



□ Dipl.-Ing. Johannes Bergsmann
 (johannes.bergsmann@software-quality-lab.com)
 ist Geschäftsführender Gesellschafter, Berater & Trainer
 bei Software Quality Lab.

„Definition of Ready“ für agile Requirements

Die „Definition of Ready“ (DoR) für das Requirements Engineering in agilen Vorgehensweisen ist als Eingangskriterium bzw. als Qualitäts-Checkliste für umzusetzende Requirements zu sehen. Nur Requirements, die diesen Qualitätslevel erfüllen, dürfen in die Umsetzungsplanung des Sprints genommen werden. Dieser Beitrag erklärt, was dabei zu beachten ist und wie eine Definition of Ready formuliert werden kann.

„Definition of ...“ – Quality-Gates im agilen Umfeld

In agilen Vorgehensweisen gibt es eine Reihe von Quality Gates, die sicherstellen sollen, dass die Umsetzung der Anforderungen und die Abwicklung eines Projekts in ausreichender Qualität passieren. In [Abbildung 1](#) sind einige Beispiele dafür angegeben. Auf der Ebene der Anforderungen sind insbesondere die Definition of Ready (DoR) sowie die Definition of Done (DoD) wichtig.

Definition of Ready (DoR)

Bevor die Anforderungen vom Team geschätzt und für die Umsetzung eingeplant werden können, muss sichergestellt sein, dass sie verständlich, vollständig und detailliert genug beschrieben sind. Dies ist Aufgabe des Produkt Owners (PO) in Abstimmung mit dem Entwicklungsteam.

Neben der Verständlichkeit, Vollständigkeit und Detailliertheit gibt es im Requirements Engineering eine Reihe von weiteren Qualitätskriterien, die erfüllt sein müssen, sowie Techniken zu deren Überprüfung. Eine Voraussetzung für die Erstellung eines Produktinkrements ist, dass die dazu eingeplanten Anforderungen bereit für die Umsetzung sind.

Welche Qualitätskriterien eine Anforderung

tatsächlich erfüllen muss und welche Analysen durchgeführt sein müssen, damit sie als „bereit für die Umsetzung“ bezeichnet werden darf, wird in einer „Definition of Ready“ festgelegt. Analog zur DoD werden darin diejenigen Kriterien explizit sichtbar gemacht, die eine Anforderung erfüllen muss, bevor sie für die Umsetzung in der nächsten Iteration akzeptiert wird. Diese können helfen, Miss-

verständnisse zwischen Team und Product Owner von vornherein zu vermeiden und die Effizienz weiter zu steigern.

Die DoR ist eine Liste mit Kriterien für die Aufnahme eines Requirements aus dem Product Backlog in den Sprint Backlog. Sie dient der Freigabe dieser Requirements für die Umsetzung.

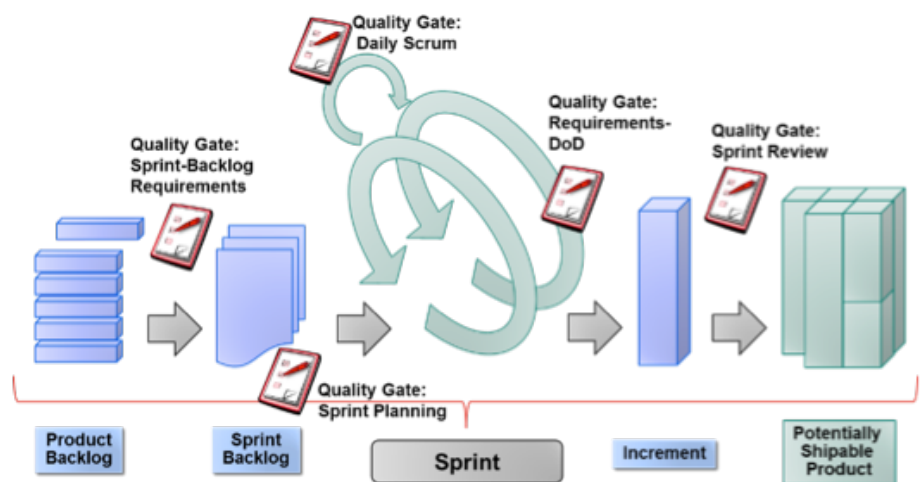


Abb. 1: Quality Gates in Scrum
 (hier Beispiel funktionierender Hochwasserschutz in agilen Projekten)

