verlustfreie Anforderungen – das Wichtigste zu ReqIF**

## **Effektivere Systementwicklung mit dem offenen Standard**

Dr. Michael Jastram, Formal Mind

**Die Produktentwicklung heutzutage steht enorm unter Druck: Steigende Komplexität, steigende Anzahl von Lieferanten, Vorschriften, jahrzehntelange Verfügbarkeit von Artefakten der Entwicklung. Der ReqIF-Standard adressiert diese Herausforderungen zweifach: Zum einen stellt die Standardisierung sicher, dass Anforderungen auch nach Jahrzehnten noch lesbar sind; zum anderen ermöglicht der Standard eine Integration zum verlustfreien, nachverfolgbaren Austausch von Anforderungen.**

### **Motivation**

Der Austausch von Anforderungen wird zunehmend wichtiger, aus primär zwei Gründen: Zum einen wird zunehmend mit Zulieferern und externen Dienstleistern gearbeitet. Zum anderen werden die Anforderungen bezüglich Nachverfolgbarkeit immer strenger, insbesondere in Entwicklungen mit Aspekten der funktionalen Sicherheit.

Bis vor wenigen Jahren wurden Anforderungen entweder in verlustbehafteten Formaten ausgetauscht (PDF, Excel, Word), oder es mussten proprietäre Lösungen genutzt werden. Doch seit 2011 gibt es mit ReqIF einen offenen Standard, der den verlustfreien Austausch von Anforderungen ermöglicht [ReqIF].

ReqIF steht für „Requirements Interchange Format“ und stammt aus der Automobilindustrie. Dort wurden vermehrt Anforderungen an Zulieferer geschickt, die diese bewerten und kommentieren sollen. Dieses Szenario kann mit ReqIF umgesetzt werden. Dazu schickt der Hersteller die Anforderungen als ReqIF zum Zulieferer, der die Bewertungsattribute mit Werten befüllt und an den OEM zurückschickt.

### **Die Geschichte**

ReqIF 1.0.1 wurde 2011 als Standard der Object Management Group (OMG) veröffentlicht und darf von Nutzern und Werkzeugherstellern lizenzfrei genutzt werden. Doch die Geschichte von ReqIF begann bereits 2004, als ein Konsortium mehrerer OEMs die erste Version des Standards mit dem Namen RIF in Leben rief und die prototypische Entwicklung eines Werkzeugs finanzierte, mit welchem ein RIF-basierter Anforderungsaustausch durchgeführt werden konnte.

Der Standard durchlief mehrere Iterationen, bevor er 2011 an die OMG übergeben wurde. Da die OMG bereits einen Standard mit dem Namen RIF hatte, wurde das Format in ReqIF unbenannt, und auch die Versionsnummer auf 1.0 zurückgesetzt.

Auch wenn es schon vor 2011 die Unterstützung von kommerziellen Werkzeugen gab, so sind die Implementierungen erst jetzt kurzem wirklich praxistauglich. Auch findet das Format langsam auch außerhalb der Automobilindustrie Beachtung.

Die aktuelle Version von ReqIF ist 1.2. Allerdings ist diese identisch zu ReqIF 1.0, was manchmal zu Verwirrungen führt. Geändert hat sich lediglich der begleitende Text, jedoch nicht das Format selbst.

Mit dem Eclipse Requirements Modeling Framework (RMF) [RMF] gibt es auch bereits seit 2009 eine Referenzimplementierung von ReqIF, was ebenfalls zur Verbreitung von ReqIF beigetragen hat [OS13]. Dieses Open Source-Projekt wird sowohl von mehreren kommerziellen Werkzeugen als Bibliothek zum Verarbeiten von ReqIF eingesetzt, und kann auch eigenständig unter dem Namen ReqIF Studio als einfacher Requirementseditor eingesetzt werden [Studio].

