ISO 29119 und der agile Ansatz: Geht das zusammen?

Probleme und Lösungsansätze im agilen Umfeld

Dr. Richard Kölbl, Mixed Mode

Bis 2015 wurden fünf Teile der Norm zum Softwaretest ISO/IEC/IEEE 29119 veröffentlicht. Seit Beginn ihrer Erarbeitung gab es besonders aus den Reihen der Befürworter der agilen Entwicklung Widerstand dagegen: Die Norm sei zu schwerfällig, veraltet, behindere die agile Entwicklung und überhaupt brauche der Test keine Normierung. Aber stimmt das wirklich? Sind die ISO 29119 und agile Prinzipien auch beim näheren Hinsehen unverträglich?

Die Norm zum Softwaretest ISO/IEC/IEEE 29119 wurde seit 2007 erarbeitet und ist aktuell (2017) in fünf Teilen veröffentlicht (Abb.1, [1]). Es ist eine Normenreihe, die von Testentwurf und -durchführung bis zum einzelnen Testverfahren Standards zur professionellen Durchführung von Softwaretests bietet. Normen zielen auf Rationalisierung, Kostensenkung und Verständigung - allgemein gesprochen auf Vorteile der Industriegesellschaft. Sie können verstanden werden als eine systematisch konsolidierte und theoretisch begründete Sammlung von Erfahrungen und bewährten Praktiken. Sie werden zudem - anders als reine Industriestandards - nach einem international festgelegten Verfahren erarbeitet, verabschiedet und fortlaufend aktualisiert. Damit dürfte die Berechtigung für verbindliche Regelwerke dieser Art gezeigt sein. Motivation für die Erarbeitung dieser Normenreihe war u.a. dem bisherigen De-facto-Standard ISTQB, dessen Schwerpunkt auf der Testerausbildung liegt, ein verbindliches Regelwerk zur Seite zu stellen. Es sollte auch in sich besser abgestimmt sein als es bei den Standards und Normen der Fall ist, auf denen der ISTQB basiert.

Protest im Netz fordert Einstellung der Normentwicklung

Es stellt wohl einen relativ seltenen, wenn nicht einzigartigen Fall dar, dass schon in der Frühphase der Erarbeitung einer Norm sich im Netz Initiativen dagegen formierten und Petitionen starteten, die sie letztlich verhindern sollten. Es ist nicht schwierig, entsprechende Einträge (insbesondere von 2014) zu finden. Sie belegen, dass die Gegenbewegung in erster Linie aus dem agilen Bereich herrührt. Die wesentlichen Gegenargumente lauten:

1. Die Norm schreibe veraltete und zu schwerfällige Testverfahren fest.
2. Sie lege zuviel Gewicht auf Prozesse und Dokumentation.
3. Softwaretest sei in erster Linie ein kreativer Prozeß, der keine Normen brauche. Die Norm lasse dem explorativen Test zu wenig Freiraum.

Im Folgenden soll auf diese Argumente eingegangen und die These aufgestellt werden, dass sich agiles Vorgehen und die Anwendung einer Norm nicht unbedingt widersprechen müssen. Das soll zunächst auf Basis der fundamentalen Aussagen beider Parteien gemacht und anschließend anhand ausgewählter Regulatorien aus Teil 2 bzw. 4 der Norm im Detail betrachtet werden. Im gegebenen Rahmen kann natürlich nicht die ganze Norm auf ihre Konformität zu agilen Prinzipien untersucht werden; vielmehr sollen Beispiele die Herangehensweise zeigen, wie dies gelingen kann.

Grundlage der agilen Entwicklung
Zuvor ein Blick auf die agilen Prinzipien, die scheinbar mit der Anwendung einer Testnorm unverträglich sind. Die Grundprinzipien der agilen Softwareentwicklung sind im Agilen Manifest niedergelegt (www.agilemanifesto.org, Abb. 2). Das Agile Manifest versteht sich als ein Wertesystem und gibt keine unmittelbaren *technischen* Anleitungen, wie zu entwickeln und zu testen sei - was andererseits Aufgabe einer Norm ist. Auch das Ziel beider Artefakte dürfte sinngemäß übereinstimmen: Bessere Software zu schreiben und ihre Qualität sicherzustellen.

