Matthias Daigl · Rolf Glunz

ISO 29119

Die Softwaretest-Normen verstehen und anwenden

Openbook
zum Teil 5 der Norm:
Keyword-Driven Testing

dpunkt.verlag

ISO 29119 – Die Softwaretest-Normen verstehen und anwenden

Matthias Daigl ist Product Owner bei der imbus AG. In rund 20 Jahren als Tester, QS-Berater, Coach und Trainer hat er Erfahrungen in den verschiedensten Softwareprojekten gesammelt. Er ist begeisterter Botschafter des Einsatzes internationaler Normen und vertritt das DIN in der Arbeitsgruppe 26 des ISO/IEC JTC1 SC7 bei der Erstellung der ISO/IEC 29119. Teil 5 der ISO 29119 verantwortet er als Editor.

Matthias Daigl

ISO 29119 – Die Softwaretest-Normen verstehen und anwenden

Openbook zum Teil 5 der Norm: Keyword-Driven Testing

dpunkt.verlag

Matthias Daigl
ISO29119@daigl.de

Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Matthias Daigl
Herstellung: Stefanie Weidner
Umschlaggestaltung: Helmut Kraus, *www.exclam.de*

Dieses Openbook basiert auf dem Buch »ISO 29119 –
Die Softwaretest-Normen verstehen und anwenden«,
dpunkt.verlag, 2016 (ISBN 978-3-86490-237-6).

Wieblinger Weg 17
69123 Heidelberg

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: hallo@dpunkt.de.

Vorwort

In den Jahren 2013 und 2015 erschienen die ersten vier Teile der Normenfamilie ISO 29119 ([6], [7], [8], [9]) zum Softwaretest. Zwar war es zu diesem Zeitpunkt keine Sensation, dass es Normen zu diesem Themenbereich gab. Allerdings waren die bestehenden Normen in die Jahre gekommen. Man hatte sich daran gewöhnt und sich mit manchen Eigenheiten abgefunden (etwa dem scheinbaren Zwang, die Testkonzepte in Master Test Plan und Level Test Plan zu gliedern), aber trotz mancher Überarbeitung waren die verfügbaren Normen (beispielsweise die IEEE 829 [5] und BS 7925-1 [1]) nicht mehr zeitgemäß – weder wurde agilen Vorgehensweisen Rechnung getragen, noch angemessen auf Prozesse eingegangen.

Mit der Normenreihe ISO/IEC/IEEE 29119[1] gibt es nun Normen, die diese Punkte aufgreifen. Anstatt aber ein weiteres Mal evolutionär kleinere Verbesserungen vorzunehmen, hat man, wie es die neue Nummer und die gemeinsame Veröffentlichung durch die drei Organisationen ISO, IEC und IEEE impliziert, einen echten Neustart gewagt. Die neue Struktur erlaubt es jetzt, auch bisher fehlende Themen angemessen zu betrachten.

Trotz großer Bemühungen der Editoren um die Lesbarkeit und Verständlichkeit der Texte der Norm fanden Rolf Glunz und ich, dass Hintergrundwissen und Hilfestellungen die Anwendung der ISO 29119 erleichtern würden, und so entschieden wir uns, das Buch »ISO 29119« [2] zu verfassen, das zum Zeitpunkt der Drucklegung die Teile 1 bis 4 umfasste. Teil 5 war noch in Arbeit.

In der Zwischenzeit ist mit der ISO 29119-5 zum Thema »Keyword-Driven Testing« [10] eine zusätzliche Norm im Testumfeld erschienen. Weitere Normen sind für diesen Bereich in Vorbe-

[1] »ISO/IEC/IEEE 29119« ist die korrekte Bezeichnung, da sie die drei Normungsorganisationen, die gemeinsam die Norm herausgeben, widerspiegelt. Der besseren Lesbarkeit halber verzichten wir im Folgenden auf die korrekte Schreibweise und verwenden meist das umgangssprachlich übliche »ISO 29119«.

reitung – beispielsweise zu den Themen »Statische Analyse« oder »Performance Testing«. Und nichts ist perfekt: An den ersten vier Teilen der ISO 29119 wurde Verbesserungspotenzial festgestellt, also sind für 2020 oder 2021 überarbeitete Fassungen geplant.

Es gibt also einiges zu tun, wenn man bezüglich der Testnormen auf dem Laufenden bleiben möchte.

Von besonderem Interesse scheint für die Welt der Tester die hier im Openbook erläuterte ISO 29119-5 zu sein. Jedenfalls wurde sowohl der dpunkt.verlag als auch die Autoren gefragt, wann denn eine Ergänzung unseres Buches zu dieser Norm erscheinen würde. Dieses Openbook soll diese Lücke nun schließen.

Zu diesem Zweck wird hier erläutert, welche Inhalte und Ziele die Norm hat, wie sie zu verstehen ist und wie man sie anwenden kann. Das Ziel ist dabei nicht, Ihnen als Leser die Anschaffung der Norm zu ersparen. Das Openbook kann nur einen ersten Eindruck vermitteln. Sie erhalten im Text auch einige Erläuterungen zum Keyword-Driven Testing selbst, die Sie nicht in der Norm finden – und so ergänzen sich beide.

Über das Thema Keyword-Driven Testing hinaus wird im zweiten Teil des Openbook zumindest in einer Momentaufnahme gezeigt, zu welchen weiteren Themen im Rahmen der Normenfamilie ISO 29119 oder in anderen Normen Softwaretester in den nächsten Jahren Unterstützung erwarten dürfen.

Liebe Leser, ganz besonders spannend wäre für mich Ihr Feedback – wie nutzen Sie Keyword-Driven Testing? Was erwarten Sie sich dabei?

Fürs Erste hoffe ich, den gröbsten Wissensdurst hinsichtlich der jüngsten Testnormen und auch ein wenig in Bezug auf Keyword-Driven Testing gestillt zu haben.

Aber bitte, liebe Leser, belassen Sie es nicht dabei: Sekundärliteratur – darum handelt es sich mit diesem Openbook – ist nützlich zum Verstehen eines Themas, aber lassen Sie die Primärliteratur, also die Normen, nicht links liegen! Denn wenn Sie damit arbeiten wollen, werden Sie ohne die Normen nicht auskommen.

Viel Spaß bei der Lektüre — ich wünsche Ihnen Inspiration für Ihre tägliche Arbeit und viel Erfolg bei der Umsetzung!

Matthias Daigl
Dezember 2019

Inhalt

Teil I

Keyword-Driven Testing in der ISO 29119

1 Keyword-Driven Testing — Grundprinzipien

Bevor darauf eingegangen wird, was konkret die Norm ISO 29119-5 zum Thema »Keyword-Driven Testing« – bekannt auch als »Schlüsselwortbasierter Test« – zu sagen hat, sollte vielleicht erst einmal geklärt werden, wofür dieser Begriff eigentlich steht.

Die zugrunde liegende Idee von Keyword-Driven Testing (KDT) wird gerne über eine Analogie mit Bausteinen erläutert.

So, wie ein Kind aus einer begrenzten Zahl geeigneter Bausteine – nur durch die Fantasie eingeschränkt – fast unendlich viele unterschiedliche Dinge – Häuser, Flugzeuge, Tiere – zusammensetzen kann, so kann ein Tester aus einer begrenzten Zahl an Keywords all die Testfälle zusammensetzen, die benötigt werden.

Genauso wie das Kind dazu einen Kasten voller Bausteine braucht, benötigt man als Tester hierzu eine Bibliothek geeigneter Keywords.

Das klingt vielleicht nach einer Mischung aus Spielerei und Over-Engineering, aber es ist alles andere als das. Worum es nämlich geht, ist eine Vereinheitlichung (Normierung) der Testfälle, und es wird eine für alle Beteiligten gemeinsame Sprache, maßgeschneidert für genau den jeweiligen Anwendungsfall, geschaffen.