Die Definition of Ready wird, wie die Definition of Done, zwischen PO und dem Umsetzungsteam vereinbart. Wie bei der DoD, kann es auch bei der DoR sein, dass einzelne Kriterien nicht für jedes Requirement angewendet werden müssen. So ist es möglicherweise sinnvoll, für unterschiedliche Risikoklassen von Requirements auch unterschiedliche DoR-Kriterien anzuwenden.

Werden für ein Projekt oder eine agile Organisation Qualitätskriterien für Requirements definiert, so sollten diese jedoch auch in die DoR aufgenommen werden.

Die **Tabelle** zeigt **Beispiele für Kriterien**, die in eine Definition of Ready aufgenommen werden können. Für jedes Kriterium ist beispielhaft auch angegeben, für welche Risikoklasse dieses Kriterium bei der Requirements-Freigabe angewendet werden muss.

Die DoR kann über den Projektverlauf angepasst, erweitert oder auch reduziert werden, wenn sich das Team, der PO und die Stakeholder „eingeschwungen“ haben und die Zusammenarbeit so gut funktioniert, dass auch ohne laufendes Durchgehen einer Checkliste die Qualitätskriterien selbstständig eingehalten werden.

Die DoR kann als Checkliste beim Taskboard aufliegen. Die Storys werden dann in der Iterationsplanung entsprechend gemeinsam bewertet und bei Erfüllung der Kriterien in den Sprint Backlog aufgenommen. Die DoR-Bewertungen werden in diesem Fall typischerweise nicht niedergeschrieben. Durch das Aufnehmen der Story in den Sprint Backlog ist implizit dokumentiert, dass sich das ganze Team einig war, dass die Kriterien auch erfüllt sind.

In manchen Fällen, wie z. B. bei sicherheitskritischer Software oder anderen haftungsrelevante Umgebungen, kann es notwendig sein, die DoR-Durchführung nachvollziehbar zu dokumentieren. Um die DoR hier effizient und nachvollziehbar durchführen zu können, ist es sinnvoll, dies toolunterstützt zu machen, z. B. mit einem entsprechenden Requirements-Management-Werkzeug oder einem elektronischen Taskboard.

Aufpassen muss man bei der DoR, dass diese nicht als Ausrede verwendet wird, um Diskussionen über User Storys vom Entwickler-Team fernzuhalten, was dann genau den agilen Grundsätzen widersprechen würde.

Allgemeine Requirements-Qualitätskriterien

Nachfolgend werden einige allgemeine

Checkliste:

Prüfpunkt	für Risiko
Ist die Anforderung vollständig beschrieben? Sind UI-Masken ausreichend beschrieben? ■ Grafischer Maskenentwurf / Skizze ■ Mögliche Aktionen ■ Angezeigten Daten ■ Regeln ■ Berechtigungen zur Anzeige / Bearbeitung	H,M
Sind die externen Systemschnittstellen ausreichend beschrieben? ■ Datenobjekte und Felder inkl. möglicher Feldwerte ■ Austauschformat ■ Austauschprozess ■ Erwartete Datenmengen ■ Fehlerbehandlung	H,M,L
Sind Berichte ausreichend beschrieben? ■ Grafischer Entwurf / Skizze ■ Angezeigte Daten ■ Berechtigungen zur Anzeige	H
Sind nicht-funktionale Anforderungen definiert und geprüft?	H,M
Ist die Story verständlich und eindeutig formuliert?	H,M,L
Werden keine Universalquantoren (alle, nichts, etc.), Konjunktive, Passivkonstruktionen und Relativierungen verwendet?	H,M,L
Wird die Satzschablone für User Stories eingehalten („Als ROLLE möchte ich AKTION, um NUTZEN“)?	H,M,L
Ist die Story bewertet?	H,M,L
Herrscht Übereinstimmung bei allen Stakeholdern, dass die Story wichtig für das Produkt ist?	H,M,L
Sind folgende Priorität, Business Value und Risiko bestimmt?	H,M,L
Sind die relevanten Akzeptanzkriterien definiert? <i>Manche Akzeptanzkriterien ergeben sich erst im Zuge der Umsetzung, die wesentlichen müssen aber schon vorher definiert sein.</i>	H,M,L
Sind die Risiken für die Story identifiziert und bewertet?	H,M
Sind die Risiken in einer Risikoliste dokumentiert?	H,M,L
Sind für Risiken mit Wert > 15 Maßnahmen definiert? Wurden Risiken mit Wert > 20 an die GF gemeldet?	H,M
Wird die Story vom Tester als „testbar“ eingestuft?	H,M,L
Wurden die Testfälle erstellt? <i>Basierend auf den Akzeptanzkriterien muss ein Testplan für die Prüfung der Story erstellt werden.</i>	H,M
Wurde überlegt, wer die Story abnimmt und wie das Team sie demonstrieren kann? <i>Oft wird eine Story nicht vom PO, sondern vom Kunden abgenommen. In diesem Fall muss dies vorher geplant werden.</i>	H,M,L
Ist das Entwicklungssystem bereit für die Story? <i>Kann aus personeller, technischer und Infrastruktursicht mit der Umsetzung der Story begonnen werden?</i>	H,M,L
Erfüllt die Story die INVEST Kriterien? <i>Die INVEST-Kriterien werden weiter hinten in diesem Beitrag beschrieben.</i>	H,M,L
Sind alle Abhängigkeiten* identifiziert? <i>Abhängigkeiten können die Umsetzung einer Story verzögern und die Prüfung erschweren.</i> <i>* Dies widerspricht eigentlich dem INVEST-Kriterium „independent“. In der Praxis kann es jedoch vorkommen, dass Storys nicht ganz unabhängig sind. In diesem Fall sollten dann die Abhängigkeiten angegeben sein.</i>	H
...	

Tab: Beispielkriterien für eine Definition of Ready Checkliste

Qualitätskriterien für Requirements angeführt, die ebenfalls als Quelle für die in die DoR aufzunehmenden Kriterien herangezogen werden können.

Qualitätskriterien nach IEEE 830-1998 und IREB

Qualitätskriterien sind unter anderem im Standard IEEE 830-1998 oder von IREB (IREB CPRE FL 2012, kein Datum) definiert:

■ Verständlich und eindeutig

Alle Anforderungen müssen spätestens vor der Schätzung in einem Zustand sein, dass jeder im Team sie versteht. Sie müssen auch so eindeutig beschrieben sein, dass es keine unterschiedlichen Interpretationen über den Inhalt der Anforderung gibt. Hier kann es helfen, bestimmte mehrdeutige Worte und Satzkonstrukte zu *vermeiden*:

- Universalquantoren („alle“, „jeder“, „nichts“, „immer“, etc.)
- Passive Formulierungen
- Konjunktive und relativierende Formulierungen („meistens“, „manchmal“, „sollte“, „dürfte“).

■ Vollständig und angemessen

Um Vollständigkeit zu erlangen, müssen *alle* Anforderungen *aller* relevanten Stakeholder gesammelt werden.

Damit keine Stakeholder übersehen werden, ist im Vorfeld eine Stakeholderanalyse sinnvoll.

Die Vollständigkeit kann durch gute Fragetechnik des Requirements-Erstellers, durch entsprechendes Hintergrundwissen und Einblick in die zu spezifizierende Domäne gefördert werden. Leider ist es jedoch nicht möglich, die Vollständigkeit endgültig sicherzustellen.