Gibt es prinzipielle Widersprüche zwischen Agil und Norm?
Der scheinbare Widerspruch kommt bei der Betrachtung von Zeile 1, 2 und 4 des Mittelteils in Sicht, in dem Dokumentation, Prozesse und Pläne niedriger priorisiert werden als die jeweils links stehenden Werte. Dagegen ist nichts einzuwenden. Der Schlußsatz des Agilen Manifests gesteht Dokumentation, Prozessen und Plänen ihren eigenen Wert denn auch zu. Anders ausgedrückt: Es ist hier nirgends die Aussage zu finden, dass Dokumentation, Prozesse und Pläne grundsätzlich verwerflich und abzuschaffen seien. In einer Schulung zum Agilen Testen, die der Autor absolviert hat, wurde die Faustregel "Just enough - soviel Dokumentation wie nötig" gegeben. Sie kann nur als vernünftig bezeichnet werden: Dokumentation um ihrer selbst willen ist auch in der traditionellen Entwicklung nicht zielführend. Aus dem Gesagten folgt: Auch in der agilen Softwareentwicklung haben Dokumentation sowie die Befolgung von Prozessen und Plänen ihren Platz, wenn auch deutlich anders gewichtet als im nichtagilen Tun.

Ein weiteres Argument findet sich im Scrum Guide (http://www.scrumguides.org/scrum-guide.html, 15.10.2017): "Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques." Mit anderen Worten: Sowohl das Agile Manifest, als Fundament aller agilen Ansätze, als auch einer der am weitesten verbreiteten Ansätze, nämlich Scrum, sind absichtlich so gehalten, dass sie Werte und Leitlinien auf einer *über* den technischen Details gehaltenen Ebene vermitteln und gleichzeitig Raum lassen für eben auf dieser technischen und prozessualen Ebene operierenden Anleitungen - sprich: auch für die Anwendung einer Norm zum Softwaretest. Der aufmerksame Leser wird bemerkt haben, dass auch der Scrum Guide den Begriff *processes* nicht vermeidet.

Wie sieht es aufseiten der Norm aus? Schließt die Norm das Arbeiten nach agilen Prinzipien aus? Definitiv nicht. Sowohl die Norm selbst als auch einführende Werke gehen ausdrücklich und positiv auf agile Ansätze ein. Die Norm bietet z.B. im Annex C "Testing in Different Life Cycle Models" einen Abschnitt zur agilen Entwicklung: C.2 "Agile Development and Testing". Er gliedert sich in drei Unterkapitel:
C.2.1: Agile Development Principles
C.2.2.: Test Management in Agile Development
C.2.3. Test sub-processes in Agile Development.

Das Vorwort zur Norm führt aus: "It is recognized that there are many different types of software [...] and methodologies. [...] Software methodologies include object-oriented, traditional, data-driven and agile. [...] This series of international standards can support testing in many different contexts." ([1]1:2013(E), Foreword, S. vi). Ein einführendes Werk zur ISO 29119 gibt ergänzend ein ausführliches Beispiel für die

Anwendung der Norm im agilen Kontext [2, S. 199]. Die Norm erkennt somit klar Softwareentwicklung im agilen Kontext an.

Aus dem bisher Gesagten lässt sich festhalten, dass auf dieser fundamentalen Ebene kein grundsätzlicher Widerspruch zwischen Manifest und Norm existieren sollte. Es ist in keinem der betrachteten Texte eine Aussage zu finden, die die Anwendung der jeweils anderen Seite kategorisch ausschließen würde.

Beispiele aus Teil 2 der ISO 29119

Wie sieht es nun im Detail aus? Hierzu sollen repräsentative Aussagen aus Teil 2 (Test processes) und 4 (Test techniques) betrachtet werden. Diese Auswahl begründet sich daraus, dass Teil 1 (Concepts und definitions) nicht normativ ist und Teil 3 (Test documentation) nur zur Festlegung des Umsetzungsumfangs normative Aussagen enthält. Grundsätzlich lässt die Norm sowohl die vollumfängliche Umsetzung (full conformance) als auch die maßgeschneiderte Umsetzung (tailored conformance) zu. Maßgeblich ist für beides die Beachtung der normativen Sätze, die durch Verwendung des Schlüsselbegriffs SHALL gekennzeichnet sind. Während bei der full conformance alle SHALL-Sätze umgesetzt werden müssen, ist es bei der tailored conformance nur eine Teilmenge. Allerdings müssen die nicht umgesetzten normativen Anforderungen bewertet und mit Begründung ausgeschlossen werden, damit auch in diesem Fall Konformität zur ISO 29119 festgestellt werden kann. Hierin liegt die Flexibilität der Norm, wodurch die Umsetzung im agilen Kontext gelingen kann.

