



Frank Witte

Strategie, Planung und Organisation von Testprozessen

Basis für erfolgreiche
Projektentwicklung im Softwaretest

EBOOK INSIDE

 **Springer** Vieweg

Strategie, Planung und Organisation von Testprozessen

Frank Witte

Strategie, Planung und Organisation von Testprozessen

Basis für erfolgreiche Projektabwicklung
im Softwaretest



Springer Vieweg

Frank Witte
Landshut, Deutschland

ISBN 978-3-658-31227-5 ISBN 978-3-658-31228-2 (eBook)
<https://doi.org/10.1007/978-3-658-31228-2>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Vieweg

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2020

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung: Petra Steinmüller

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Vorwort

Mit meinem ersten Buch „Softwaretest und Testmanagement“ hatte ich mich mit einem Überblick über den gesamten Testprozess beschäftigt und mit meinem Buch „Metriken für das Testreporting“ Analysen und geeignete Verfahren vorgestellt, die während und am Ende des Testprozesses zur Steuerung und Kontrolle Anwendung finden. Mit dem vorliegenden Buch möchte ich mich verstärkt mit dem Anfang des Testprozesses beschäftigen, ich beschreibe vor allem eine Testphase, bei der noch nicht unbedingt ein Testobjekt vorhanden ist und bei der es um die Teststrategie, die Testkonzeption und die Planung geht. Die Organisation und Planung von Projekten erstreckt sich grundsätzlich über den gesamten Projektzeitraum, da sie ständig an neue Gegebenheiten angepasst werden muss. Ein Schwerpunkt liegt jedoch in der Konzeptphase zu Anfang des Projektes.

In früheren Zeiten gab es immer noch Unternehmen, die ernsthaft die Notwendigkeit von Software-Systemtests infrage stellten. Das hat sich immerhin flächendeckend im Bewusstsein gewandelt, ob Kleinunternehmen oder Konzern. Jeder professionelle Software-Entwickler lernt bereits im Studium seine Software zu testen. Abhängig davon, ob es sich um ein erstes Release oder ein Update handelt, gilt es die Software in Gänze oder im Rahmen von Regressionstests so weit wie sinnvoll zu testen. Das nicht zu tun, ist so ähnlich, wie wenn ein Chirurg diskutiert, ob das Desinfizieren der Hände vor der OP nun wirklich notwendig sei.

Im Projektmanagement gilt die Regel, dass 25–35 % der voraussichtlichen Projektgesamtdauer für Planungen aufgewendet werden soll. Denn je klarer Ziele, Rahmenbedingungen und Anforderungen beschrieben sind, desto einfacher und schneller lässt sich das Projekt später umsetzen. So wie Projektplanung die Hauptaufgabe des Projektmanagements ist, ist auch eine gut durchdachte und fundierte Testplanung ein Schlüssel für ein erfolgreiches Testprojekt.

Das folgende Buch behandelt Strategie, Planung und Organisation von Testprozessen.

Unter Strategie wurde ursprünglich die Kunst der Heeresführung verstanden. Clausewitz entwickelte den militärischen Begriff politisch und wirtschaftlich weiter. In der Wirtschaft werden die geplanten Verhaltensweisen der Unternehmen zur Erreichung ihrer Ziele als Strategie verstanden. Mit einer Teststrategie werden folglich Rahmenbedingungen

und Vorgehensweisen, aber auch notwendige Einzelaktivitäten für ein Projekt definiert, die zur erfolgreichen Umsetzung des Ziels erforderlich sind.

Planung will die Zukunft gestalten und soll berücksichtigen, mit welchen Mitteln welches Ziel erreicht werden kann. Das bedeutet für den Softwaretest, dass Ziele, Ressourcen und Prozesse definiert und dokumentiert werden müssen. Mit Hilfe der Planung werden Abläufe festgelegt, die im betrieblichen Umfeld strukturiert ablaufen. Die Ergebnisse der Planung werden in der Organisation laufend umgesetzt. Das bedeutet, dass Testprozesse laufend gesteuert werden müssen, also den aktuellen Testfortschritt mit der Planung permanent abgeglichen werden müssen und ggf. Korrekturmaßnahmen einzuleiten sind.

Für die Planung von Testaktivitäten in einem Softwareprojekt wird ein Testkonzept erstellt. Das Testkonzept kann man mit einem Bauplan vergleichen. Die Struktur des Buches orientiert sich daher am Aufbau eines Testkonzeptes nach **Norm IEE829**. Zusätzlich gebe ich Hinweise zur Teststrategie, zur Projekt- und Testplanung und Methoden zur Bewertung. Neue Trends wie das agile oder kognitive Testen wirken sich ebenfalls auf die Testorganisation aus.

Die Definition „IEEE 829 Standard for Software Test Documentation“ ist ein vom IEEE (Institute of Electrical and Electronic Engineers) veröffentlichter Standard, der einen Satz von acht Basis-Dokumenten zur Dokumentation von Softwaretests beschreibt. Die aktuelle Version ist die IEEE 829-2008. Die Struktur nach IEEE 829 kann man anpassen und erweitern, nicht alle Teile sind in gleichem Umfang relevant. Es ist aber sinnvoll, sie als grobe Leitlinie und Kompass zu verwenden.

Das Testkonzept bestimmt Abgrenzung, Vorgehensweise, eingesetzte Mittel und enthält einen Ablaufplan der Testaktivitäten. Das Testkonzept identifiziert die Testobjekte, die zu testenden Features und die Testaufgaben. Es bildet die Grundlage, auf der die Testorganisation und die Testinfrastruktur bereitgestellt und die Tests durchgeführt werden.

In früheren Zeiten wurden Testaktivitäten meist erst zu einer späteren Projektphase gestartet. Dadurch kam die Testabteilung von Anfang an in Terminkonflikte. Inzwischen hat sich auf breiter Front die Erkenntnis durchgesetzt, dass der Testmanager schon von Anfang an im Projekt mit eingebunden sein soll, also von Anfang an den Entwicklungsprozess mit begleitet, die Planungsphase folglich am Beginn des Projektes steht. Daher soll auch das Testkonzept erstellt werden, bevor bzw. während die Systementwicklung in Arbeit ist und die Projektphase der Definition geeigneter Testfälle und die Phase der Testdurchführung noch nicht begonnen haben.

Für die Testplanung gilt sinngemäß dasselbe wie für die Gesamtplanung von Projekten: für eine effiziente und effektive Projektplanung empfiehlt sich eine „**rollierende**“ **Vorgehensweise**. Das bedeutet, dass zu Beginn des Projektes eine grobe Planung für das gesamte Projekt vorgenommen wird, welche jeweils für die einzelne anstehende Projektphase weiter detailliert wird. Vorteile dieser Vorgehensweise sind, dass im Laufe der Untersuchung gewonnene Erkenntnisse in die Planung einfließen können und der Planungsaufwand vertretbar bleibt [ORGH2019].

Im Folgenden möchte ich aber nicht nur die Inhalte des Testkonzeptes lehrbuchartig aufzeigen, sondern aus Erfahrung auf die Fallstricke und Besonderheiten in der Praxis

hinweisen. Bereits bei der Erstellung des Testkonzeptes kann man, bei geübter Betrachtung, im Projektverlauf entstehende Risiken im Vorfeld erkennen und mögliche Gegenmaßnahmen von Anfang an definieren. Man kann die Schwächen im Prozess aufzeigen und die Rahmenbedingungen transparent machen, um bei eventuell auftretenden Problemen von vornherein gerüstet zu sein. Es geht darum, als Testmanager nicht als Getriebener im Reaktionsmodus zu sein, sondern von Anfang an aktiv agieren zu können.