Angemessenheit ist relativ schwierig zu erlangen, weil es subjektiv sehr unterschiedlich sein kann, ob eine Anforderung angemessen ist – vor allem auch bei Betrachtung durch verschiedene Stakeholder. Was eventuell für den einen Stakeholder wichtig ist, ist für den anderen reine Zeit- und Geldverschwendung.

■ Abgestimmt

Der PO oder auch andere Auftraggeber sind alleine manchmal nicht in der Lage, eine Anforderung korrekt, vollständig und für die Umsetzer verständlich zu beschreiben. Daher sollten die Anforderungen mit allen relevanten

Stakeholdern abgestimmt werden. Möglichkeiten dazu sind beispielsweise in Review-Meetings, im Sprint Planning Meeting aber auch jederzeit bei Bedarf vor der Umsetzung der jeweiligen Anforderungen.

■ Gültig und aktuell

Anforderungen können sich laufend ändern. Vor jeder Umsetzungsiteration muss daher der PO neu prüfen, ob die Anforderungen den aktuellen Stand der Kundenwünsche widerspiegeln und noch gültig sind, bevor diese in einen Sprint für die Umsetzung aufgenommen werden.

■ Realisierbar

Mit Anforderungen, die nicht umsetzbar sind, sollte sich das Team gar nicht weiter beschäftigen. Diese Anforderungen sollten durch den „DoR-Filter“ gar nicht erst in den Umsetzungsprint gelangen. Gegebenenfalls muss die Umsetzbarkeit eines Requirements vorab durch eine Machbarkeitsanalyse geprüft werden.

■ Bewertet

Für die Release- und Iterationsplanung muss jede Anforderung bewertet sein. Ein wichtiges Eingangskriterium in den Umsetzungsprint ist die Bewertung, ob das Requirement die DoR erfüllt.

Weitere Bewertungsaspekte können z. B. Risiko, Dringlichkeit, Umsetzungsaufwand, etc. sein.

■ Korrekt

Eine Anforderung ist dann korrekt, wenn sie die *tatsächlichen* Bedürfnisse der Stakeholder widerspiegelt.

■ Konsistent

Die Anforderungen sollen in sich und über alle Anforderungen hinweg konsistent, also widerspruchsfrei sein.

Da das Product Backlog lediglich die noch offenen Anforderungen enthält und somit keine Gesamtspezifikation existiert, kann mit dem Backlog alleine die Widerspruchsfreiheit nicht geprüft werden. Des Weiteren ist der Detaillierungsgrad weiter in der Zukunft liegender Anforderungen recht grob und diese damit kaum auf Konsistenz prüfbar. Die Prüfung auf Konsistenz wird daher durch den PO und das Team gemeinsam durchgeführt. Durch die enge Zusammenarbeit zwischen PO (fachlich)

und Team (technisch) erfolgt ein ständiger Austausch und ein ständiges Prüfen, ob die gerade in Umsetzung befindlichen Storys auch mit dem bisher Erreichten zusammenpassen. Bei der Konsistenzprüfung werden nicht nur die spezifizierten Anforderungen berücksichtigt, sondern auch das implizite Wissen in den Köpfen des PO und des Teams über die noch ungenau spezifizierten Anforderungen im Backlog sowie über bereits fertige Softwareteile.

■ Prüfbar

Jede Anforderung, die in einen Umsetzungsprint aufgenommen wird, muss so beschrieben sein, dass am fertigen Produktinkrement geprüft werden kann, ob sie auch richtig umgesetzt wurde. Die Kriterien sollten vor der jeweiligen Iteration definiert werden durch:

- Akzeptanzkriterien zu jeder Anforderung
- Definition of Done
- Idealerweise Testfälle oder konkrete Beispiele.

Zumindest die Akzeptanzkriterien und die Definition of Done müssen immer vorhanden sein.

Wenn eine nachhaltig qualitätsorientierte Entwicklung angestrebt wird, kann nur dringend empfohlen werden, die Prinzipien des testgetriebenen Vorgehens zu berücksichtigen und die Definition von passenden Testfällen als verbindliches Kriterium für die Umsetzungs freigabe einer Anforderung festzulegen.