Das erfordert, zugegeben, einen gewissen Aufwand, der zahlt sich erfahrungsgemäß jedoch aus.

Auf dem Weg dahin kann manches schiefgehen, es gibt einfach viele Möglichkeiten. Und bevor hier allzu viel in Irrwege investiert wird, ist es eine gute Idee, sich der Erfahrung derer zu bedienen, die diese Irrwege schon vorher ausprobiert haben. Solche konsolidierten Erfahrungen findet man nun erfreulicherweise auch zu Keyword-Driven Testing in einer Norm verdichtet.

2 Inhalt ISO 29119 – Teil 5: Keyword-Driven Testing

2.1 Überblick über die Norm ISO 29119-5

In der Norm ISO/IEC/IEEE 29119-5 wird Keyword-Driven Testing beschrieben.

Die Normenreihe ISO 29119 wurde ursprünglich basierend auf einem Konzept geschaffen, das mit den Teilen 1 bis 4 eine solide Grundlage für Testen in jeglichem Kontext bieten sollte – so die Hoffnung.

Dabei entschied man sich für folgende inhaltliche Aufteilung:

- Teil 1 enthält Begriffsdefinitionen und beschreibt Grundkonzepte.
- Teil 2 beschreibt die wichtigsten testspezifischen Prozesse.
- Teil 3 beschreibt die Dokumente, die aus den Prozessen im zweiten Teil entstehen.
- Teil 4 erläutert dynamische Testverfahren und wie diese mit den Prozessen aus dem zweiten Teil und den Dokumenten im dritten Teil zusammenspielen.

Erste Ergänzung der Normenreihe ISO 29119

Dieses ursprüngliche Vorhaben wurde durch die Veröffentlichung von Teil 4 zunächst abgeschlossen. Doch schon während der Entwicklung der ersten Teile wurde klar, dass es weitere Themen beim Softwaretesten gibt, die normungswürdig sind, und glücklicherweise besteht keine Vorgabe, dass eine Normenreihe mit vier Teilen enden muss. Es gab also Bedarf und die Möglichkeit, weitere Normen der Familie hinzuzufügen. Die erste Norm ist die ISO/IEC/IEEE 29119-5 zu Keyword-Driven Testing.

Keyword-Driven Testing in ein Ansatz zur Spezifikation von Tests, der sich unter anderem durch besondere Wartungsfreundlichkeit auszeichnet. Die Norm ISO 29119 beschreibt diesen Ansatz im Grunde für zwei Anwendergruppen:

Zielgruppen der Norm

- Personen, die Tests spezifizieren und gestalten, also Tester oder Testdesigner – sie erhalten Unterstützung beim Einsatz von Keyword-Driven Testing und bei der Gestaltung ihrer eigenen, ganz speziellen Variante davon.
- Hersteller von Frameworks, die den Anwendern bei Keyword-Driven Testing die Arbeit erleichtern und für den Einsatz von Keyword-Driven Testing Voraussetzung sind. Die Anwender können ihre Produkte an der Norm messen, die Produkte mit dem Ergebnis bewerben und zum Ausdruck bringen, bis zu welchem Grad Keyword-Driven Testing unterstützt wird, und die Kompatibilität mit anderen Produkten sicherstellen.

Selbst wenn Hersteller von Werkzeugen, die Teil eines Frameworks für Keyword-Driven Testing sein können, das Potenzial der Norm für sich nicht erkennen und sie nicht von sich aus zur Bewertung und Bewerbung der Möglichkeiten ihrer Produkte einsetzen, können die potenziellen Anwender immerhin selbst die Norm zur Bewertung der Werkzeuge und möglicher Alternativen nutzen.

Eine Übersicht über die Inhalte der ISO 29119-5 vermittelt Abbildung 2-1. Die stärkeren Umrandungen der etwas heller gehaltenen Kästen markieren in der Abbildung die normativen Teile.

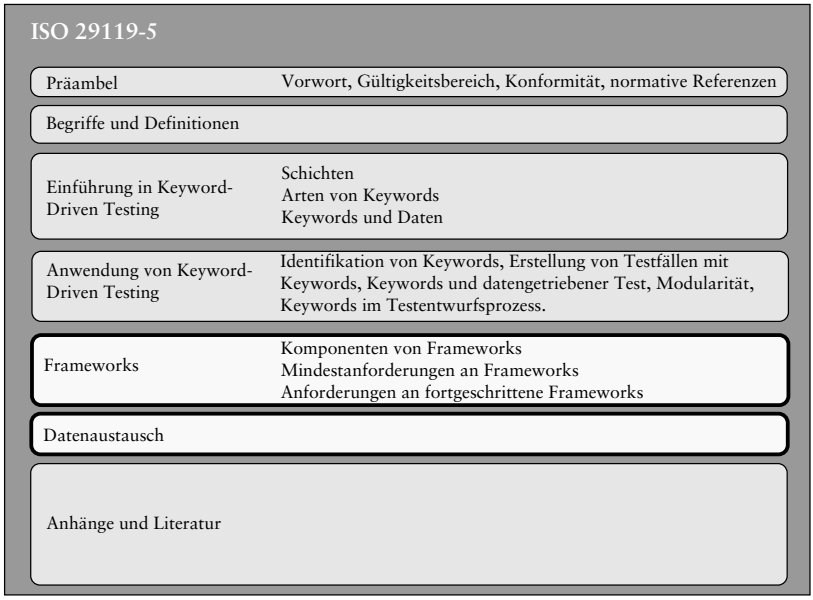


Abbildung 2-1 *Überblick über die Inhalte von ISO 29119-5*

Bei Teil 5 handelt es sich hierbei um die Abschnitte über das Framework und das Thema Datenaustausch. Damit sind in erster Linie die Teile normativ gehalten, die für die Hersteller von Werk-

zeugen relevant sind – sofern diese anstreben, Konformität ihrer Produkte mit der ISO 29119 für sich in Anspruch zu nehmen.

2.2 Normativ oder Informativ

Im Zusammenhang mit Normen tauchen immer wieder zwei Begriffe auf: »normativ« und »informativ«.

Abschnitte einer Norm können beides sein – so eigenartig es auch zunächst erscheinen mag, dass ein Teil einer Norm nicht normativ ist.

Aber damit wird die Unterscheidung getätigt, an welcher Stelle die Norm entweder

- einfach nur helfen will und Empfehlungen gibt, die nicht unbedingt einzuhalten sind,
- oder zwar auch helfen will, aber jetzt harte Forderungen stellt, die einzuhalten sind, wenn man konform mit der Norm sein möchte.

Normenautoren überlegen sich sehr sorgfältig, was das eine und was das andere ist.

»Shall« markiert Anforderungen

»Normativ« bedeutet also, dass der entsprechende Teil eine Anforderung darstellt, die – in unserem Kontext – für Keyword-Driven Testing unabdingbar ist. Die Anforderungen im Einzelnen sind immer mit dem Keyword »shall« gekennzeichnet.

Konformität

Konformität mit einer Norm erreicht man durch die vollständige Erfüllung aller darin enthaltenen (über »shall« markierten) Anforderungen.

2.3 Entstehung der ISO 29119-5

Die Normenreihe ISO/IEC/IEEE 29119 war zunächst als vierteilige Reihe gedacht. Mit den ersten vier Teilen zu Definitionen und Konzepten, Prozessen, Dokumenten und (dynamischen) Testtechniken wurden die Grundlagen des Softwaretestens betrachtet. Abgedeckt wurde zunächst das, was schon in den Vorgängernormen behandelt wurde, erweitert um das, was bei diesen offensichtlich fehlte.