Jedes Testprojekt ist in einem bestimmten Umfeld eingebunden, und ich empfehle jedem Testmanager, sich erst die Rahmenbedingungen und die besonderen Probleme zu vergegenwärtigen:

Da ist zum einen das Thema Personal: Stehen überhaupt genügend Tester zur Verfügung? Sind die Tester grundsätzlich motiviert? Sind gewisse Ressourcen erfahrungsgemäß überlastet? Wie ist die Auslastung im Zeitverlauf? Wenn ein Mitarbeiter mit 40 Arbeitstagen von 1.3. bis 30.4. geplant ist, aber sein Arbeitspaket Voraussetzungen hat, die erst am 15.3. vorhanden sind, und der Mitarbeiter am 15.4. seine Testergebnisse bereits liefern muss, ist er zeitweise massiv überlastet, obwohl die Gesamtverfügbarkeit stimmt. Gibt es (evtl. aus der Vergangenheit) persönliche Differenzen im Team oder Koalitionen? Stehen die Mitarbeiter dem Projekt exklusiv zur Verfügung, oder greifen unterschiedliche Projekte auf die einzelnen Tester zu? Dabei ist zu bedenken, dass jeder Stakeholder mehr die Kommunikation komplizierter macht: Bei nur 10 Beteiligten gibt es rein rechnerisch 45 Kommunikationswege zwischen allen Beteiligten, bei 20 bereits 190. Vor allem der „Flur Funk“ kann in größeren Organisationseinheiten viel zerstören.

Wer wird den Test durchführen, sind die Tester erfahren oder neu im Projekt? Sind sie motiviert oder eher desillusioniert, machen sie vielleicht sogar schlechte Stimmung gegen das Projekt? Gibt es persönliche Differenzen innerhalb der Stakeholder oder Allianzen im Unternehmen, zum Beispiel Entwickler gegen Tester?

Wie steht es um das Projekt, ist es politisch gewollt, hat es hohe Priorität? Hat das Projekt Förderer im Unternehmen, oder kann es sein, dass es aufgrund anderer Prioritäten ausgebremst, geblockt, hintangestellt wird? Gibt es andere Projekte, die eine höhere Priorität haben, Voraussetzung für das anstehende Projekt haben oder vielleicht sogar im Widerspruch dazu stehen?

Gibt es technische Risiken, Neuerungen, unbekannte Nebeneffekte? Eine Umstellung der Server oder der Datenhaltung in der Cloud hat meist nicht direkt mit der zu testenden Software zu tun, aber in vielen Fällen gibt es Auswirkungen. Ähnlich verhält es sich zum Beispiel bei einer Änderung der Datenbanken. Laufen parallel andere Projekte, die einen technischen oder fachlichen Einfluss auf das eigene Projekt haben, oder ist das betrachtete Projekt sogar auf Gedeih und Verderb ihnen ausgeliefert? Welche Voraussetzungen müssen vorhanden sein, um gewisse Aktivitäten überhaupt zu starten bzw. erfolgreich durchführen zu können? Sind andere Projekte vom eigenen Projekt abhängig?

Wie liefen Projekte in der Vergangenheit ab? Gibt es unternehmensspezifische oder branchenspezifische Probleme? Gibt es dazu „Lessons-Learned-Protokolle“?

Ein Testmanager, der von außen (z. B. extern) in ein Projekt kommt, kennt oft nicht genügend die technischen und fachlichen Besonderheiten und die Stärken und Schwächen

der einzelnen Mitarbeiter. Jemand, der schon lange dabei ist, kennt das alles vielleicht, kann aber wiederum wegen Erfahrungen aus früheren Projekten vorbelastet sein. Wie ist die Stellung der Testabteilung im Unternehmen, ist sie stark oder wird der Test eher als notwendiges Übel betrachtet, der ein Schattendasein fristet?

In der Vergangenheit wurde das Testen von Anwendungen vielfach als notwendiges, aber wenig nutzbringendes Anhängsel klassischer Software-Entwicklungsprojekte gesehen. Diese Betrachtungsweise hat sich nicht zuletzt vor dem Hintergrund geändert, dass Testen, über den Lebenszyklus einer Applikation betrachtet, zwischen 25 und 50 % der anfallenden Gesamtkosten in Anspruch nimmt. Auch Software-Qualitätsprobleme können schwerwiegende Konsequenzen nach sich ziehen, wie beispielsweise eine qualitätsbedingte Projektverzögerung, die oftmals die Time-to-Market verlängert. Größte kommerzielle und materielle Schäden können entstehen, wenn beispielsweise eine Buchungsplattform aus softwaretechnischen Gründen ausfällt, wichtige Daten verloren gehen bzw. verfälscht werden oder Sicherheitslücken im System zum Diebstahl von Kunden- oder Kreditkarteninformationen führen. Darüber hinaus steigen durch die immer engere Verzahnung von Business und IT die Anforderungen an Time-to-Market und Qualität, was den Leistungsdruck auf die Testorganisationen erhöht. Zugleich werden Systeme und IT-Landschaften immer komplexer und die Software-Entwicklungs- und -Lebenszyklen schneller. Hinzu kommen erweiterte regulatorische und sicherheitsspezifische Anforderungen. Insofern ist es nur konsequent, dass seit einigen Jahren das Testen von Anwendungen zu den aktuellen Fokusthemen der IT zählt [IODT2019].

Aus meiner Sicht sind vor allem unklare Aufgabenstellungen ein wesentlicher Problempunkt in zahlreichen Testprojekten. Viele der Kommunikationsprobleme entzündeten sich daran, dass man von Anfang an nicht exakt definiert hat, wie das Testobjekt auszusehen hat, wie es abgegrenzt wird und was die Erwartungshaltung an den Softwaretest ist. Anforderungen, die unvollständig, interpretierbar oder widersprüchliche Inhalte mit sich führen, sind erfahrungsgemäß leider eher die Regel als die Ausnahme. Diese Probleme führen dann zu Fehlern bei der Entwicklung und Rückfragen und Fehlern im Testprozess.

Welche Erwartungen werden vom Management an das Projekt gestellt? Sind diese Erwartungen realistisch hinsichtlich Terminsituation, Kosten und Qualität? Wo liegt die Kostenverantwortung? Dabei muss gerade der Testmanager, der die Qualität der Software sichern soll, von Anfang an vor übertrieben optimistischen Erwartungen warnen.

Erst wenn man entsprechende Aussagen mit Zahlen hinterlegt, werden sie auch konkret greifbar.

Für die Bewertung empfehle ich, die vorhandenen Risiken immer in echte EUR umzurechnen. Eine Aussage wie „das Risiko beträgt 750.000,- EUR wegen mangelhafter Testabdeckung“ ist wesentlich eindringlicher als eine Aussage wie „wahrscheinlich können 5 % der geplanten Testfälle nicht termingerecht getestet werden“.

Viele Fragestellungen kann man sich im Vorfeld stellen und an geeigneter Stelle im Testkonzept berücksichtigen. Typische Problemfälle will ich auf den folgenden Kapiteln dabei benennen. Teilweise empfiehlt es sich, entsprechende „Hilfsdokumente“ dafür zu erstellen.

Dieses vorliegende Buch soll dabei helfen, das Testkonzept mit Leben zu füllen.

Manchmal wird das Testkonzept lediglich aus Gründen der Revisionssicherheit erstellt. Manchmal ist es als theoretisches Dokument vorhanden, spiegelt aber nicht den tatsächlichen Zustand der Testaktivitäten wider. Dadurch wird die Chance vertan, den Testprozess mit dem notwendigen Pragmatismus individuell zu gestalten und begründet zu dokumentieren. Ein Testkonzept mit Alibifunktion erfüllt seinen Zweck nicht.