Dabei geht es nicht darum, dass alle theoretisch denkbaren Abnahmekriterien und Testfälle vorab schon festgelegt werden. Es geht hier um diejenigen Testfälle, die das prüfen, was dem Kunden wichtig ist (z. B. ob die Funktion auch den korrekten Wert liefert) und das, was nach der Erfahrung guter Tester auch ohne expliziten Kundenwunsch sinnvollerweise getestet werden sollte (z. B. grenzwertige Eingaben, die erfahrungsgemäß oft zu Fehlern führen oder risikoreiche Eigenschaften oder Anwendungen der Anforderung, auch wenn sie nicht unmittelbar im Anwendungsfokus stehen).

Es gilt in der Softwareentwicklung der Grundsatz: Solange man für eine Anforderung nicht genau weiß, wie man sie später abnehmen und testen wird, ist sie nicht reif für die Umsetzung!

Beispiel für nicht prüfbare Anforderungsdefinition:

Ein Stakeholder hat als grobe Anforderung definiert: „Als Mitarbeiter möchte ich meine Zeiten erfassen können“. Er wird dann vom PO gefragt, wie er das testen und abnehmen möchte. Wenn als Antwort kommt: „Weiß ich nicht“, dann kann es sein, dass der Stakeholder noch keine Zeit hatte, sich detailliertere Gedanken zu machen. Der PO sollte hier so lange nachfragen bzw. Testfälle mit dem Stakeholder festlegen, bis diesem klar ist, was er eigentlich möchte.

Der andere Fall ist, dass der Stakeholder bewusst nicht genau spezifizieren möchte, ganz nach dem Motto „Ist mir egal, ich habe keine Zeit darüber nachzudenken, liefern Sie mir eine brauchbare Lösung!“. Dies ist ein kritischer Fall für den PO. Damit es nachträglich nicht zu Ärger kommt, muss der PO dem Stakeholder klar machen, dass er bei dieser ungenauen Anforderungsspezifikation am Ende dann auch die Abnahme nicht verweigern kann, wenn die Mitarbeiter nur irgendwie die Zeiten erfassen können, dies jedoch gar nicht seinen Vorstellungen entspricht.

■ Verfolgbar

Das Kriterium Verfolgbarkeit (Traceability) definiert die Verknüpfung von Anforderungen mit anderen Artefakten, wie z. B. Ziele, Architekturmodellen, Testfällen, Quellcode, Normen, externe Artefakte, usw.

Diese Verweise zu anderen Artefakten können nur mit einem professionellen Anforderungsmanagement- bzw. ALM-Werkzeug sinnvoll verwaltet werden.

Zusätzlich zu den Qualitätskriterien in IEEE 830-1998 und IREB können **noch weitere Eigenschaften als Kriterien** in der DoR definiert werden:

■ Relevante Stakeholder

Vor allem für den PO ist es wichtig zu wissen, welche Kunden und Stakeholder an einer Anforderung interessiert sind bzw. diese definiert haben.

Es sollte für jede Anforderung klar sein:

- Wer hat sich diese Anforderung gewünscht?
- Wer trifft die Letztentscheidung bei fachlichen Fragen?
- Wer nimmt die Story ab?

- Wer soll zusätzlich informiert werden bzw. ist betroffen?

- Wer ist die Zielgruppe für die Anforderung?

Da es möglich und auch oft üblich ist, dass Anforderungen stellvertretend von jemand anderem formuliert werden, sollte bei jeder Anforderung jedenfalls noch der sogenannte „Anforderungsprovider“ – also derjenige, der diese Anforderung explizit haben möchte –, angegeben werden, damit man bei Fragen weiß, an wen man sich wenden kann. Anstelle von Gruppenbezeichnungen (z. B. „Team A“, „die IT“, „Abteilung X“) sollten immer konkrete Namen (z. B. „Hans Mair“, „Franz Schmidt“) angegeben werden, damit der Stakeholder bzw. Anforderungsprovider „greifbar“ bleibt.