Schon während der Entwicklung der ersten vier Teile kam die Idee auf, auch einen Teil zum Thema Keyword-Driven Testing zu erstellen. Das Vorhaben wurde durchaus kritisch hinterfragt.

Es wurde die Frage gestellt, ob Keyword-Driven Testing überhaupt schon reif für die Normung wäre – würde es denn in den, wie es immer heißt, »industries« (den regionalen oder weltweiten Märkten) schon eingesetzt? Die Antwort lautete ganz klar: Ja! Denn als diese Frage gestellt wurde, da war Keyword-Driven Testing schon zwanzig Jahre verfügbar. Und zwar in ganz unterschiedlichen Ausprägungen.

Und doch waren die Konzepte, jedenfalls in ihrer Breite, nicht allgemein bekannt oder verstanden – und hier sollte Abhilfe geschaffen werden: um ein Verständnis für das Thema zu erleichtern, Begriffe zu vereinheitlichen und um dazu beizutragen, dass durch ein einheitliches Verständnis auch eine bessere Werkzeugunterstützung verfügbar werden kann.

Auch Werkzeuge unterstützen Keyword-Driven Testing schon eine ganze Weile. Ein Vorreiter war die imbus TestBench, die schon sehr früh (seit Anfang des Jahrtausends) eine umfassende Unterstützung für KDT bot. Schon zuvor gab es von Logica ein Framework (»Action Words«). Und später erkannten weitere Werkzeughersteller wie seinerzeit Mercury (zwischenzeitlich unter der Flagge von HP und nun Microfocus) oder Ranorex das Potenzial und boten, alle auf ihre Weise und kaum miteinander vergleichbar, Unterstützung für Keyword-Driven Testing.

Hier liegt eine weitere Zielsetzung der Norm: Sie möchte uns Anwendern dabei helfen, objektiver zu beurteilen, in welcher Breite und Tiefe Werkzeuge Keyword-Driven Testing unterstützen.

Keyword-Driven Testing ist nicht das einzige Thema, für das während und nach der Ausarbeitung der ersten vier Teile die Notwendigkeit zur Normung erkannt wurde – weitere verfügbare und in Arbeit befindliche Normen werden in Kapitel 7 des Openbook angesprochen.

2.4 Bezug zu den Teilen 1-4 der Normenreihe ISO 29119

Der fünfte Teil der Normenreihe ISO/IEC/IEEE 29119 baut auf den Grundlagen der ersten vier Teile auf[1]:

- ISO 29119-1: Software and systems engineering – Software testing – Part 1: Concepts and definitions

[1] Hier wird nicht weiter auf die Inhalte der ersten vier Teile eingegangen. Wer mehr wissen möchte, sei auf [2] verwiesen.

- ISO 29119-2: Software and systems engineering – Software testing – Part 2: Test processes
- ISO 29119-3: Software and systems engineering – Software testing – Part 3: Test documentation
- ISO 29119-4: Software and systems engineering – Software testing – Part 4: Test techniques

Teil 1 liefert Definitionen für verwendete Begriffe und beschreibt allgemeine Grundlagen, die für das Verständnis von Testen hilfreich sind.

Von der Verwendung der Testtechniken aus Teil 4 wird zwar ausgegangen, aber hier gibt es tatsächlich wenig Berührungspunkte – denn letztlich ist es für die Anwendung von Keyword-Driven Testing nicht entscheidend, wie man die Testfälle gefunden hat.

Der stärkste Bezug besteht zu Teil 2 – Testprozesse und zu Teil 3 – Dokumente.

Bei den Prozessen aus Teil 2 spielt vor allem der »Test Design & Implementation Process« eine Rolle. Was im Bereich Keyword-Driven Testing beschrieben wird, ist eine spezielle Umsetzung der Aktivität TD4 (Derive Test Cases) und ein wenig TD5 (Assemble Test Sets) und TD6 (Derive Test Procedures).

Das Dokument aus Teil 3, das den größten Bezug zu Keyword-Driven Testing hat, ist naheliegenderweise die »Test Case Specification«. Betrachtet man das Thema etwas abstrakter, so können aber auch andere Dokumente tangiert sein – beispielsweise das Testkonzept (auf englisch: Test Plan), denn das ist ein guter Ort, um die Verwendung von Keyword-Driven Testing zu verankern.

2.5 Gründe für Keyword-Driven Testing

Was also sind die Vorteile von Keyword-Driven Testing?

An dieser Stelle wird oft und als Erstes Testautomatisierung genannt. Unbestritten hat Keyword-Driven Testing hier viel Potenzial. Um aber klarzumachen, dass die Vorteile nicht davon abhängen, ob jemand Testautomatisierung betreibt, beginnen wir ganz anders:

Traditionell werden Testfälle in natürlicher Sprache geschrieben. Das ist schließlich das Naheliegendste. Ein Beispiel hierfür könnte wie folgt aussehen:

Testfall 08/15 – Prosa, Variante 1

Um die Login-Funktion zu testen, gib in der Login-Maske als Benutzernamen »admin« und als Passwort »abcdef« ein. Nach erfolgreichem Login soll die Hauptmaske angezeigt werden.

Das kann zunächst einmal jeder verstehen und wohl auch am Testobjekt nachvollziehen. Also ist es wohl zweckmäßig. Kein Problem.

Natürlich ist es nur ein *Beispiel*. Es gefällt Ihnen nicht, und Sie würden es anders formulieren? Natürlich, warum nicht! Es gibt so viele Möglichkeiten ... Vielleicht würden Sie den Testfall eher so schreiben wie Variante 2?

Testfall 08/15 – Prosa, Variante 2

Teste das Login.
Nimm dafür als Zugangsdaten: »admin«, »abcdef«.
Das Login muss erfolgreich sein.
Check: Die Hauptmaske wird angezeigt.

Vielleicht ist auch das nicht Ihr Stil. Deutlich wird aber: Es gibt jede Menge Spielraum für die Art, einen Testfall zu formulieren. Jeder Testdesigner macht es wohl ein bisschen anders. Jeder hat seinen Weg, und manche machen es etwas besser als die anderen.

Und so wie jeder den Testfall ein bisschen anders beschreibt, so kann es auch leicht passieren, dass dieser Testfall – oder dessen Geschwister aus der Feder anderer Testdesigner – von jedem Leser anders verstanden wird. Dann wird er auch ein bisschen anders durchgeführt. Und schon haben wir ein Konsistenzproblem, die Wiederholbarkeit ist nicht sichergestellt.

Wie wird zum Beispiel das Login ausgeführt? Über Betätigung eines Buttons mit der Maus oder über die Enter-Taste? Vermutlich soll ja beides gehen, aber streng genommen sind das zwei unterschiedliche Testfälle. Wenn zwei Tester unterschiedliche Testfälle

unter dem Deckmäntelchen eines einzigen Testfalls durchführen, kann man die Ergebnisse eigentlich nicht vergleichen.

Natürliche Sprache ist oft ungenau.

Dazu kommt, dass der durchführende Tester die mehr oder weniger blumige und uneinheitliche Sprache der verschiedenen Autoren interpretieren muss: Wenn es heißt, ein »User« in einer Projektverwaltung solle »entfernt« werden – soll er dann nur aus dem Projekt entfernt werden oder doch aus dem System gelöscht werden?

Auch wenn dergleichen nicht unbedingt so unklar sein muss wie in diesem Beispiel, kostet die notwendige Interpretation immer Zeit und Kraft.