Die dabei freigegebene Version ist ein Plan, mit dem man startet. Es ist normal, dass man im Laufe der Testaktivitäten neue Erkenntnisse gewinnt und dadurch nicht mehr strikt nach dem vorgezeichneten Wege testen kann. Man muss so viel Flexibilität besitzen, pragmatisch vorzugehen und es kann immer gute Gründe geben, vom ursprünglichen Plan abweichen zu müssen. Es ist dann aber auch notwendig, das Testkonzept nachträglich anzupassen, damit das tatsächliche Testvorgehen daraus hervorgeht. Wenn man bei einem neuen Haus auf der Baustelle eine Wand nachträglich einzieht oder eine Tür versetzt, muss auch der Bauplan geändert werden. Genauso verhält es sich mit dem Testkonzept.

Je mehr die einzelnen Mitarbeiter im Team bei offenen Fragen rund um den Softwaretest das Testkonzept heranziehen, je mehr sie sich damit beschäftigen und kritische Anmerkungen geben, desto eher kann auch das gesamte Testprojekt in die richtige Richtung gelenkt werden. Gerade dem Testmanager obliegt hier die Verantwortung, darauf zu achten, dass das Testkonzept bei Bedarf geändert wird, wenn aktuelle Entwicklungen den Plan überholen. Oft erfährt man eher nebenbei, dass bestimmte Annahmen, die im Testkonzept definiert wurden, gar nicht zutreffend sind.

Das Testkonzept ist kein Produkt, das am grünen Tisch entsteht. Ein Testmanager sitzt nicht wie ein musikalisches Genie im stillen Kämmerlein und entwirft dabei sein großes Werk. Ein Testkonzept entsteht schrittweise im Dialog mit Experten und zukünftig Betroffenen im Projekt. Bei kleinen Vorhaben reicht dafür möglicherweise schon wenige Arbeitstage (fairer Weise sei hier erwähnt), dass eine Testkonzeption im Mittel zwischen 5 bis 20 PT Aufwand benötigt, bei besonders umfangreichen Vorhaben (z. B. Multiprojekten oder großen Softwarepaketen) auch deutlich mehr. Um ein Testkonzept zu erarbeiten, muss eine Vielzahl nötiger Fragen mit den richtigen Experten geklärt werden. Sicher sind im Anschluss noch einzelne Details auszuarbeiten, aber das wichtigste sollte angerissen und einmal gemeinsam durchdacht sein. Alternativ kann ein Testmanager, Testkoordinator, Testverantwortlicher (oder wie auch immer man ihn nennen mag) in Einzelinterviews die Fragen klären und alles nach und nach zusammentragen [PERF2017].

In agilen Teams, in denen iterativ gearbeitet wird, müssen nicht alle Fragen vorweg beantwortet sein. Es reicht auch, sie anzureißen und ihre Lösungen nach Prioritäten der Reihe nach anzugehen. So ist nicht einzusehen, warum ein Team von Anfang an genau definieren muss, welche Testartefakte (Testfälle, Testprotokolle, Berichte, ...) es führen will. Wenn aber ein Akzeptanztest durch den Fachbereich zum nächsten Release erforderlich ist, sollte man die Zielgruppe schon jetzt lieber buchen (und sich damit unweigerlich festlegen). Die o. g. Norm in Form einer Taskliste kann die nötigen Klärungen gut visualisieren und dafür sorgen, dass man im Laufe der Iterationen nichts zu spät löst bzw.

entscheidet. Die zunehmende Agilisierung und künftige Trends führen zu neuen Umweltbedingungen, mit denen sich auch das Testkonzept anpassen muss.

Als zentraler Punkt gilt bei allen Ausarbeitungen und Festlegungen: Man braucht die unbedingte Zustimmung der Beteiligten! Nur wenn ein Konsens im Unternehmen besteht, ist es möglich, die Ziele der Qualitätssicherung, die mit Hilfe von Tests umgesetzt werden, überhaupt zu erreichen.

Besonders wichtig ist in diesem Zusammenhang, das Testkonzept einem Review zu unterziehen und den Review Prozess exakt zu definieren. Dafür sind Review Sitzungen mit allen Beteiligten unumgänglich. Ein Testkonzept nur in die Runde zu senden und auf Feedback zu warten, ist nicht zielführend. Nicht jeder Beteiligte kann zu allen Punkten im Testkonzept etwas sagen, wenn man aber nur die Rückmeldungen einholt läuft man Gefahr, dass einzelne Seiten gar nicht angeschaut werden, weil ein eingeladener Reviewer für bestimmte Kapitel jemand anderen für die Beurteilung des Inhalts als Verantwortlichen betrachtet. Papier ist eben geduldig. Gerade bei Reviews habe ich schon oft beobachtet, dass man meint, keine Zeit zu haben und deswegen notwendige Aktivitäten abkürzen oder unterlassen wollen. Man verbraucht dadurch aber am Ende noch mehr Zeit – frei nach dem Motto „wir haben Zeit die Sachen dreimal falsch zu machen aber nicht um sie einmal richtig durchzuführen“.

Ein Review-Zyklus dient also nicht der Fehlerfindung im Konzept allein, sondern vor allem der Zustimmung. Diese muss man sich aktiv einholen. Viele machen hier den Fehler, ein Konzept auf den Tisch zu knallen und dann davon auszugehen, dass sich alle (gefälligst) daran halten. Systemisch liegt dadurch die Verantwortung bei dem Ersteller des Testkonzeptes und er wird sich um alles selbst kümmern müssen, was selten sinnvoll ist [PERF2017].

Jedes eigene Projekt sollte ein eigenes Testkonzept haben, je nach Projektumfang unterschiedlich umfangreich.

Ein **Projekt** ist durch **Norm DIN 69901** folgendermaßen definiert: Es ist ein Vorhaben, das im Wesentlichen durch die Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist: Für das Projekt existiert eine konkrete Zielvorgabe, es ist zeitlich begrenzt, es benötigt spezifische Ressourcen (z. B. finanziell, personell), es existiert eine eigenständige Ablauforganisation, die sich von der Standardorganisation des Unternehmens abgrenzt, es werden nicht routinemäßige Aufgaben behandelt, es besteht aus mindestens 3 Teilnehmern und hat eine Mindestdauer von 4 Wochen.

Ein Projekt hat einen Anfang und ein Ende, im Gegensatz zu einem Prozess, der fortlaufend ist. Ein Projektstrukturplan (PDF) ist in Arbeitspakete gegliedert, in denen die einzelnen Vorgänge, also auszuführende Schritte, beschrieben sind. Häufig wird am Ende eines Arbeitspakts ein Meilenstein gesetzt – ein Zeitpunkt zu dem der Fortschritt des Projektes analysiert und gegebenenfalls den Auftraggebern in einem Meeting vorgestellt wird. Die Überwachung des Ablaufs führt der Projektmanager durch, alle Mitglieder des Projektteams müssen ihm zuarbeiten und sind für die ihnen zugewiesenen Vorgänge verantwortlich. Auch das Krisenmanagement und das Erstellen einer Dokumentation fallen in die Aufgaben des Managers [KARB2019].