■ Identifizierbarkeit

Die Referenzierung einer Anforderung wird enorm erleichtert, wenn diese eine eindeutige ID hat. Im einfachsten Fall ist dies eine Nummer, es können aber auch beliebig strukturierte Kennzeichnungen verwendet werden.

Wichtig dabei ist nur, dass die Kennzeichnung innerhalb des Bereichs, in dem referenziert werden soll (z. B. Task-Board, gesamter Backlog, gesamte Produktentwicklung mit allen Artefakten oder auch gesamtes Unternehmen oder weltweit) eindeutig ist.

INVEST-Qualitätskriterien

Speziell für die Beurteilung der Qualität von User Storys hat sich ein Set von Qualitätskriterien herauskristallisiert, die unter dem Akronym INVEST bekannt geworden sind [Lef11].

■ Independent

Die Storys sind unabhängig voneinander. Jede Story kann als eigenständiges Paket geschätzt, geplant und umgesetzt werden.

■ Negotiable

Die Storys sind verhandelbar (und verhandelt). Sie stellen keinen fixen Vertrag dar. Stellt sich im Zuge der Umsetzung heraus, dass ein Detail anders einfacher oder benutzerfreundlicher umgesetzt werden kann, so wird dies mit dem PO abgestimmt und dann geändert.

■ Valuable

Jede Story muss einen erkennbaren Wert für den Kunden liefern.

■ Estimatable

Der Aufwand für die Story ist zumindest relativ schätzbar (und geschätzt). Dies ist notwendig, um die Story planen zu können.

■ Small

Jede Story muss innerhalb einer einzigen Iteration umsetzbar sein. Sie sollte daher, je nach Sprintlänge, eine Größe von maximal fünf Umsetzungstagen haben.

■ Testable

Für jede Story sind klare Kriterien festgelegt, wann sie fertig ist. Diese Kriterien sind in den Akzeptanzkriterien bei der Story selbst und in der „Definition of Done“ festgeschrieben.

Qualität der Kriterien für die DoR

Wenn nun Kriterien in die DoR aufgenommen werden, sollte darauf geachtet werden, dass diese „SMART“ sind:

■ Spezifisch

Die DoR-Kriterien sollen eindeutig und so präzise wie möglich definiert sein und nicht vage.

■ Messbar

Die Kriterien müssen messbar bzw. testbar sein.

■ Akzeptiert

Die DoR-Kriterien müssen angemessen sein und von den Betroffenen akzeptiert werden.

■ Realistisch

Die Einhaltung bzw. Erreichung muss möglich sein.

■ Testbar

Es muss klar sein, wie getestet und abgenommen wird.

Es kann auch sein, dass einzelne DoR-Elemente nicht für jede Anforderung angewendet werden. So ist es möglicherweise sinnvoll, je nach Risiko einer Anforderung unterschiedliche DoR-Kriterien zu haben.

Damit man in die DoR nicht zu wenig, aber auch nicht zu viele oder vielleicht sogar unsinnige Kriterien aufnimmt, sollte man sich vorab bei der Erstellung der DoR schon intensiv Gedanken machen. Unterstützt werden kann dies durch ein sogenanntes „Done Thinking Grid“



Abb. 2: Beispiel für „Done Thinking Grid“

[DTG]. Dies ist eine Technik, mit der auf einem Board Ideen zu sinnvollen Done-Kriterien gesammelt werden. Zur Strukturierung können evtl. schon Themenbereiche und Ebenen vorgegeben werden (z. B. als „Swimlanes“, vgl. **Abbildung 2**).

In der Literatur und im Internet finden sich oft unklare Begriffsdefinition bzw. Beispiele für DoR's, die dazu verleiten, DoD's zu oberflächlich oder ungeeignet zu erstellen. Diese oberflächlichen Beispiele führen in der Praxis zu dem Problem, dass unterschiedliche Stakeholder, Entwickler bzw. Teams oder der Kunde diese unterschiedlich interpretieren.