Betrachten wir Beispiele aus Teil 2 der Norm, in dem der Testprozess geregelt wird. Beschränken wir uns auf Abschnitt 7.3 "Test Monitoring and Control Process". Tabelle 1 listet einige normative Sätze auf, deren Agil-Konformität am Beispiel von Scrum nachfolgend untersucht und begründet wird (Leser, denen einige Fachbegriffe des Scrum fremd sein mögen, werden gebeten, diese im Netz nachzuschlagen).

Clauses 7.3.4 - 7.3.4.2 b) fordern, dass von dem oder den Verantwortlichen eine Reihe von Aktivitäten und Aufgaben implementiert werden, die das Testen beobachten und steuern. Der Fortschritt gegenüber dem Testvorhaben soll über die übereingekommenen Testmaßnahmen beobachtet werden.

Agil: Auch im Agilen findet Testen nicht losgelöst von allen anderen Aktivitäten statt. Im Gegenteil liefert der allnächtliche Test (z.B. Continuous Integration) regelmäßige Rückmeldung über den Softwarestand. Rot-Grün-Dashboards oder Lavalampen sind im Agilen eingesetzte Rückmeldetechniken.

Clause 7.3.4.2 c) betont die Notwendigkeit, dass Abweichungen vom Testvorhaben wahrgenommen werden, insbesondere auch jegliche Blockaden.
Clause 7.3.4.2 d) setzt einen Grundzug der Norm um, nämlich Risiken stets im Auge zu behalten und neu entstehende zu identifizieren und festzuhalten. Dieses Risikobewusstsein zieht sich durch sehr viele normative Sätze der Norm.

Agil: Hier treffen sich nach Einschätzung des Autors das agile Prinzip und die Norm insofern, als gerade dieses ständige Beobachten des Standes z.B. in den täglichen Meetings (Standup, Daily etc.) und den Sprintplanungen bei Scrum ebenso ein zentrales Element darstellt. Dass Projekt- und Produktrisiken von

verantwortungsbewussten Product Ownern bzw. Scrum Mastern im Auge behalten werden, sollte eine Selbstverständlichkeit sein. Und schließlich kann der *Test Plan* aus Clause 7.3.4.2 durchaus in den Definitions of Done einer User Story enthalten sein. Es gibt keine normative Aussage in der ISO 29119, die dies verbieten würde.

Clause 7.3.4.3 a) bis d) zielen auf eine Fortschrittskontrolle beim Testen ab. Satz a) betont lapidar, dass das, was zu tun ist, auch getan werden möge (die Praxis zeigt in der Tat, dass manches Selbstverständliche durchaus auch einen Anstoß braucht, damit es durchgeführt wird). Satz b) bringt die jeweilige Firmenstruktur mit ins Testgeschehen, während Satz c) Maßnahmen zur Beherrschung von Abweichungen und Satz d) von Risiken fordern.

Agil: Auch agile Ansätze können sich nicht über übergeordnete Firmenleitlinien hinwegsetzen, während Standup meetings, Groomings, Sprint Retrospectives etc. für sich als Maßnahmen gesehen werden können, die die Sätze c) und d) erfüllen oder sie eigens als Tagesordnungspunkt vorsehen.