Der Testmanager ist für das Projektmanagement aller Teststufen verantwortlich, koordiniert und überwacht die Testaktivitäten. Eine gute Zusammenarbeit und enge regelmäßige Abstimmung mit dem Projektleiter sollte deshalb gegeben sein, sonst wird es erfahrungsgemäß schwierig bis unmöglich gute Fortschritte und Ergebnisse beim Testprozess zu erreichen. Differenzen mit dem Projektleiter sollten gar nicht erst entstehen bzw. müssen sofort ausgeräumt werden. Wenn Projektleiter und Testmanager nicht an einem Strang ziehen, werden das andere Stakeholder schnell bemerken, die Produktivität im Projekt wird darunter leiden oder sogar das gesamte Projekt scheitern.

Man kann aber zusätzlich ein generisches Testkonzept für das gesamte Unternehmen bzw. die gesamte zu betrachtende Organisationseinheit erstellen, vor allem wenn es sich um wiederkehrende Tätigkeiten handelt.

Die IEEE 829 dient dabei als ein Leitfaden, es ist aber notwendig, das Testkonzept an die betrieblichen Gegebenheiten und individuellen Besonderheiten anzupassen. Ich habe bereits eine Reihe von Testkonzepten erstellt, die sich zwar in der Grobstruktur ähneln aber im Schwerpunkt je nach Branche, Unternehmen und Testobjekt unterscheiden.

Erfahrungsgemäß kommt es dabei zu zahlreichen Fragen, die sich während der Erstellung des Testkonzeptes bereits klären lassen. Je stärker man die Stakeholder dabei schon in der Planungsphase einbezieht, umso mehr Aufmerksamkeit und Vertrauen werden bereits im Vorfeld erzeugt, bevor es mit der eigentlichen Testdurchführung losgeht.

Bei den offiziellen Vorgaben und Normen geht man in der Regel von einem optimalen Arbeitsumfeld aus. Dies ist oftmals nicht gegeben, sondern man hat von Anfang an Rahmenbedingungen, die das Projekt negativ beeinflussen, wie z. B. unklare Aufgabenstellungen, Terminverzug, mangelndes Know-how im Testteam; dafür sind meist Kommunikationsprobleme verantwortlich. Die Beispiele in vielen Lehrbüchern gehen von idealen Umgebungsvoraussetzungen aus, die in der Praxis so nicht gegeben sind. Ich werde im Folgenden im Detail darauf eingehen, wie man gerade bei schwierigen Randbedingungen den Softwaretest doch noch zum Erfolg führen bzw. das Schlimmste verhindern kann. Viele Projekte erreichen ihr Ziel nicht in Time and Budget, oft, weil die Planung von Anfang an zu optimistisch war und/oder zu wenig Parameter bei der Betrachtung eingeflossen sind. Man kann aber mit geeigneten Methoden darauf hinwirken, dass Terminverzug, Kostenüberschreitung oder Qualitätsmängel immerhin deutlich minimiert werden. Man muss dazu den Königsweg zwischen einem geplanten, strukturierten Vorgehen und der notwendigen Flexibilität beschreiten. Ein pragmatischer Umgang mit den vorhandenen Realitäten ist genauso wichtig wie die Einhaltung von Regeln und Normen.

Der Umfang eines Testkonzeptes hängt sicherlich vom Umfang des Projektes ab. Aber selbst bei kleinen Projekten muss man einen Plan haben, wie man beim Testen vorgeht und die entscheidenden Rahmendaten dokumentieren.

Falls Informationen bereits existieren oder an anderer Stelle gepflegt werden, ist zu vermeiden, sie redundant zu beschreiben. Eine Stakeholder Liste mit Namen, Abteilungsbezeichnungen und Telefonnummern muss bei einer betrieblichen Umorganisation jeweils aktualisiert werden. Wenn man jedoch nur die Rolle und den Namen beschreibt und dabei auf ein Dokument verweist, das dafür ohnehin gepflegt wird, kann das gerade bei großen

Projekten besser sein. Das gilt auch für Terminpläne. Eine redundante Datenpflege beinhaltet immer das Risiko, dass unterschiedliche und damit widersprüchliche Informationen vorgehalten werden, auf denen aufgebaut wird.

Es empfiehlt sich für eine Organisationseinheit, die häufiger Tests durchführen soll, ein Template für das Testkonzept vorzubereiten. Wenn einzelne Teile nicht relevant sein sollten, kann man das dann innerhalb dieser Vorlage an geeigneter Stelle anmerken. Dadurch wird nichts vergessen, und man bekommt bei der Erstellung des Testkonzeptes bereits Hinweise, an was man denken sollte.

Man kann sich an ähnlich gelagerten Projekten aus der Vergangenheit orientieren und Schwachstellen, die damals auftraten, bereits zu Projektbeginn untersuchen: wenn man schon weiß, dass die Bereitstellung der Testumgebung bei früheren Releases mehrfach verzögert war oder man schon ahnt, dass es bei der Integration bestimmter Komponenten immer wieder zu Fehlern kommt, kann man bereits während der Planungsphase ausreichend Puffer einplanen und entsprechende Risiken schon zu Projektbeginn aufzeigen.

Wenn im Folgenden von Testmanagern, Testkoordinatoren, Testern usw. die Rede ist, meine ich immer auch die weibliche Form. Die männliche Form wird nur aufgrund der Lesbarkeit verwendet.

Zu bedenken ist, dass eine gute Testorganisation und ein durchdachtes Testkonzept Zeit benötigen. Die Schätzungen für den dafür benötigten Aufwand sollen sich an Erfahrungsschatz, Projektumfang und dem Einsatzzweck der Anwendung orientieren. Hier gilt es für den Testmanager, hinter seinen Zahlen zu stehen, diesen Aktivitäten die nötige Wichtigkeit im Unternehmen zu verschaffen und manchen übertriebenen Optimismus zu bremsen.

Literatur

- [KARB2019]: <https://karrierebibel.de/projektarbeit>, zugegriffen am 09.05.2020
- [PERF2017]: <https://www.informatik-aktuell.de/entwicklung/methoden/das-perfekte-testkonzept-in-6-schritten.html>, zugegriffen am 09.05.2020
- [IODT2019]: <https://www.iot-design.de/allgemein/transformation-von-testorganisationen/>, zugegriffen am 09.05.2020
- [ORGH2019]: https://www.orghandbuch.de/OHB/DE/Organisationshandbuch/2_Vorgehensmodell/21_Projektvorbereitung/212_Projektplanung/projektplanung-node.html, zugegriffen am 09.05.2020

Inhaltsverzeichnis

- 1 Testdokumente nach IEEE 829** 1
 - 1.1 Normen 1
 - 1.2 Testkonzept 2
 - 1.3 Testspezifikation („test specification“) 3
 - 1.4 Testbericht („test reporting“) 4
 - 1.5 Integritätslevel 6
 - Literatur. 8
- 2 Teststrategie** 9
 - 2.1 Merkmale der Teststrategie 9
 - 2.2 Integrationsstrategie 13
 - 2.3 Testspezifikationen 13
 - 2.4 Einsatz von Tools 14
 - 2.5 Fragen zur Umsetzung der Teststrategie 15
 - 2.6 Balanced Scorecard 16
 - Literatur. 19
- 3 Testziele** 21
 - 3.1 Bedeutung von Testzielen 21
 - 3.2 Testkenngrößen 23
 - 3.3 Test-Reporting 24
 - Literatur. 27
- 4 Testplanung** 29
 - 4.1 Inhalte der Testplanung 29
 - 4.2 Teilpläne für IT-Projekte 30
 - 4.3 Testvoraussetzungen 32
 - 4.4 Rahmenbedingungen der Testplanung 33
 - 4.5 Testwerkzeuge 34
 - 4.6 Testaufwände und Testautomatisierung 34
 - Literatur. 35