Vereinheitlichung ist also hilfreich. Hier setzt Keyword-Driven Testing an. Eine Umsetzung des Prosa-Testfalls von eben könnte so aussehen:

Testfall 08/15 – Keyword-Driven

- OpenLogin
- EnterUsername »admin«
- EnterPassword »abcdef«
- PressButton »LOGIN«
- CheckWindow »Hauptmaske«

Was ist daran jetzt so viel anders? Die Ungenauigkeit hinsichtlich des Buttons »LOGIN« hätte man auch anders eliminieren können. Stimmt schon.

Aber was hier grundsätzlich anders ist: Obwohl die Keywords alle beliebig nahe an natürlicher Sprache formuliert sind[2], sind sie doch formalisiert. Das Ziel ist, dass jeder Testschritt nur einmal in nur einem Keyword abgebildet ist. Damit hat jedes Keyword eine klare Bedeutung – die ich mir als Tester zwar vielleicht genauso erschließen muss wie beim Prosa-Testfall, aber eben nur einmal, dann ist das geklärt.

Und so sind Keyword-Driven Tests wesentlich präziser und gleichzeitig einfacher zu verstehen als Prosa-Testfälle.

[2]Gemeint ist, dass es einem freisteht, wie man die Keywords benennt. Im Beispiel werden englische Vokabeln benutzt. Selbstverständlich kann man hier jede beliebige Sprache nehmen!

Das sind nur zwei der Vorteile von Keyword-Driven Testing. Nachfolgend eine nicht unbedingt vollständige Aufstellung weiterer Vorteile:

- Höhere Präzision als Prosa – das wurde gerade besprochen.
- Bessere Verständlichkeit als Prosa – auch das.
- Bessere Verständlichkeit als programmierte Tests – da werden manche widersprechen. Entwickler finden (ihren) Code meist auch sehr gut lesbar. Aber nicht alle Tester haben einen Entwicklerhintergrund. Mit einem durchdachten Satz von Keywords hat man eine DSL (Domain Specific Language). Diese DSL verbindet, wenn sie gut gemacht ist, die Vorteile der Formalisierung mit einer Verständlichkeit für jeden.
- Trennung von Fachlichkeit und Technik – gerade mit hierarchischen Keywords (mehr dazu in Abschnitt 3.1) ist es sehr gut möglich, Arbeitsteiligkeit zu erreichen. Fachspezialisten können spezifizieren, Techniker können die Testfälle dann implementieren. KDT dient als Schnittstelle. In manchen Umgebungen ist das ein Killerargument!
- Wiederverwendbarkeit – jetzt kommen wir auf das Thema Testautomatisierung zurück. Das lässt sich mit Keyword-Driven Testing nämlich ganz famos umsetzen. Weil das Thema meiner Erfahrung nach für viele sehr interessant ist, wird es in Kapitel 4 etwas genauer erläutert.

Was hier noch klargestellt werden sollte: Es geht hier überhaupt nicht um die Form! Welche Form man wählt, ist ziemlich egal. Sie sollte nur zweckmäßig sein, und man sollte sich auf eine Form, also ein gemeinsam genutztes Format, verständigen.

Oben wurde der Testfall mit Keywords gezeigt, aber in einer ziemlich informellen Schreibweise. Manchmal ist stattdessen eine tabellarische Notation hilfreich, die den Testfall formaler und technischer darstellt wie in Tabelle 2-1:

Tabelle 2-1 *Testfall 08/15 — Keyword-Driven als Tabelle*

Keyword	Parameter 1	Parameter 2
`OpenLogin`		
`EnterUsername`	`admin`	
`EnterPassword`	`abcdef`	
`PressButton`	`LOGIN`	
`CheckWindow`	`Hauptmaske`	`Aktiv`

Hier ist eine Spalte für die Keywords vorhanden und zwei Spalten für die Parameter der Keywords. Die zweite Spalte mit dem Parameter »Aktiv« (es soll geprüft werden, ob das Fenster aktiv ist) für das Keyword »CheckWindow« ist eine Ergänzung zur Verdeutlichung, dass natürlich mehr als ein oder auch zwei Parameter sinnvoll und notwendig sein können. Für jeden weiteren Parameter würden wir in dieser Darstellung eine weitere Spalte benötigen.

2.6 Anwendung der ISO 29119-5

Für die Anwendung der Norm gibt es verschiedene Szenarien.

1. Szenario »Ausbildung«: Sie kann als Möglichkeit verstanden werden, das Thema Keyword-Driven Testing als Vorbereitung für den Einsatz von Keyword-Driven Testing kennenzulernen.
2. Szenario »Prozessverbesserung«: Sie kann dafür genutzt werden, den eigenen Einsatz von Keyword-Driven Testing zu hinterfragen. Das kann etwa im Hinblick darauf geschehen, ob die aktuelle Vorgehensweise schon alle Möglichkeiten ausschöpft, die Keyword-Driven Testing bietet – sofern sie für den eigenen Kontext hilfreich sind.
3. Szenario »Werkzeugauswahl«: Sie kann dafür genutzt werden, bei der Auswahl von Werkzeugen zum Keyword-Driven Testing die infrage kommenden Werkzeuge daraufhin abzuklopfen, welches davon mit seinen Features den geplanten Einsatz am besten unterstützt.
4. Szenario »Anregung des Wettbewerbs«: Und sie kann – vielleicht! – Werkzeughersteller in ihrem Wunsch nach Konformität[3] dahin bewegen, mit ihren Produkten Keyword-Driven Testing in größerer Tiefe und nicht nur pro forma zu unterstützen. Das wäre ein Riesenerfolg!

[3] Und wenn die Konformität nur aus Marketing-Gründen gesucht wird – auch recht!

3 Keywords

3.1 Schichten

Layer als Abstraktionsmodell

In vielen Einsatzfällen von Keyword-Driven Testing werden Keywords in mehreren Schichten als hierarchische Keywords gebildet. Das bedeutet, dass Keywords sich aus Keywords zusammensetzen. Dadurch entsteht ein Schichtenmodell. Die Norm spricht hier von »Layers«.

Man muss so nicht vorgehen – Keywords können grundsätzlich auch auf einer einzigen Abstraktionsebene angesiedelt sein. Viele Implementierungen für Keyword-Driven Testing begnügen sich damit.

Einfache Lösung

Damit kann man arbeiten, und wenn die Bedürfnisse der Anwender nicht darüber hinausgehen, ist es nicht nötig, eine komplexere Lösung zu suchen. Einfach kann einfach genug sein.

Allerdings wird das volle Potenzial des Keyword-Driven Testing nicht ausgeschöpft, wenn man sich auf eine Abstraktionsebene beschränkt:

- Arbeitsteiligkeit ist in höherem Maße möglich, wenn Fachexperten und Techniker auf unterschiedlichen Abstraktionsebenen tätig sein können.
- Die Implementierung einer Testautomatisierung wird robuster, wenn kleinere Einheiten, also kleinere Keywords, umgesetzt werden.
- Die Verständlichkeit der Keywords und der daraus entstehenden Testfälle wird aber besser, wenn die Keywords nicht zu feingranular sind. Diese beiden Punkte zusammenzubringen erfordert hierarchische Keywords, also Ebenen.
- Die Strukturierung der Keywords fällt leichter, wenn Abstraktionsebenen nutzbar sind.

Fortgeschrittenere Ansätze sehen daher vor, dass Keywords hierarchisch sein können.

Arbeitsteiligkeit

Was ist hier mit Arbeitsteiligkeit gemeint?

In einem Team können Menschen mit sehr unterschiedlicher Qualifikation zusammenarbeiten. Nicht jeder kann notwendigerweise alles oder zumindest nicht gleich gut.

Mit Arbeitsteiligkeit ist gemeint, dass beispielsweise jemand mit tiefen Kenntnissen der Fachlichkeit diese bei der Entwicklung der Testfälle anwenden kann, ohne in die Technik eintauchen zu müssen.