Clause 7.3.5 ist ein Beispiel für eine scheinbar für den agilen Ansatz anstößige Forderung. Nur: Zum einen kann sie mit Begründung in einer tailored conformance ausgelassen und dennoch für sich die Konformität des Testens mit der ISO 29119 in Anspruch genommen werden. Zum anderen kommt Teil 3 dem agilen Ansatz grundsätzlich sehr weit entgegen: Er enthält viele Sätze mit "may, must, should", aber bei der Beschreibung, wie solche Dokumente auszusehen haben, keine einzige normative Aussage. Im Gegenteil zeigen gerade die zahlreichen Annexe des Teil 3 an erster Stelle Beispiele für eine Umsetzung im agilen Test. Schließlich betont Teil 3 z.B. "In an agile project, the Test Status Report might not be a written document. For example, its contents could be discussed at iteration meetings and supplemented by information stored on activity boards and burn-down charts". (NOTE zu 6.3.1, S. 21) oder "There are many documentation styles and names, e.g. in agile, session sheets and charters with test ideas." (7.1, NOTE, S. 25). Auch hier nennt die Norm gerade den agilen Ansatz an erster Stelle, sodass für den Teil 3 der starke Eindruck entsteht, als sei er unter anderem gerade auf die Proteste aus dem agilen Bereich hin konzipiert worden.

Es sollte an diesen wenigen, repräsentativen Beispielen eine Herangehensweise demonstriert werden, wie die normativen Aussagen des Teils 2 auf ihre agil-Konformität hin untersucht werden können. Ohne dem Befund eigener Untersuchungen der Leser vorgreifen zu wollen: Ein Großteil der SHALL-Sätze des Teils 2 befindet sich wie in den Beispielen ganz in Übereinstimmung mit agilen Prinzipien oder eine Anpassung kann gefunden werden. Es dürften nur wenige normative Anforderungen als nicht übereinstimmend bewertet werden, aber auch das lässt die Norm zu.

Beispiel für ein normiertes Testverfahren

Betrachten wir nur ein Beispiel aus Teil 4 der ISO 29119, in dem aktuell 14 Testverfahren beschrieben sind (Abb. 3). Da alle Testverfahren nach demselben Muster abgefasst sind, genügt der Blick auf einen Repräsentanten (Abb. 4). Der Eintrag enthält jeweils einen Abschnitt, in dem normative Aussagen zur Ableitung der Testbedingung (quasi die Testidee, TD2), der Testüberdeckungselemente (TD3) sowie der Testfälle selbst (TD4) gemacht werden. Dazu gibt es Beispiele und

zusätzliche Hinweise. Die Anwendung der Norm kann hier sehr hilfreich sein, einen Äquivalenzklassentest nach state-of-the-art zu entwerfen. Die Praxiserfahrung zeigt, dass Anleitungen dieser Art nicht immer völlig überflüssig sind. Gerade Aspekte der Vollständigkeit eines Einzeltests wie Testüberdeckung oder die ausreichende Beachtung der Testbasis werden bei weniger erfahrenen Testern immer wieder vernachlässigt. Da das Agile Manifest "better ways of developing software by doing it and helping others do it" zum Ziel hat, sollte die Verbesserung und Professionalisierung des Testens dazu nicht im Widerspruch stehen. Damit dürfte gezeigt sein, dass die normativen Sätze des Teils 4 insgesamt in Übereinstimmung mit den agilen Prinzipien zu bringen sein sollten. Im Gegenteil wagt der Autor die These, dass von der Anwendung des Teils 4 die Abfassung von Definitions of Done und Test-Userstories profitieren kann: Im agilen Bereich gilt das Stichwort vom T-shaped Professional, der zwar sein Spezialwissen hat (der senkrechte Teil eines T), von vielen Dingen aber nur durchschnittliche Kenntnisse (der waagrechte Balken). Das ist ein realistischer Ansatz. Für das Testerwissen bedeutet das, dass jeder, der eine Testaufgabe zu lösen hat, vom in der Norm festgehaltenen Wissen profitieren kann.

Wie sieht es also aus mit Norm und Agil?