5	Bezeichnung des Testkonzeptes und Einleitung	37
5.1	Bezeichnung und Ablage des Testkonzeptes	37
5.2	Testkonzepte für einen längeren Zeitraum	38
5.3	Formale Grundlagen	39
5.4	Einleitung Testkonzept	40
5.5	Im Testkonzept referenzierte Dokumente	42
5.6	Testrahmen mit Testvoraussetzungen, Fehlerklassen und Startbedingungen	46
	Literatur	47
6	Testorganisation	49
6.1	Wesentliche Kriterien der Testorganisation	49
6.2	Testpersonal und Testereinsatzplanung	52
6.3	Rechenzeit	53
6.4	Datenbereitstellung	54
6.5	Benötigte Hardware	54
6.6	Projektorganisation	54
6.7	Stakeholder und Rollen in der Projektorganisation	55
6.8	Organisationsstruktur	55
6.9	Projektlenkungsausschuss	55
	Literatur	57
7	Prozessbeschreibung	59
7.1	Testprozess nach ISTQB	59
7.2	Planung und Steuerung	60
7.3	Meetings zu Projektstart	61
7.4	Analyse und Design	63
7.5	Realisierung und Durchführung	63
7.6	Auswertung und Bericht	64
7.7	Testabschluss	64
7.8	Dokumente als Testvoraussetzung	65
	Literatur	66
8	Testobjekte und Testphasen	67
8.1	Testobjekte	67
8.2	Testphasen	68
8.3	Einflussfaktoren für die Ermittlung des Testaufwands	70
	Literatur	72
9	Teststufen	73
9.1	Klassische Teststufen	73
9.2	Exploratives Testen	76
9.3	Teststufen in einem agilen Projekt	77
9.4	Integration der Testaktivitäten mit anderen Projektaktivitäten	81
	Literatur	82

10	Zu testende Leistungsmerkmale	83
10.1	Arten von Systemtests	83
10.2	Spezielle Testverfahren	86
	Literatur.	87
11	Leistungsmerkmale, die nicht getestet werden	89
11.1	Testabdeckung durch Testverfahren.	89
11.2	Nachweise durch andere Methoden.	90
	Literatur.	91
12	Priorisierung von Testfällen	93
12.1	Testumfang.	93
12.2	Priorisierung beim Systemtest und beim Abnahmetest	94
12.3	Priorisierung beim Modultest und Integrationstest.	94
12.4	Priorisierung nach unterschiedlichen Testarten	96
12.5	Risikobasiertes Testen	98
	Literatur.	99
13	Permanente Testorganisation	101
13.1	Wesen der permanenten Testorganisation	101
13.2	Bedingungen für eine permanente Testorganisation	103
13.3	Kritische Punkte für permanente Testorganisationen	104
13.4	Testrichtlinien	104
	Literatur.	105
14	Abnahmekriterien	107
14.1	Merkmale von Abnahmekriterien	107
14.2	Abnahmekriterienkatalog.	108
14.3	Organisation der Abnahme.	110
14.4	Definition geeigneter Akzeptanzkriterien	111
14.5	Eigenschaften von Akzeptanztests.	113
	Literatur.	114
15	Kriterien für Testabbruch und Testfortsetzung	115
15.1	Fälle für Testabbrüche	115
15.2	Wiederanlauf und Testfortsetzung nach Testabbruch	116
16	Testrisiken	119
16.1	Projekterfolg	119
16.2	Bewertung von Risiken	120
16.3	Mögliche Testrisiken	121
	Literatur.	127
17	Testdaten	129
17.1	Testdatenmanagement	129

17.2	Ansätze zur Testdatenerstellung	131
17.3	Werkzeuge zur Testdatenerstellung	133
17.4	Testdatentypen	134
	Literatur	134
18	Testdokumentation	135
18.1	Ziele der Testdokumentation und Integritätslevel	135
18.2	Struktur der Testdokumentation	137
18.3	Testskript	137
18.4	Testfall	138
18.5	Testprotokoll	138
18.6	Zusammenfassung der Testdokumentation	139
18.7	Vernachlässigung der Testdokumentation	139
18.8	Testabschlussbericht	141
18.9	Metriken	141
18.10	Verfügbarkeit des Testsystems	143
	Literatur	145
19	Testaufgaben	147
19.1	Arten von Testaufgaben	147
19.2	Rollenbeschreibungen	148
19.3	Norm IEC 62034	150
19.4	Problemlösungsprozess	151
19.5	Testaufgaben im Regressionstest	152
19.6	Testtools	153
19.7	Aufgaben für die einzelnen Stakeholder	154
	Literatur	154
20	Testumgebung	157
20.1	Notwendigkeit von Testumgebungen	157
20.2	Angemessenheit der Testumgebung	158
20.3	Testumgebungsmanagement	158
20.4	Externe Testumgebungen	160
20.5	Testumgebung in der Cloud	161
	Literatur	164
21	Verantwortlichkeiten, Zuständigkeiten und Kommunikation	165
21.1	Festlegung der Verantwortungsbereiche	165
21.2	Kommunikation im Projekt	166
21.3	Definition von Zuständigkeiten	167
21.4	Kommunikationsmatrix	168
21.5	Auswahl der Informationsempfänger	170
21.6	Definition von Kommunikationszielen	172
21.7	Festlegung von Informationsinhalten	172

21.8	Vorteile der Kommunikationsmatrix	173
21.9	Anforderungen an die Projektkommunikation	173
21.10	RACI-Methode	174
21.11	Häufige Fehlerquellen	176
	Literatur	176
22	Personal, Einarbeitung, Ausbildung	177
22.1	Testpersonal	177
22.2	Konfliktmanagement	178
23	Zeitplan/Arbeitsplan	181
23.1	Ermittlung der Testdauer	181
23.2	Detaillierte Arbeitsplanung	183
23.3	Kostenschätzung	184
23.4	Planung der einzelnen Testphasen	184
23.5	Personalauslastungsplan	187
23.6	Detaillierte Testaufwandsplanung	188
23.7	Schätzung der Testproduktivität mit der COCOMO-II-Methode	190
	Literatur	192
24	Planungsrisiken und Unvorhersehbares	193
24.1	Unvorhersehbare Risiken	193
24.2	Zielbeiträge der Handlungsalternativen sichtbar machen	194
24.3	Methoden und Werkzeuge als Entscheidungstechniken	194
24.4	Methoden der Investitionsrechnung	195
24.5	Argumentenbilanz	195
24.6	SWOT-Analyse	196
24.7	Konsequenztabelle	197
24.8	Entscheidungsbaum	198
24.9	Nutzwertanalyse	199
24.10	Risikoanalyse	201
24.11	Methoden und Werkzeuge zur Entscheidung auswählen und nutzen	201
24.12	Entscheidungsprobleme, Handlungsalternativen und mögliche Folgen aufzeigen	202
25	Genehmigung und Freigabe	205
25.1	Freigabeempfehlung	205
	Literatur	207
26	Projektorganisation	209
26.1	Stablinien-Projektorganisation	209
26.2	Reine Projektorganisation	211
26.3	Matrixorganisation	212