DoD's sollten nur Qualitätskriterien an den Requirements-Prozess und allgemeine Qualitätskriterien für die Requirements selbst enthalten. Sie sollen keine Testfallbeschreibungen oder inhaltlichen Abnahmekriterien enthalten oder diese ersetzen.

Ebenso sollten Kriterien, die spezifisch zu einer Anforderung gehören, in deren Akzeptanzkriterien aufgenommen werden und nur übergreifende Kriterien in die DoR. Für eine einfache und klare Anwendung der Definition of Done sollte vermieden werden, Testfälle und DoD-Kriterien zu vermischen!

Tests und inhaltliche Anforderungen jeglicher Art sollten in den jeweiligen Anforderungs- und Test-Artefakten definiert werden und nicht in einer DoR. Die DoD soll ausschließlich qualitative

Kriterien und keine inhaltlichen Anforderungskriterien an das zu erstellende System oder einzelne Artefakte enthalten!

Folgende Tipps helfen auch dabei, DoD's passend zu erstellen:

- Ein einziger DoR-Eintrag ist in den meisten Fällen zu wenig!
- Jedes Requirements-Artefakt durchläuft verschiedene Phasen/Ebenen. Es sollten daher verschiedene DoR-Checklisten für jede dieser Phasen/Ebenen erstellt werden.
- „Think twice“: Nach dem ersten DoR-Entwurf sollte man nach einer gewissen Zeit selbst nochmal darüber nachdenken und die DoR von anderen Personen reviewen lassen.
- Regelmäßige Anpassung der DoR-Checkliste.
- Berücksichtigen der Projektkritikalität und des Projektumfelds.

DoR's sollten so in den Prozess integriert sein, dass sie möglichst zwingend eingehalten werden müssen, z. B. durch ...

- Vier-Augen-Prinzip in verschiedenen Prozessphasen
- Die DoR nicht alleine abhaken und das Requirement alleine freigeben (Das Team prüft und gibt frei!).
- Der unternehmensweite Prozessverantwortliche sollte die Einhaltung der DoR-Anwendung übergreifend prüfen.

- Der projektbegleitende Quality-Manager sollte die Anwendung der DoR in einem spezifischen Projekt prüfen.
- Die DoR-Checklisten sollte – soweit dies möglich ist – in die verwendeten Tools eingebaut werden.
- Ein Workflow-Tool kann verwendet werden, um die DoR dadurch direkt in die Prozesssteuerung zu integrieren.

Eine Prüfung und (Selbst-)Kontrolle bei der DoR-Umsetzung ist wichtig, da die Anwendung viel Aufwand bedeuten kann und damit zu rechnen ist, dass die Durchführungsdisziplin bei fehlender Prozessabsicherung und -prüfung nachlässt. Damit DoR's auch gelebt werden, ist es wichtig, diese vor Projektbeginn unter Einbeziehung aller relevanten Beteiligten zu definieren. Im Laufe des Projekts sollte das Team die DoR's in den Retrospektiven immer wieder kritisch hinterfragen und sie bei Bedarf weiter anpassen.

Ziel ist es, kontinuierlich besser zu werden und die Qualitätsansprüche in den DoR's laufend anzuheben! ■

Literatur & Links

[Ber14] Bergmann, Unterauer, Requirements Engineering für die agile Softwareentwicklung, dPunkt, 2014.
 [IEEE] IEEE 830-1998, Recommended Practice for Software Requirements Specifications.
 [IREB] IREB, International Requirements Engineering Board <http://www.certified-re.de/>
 [Lef11] Leffingwell, Dean, Agile Software Requirements, Addison Wesley, 2011.
 [DTG] <http://www.scrumalliance.org/community/articles/2008/september/definition-of-done-a-reference>

Der Beitrag wurde erstmalig auf unserer Website veröffentlicht unter: <http://www.software-quality-lab.com/download/knowledge-letter/>