Jetzt können die eingangs erwähnten drei Argumente gegen die Norm vor dem Hintergrund des bisher Gesagten betrachtet werden. Für Argument 1, "veraltete Testmethoden" findet sich nach Ansicht des Autors kein stichhaltiges Argument: Testen ist im Kern der Versuch, ein logisches Konstrukt (z.B. Funktion, Modul) zu falsifizieren. Testverfahren (Abb. 3) sind nichts anderes als auf bestimmte Typen von logischen Konstrukten optimierte Falsifizierungsmethoden: Eine Funktion, deren Eingaben entweder im gültigen oder ungültigen Bereich liegen, kann nach wie vor z.B. geeignet mit der Grenzwertanalyse getestet werden. Das im agilen Bereich bevorzugte explorative Testen ist eine Test*strategie*, in der prinzipiell jedes Test*verfahren* anwendbar ist; somit gibt es hier keinen prinzipiellen Widerspruch. Exploratives Testen bringt sehr wohl Vorteile wie schnelle Reaktion auf Testergebnisse ("fast feedback"), aber wie die tägliche Praxis zeigt liegt seine Achillesferse bei der systematischen Testüberdeckung und -vollständigkeit. Das steckt auch im Argument "Schwerfälligkeit": Natürlich bringt systematisches, geplantes Testen einen entsprechenden Aufwand mit sich. Das ist aber weder der Norm an sich noch irgendeinem Entwicklungsansatz geschuldet: Wie hoch der notwendige Testaufwand ist, liegt in der Natur des Testobjekts und den Qualitätsanforderungen der Entwicklungsabteilung, ist also letztlich auch ein Managementthema. Zu Argument 2 ist zu sagen: Die Norm denkt in der Tat prozessual und hat planmäßiges Vorgehen im Blick, aber dies in äußerst weitem, flexiblem Rahmen. Beim Teil 3 der ISO 29119, der zur Form der Dokumentation keine einzige normative Aussage enthält, drängt sich der Eindruck auf, er sei geradezu auf die Einwände der agilen Seite hin zumindest überarbeitet worden. Von daher trifft die Behauptung, die Norm lege zuviel Wert auf Dokumentation, in der Praxis nicht wirklich zu. Zum Argument 3: Auch die Norm hat ihre Grenzen. Sie legt zwar z.B. fest, wie ein Äquivalenzklassentest zu entwerfen ist, aber für die *sachliche Richtigkeit*, dass z.B. die Grenzen dieser Klassen korrekt festgelegt werden, ist und bleibt der Tester verantwortlich. Insofern bleibt für Kreativität in der Praxis genug Raum. Zudem verbietet die Norm nirgends, dass hier nicht aufgeführte Testverfahren zusätzlich angewendet werden.

Die Praxis

Die praktische Erfahrung zeigt, dass sich tatsächlich am ehesten an Nebensächlichkeiten wie den Arbeitsbezeichnungen "Projekt", "Rolle" und - trotz obenstehender Argumente - an den Dokumentationsforderungen (z.B. bei Testspezifikationen) recht divergierende Ansichten entwickeln, die zu überbrücken einiges an Diplomatie erfordert. Letztlich sind es drei Lösungsansätze, über die man in der Praxis eine Vereinbarkeit von Norm und agilem Ansatz erleichtern kann:

1. Vermeidung von starren Formalismen (von Bezeichnungen wie "Prozess" bis hin zu Rollenbeschreibungen der Mitarbeiter),
2. Vermeidung von jeweils allzu extremen Auslegungen von Norm und agilem Ansatz und
3. besondere Betrachtung der Grenzflächen agil - nicht agil (z.B. Softwareplattform - Hardwareplattform), die je nach lokalen Gegebenheiten wieder ganz unterschiedliche Maßnahmen erfordern.

Resumée

Die vorstehenden Überlegungen zeigen nach Ansicht des Autors, dass zwischen den agilen Prinzipien und denen der ISO 29119 keine fundamentalen Unverträglichkeiten bestehen. Auch die detaillierte Betrachtung der normativen Anforderungen bringen weit weniger Diskrepanzen ans Licht, als der erste Anschein zeigen mag. Die ISO 29119 ist an vielen Stellen flexibel genug, auch durch die Möglichkeit der tailored conformance, agile Ansätze und normative Anforderungen zur Deckung zu bringen. In der praktischen Umsetzung sind es ein gesundes Maß an Pragmatismus und der Blick auf das gemeinsame Ziel, qualitativ hochwertige Software zu entwickeln, die die besten Ergebnisse versprechen.