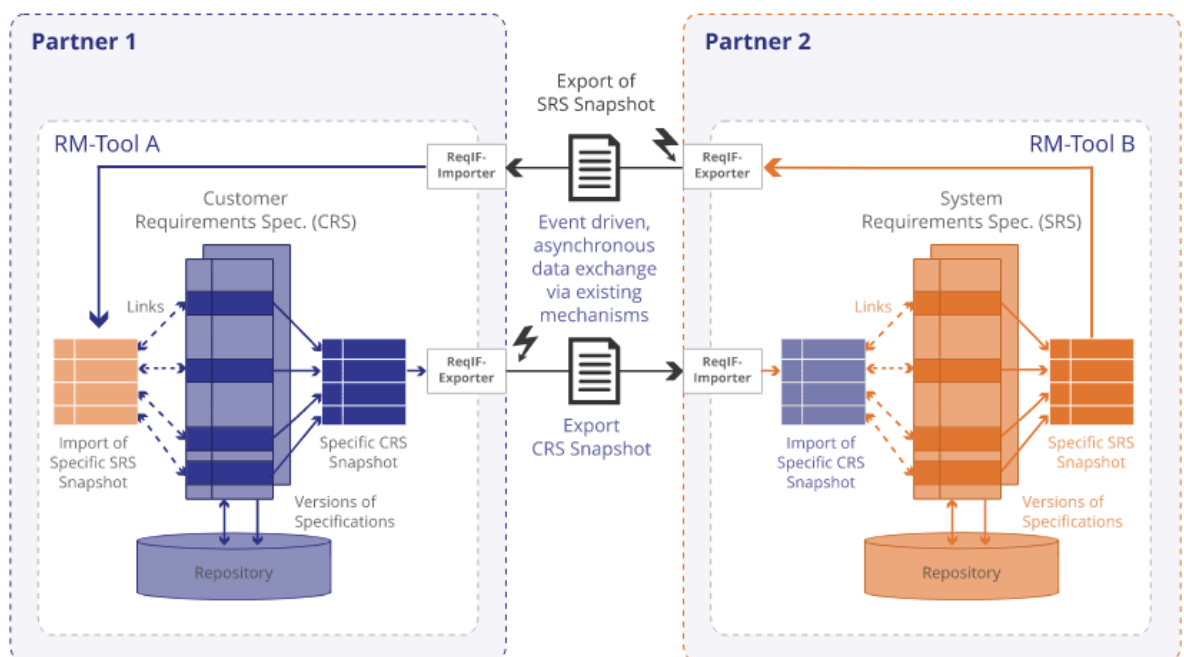
### Das Requirements Interchange Format

ReqIF ist ein XML-basiertes Dateiformat. Im ReqIF-Datenmodell werden Anforderungen „SpecObjects“ genannt und können beliebig viele, typisierte Attribute haben. Sichten auf die SpecObjects werden über „Specifications“ realisiert, die diese in einer Baumstruktur anordnen. Das Datenmodell unterstützt die Verlinkung von SpecObjects über „SpecRelations“, die ebenfalls beliebig viele, typisierte Attribute haben.

Eine ReqIF-Datei enthält nun die zu übertragenden Spezifikationen, die verlinkte Anforderungen enthalten. Die ReqIF-Datei enthält allerdings auch die Definitionen der Elemente, also die verwendeten Anforderungs- und Linktypen. Dabei werden alle gängigen Attributtypen unterstützt, wie einfacher und formatierter Text, Aufzählungen, numerische Werte, usw. Abgerundet wird das Format mit einem einfachen Rechtekonzept.

### ReqIF-Anwendungsfälle

Der wichtigste Anwendungsfall ist in der Abbildung zu sehen. Dort ist ein Roundtrip zwischen zwei Partnern gezeigt. Ein typischer ReqIF-Austausch läuft folgendermaßen ab:



- Partner 1 konfiguriert in dessen Werkzeug den Export. Dabei wird definiert, welche Spezifikationen, Anforderungen und Attribute übertragen werden sollen. Typischerweise soll der Partner nicht alle Anforderungen sehen, und auch nicht alle Attribute der zu übertragenden Elemente.
- Partner 1 führt den Export aus, der eine ReqIF-Datei (.reqif) produziert. Falls der Export auch Bilder oder Anhänge enthält, wird ein ReqIF-Archiv (.reqifz) generiert.
- Partner 1 überträgt den Export an Partner 2. Das könnte im einfachsten Fall per E-Mail realisiert werden.
- Partner 2 importiert das ReqIF in deren System. Im einfachsten Fall handelt es sich dabei um einen simplen Import, es ist aber auch möglich, die Daten in bestehende Anforderungen einzupflegen. Falls bspw. der Roundtrip in der Vergangenheit schon mal durchgeführt wurde, könnte diese Daten nun aktualisiert werden. Auch ist es für Partner 2 möglich, die importierten Anforderungen mit eigenen Anforderungen oder Attributen zu ergänzen, oder mit anderen Spezifikationen zu verlinken.
- Partner 2 verarbeitet nun die Daten, wobei im nächsten Kapitel die verschiedenen Anwendungsfälle vorgestellt werden. Wenn dies geschehen ist, kann das Ergebnis wieder als ReqIF exportiert und an Partner 1 zurückgeschickt werden.
- Partner 1 kann nun das Ergebnis in dessen Datenhaltung automatisiert einpflegen.

Basierend auf dem eben beschriebenen Prozess können nun verschiedene Anwendungsfälle umgesetzt werden. Die wichtigsten drei werden hier beschrieben, viele andere sind denkbar.

### **ReqIF-basierter Review**

Im einfachsten Fall soll ein Zulieferer die Anforderungen des Herstellers bewerten. Dazu fügt der Hersteller den Anforderungen Bewertungsattribute hinzu, typischerweise ein Status-Attribut („Approved“, „Rejected“, „Clarify“) sowie ein Kommentarattribut, über das zusätzliche Information als Freitext hinzugefügt werden kann.

Da dieser Anwendungsfall so wichtig ist, hat die Arbeitsgruppe [HIS], die RIF erstellt hat, auch einen Standardprozess entworfen, den sogenannten „HIS-Prozess“. Der HIS-Prozess spricht Empfehlungen für die Bewertungsattribute aus und wurde von der Arbeitsgruppe im Internet veröffentlicht.

### **Offene Punkte-Liste**

Ein weiterer Anwendungsfall ist das gemeinsame Verwalten einer Offenen Punkte-Liste. Dies bietet sich für zusammenarbeiten an, bei denen es um große Mengen von Anforderungen geht, aber einer kleinen Menge von offenen Punkte. Hier können beide Parteien offene Punkte anlegen, die dann über eine Traceability mit den betroffenen Anforderungen verknüpft werden. Dies ist möglich, da ReqIF auch Verlinkungen übertragen kann.

### **Lastenheft-Pflichtenheft**

Es ist auch möglich, zwei Spezifikationen gleichzeitig zu übertragen, wie ein Lastenheft und ein Pflichtenheft. Auf diesem Wege ist es möglich, beide

Spezifikationen regelmäßig zu aktualisieren, wobei der Hersteller die Hoheit über das Lastenheft hat und der Zulieferer über das Pflichtenheft. Die Traceability zwischen den beiden kann ebenfalls geteilt werden, was ohne ReqIF nur schwer möglich wäre.

### **Kompatibilität**

Leider bedeutet Standard nicht gleich Kompatibilität. Und leider hat ReqIF auch mehrere Eigenschaften, die zu Inkompatibilitäten zwischen Werkzeugen führen. Zum Beispiel ist das Format so mächtig, dass es Features unterstützt, die viele Werkzeuge selbst nicht unterstützten. Wenn ein (valides) ReqIF nun ein solches Feature nutzt, kann das importierende Werkzeug Schwierigkeiten haben.

Zum Glück wurde dies frühzeitig erkannt, und mit der Veröffentlichung von ReqIF wurde auch ein Industriekonsortium gegründet, welches die Kompatibilität von ReqIF-Implementierungen sicherstellen sollte. Das ReqIF Implementor Forum [RF] hat zu diesem Zweck einen Leitfaden für Hersteller von RE-Software veröffentlicht, welches die Kompatibilität erhöhen soll.