Ergänzend kann jemand anderes mit tiefgehenden technischen Kenntnissen und vielleicht ohne die Fachlichkeit wirklich zu verstehen, die Automatisierung umsetzen.

Diese beiden Welten zusammenzubringen ist die Herausforderung, bei der Keyword-Driven Testing hilft.

Diese Schichten zeichnen sich einerseits dadurch aus, dass sie unterschiedlich hoch in der Abstraktionebene des Schichtenmodells liegen – Keywords in den Schichten enthalten eben andere Keywords oder sind ihrerseits Bestandteil von Keywords.

Andererseits unterscheiden sie sich in der Bedeutungsebene: Keywords auf der einen Ebene sind eher technischer Natur, auf der anderen Ebene repräsentieren sie die Geschäftslogik.

Dieser Sachverhalt trifft auch dann zu, wenn keine Hierarchie von Keywords gebildet wird und wenn es nur eine einzige Ebene gibt.

Die Norm benennt diese Ebenen anhand ihrer Semantik folgendermaßen:

Spezifische Layer mit Semantik

- »Domain Layer«: Die Ebene der Geschäftslogik in der Sprache der Domäne. Das ist in der Regel die abstrakteste Schicht.
- »Intermediate Layer«: Eine Zwischenschicht, die der Verknüpfung anderer Schichten dient. Davon kann es keine, eine oder auch mehrere geben.
- »Test Interface Layer«: Diejenige Schicht, die den Zugriff auf die Testschnittstelle umsetzt. Die Norm bezeichnet sie als »Test Interface«, manche kennen hier die Begriffe »Point of Control/Point of Observation« (POC/POO). Das ist in der Regel die unterste Schicht.

3.2 Klassifikation von Keywords

Keywords lassen sich nach mehreren Gesichtspunkten klassifizieren. Zwei Beispiele:

- Falls es in einem Framework mehrere Schichten von Keywords gibt, folgt daraus unmittelbar, dass es mindestens zwei Typen von Keywords gibt – solche, die aus anderen zusammengesetzt sind, und solche, für die das nicht zutrifft.
- Eine weitere Klassifikation von Keywords sieht die Norm darin, dass Keywords manchmal (nur) zu Navigationszwecken dienen, andere aber das Ziel haben, das Testergebnis zu bestimmen.

Die erste Klassifikation zielt auf die Struktur ab, die zweite auf den Inhalt. Beides kann man sich praktisch – mithilfe eines geeigneten Frameworks – zunutze machen:

- Der Editor kann für zusammengesetzte oder atomare Keywords verschiedene Optionen anbieten.
- Bei der Ausführung kann definiert werden, dass die Auswirkung eines Fehlers beim Ausführen eines Keywords zur Navigation nicht den Testfall auf »FAIL« setzt, beim Ausführen eines prüfenden Keywords aber wohl.

3.3 Identifikation von Keywords

Beim Thema »Identifikation von Keywords« beschäftigt sich die Norm damit, wie Keywords – abhängig vom gewählten Schichtenmodell – strukturiert werden, und damit, wie sie benannt werden (Letzteres im Anhang der Norm).

Natürlich ist das am Ende jedem selbst überlassen, aber organisationsinterne Richtlinien können hier helfen, damit nicht Kraut und Rüben entstehen. Namen sind manchmal eben nicht Schall und Rauch.

Einheitliche Namen können dafür sorgen, dass es weniger Dubletten gibt. Leider stellen Dubletten innerhalb des Keyword-Repositorys in den meisten Fällen zu irgendeinem Zeitpunkt ein Problem dar. Einigt man sich aber darauf (das Folgende ist ein Beispiel!), dass ein Keyword immer aus einem Verb, gefolgt von einem Substantiv besteht, und zwar in dieser Reihenfolge, dann

wird nicht so leicht übersehen, dass zwei Keywords unterschiedlich heißen, aber dasselbe bedeuten.

Beispiel für Dubletten

Die Keywords »create_user« und »user_creation« bedeuten dasselbe und sind daher Dubletten.

Sieht man sie, wie hier, unmittelbar nebeneinander, ist das offensichtlich. In einer größeren alphabetisch sortierten Liste aber nicht unbedingt!

Des Weiteren geht es in diesem Abschnitt darum, welche Information zu einem Keyword hinterlegt werden sollte. Das ist ein schönes Beispiel dafür, wie eine Norm unterschiedlich nutzbar ist:

- Als Anwender kann ich den Vorschlag der Norm lesen, prüfen, für gut oder schlecht befinden und davon umsetzen, was mir sinnvoll erscheint.
- Als Anwender auf der Suche nach einem Werkzeug kann ich den Vorschlag der Norm, ergänzt um meine eigenen Wünsche, als Kriterium zur Auswahl eines Werkzeugs (bietet es genug?) heranziehen.
- Als Hersteller eines Werkzeugs kann ich das Werkzeug so gestalten, dass es zumindest die Forderungen der Norm erfüllt – ergänzt um Weiteres, falls es mir sinnvoll erscheint. Dadurch kann ich mich vom Wettbewerb abheben, gleichzeitig aber mit Normkonformität punkten.

3.4 Keywords und datengetriebener Test

Datengetriebene Tests gibt es auch ohne Keywords, und Keyword-Driven Testing muss nicht notwendigerweise datengetrieben sein.

Trotzdem: Seine Stärken entfaltet Keyword-Driven Testing erst dann, wenn es in Verbund mit datengetriebenem Testen (Data-Driven Testing, DDT) verwendet wird. Aus diesem Grund betrachtet auch die Norm DDT als Bestandteil des Keyword-Driven Testing.

Das bedeutet nicht, dass jeder, der Keyword-Driven Testing betreiben will, notwendigerweise auch datengetrieben arbeiten muss. Aber es ist davon auszugehen, dass die meisten das wol-

len, und daher wird auch von den Frameworks erwartet, dass sie das unterstützen.

In Tabelle 3-1 zeigen wir ein weiteres Mal das Beispiel eines abstrakten Testfalls für Login, nun aber um den Faktor »datengetrieben« erweitert:

Keyword	Parameter 1	Parameter 2
OpenLogin		
EnterUsername	$USER$	
EnterPassword	PWD	
PressButton	LOGIN	
CheckWindow	Hauptmaske	Aktiv

Tabelle 3-1 *Testfall 08/15 — Keyword-Driven & DDT*

Testfall-Name	$USER$	PWD
valid1	admin	abcdef
valid2	peter	secret
valid3	Maria	%a$zlpx_37

Jetzt sind schon zwei Tabellen zur Beschreibung des Testfalls notwendig. Aber hoppla, es ist nicht mehr ein Testfall – es sind drei verschiedene, denn mit jedem Eintrag in der Spalte »Testfall-Name« (»valid1«, »valid2« und »valid3«) erhält man, kombiniert mit den Schritten der ersten Tabelle, einen eigenen konkreten Testfall.

Und noch etwas kann man hier zeigen: Wenn wir eine vierte Zeile »invalid1« anlegen, mit ungültigen Login-Daten, dann können wir mit demselben Ablauf Positiv- und Negativtestfälle abdecken. Aber sicherlich muss dann der Ablauf flexibilisiert werden, um beide Sollreaktionen – Akzeptieren und Zurückweisen des Login-Vorgangs – prüfen zu können.

Es wird also deutlich, dass wir so eine Menge Flexibilität und Effizienz gewinnen, aber dies geschieht auf Kosten einer höheren Komplexität.

Die Vorstellung, langfristig manuell eine Vielzahl solcher Tabellen zu verwalten, die auch noch verschiedene Semantik haben (Testfälle, Daten), kann beängstigend sein. Die Schlussfolgerung ist: Eine Unterstützung durch Werkzeuge wird wohl notwendig sein.