26.4	Ausbalancierte Matrixorganisation	213
26.5	Befugnisse und Zieleverantwortung des Projektleiters.	213
26.6	Formeln zur Bewertung des Projektfortschritts	214
	Literatur.	215
27	Testmethoden.	217
27.1	Einsatz von Testmethoden	217
27.2	Geeignete Testmethoden und Teststrategie	218
27.3	Testing Maturity Model (TMMI))	219
27.4	TMap (Test Management Approach).	221
27.5	Vorteile von TMap	224
	Literatur.	224
28	Reifegrad des Testmanagements nach TPI Next	225
28.1	Ermittlung des Reifegrads des Testprozesses	225
28.2	Testorganisation im Reifeprozess nach TPI Next	227
28.3	Einteilung der Reifegrade	228
28.4	Reifeprozess für die Testorganisation	228
28.5	Reifegrad „Initial“	229
28.6	Reifegrad „Kontrolliert“	229
28.7	Reifegrad „Effizient“	230
28.8	Reifegrad „Optimierend“	231
28.9	Unterschiede TMMI und TPI.	232
	Literatur.	233
29	Besonderheiten der Testorganisation in agilen Projekten.	235
29.1	Agile Projekte	235
29.2	Agile Methoden	236
29.3	Organisation und Ziele agiler Projekte	238
29.4	Agile Projekte und traditionelle Organisation	239
29.5	Notwendige Voraussetzungen zur Testorganisation in agilen Projekten	239
29.6	Grundsätze für agiles Testen	240
	Literatur.	241
30	Künstliche Intelligenz und kognitives Testen	243
30.1	Künstliche Intelligenz	243
30.2	Anwendungsbereiche künstlicher Intelligenz	244
	Literatur.	245
	Nachwort	247
	Stichwortverzeichnis.	253

Zusammenfassung

Der IEEE 829 definiert Basis-Dokumente für den Softwaretest. Das Testkonzept beschreibt die Ergebnisse der Testplanung. Unterschiedliche Arten von Testdokumenten dokumentieren die Testinhalte, den Testfortschritt und die Ergebnisse der Testdurchführung.

1.1 Normen

Normen helfen bei der Standardisierung von Geschäftsprozessen und Abläufen. Eine Norm ist ein Dokument, das Anforderungen an Produkte, Dienstleistungen oder Verfahren festlegt. Sie schafft somit Klarheit über deren Eigenschaften, erleichtert den freien Warenverkehr und fördert den Export. Sie unterstützt die Rationalisierung und Qualitätssicherung in Wirtschaft, Technik, Wissenschaft und Verwaltung. Sie dient der Sicherheit von Menschen und Sachen sowie der Qualitätsverbesserung in allen Lebensbereichen [DIN2019].

Der Standardisierungsprozess im Softwaretest hat wesentlich später eingesetzt als in anderen Industrien. So wurde z. B. der Zertifizierungsprozess im Bereich von Softwaretestern erst ab Mitte der 1990er-Jahre in Großbritannien begonnen, woraus im Jahre 1997 das ISEB-Zertifizierungsprogramm erwuchs (ISEB = „Information System Examination Board“).

Im Jahre 1998 wurde eine erste Variante des IEEE 829 beschrieben, um damit bestimmte Testdokumente überhaupt zu etablieren. Diese Variante wurde im Jahr 2008 erweitert und aktualisiert.

Die Definition **IEEE 829 Standard for Software Test Documentation** ist ein vom IEEE (Institute of Electrical and Electronics Engineers) veröffentlichter Standard, der ei-

nen Satz von acht Basis-Dokumenten zur Dokumentation von Softwaretests beschreibt. Die aktuelle Version ist die IEEE 829-2008. Der Standard beschreibt Form und Inhalt der jeweiligen Dokumente. Er schreibt jedoch nicht vor, welche der jeweiligen Dokumente zwingend verwendet werden müssen.

Dieser Standard kann für traditionelle, nach dem Wasserfallmodell ablaufende Softwareprojekte genauso wie für agile Vorhaben angewendet werden.

Im September 2013 wurden die ersten Teile des internationalen Standards **ISO/IEC/IEEE 29119** veröffentlicht, der den IEEE 829 Standard for Software Test Documentation international ersetzt.

Der Standard beschreibt acht Dokumente, die sich wie folgt in drei Kategorien unterteilen lassen.

1.2 Testkonzept

Testkonzept (Test Plan): Das Testkonzept bestimmt Abgrenzung, Vorgehensweise, Mittel und Ablaufplan der Testaktivitäten. Es bestimmt die Elemente und Produktfunktionen, die getestet werden sollen, die Testaufgaben, die durchgeführt werden müssen, das verantwortliche Personal für jede Aufgabe und das Risiko, das mit dem Konzept verbunden ist.

Das Testkonzept dient als Leitfaden, um Testaktivitäten zu planen und zu organisieren.

Im Folgenden ist eine Struktur angegeben, die sich für die Erstellung von Testkonzepten als sinnvoll erwiesen hat.

1. Einführung
 - 1.1. Identifikation des Testkonzeptes
 - 1.2. Geltungsbereich und Umfang
 - 1.3. Referenzen
 - 1.3.1. Externe Referenzen
 - 1.3.2. Interne Referenzen
 - 1.4. Zu testendes System und Testobjekt
 - 1.5. Überblick über die Testaufgaben
 - 1.5.1. Organisation
 - 1.5.2. Projekttestplan
 - 1.5.3. Integrationsstufen
 - 1.5.4. Ressourcenübersicht
 - 1.5.5. Zuständigkeiten
 - 1.5.6. Werkzeuge, Techniken, Methoden, Metriken
2. Details
 - 2.1. Testprozess und Teststufen
 - 2.2. Dokumente
 - 2.3. Abweichungs- und Änderungsmanagement
 - 2.4. Berichtswesen

3. Allgemeines

3.1. Glossar

3.2. Änderungsdienst und Historie [QYTE2019]

Das Testkonzept definiert Gültigkeitsbereich, Vorgehensweise, Zeitplanung und die benötigten Ressourcen der beabsichtigen Testaktivitäten. Dabei werden Testobjekte, zu testende Features und die einzelnen Testaufgaben identifiziert.

Eine Zuordnung der Testaufgaben auf einzelne Mitarbeiter kann ebenfalls vorgenommen werden. Manchmal wird dafür auch auf ein entsprechendes eigenes Planungsdokument verwiesen.

Der Test soll unabhängig von der Entwicklung stattfinden. Nur wenn der Test eine neutrale, eigene Instanz ist kann er diese Unabhängigkeit gewährleisten. Die daraus folgende organisatorische Festlegung wird ebenfalls im Testkonzept beschrieben.

Darüber hinaus werden die Testumgebung, das Testentwurfssverfahren und die anzuwendenden Verfahren zur Messung der Testaktivitäten und Testergebnisse beschrieben und deren Auswahl begründet. Das Testkonzept enthält ein Kapitel über Risiken und mögliche Gegenmaßnahmen, wenn diese Risiken eintreten.

Das Testkonzept dokumentiert somit die Ergebnisse des Testplanungsprozesses.

1.3 Testspezifikation („test specification“)

Die Testspezifikation enthält die Dokumente Testentwurfsspezifikation, Testfallspezifikation und Testablaufspezifikation.

Die **Testentwurfsspezifikation** („test design specification“) verfeinert die Beschreibung der Vorgehensweise für das Testen der Software. Sie identifiziert Features und Funktionen, die von definierten entsprechend zugeordneten logischen Testfällen abgedeckt werden müssen. Sie beschreibt außerdem die Testfälle und Testabläufe, die benötigt werden, um die Testfälle als erfolgreich durchgeführt betrachten zu können. Die Kriterien, nach denen Tests bestanden oder nicht bestanden sind, werden dabei pro Funktion definiert [SETH2019].