Die bereits erwähnte Referenzimplementierung [RMF] erhöht ebenfalls die Kompatibilität. Zum einen sind Werkzeuge, die RMF konsumieren, de Facto kompatibel, da RMF 100% kompatibel ist. Weiterhin kann RMF auch zum Testen eingesetzt werden. Zu RMF gehört ein Validator [Val]. Dieser überprüft nicht nur, ob eine ReqIF-Datei korrektes ReqIF enthält, sondern auch, ob die Konventionen des Implementor Forums eingehalten werden.

### **ReqIF effektiv nutzen**

Viele Nutzer, insbesondere Zulieferer, haben kaum eine Wahl: Wenn der Kunde einen ReqIF-basierten Anforderungsaustausch verlangt, dann gibt es kaum eine Wahl. Falls dies auf die Einführung eines neuen Werkzeugs hinausläuft, sollte unbedingt das Thema Kompatibilität genauer betrachtet werden.

Aber selbst wenn der Druck von außen nicht vorhanden ist, könnte ein Einsatz für die folgenden Zwecke interessant sein:

- Einsatz mit den eigenen Zulieferern. Insbesondere, wenn der Gesetzgeber die Nachverfolgbarkeit zum Zulieferer vorschreibt.
- Langfristige (verlustfreie) Archivierung von Anforderungen.
- Falls aus anderen Gründen ein neues Werkzeug eingeführt wird, sollte aus ReqIF-Unterstützung geachtet werden, um Vendor-Lock-in zu vermeiden

### **Fazit**

ReqIF ist eine Technologie die erwachsen ist, von immer mehr Werkzeugen unterstützt wird und auch immer mehr eingesetzt wird. Wer mit Anforderungen arbeitet, sollte die Möglichkeiten und Grenzen von ReqIF kennen.

### **Literatur und Quellen**

[HIS] Der HIS-Prozess war für lange Zeit auf der Webseite der HIS verfügbar (<http://www.automotive-his.de/>). Diese ist allerdings schon seit 2016 nicht mehr erreichbar. Alternativ (registrierungspflichtig) unter <https://reqif.academy/reference/his-process-documentation/>

[ReqIF] <http://www.omg.org/spec/ReqIF/1.2/>

[RF] <http://www.prostep.org/projekte/reqif-implementor-forum/>

[RMF] <https://www.eclipse.org/rmf/>

[OS13] Michael Jastram: „ReqIF-OLUTION: Mit Eclipse und ReqIF zur Open-Source ALM-Werkzeugkette“, ObjektSpektrum, 2013

[Studio] ReqIF Studio Download, registrierungspflichtig unter <https://reqif.academy/software/reqif-studio/>

[Val] Consequent ist der ReqIF Validator. Der Quellcode ist unter [RMF] erhältlich. Das ausführbare Programm kann heruntergeladen werden (registrierungspflichtig) unter <https://reqif.academy/software/consequent/>

### **Michael Jastram**

Dr. Michael Jastram ist Systems Engineer mit Schwerpunkt auf die Anforderungsmodellierung. Er ist Gründer und Projekt Lead des Eclipse Requirements Modeling Frameworks, einem Open Source-Projekt zur Anforderungsmodellierung, das als Referenzimplementierung den offenen ReqIF-Standard umsetzt. Als Advokat für Offenheit teilt er sein Wissen über Bücher, Fachartikel, Vorträge und als Veranstalter, sowie über seinen wöchentlichen deutschsprachigen Blog [System Engineering Trends](#) und den englischsprachigen [Formal Mind Blog](#). Herr Jastram hat zwanzig Jahre Berufserfahrung, davon zehn in den USA, wo er einen Abschluss am M.I.T. erwarb und als Software-Ingenieur und -Architekt in mehreren Start-Ups arbeitete. Er ist Gründer der Düsseldorfer Java User Group ([rheinjug e.V.](#)). Heute ist er Geschäftsführer der Formal Mind GmbH und betreibt die [ReqIF.academy](#), einer Online-Bibliothek rund um den Austausch von Anforderungen. Weiterhin unterstützt er [Jama Software](#) als Senior Solutions Architect.