4 Testautomatisierung mit Keywords

Testautomatisierung kennt, seitdem es sie gibt, eine große Herausforderung: die Wartbarkeit der Automatisierung. Schließlich sollen automatisierte Tests oft durchgeführt werden, damit sich der Aufwand, den man in die Automatisierung investiert hat, überhaupt lohnt.

Das lässt sich mit einer Modellrechnung leicht nachvollziehen. Wir gehen von einigen festen Werten für die Kosten der manuellen und automatisierten Durchführung eines einzelnen Testfalls aus (Tabelle 4-1).

Parameter	manuell	automatisiert 10% Wartung	automatisiert 30% Wartung
Einführung	0	10000	10000
Aut. 1 Testfall	0	500	500
1 Durchführung	200	10	10
Anzahl Testfälle	100	100	100
Wartung	0	50	150

Tabelle 4-1 *Werte für Modellrechnung*

Verglichen wird hier ein manueller Test mit zwei Varianten der Testautomatisierung, die sich voneinander nur im Wartungsaufwand unterscheiden – einmal gehen wir von 10 % durchschnittlichem Wartungsaufwand im Verhältnis zur Neuerstellung der Automatisierung aus, einmal von 30 %.[1]

[1] Die restlichen Zahlen sind nicht unrealistisch, spielen aber auch gar keine so große Rolle – probieren Sie es aus, Änderungen hier bewirken erstaunlich wenig.

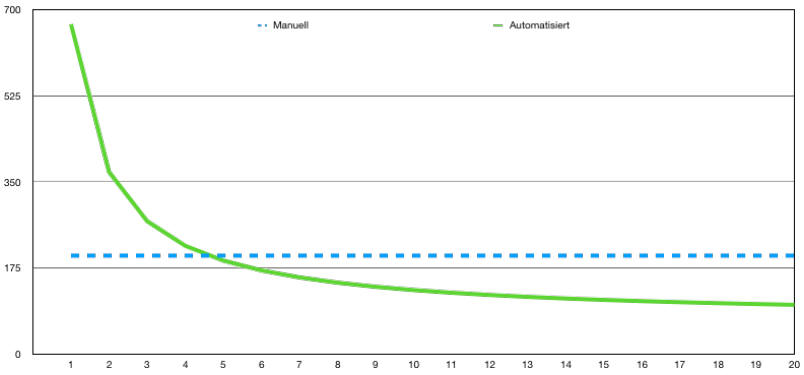


Abbildung 4-1 *Modellrechnung 10 % Wartung*

Füttern wir diese Werte in eine Tabellenkalkulation und berechnen die Kosten[2] pro Durchführung, lassen sich Graphen wie in Abbildung 4-1 und Abbildung 4-2 erzeugen. Die gestrichelte Linie steht für manuellen Test, die durchgezogene Linie für automatisierten Test. Man sieht, dass im »guten« Fall, bei nur 10 % Wartungsaufwand, die Testautomatisierung bei rund fünf Durchführungen wirtschaftlich rentabel wird. Bei 30 % Wartungsaufwand – zugegeben hoch, aber es gibt schlimmere Fälle! – benötigt man dagegen 20 Wiederholungen.

Diese Modellrechnung sieht vielleicht zu simpel aus, um die Realität abzubilden — aber tatsächlich deckt sie sich mit den Praxiserfahrungen!

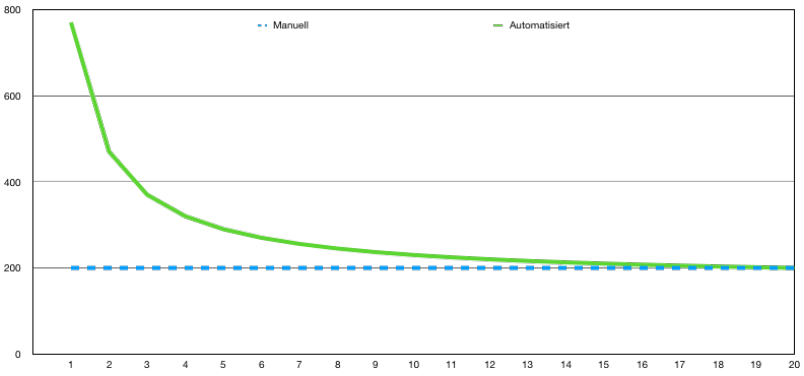


Abbildung 4-2 *Modellrechnung 30 % Wartung*

Damit sollte plausibel sein, warum geringe Wartungskosten so essenziell sind für eine wirtschaftlich rentable Testautomatisierung.

[2] Die Währungseinheit fehlt in der Tabelle und in den Graphen? Das stimmt. Setzen Sie € oder $ ein, es geht hier eher um das Verhältnis der Zahlen zueinander.

Tatsächlich ist es *der* Schlüsselfaktor.

Schlüsselfaktor

Was kann nun KDT dazu beitragen? Was KDT hier leistet, ist insbesondere eine Trennung der Testlogik von den Testskripten:

Wartungskosten

1. Die Testlogik ist in einem Keyword-Testskript aus Keywords zusammengesetzt abgebildet. Zur Änderung der Testlogik und sogar zum Erstellen neuer Testfälle muss die technische Implementierung nicht berührt werden.
2. Die Automatisierung ist in viele kleine Skriptbausteine aufgeteilt, die sich inhaltlich möglichst nicht überlappen. Eine technische Änderung der Testinterfaces (z.B. das User Interface, UI) erfordert daher im besten Fall nur die Anpassung eines einzigen kleinen Automatisierungsskripts. Die Testlogik bleibt unbeeinflusst.

Bei monolithischen Automatisierungsskripten pro Testfall muss bei jeder Änderung, sei es in der Testlogik oder in der Technik, immer sowohl Logik als auch Technik angefasst werden, und das oft an vielen Stellen. Das bedeutet hohen Wartungsaufwand.

Keyword-Driven Testing unterstützt durch systematische Modularisierung das Erreichen eines minimalen Wartungsaufwandes. Mancherorts reicht das bereits als alleiniger Grund, KDT zu verwenden, auch wenn es viele weitere gibt (wer sich fragt, welche: bitte zurück nach Abschnitt 2.5, gehen Sie nicht über LOS).

5 Frameworks und ihre Komponenten

Ein zentraler Bestandteil der Norm ISO 29119-5 ist die Definition eines Frameworks für Keyword-Driven Testing.

Unter einem Framework verstehen wir hier eine Umgebung, in der Keyword-Driven Testing abgewickelt werden kann. Dazu gehört auch Organisatorisches, der Fokus liegt aber auf technischen Hilfsmitteln (Werkzeugen), die zusammenspielen müssen.

Dieses Framework ist so generisch wie möglich gehalten, um möglichst keine vorhandene Werkzeugkette von einer Betrachtung auszuschließen.

Zu diesem Zweck wurden Komponenten definiert, die für Keyword-Driven Testing – je nachdem, wie man es betreiben möchte – notwendig sind.