Abbildungsverzeichnis:

IEEE 29119-1-2013: Software and systems engineering Software testing Part 1: Concepts and definitions.
IEEE 29119-2-2013: Software and systems engineering Software testing Part 2: Test processes.
IEEE 29119-3-2013: Software and systems engineering Software testing Part 3: Test documentation.
IEEE 29119-4-2015: ISO/IEC/IEEE International Standard for Software and systems engineering--Software testing--Part 4: Test techniques.
IEEE 29119-5-2015: ISO/IEC/IEEE 29119-5-2015: Software and systems engineering - Software testing - Part 5: Keyword-Driven testing.

Abb. 1: Bisher veröffentlichte Bestandteile der Norm zum Softwaretest ISO/IEC/IEEE 29119.

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Abb. 2: Text des Agilen Manifests (www.agilemanifesto.org, Zugriff 15.10.2017)

Spezifikationsbasiert	Strukturbasiert	Erfahrungsbasiert
Äquivalenzklassen	Anweisungstest	Fehlererwartung
Klassifikationsbaum	Zweigtest	
Grenzwertanalyse	Entscheidungstest	
Kombinatorik	Bedingungstest	
Entscheidungstabellen	Mehrfachbedingungstest	
Zustandsbasiert	Datenflusstest	
Szenariotest		

Abb. 3: Die im Teil 4 der ISO 29119 beschriebenen Testverfahren, eingeteilt in drei Kategorien.

5.2.1 Equivalence Partitioning (Äquivalenzklassenmethode)

5.2.1.1 Derive Test Conditions (TD2):

Equivalence partitioning (BS 7925-2:1998; Myers 1979) uses a model ... .

These equivalence partitions shall be derived from the test basis where each partition is chosen such that ...

EXAMPLE

NOTES

5.2.1.2 Derive Test Coverage Items (TD3):

Each equivalence partition shall be identified as ...

5.2.1.3. Derive Test Cases (TD4):

Test cases shall be derived to exercise the test coverage items.

The following steps shall be used:

a) bis e)

Abb. 4: Abgekürztes Beispiel eines Eintrags für ein Testverfahren in der ISO 29119. Dieses Beispiel soll nur das grundsätzliche Muster veranschaulichen, nach dem alle Einträge zu Testverfahren aufgebaut sind.

Clause 7.3.4	The person(s) responsible for test monitoring and control shall implement the following activities and tasks in accordance with applicable organization policies and procedures with respect to the Test Monitoring and Control Process.
Clause 7.3.4.2.	b) Progress against the Test Plan shall be monitored using the collected test measures.
	c) Divergence from planned testing activities shall be identified and any factors blocking progress recorded.
	d) New risks shall be identified and analysed to identify thise that require mitigation by testing and those that need to be communicated to other stakeholders.
Clause 7.3.4.3	a) Those actions necessary to implement the test plan shall be performed.
	b) Those actions necessary to implement control directives received from higher level management processes shall be performed.
	c) Those actions necessary to manage the divergence of actual testing from planned testing shall be identified.
	d) Means of treating newly-identified and changed risks shall be identified.
Clause 7.3.5	As a result of carrying out this process, the following information items shall be produced: a) Test status reports; b) Test plan updates; c) Control directives (e.g. changes to the testing, the test plan, test data, test environment and staffing); d) Project and product risk information.

Tab. 1: Beispiele für normative Sätze aus der ISO 29119, Abschnitt 7.3. Erläuterung siehe Text

Literaturverzeichnis

[1] ISO/IEC/IEEE 29119-1 to 5, International Organization for Standardization, Genf, Schweiz, 2013 - 2015

[2] Daigl, M., Glunz, R.: ISO 29119 - Die Softwaretest-Normen verstehen und anwenden. dpunkt, 2016

Autor

Dr. Richard Kölbl ist seit 2001 in der Softwareentwicklung tätig, mit Schwerpunkt Softwaretest und Qualitätssicherung. Er ist ISTQB-zertifizierter Tester (Basis, agil). Seit über 10 Jahren ist er für Fa. Mixed Mode tätig, wobei er zum einen für Firmen im Münchner Raum Testprojekte (konventionell und agil) durchführt, verschiedene Seminare zum professionellen Softwaretest abhält und Erfahrung bei der Einführung der ISO29119 in einem agil entwickelnden Unternehmen gesammelt hat.

Kontakt

Internet: www.mixed-mode.de
Email: info@mixed-mode.de