Die **Testfallspezifikation** („test case specification“) definiert die zu verwendenden Eingabewerte und die der zu erwartenden Ausgabewerte. Außerdem werden in der Testfallspezifikation die dazu notwendigen Vor- und Nachbedingungen, die Ziele und Testaktionen beschrieben. Die Priorität und Dauer der Testdurchführung sollen, vor allem bei manuell durchzuführenden Testfällen, ebenfalls dokumentiert werden. Bei Bedarf ist auch zu beschreiben, wie nach der Testdurchführung das System auf einen Anfangszustand zurückgesetzt werden muss.

In der Norm IEEE 829 sind Testfälle vom Testdesign getrennt. Das erlaubt die Verwendung der Testfälle in unterschiedlichen Designs und die Wiederverwendung in anderen Situationen. In der Praxis sind häufig die Testentwurfsspezifikation und Testfallspezifikation zusammen in einem Dokument gespeichert und bestehende Testfälle werden dabei für

ein neues Projekt nur kopiert. Der modulare Aufbau in unterschiedlichen Dokumenten führt aber zu einer höheren Qualität, auch wenn es anfangs einen Mehraufwand darstellt.

Die **Testablaufspezifikation** („test procedure specification“) ist die Beschreibung aller Schritte zur Durchführung der spezifizierten Testfälle und Implementierung des zugehörigen Test-Designs. Bei vielen Testfällen sind zunächst die Voraussetzungen herzustellen (z. B. mehrere Buchungsschritte), bevor mit dem eigentlichen Testfall überhaupt begonnen werden kann. Manche Tests bestehen aus einer Reihe von Schritten. Die Testablaufspezifikation wird manchmal auch als Testprozedurbeschreibung, Testskript oder Testdrehbuch bezeichnet. Oft werden auch Mischformen von Testfallspezifikation und Testablaufspezifikation verwendet.

In vielen professionellen Testmanagement-Tools werden der Testablauf und die Trennung von Testfall- und Testablaufspezifikation mittels verschiedener Bereiche abgebildet.

1.4 Testbericht („test reporting“)

Das Berichtswesen soll die Ergebnisse aller Testläufe einer Version zusammenfassen.

Der **Testobjektübergabebericht** („test item transmittal report“) beschreibt die Übergabe der Testfälle für den Fall, dass getrennte Entwicklungs- und Testteams eingebunden sind oder für den Fall, dass ein offizieller Zeitpunkt für den Beginn einer Testausführung erwünscht ist. Im Testobjektübergabebericht wird eine definierte Version an ein Testteam übergeben, und die testbaren Features sind dort aufgeführt.

Das **Testprotokoll** („test log“) dient zur chronologischen Aufzeichnung der Ereignisse während einer Testausführung. Bei jeder Testdurchführung ist zu definieren, welche Logfiles von der Entwicklung benötigt werden, um Fehlverhalten reproduzieren zu können. Das ist gerade bei automatisierten Tests, die über Nacht oder am Wochenende laufen, wichtig. Zu beachten ist aber, dass ein hoher Log Level nicht nur viele Informationen aufzeichnet und daher viel Speicherplatz benötigt, sondern auch, dass durch das verstärkte Logging die Antwortzeit und die Systemlast derart ansteigen kann, dass es zu Fehlern kommt, die unter Normalbetrieb gar nicht auftreten würden.

Der **Testabweichungsbericht** („test incident report“) beschreibt alle Ereignisse, die während einer Testausführung auftreten und weitere Nachprüfungen erfordern. Eine Abweichung ist gemäß Norm IEEE 108 (IEEE Standard for Software Unit Testing) ein Ereignis, das während des Testens auftritt und weiterer Untersuchungen bedarf.

Der **Testabschlussbericht** („test summary report“) fasst die Testaktivitäten zusammen, die mit einer oder mehreren Testentwurfsspezifikationen zusammenhängen. Wenn z. B. der Systemtest oder der Integrationstest einer definierten Software komplett abgeschlossen ist, wird der Testabschlussbericht erstellt, um die nächste Teststufe starten zu können. Wenn es dabei z. B. beim Systemtest nur eine bedingte Freigabe gibt, kann die Software zwar im Abnahmetest weiter getestet werden, aber man muss bestimmte Einschränkungen beachten. Manchmal sind auch Fehler am Ende der Testaktivitäten einer Teststufe noch nicht behoben bzw. noch nicht erfolgreich nachgetestet, so dass man zwar mit diesen Einschrän-

kungen die nächsthöhere Teststufe durchführen, das Produkt aber noch nicht in Produktion ausliefern kann solange diese Fehler noch bestehen. Der Testabweichungsbericht wird in der Praxis häufig mit dem Testabweichungsbericht in einem Dokument zusammengefasst.

Der Testabschlussbericht soll die beim Test aufgetretenen Fehler mit ihrem aktuellen Status dokumentieren, also eine Aussage darüber treffen, welche Fehler festgestellt und während der Testphase behoben wurden. Der Testabschlussbericht wird teilweise auch als **Testbewertungsbericht** bezeichnet. Dieses Dokument enthält auch eine Bewertung des Testprozesses und einen Erfahrungsbericht.

Bei umfangreicheren Projekten dauern einzelne Testphasen über mehrere Wochen bzw. Monate. Daher ist es ratsam, im wöchentlichen Abstand einen aktuellen **Statusbericht** zu erstellen. Bereits im Rahmen der Testvorbereitung sollte man sich dabei überlegen, welche Aussagen man darstellen will und welche Metriken man dazu einsetzen will. Es ist zu empfehlen, das regelmäßige Reporting im Vorfeld der Testphasen zu definieren und abzustimmen. Idealerweise werden Metriken, die nur noch für das laufende Projekt konfiguriert werden müssen, bereitgestellt. Wenn man Fragen nach dem Testfortschritt während der Testdurchführungsphase beantworten soll und dann nicht vorbereitet ist, hat man zusätzlich zu stressigen Projektphasen noch das Problem, Ereignisse ansprechend zu präsentieren und die benötigten Informationen (nicht zu viel aber auch nicht zu wenig) zu sammeln und aufzubereiten.

In diesem Zusammenhang sollte auch der Rhythmus für das Reporting bereits vor Beginn des Projektes geprüft werden: wie aktuell sind die Ereignisse, die in den Testreport einfließen? Wenn am Mittwoch der Report für die Vorwoche erst erstellt und unternehmensweit verteilt wird, oder – noch schlimmer – ein Testreport erst am 10. des Folgemonats für den laufenden Monat präsentiert wird, sind viele Informationen längst veraltet. Ein Reporting über Fortschritt, aber auch Probleme, muss immer zeitnah, also tagesaktuell erfolgen und damit auch zeitnah erstellt werden können.

Bei allen Berichten sollte man die Zeit für die Aufbereitung und Präsentation der Ergebnisse nicht unterschätzen. In aller Regel ist es schwierig, komplexe Zusammenhänge auf wenigen Seiten Reporting zusammenzufassen und grafisch ansprechend darzustellen. Das ist aber unbedingt notwendig für ein umfassendes Berichtswesen gegenüber dem Management und zuletzt für das Eigenmarketing als Testmanager. Die Aufbereitung von Fakten für das Reporting kann auch sehr zeitaufwändig werden. Meiner Erfahrung nach ist in vielen Projekten das Reporting nicht von Anfang an so definiert, dass man sinnvoll mit standardisierten Berichten, die für das individuelle Projekt nur noch konfiguriert werden müssen, darauf aufsetzen kann. Ein klares Reporting ohne Beschönigung, ohne Panikmache und ohne Schuldzuweisungen, sondern als Aufzeigen der Ist-Situation mit einer realistischen Einschätzung verbessert die Kommunikation im Projekt erheblich.

Abb. 1.1 zeigt den Zusammenhang der einzelnen Dokumente im Überblick.