Diese Komponenten sind:

- »Keyword-Driven Editor« – es muss einen Weg geben, Keywords zu Testfällen zusammenzufügen
- »Decomposer« – um zusammengesetzte Keywords in ihre Schritte zu zerlegen
- »Data sequencer« – um die Daten aus dem DDT mit den Keywords zu verknüpfen
- »Manual test assistant« – zur Unterstützung der manuellen Ausführung von Testfällen
- »Tool bridge« – um die Testfälle an ein konkretes Ausführungswerkzeug zu übermitteln
- »Test execution environment and execution engine« – zur Interpretation der Testfälle und Ausführung im Ausführungswerkzeug

Das wären die aktiven Komponenten, dazu gehören noch Repositorys:

- »Keyword library« – eine Ablage für die Keywords
- »Data« – eine Ablage für die Testdaten

- »Script repository« – eine Ablage für die Implementierung der Keywords für ein konkretes Ausführungswerkzeug

Zu all diesen Komponenten werden Anforderungen definiert, die man als Anwender des Keyword-Driven Testing stellen kann. Diese Anforderungen sind in zwei Gruppen aufgeteilt:

1. »Basic« — sehr grundlegende Anforderungen, braucht eigentlich jedes Framework
2. »Advanced« — Anforderungen für anspruchsvollere Umsetzungen

Wichtig für das Verständnis: Die zugrunde liegende Idee ist, dass diese Anforderungen – sofern eine Abdeckung gewünscht ist – vom Framework insgesamt erfüllt werden sollten. *Nicht* gefordert ist, dass ein einzelnes Werkzeug auf jedem Gebiet alle Anforderungen abdeckt.

Ein konkretes Tool kann also eine Komponente, mehrere Komponenten, Teile einer oder mehrerer Komponenten oder — nicht sehr realistisch, aber ein hübscher Gedanke — sämtliche Komponenten abdecken.

6 Ausblick

Ist Keyword-Driven Testing mit der vorliegenden Norm nun erschöpfend behandelt?

Die Frage ist nicht unberechtigt, schließlich gibt es die Konzepte nun schon eine ganze Weile, insofern kann man schon auf die Idee kommen, dass sie ausreichend von allen Seiten betrachtet wurden.

Trotzdem fehlt bislang eine angemessene Behandlung der vielen Facetten in der Literatur – das kann weder eine Norm noch dieses Openbook leisten. Aus diesem Grund ist vom Autor ein weiteres Buch in Vorbereitung, das sich dem Thema Keyword-Driven Testing umfassender widmen wird, dabei aber deutlich über den Fokus einer Norm hinausgeht und insbesondere Hilfestellung für die Praxis bieten will [3].

Auf welche weiteren Ideen oder Erweiterungen jemand für diesen Bereich Softwaretest in der Zukunft kommen wird, ist schwer zu sagen. Aus heutiger Perspektive kann man aber feststellen, dass vielerorts bereits das, was heute verfügbar und bekannt ist, noch gar nicht ausgeschöpft wird.

Das trifft auf Organisationen zu, leider auch auf Werkzeuge. Auf der Packung ist Keyword-Driven Testing vielleicht aufgedruckt, aber tatsächlich wird nur an der Oberfläche dessen gekratzt, was möglich ist.

Hier ist jedenfalls zu hoffen, dass das Ende der Fahnenstange noch lange nicht erreicht ist. Als Verbraucher, also als Nutzer von Testwerkzeugen, können wir vielleicht sogar durch Nutzung der Norm – indem wir sie als Maßstab für die Werkzeuge nehmen, die Werkzeuge damit vergleichen und die Hersteller mit dem Ergebnis konfrontieren – dafür sorgen, dass sich hier noch etwas tut!

Teil II

Entwicklungen in der Normung

7 Weitere verfügbare und kommende Normen zum Softwaretest

Das war's noch nicht.

Mit der ISO 29119-5 ist die erste Erweiterung zu den Grundlagennormen ISO 29119 Teil 1-4 erschienen, aber der Bedarf nach weiteren Normen, die in konkreten Anwendungsfällen einheitliches Verständnis schaffen und Leitplanken für die Praxis bieten, ist ins Bewusstsein gerückt, und so wurden die folgenden Normungsvorhaben durch die Normungsstellen von ISO und IEC in Angriff genommen und zum Teil (die ersten beiden in der Liste) bereits umgesetzt:

- ISO/IEC 20246:2017 Software and systems engineering – Work product reviews (International Standard – verfügbar): In dieser Norm wird das Thema Reviews betrachtet. Sie ist natürlich nicht nur auf das Thema Testen beschränkt, denn auch wenn Reviews aus Sicht der Tester als statische Testtechnik verstanden werden, sind Reviews ja in allen Disziplinen des Software Engineering wichtig.

  Es handelt sich hier nicht, wie man denken könnte, um einen Nachfolger der IEEE 1028-2008 »IEEE Standard for Software Reviews and Audits« [4], sondern um eine eigenständige Norm, die sich auch auf Konzepte der IEEE 1028 bezieht.

  Im Gegensatz zu dieser Norm wird hier allerdings ein größerer Wert auf Reviewprozesse gelegt, und es wird eine Vielzahl von Reviewpraktiken beschrieben. Insgesamt erscheint die ISO 20246 – aus meiner Sicht – deutlich praxisorientierter als die ältere ISO 1028. Auch wenn diese bestehen bleibt, bietet sie eigentlich keine Vorzüge gegenüber der neueren Norm.
- ISO/IEC 33063:2015 Information technology – Process assessment – Process assessment model for software testing (International Standard – verfügbar): Diese Norm ist ein Handwerks-

zeug für Auditoren, die Konformität mit den Prozessen der ISO 29119 prüfen müssen. Alle, die das nicht zu tun brauchen, können diese Norm getrost links liegen lassen.

- Technical Report zur Anwendung der ISO 29119 im agilen Umfeld: Dieser TR (Technical Report) soll es Anwendern in agilen Umgebungen erleichtern, zu verstehen, wie einfach sie agil und doch normkonform arbeiten können. In diesem TR werden die häufigsten agilen Praktiken in Bezug auf die Normenreihe ISO 29119 Teil 1-4 betrachtet.
- Technical Report zum Zusammenhang der ISO 29119 mit ISO 26262-6 und Automotive SPICE: Im Automotive-Bereich wird man mit vielen Normen und Standards konfrontiert, von denen viele obendrein verpflichtend sind. Wenn eine vergleichsweise neue Norm wie die ISO 29119 (Teile 1 bis 4) dazu kommt, stellt sich die Frage, wie sie einzuordnen ist, was sie neu regelt und inwiefern es vielleicht auch Konflikte gibt.

  Tatsächlich bezieht sich Automotive SPICE bereits auf die ISO 29119. Die Norm ISO 26262-6 hingegen lässt Details offen, die durch die Normenreihe ISO 29119 geklärt werden.

  Diese Zusammenhänge sind das Thema des geplanten TR.
- Technical Report zum Thema Games Testing (sic!): Auch wenn das Thema »Games Testing« für viele Softwaretester im deutschsprachigen Raum exotisch erscheinen mag, spielt es doch in anderen Ländern, in denen die Computerspiele-Industrie stark ausgeprägt ist, eine deutlich größere Rolle, beispielsweise in Japan oder Brasilien. Testen von Spielen bietet spezielle Herausforderungen aufgrund von für Spiele typischen Aspekten wie zum Beispiel die Gefühle und Emotionen der Anwender. Darauf und wie diese Herausforderungen sich auf die Lösungsvorschläge der ISO 29119 abbilden lassen, möchte dieser TR eingehen.
- Technical Report zum Thema Model Based Testing: Der modellbasierte Test (MBT) wird heute in sehr unterschiedlichem Umfang eingesetzt. In manchen Bereichen wird kein Vorteil in seinem Einsatz gesehen, in anderen Bereichen (beispielsweise bei sicherheitskritischen Embedded-Anwendungen) ist MBT hingegen unverzichtbar geworden. In Zukunft wird MBT mit Sicherheit ein fester Bestandteil im Reigen der Testpraktiken sein.

  Der hier entstehende TR bringt MBT in Zusammenhang mit den ersten fünf Teilen der ISO 29119. Interessante Berührungspunkte finden sich gerade auch im Hinblick auf Teil 5, in dem es ja um »Keyword-Driven Testing« geht – denn

die Verwendung von Keywords erleichtert den Sprung von aus Modellen generierten Testfällen zu deren automatisierter Durchführung ganz erheblich.