Die Eingangsdokumente „Proj Doc“ (**Projektdokumentationen**) und „Item Doc“ (**Projektprodukt dokumentationen**), sowie die „Test Item“ (Testgegenstand/Software)

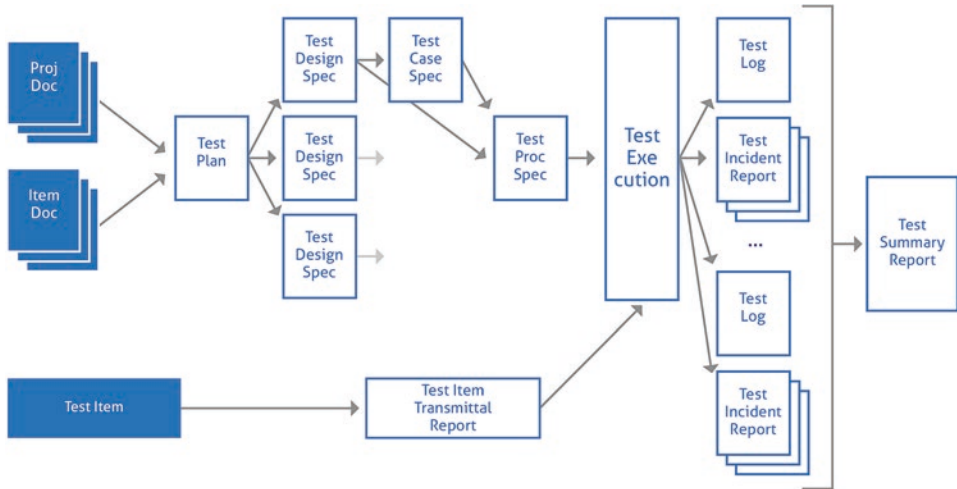


Abb. 1.1 Testdokumente und Ergebnistypen im Test

sind nicht Gegenstand des Standards, sondern werden von den im vorigen Kapitel benachbarten Prozessen definiert und erstellt.

Daneben gibt es weitere typische Dokumente, welche jedoch nicht im IEEE 829 hinterlegt sind. Dazu gehört beispielsweise der **Testfortschrittsbericht** („test progress report“), der **Teststufensteckbrief** etc. Außerdem gibt es weitere Dokumente, welche in der Regel durch das Projektmanagement definiert werden (beispielsweise Steckbrief des Teilprojektes Test, Risikoliste, Offene Punkte Liste, etc.) [PMQS2019].

In diesem Zusammenhang sind auch **Release Notes** zu betrachten. Release Notes sind Dokumente, die im Rahmen der Übergabe von der Entwicklung zum Test zu Beginn der Testdurchführung die Testobjekte identifizieren, ihre Konfiguration, ihren aktuellen Status und weitere Informationen zur eingesetzten Software enthalten.

Bei allen erstellten Dokumenten ist darauf zu achten, die Identifizierung eindeutig vorzunehmen, eine Versionierung und den Dokumentenstatus zu kennzeichnen (z. B. „in Arbeit“, „in Review“, „fertiggestellt“, „abgenommen“). Dazu sollte ein Ablaufplan über den jeweiligen Dokumentenstatus als Basisdokument dem Testprozess zugrunde gelegt werden.

1.5 Integritätslevel

IEEE 829 soll für ein agiles, ebenso wie für ein konventionelles systematisches Vorgehen geeignet sein und bezieht sich genauso auf Systeme wie Software. Mit IEEE 829 wurde aus einem bisher dokumentenorientierten Prinzip „Dokumente folgen Aktivitäten“ nun ein Dokumentationsprozess (konform zu **ISO 12207**, **ISO 15288**) mit Input und Output (aufgabenorientiert).

Mit IEEE 829 werden 4 **Integritätslevel**, je nach Fehlertoleranz des Systems, definiert. Für jedes Level werden bestimmte Dokumente vorgeschlagen:

4 – Katastrophal

Master Test Plan (statisch), Level Test Plan (dynamisch und detailliert), Level Test Design, Level Test Case, Level Test Protocol, Level Test Log, Anomaly Report, Level Integration Test Status Report, Level Test Report, Master Test Report

3 – Kritisch

Master Test Plan (statisch), Level Test Plan (dynamisch und detailliert), Level Test Design, Level Test Case, Level Test Protocol, Level Test Log, Anomaly Report, Level Integration Test Status Report, Level Test Report, Master Test Report

2 – Marginal

Level Test Plan (dynamisch und detailliert), Level Test Design, Level Test Case, Level Test Protocol, Level Test Log, Anomaly Report, Level Integration Test Status Report, Level Test Report

1 – Vernachlässigbar

Level Test Plan (dynamisch und detailliert), Level Test Design, Level Test Case, Level Test Protocol, Level Test Log, Anomaly Report, Level Test Report

In Anlehnung an das V-Modell sind für den Test mehrere Stufen vorgesehen, die Dokumente Test Design, Test Case, Test Protokoll, Test Log, Interim Test Status Report und Anomaly Report werden daher bei Bedarf jeweils für die Stufen Component, Component Integration, System und Acceptance Test erstellt.

Zur Unterstützung des Testmanagers existiert eine Vorlage für einen **Master Test Plan Metrics Report**, dieser schließt Metriken nach **ISO 9126** ein.

Eine Anpassung des Testdokumentationsprozesses an die eigene Organisation (Projekt bzw. Unternehmen) ist dabei wie bei anderen Normen ausdrücklich gewünscht, solange die Anforderungen an eine nachvollziehbare Testdokumentation erfüllt werden. Die Norm soll als Richtschnur und Orientierung dienen, die Ausgestaltung und Umsetzung ist individuell unterschiedlich. Dabei kommt es immer wieder darauf an, keine starren Vorgaben zu definieren, sondern den etablierten Abläufen und Umweltvoraussetzungen so weit wie möglich entgegenzukommen, ohne die begründeten Basisanforderungen zu vernachlässigen.

Über die Testdokumentation hinaus befasst sich der Standard IEEE 829 auch mit der Einbindung des Tests in die Organisation. Er definiert und beschreibt folgende Formen: (ISTQB konform)

- eingegliedert: komplett integriert im Entwicklerteam
- intern: Tester im Entwicklerteam
- integriert: eigenes Team parallel zu den Entwicklern in derselben Organisationseinheit

- eigenständig: eigene selbstständige Organisationseinheit innerhalb des Unternehmens
- unabhängig: komplett unabhängig, außerhalb des Unternehmens, extern vergeben [ROBU2019]

Dabei werden auch Mischformen angetroffen, etwa wenn der Integrationstest einer zugelieferten Komponente zwar an eine externe Organisation ausgelagert ist, der Systemtest und Abnahmetest aber von einer bestimmten Abteilung innerhalb des Unternehmens organisiert und durchgeführt wird.

Literatur

- [DIN2019] : <https://www.din.de/de/ueber-normen-und-standards/basiswissen>, zugegriffen am 09.05.2020
- [QYTE2019] : <https://www.qytera.de/blog/testkonzept-ieee-829-testmanagement>, zugegriffen am 09.05.2020
- [SETH2019] : http://www.sebastianthiem.de/?page=Workshop/004_SoftwareTest.html, zugegriffen am 09.05.2020
- [PMQS2019] : <https://pmqs.de/index.php/testmanagement/prozesse-tm/32-04-ergebnistypen-im-test>, zugegriffen am 09.05.2020
- [ROBU2019] : <http://robert.bullinger.over-blog.com/article-testdokumentation-nach-ieee-829-20510038.html>, zugegriffen am 09.05.2020