- International Standard zum Thema Performanztest: Der Performanztest spielt im Licht von Cloud und zunehmender Bedeutung von Webanwendungen eine eher größer werdende Rolle im Softwaretest, auch wenn er ohnehin schon immer wichtig war – und wichtiger vielleicht, als der ihm bisher zugestandene Stellenwert erahnen ließ.

  Zu diesem Thema soll nun ein neuer Standard Unterstützung bieten. Er wird sich voraussichtlich mit den speziellen Anforderungen an den Performanztest beschäftigen, eine Abbildung auf die Prozesse der ISO 29119-2 vornehmen, verschiedene Arten des Performanztests beschreiben und Metriken dazu definieren.
- International Standard zum Thema statische Analyse: Zum Thema statische Analyse gibt es bislang weder einen ISO- noch einen IEEE-Standard. Das soll sich nun ändern. Damit wird auch eine Lücke geschlossen, da »Statischer Test« nicht in den ersten vier Teilen der ISO 29119 behandelt wird. Der statische Test soll künftig in Bezug auf statische Analyse durch den hier neu entstehenden Standard abgedeckt werden, und in Bezug auf Reviews durch die ISO 20246.
- International Standard zum Thema Incident Management: Dieser Standard soll sich mit dem Thema »Incident Management« befassen, einem sehr grundlegenden Aspekt des Softwaretestens. Es geht hier um allgemeine Prozesse zum Umgang mit »Incidents« – darunter werden Dinge verstanden, die unerwartet auftreten und möglicherweise eine weitere Betrachtung erfordern. Paradebeispiel ist dabei natürlich der Fehler oder »Bug«.

  Die aktuellen Themen »AI« (Artificial Intelligence, künstliche Intelligenz) und »Autonomous Systems« sollen hierbei Berücksichtigung finden. Eine Frage könnte lauten: Sollte man von autonomen Systemen als Zulassungsvoraussetzung erwarten, dass sie selbsttätig unerwartete Ereignisse oder Betriebszustände melden, vielleicht auch an eine Aufsichtsbehörde? Dann wäre ein herstellerübergreifend einheitliches, normiertes Meldungsformat sinnvoll.

Ich höre schon die Stimmen: Braucht es das denn wirklich? So viele Normen zu diesem Thema? Ernsthaft, eine Norm zum Thema Computerspiele?

Jeder wird dazu eine eigene Meinung haben. Aber wie auch immer Sie dazu stehen — Normungsvorhaben werden nur gestartet, wenn sie genügend Unterstützung haben und es ausreichend wenig Gegenwehr gibt. Somit ist jede existierende Norm ein Beleg dafür, dass ihre Notwendigkeit von vielen Menschen oder Unternehmen gesehen wird.

Einige hier genannte Normen sind »Technical Reports«. Das bedeutet, dass sie keine normativen Bestandteile (in der Begriffswelt der Normen: »Requirements«) enthalten. Sie sind ausdrücklich als Hilfestellung gedacht.

Bei den Normen, die als »International Standards« mit normativen Bestandteilen aufwarten, kann eine Konformität beispielsweise durch Auftraggeber eingefordert werden. Wenn das geschieht, dann kann es zwar lästig sein, sich mit diesen Normen aus diesem Grund beschäftigen zu müssen. Meistens ist aber eine Auseinandersetzung mit etwas sehr fruchtbar, das wie eine Norm auf den Erfahrungen vieler beruht!

Anhang

A Referenzen und weiterführende Literatur

[1] BSI. *Software testing. Vocabulary*. BS 7925-1:1998, UK National Standards Body, August 1998.

[2] Matthias Daigl und Rolf Glunz. *ISO 29119 – Die Softwaretest-Normen verstehen und anwenden*. dpunkt.verlag, 2016.

[3] Matthias Daigl und René Rohner. *Keyword Driven Testing – Grundlage für effiziente Testspezifikation und Automatisierung*. dpunkt.verlag, ca. II. Quartal 2020.

[4] IEEE. *IEEE Standard for Software Reviews and Audits*. IEEE 1028:2008, Institute of Electrical and Electronics Engineers, New York, USA, August 2008.

[5] IEEE. *Standard for Software and System Test Documentation*. IEEE 829:2008, Institute of Electrical and Electronics Engineers, New York, USA, Juli 2008.

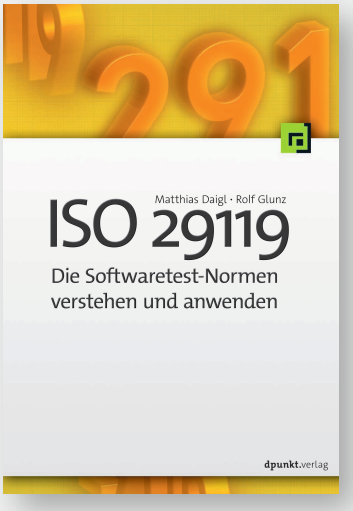
[6] ISO/IEC/IEEE. *Software and systems engineering – Software testing – Part 1: Concepts and definitions*. ISO/IEC/IEEE 29119-1:2013, International Organization for Standardization, Genf, Schweiz, September 2013.

[7] ISO/IEC/IEEE. *Software and systems engineering – Software testing – Part 2: Test processes*. ISO/IEC/IEEE 29119-2:2013, International Organization for Standardization, Genf, Schweiz, September 2013.

[8] ISO/IEC/IEEE. *Software and systems engineering – Software testing – Part 3: Test documentation*. ISO/IEC/IEEE 29119-3:2013, International Organization for Standardization, Genf, Schweiz, September 2013.

[9] ISO/IEC/IEEE. *Software and systems engineering – Software testing – Part 4: Test Techniques*. ISO/IEC/IEEE 29119-4:2015, International Organization for Standardization, Genf, Schweiz, Dezember 2015.

[10] ISO/IEC/IEEE. *Software and systems engineering – Software testing – Part 5: Keyword-Driven Testing*. ISO/IEC/IEEE 29119-5:2016, International Organization for Standardization, Genf, Schweiz, November 2016.



Matthias Daigl · Rolf Glunz

ISO 29119 – Die Softwaretest-Normen verstehen und anwenden

1. Auflage, 2016,
264 Seiten, Festeinband
€ 34,90 (D)
ISBN :
Print 978-3-86490-237-6
PDF 978-3-86491-772-1
ePub 978-3-86491-773-8

Die ISO/IEC/IEEE 29119 stellt einen Satz neuer Normen für Softwareprüfungen dar, der Vokabular, Prozesse, Dokumentation und Techniken für Softwaretesten beschreibt und mit dem Ziel entwickelt wurde, innerhalb jedes möglichen Softwareentwicklungs-Lebenszyklus verwendet werden zu können.

Dieses Buch gibt einen Überblick über diese Normen und zeigt insbesondere die Umsetzung der Anforderungen aus der ISO 29119 hinsichtlich der Testaktivitäten auf. Der Aufbau des Buches spiegelt die Struktur der Norm wider:Teil 1: Grundlagen und Definitionen, Teil 2: Testprozesse, Teil 3: Testdokumentation, Teil 4: Testtechniken. Auch auf Themen, die es (noch) nicht in die Norm geschafft haben, wie z.B. statischer Test mit Reviews und Analyse oder Security Levels, wird eingegangen.

Das Buch richtet sich in erster Linie an Praktiker, die einen leichteren Einstieg in die Norm und eine Hilfestellung für die Umsetzung der ISO 29119 in der Praxis suchen.

dpunkt.verlag
www.dpunkt.de