Vorwort zur 7. Auflage

Testen & Agilität

Im Jahr 2002 erschien die erste Auflage dieses Buches. Ein Jahr zuvor wurde das »Agile Manifest« publiziert. Sowohl das Buch als auch das agile Vorgehen in der Softwareentwicklung haben in den beiden zurückliegenden Jahrzehnten eine bemerkenswerte Entwicklung zu verzeichnen gehabt: »Basiswissen Softwaretest« hat sich als meistverkauftes Buch zum Thema Softwaretest in deutscher Sprache als Standardwerk etabliert und die agile Vorgehensweise ist zu »der« Praktik in der IT-Industrie avanciert. Da war es naheliegend, dass in der Neuauflage des Buches – und des Lehrplans – viele der agilen Praktiken, die das Testen betreffen, aufgenommen wurden.

Testkompetenz ist weiterhin erforderlich.

Softwareentwicklung ist zur Teamarbeit geworden. Während früher einzelne Aufgaben von Fachleuten (Designer, Entwickler, Tester, …) umgesetzt wurden und jeder nur für seinen Bereich verantwortlich war, ist heute die Zusammenarbeit im Team gefragt. »Den« Tester und »den« Entwickler im engeren Sinne gibt es nicht mehr. Das Fachwissen und die Kompetenzen werden aber weiterhin benötigt, sind jedoch nicht mehr direkt an einzelne Personen gebunden. Im Buch wird an einigen Stellen noch von »Tester« und »Entwickler« gesprochen, gemeint ist aber eine Person mit Test- bzw. Entwicklungskompetenz, was sich etwas sperrig liest.

In der vorliegenden 7. Auflage des Buches haben wir den Inhalt umfassend überarbeitet, aktualisiert und an die aktuelle Version 4.0 des Lehrplans zum »ISTQB® Certified Tester – Foundation Level« aus dem Jahr 2023 angepasst.

»Certified Tester«-Ausbildungsschema

Das anerkannte und sehr erfolgreiche »Certified Tester«-Ausbildungsschema besteht aus den Ausbildungsstufen »Foundation«, »Advanced« und »Expert«. Ergänzt wird es durch Module für die Arbeit in agilen Teams sowie durch Spezialisierungsmodule. Details dazu finden sich auf den Webseiten des »International Software Testing Qualifications Board« [URL: ISTQB] und des »German Testing Board e.V.« ([URL: GTB], [URL: GTB Schema]). Der »Certified Tester« hat sich zu einem

Gütesiegel in der IT-Industrie entwickelt und ist heute der De-facto-Standard für die Ausbildung im Bereich Softwarequalitätssicherung und Softwaretest – in Deutschland und weltweit.

Beeindruckende Zahlen

»*Die Weiterbildung zum Certified-Tester ist international erfolgreich. Über 118.000 absolvierte Softwaretester-Prüfungen in Deutschland (Stand 09/2023) sind gute Gründe, stolz zu sein. Weltweit sind es mehr als 1.200.000 (Stand 06/2023)*« (aus [URL: GTB Schema]).

Damit hat sich die Anzahl der Prüfungen in den letzten fünf Jahren nahezu verdoppelt: 2018 waren es weltweit über 600.000, davon knapp 67.000 in Deutschland.

Wissen in der IT-Welt und an den Hochschulen gefragt

»*Studierenden und Auszubildenden wird durch das Angebot des GTB aktuelles, von der Industrie gefordertes Wissen vermittelt: Der wachsende Anteil an Stellenausschreibungen, in denen ein ISTQB® Certified Tester Zertifikat explizit gefordert wird, beweist eine bereits jetzt vorhandene hohe Durchdringung des Marktes. Zudem haben in den letzten Jahren knapp 600 Studierende in Deutschland erfolgreich das Certified Tester Zertifikat abgelegt*« (aus [URL: GTB Hochschulen]).

Der »Certified Tester« hat sich im Laufe der Jahre zu einem festen Bestandteil der Informatikausbildung an vielen Hochschulen entwickelt: Von A wie Aachen bis Z wie Zittau wird der Lehrstoff im deutschsprachigen Raum vermittelt. Welche Hochschulen aktuell entsprechende Lehrveranstaltungen anbieten oder planen, kann auf den Seiten des GTB nachgelesen werden [URL: GTB Hochschulen].

Ergänzende Literatur

Tilo und Andreas haben zwei weitere Bücher veröffentlicht, auf die hier hingewiesen werden soll, da sie eine gute Ergänzung bzw. Vertiefung darstellen. Andreas – als Norddeutscher – würde sagen: »Butter bei die Fische«, denn beide Bücher vertiefen die im Basiswissen-Buch vorhandenen Beschreibungen zur Agilität bzw. zu Testverfahren.

Buch: Testen in agilen Projekten

In seinem aktuellen Buch »Testen in agilen Projekten – Methoden und Techniken für Softwarequalität in der agilen Welt« (3. Auflage, 2024) zeigt Tilo, wie das Testen in agile Projekte integriert werden kann. Das Buch deckt damit die ISTQB®-Lehrpläne zum agilen Testen ab.

Buch: Lean Testing für C++-Programmierer

Andreas hat zusammen mit Ulrich Breymann[1] das Buch »Lean Testing für C++-Programmierer – Angemessen statt aufwendig testen« geschrieben. In dem Buch werden alle für den Komponententest relevanten Testverfahren des ISO-Standards 29119 ausführlich beschrieben. Die Vorgehensweisen zum Testfallentwurf werden konkret mit den entspre-

1. Ulrich Breymann ist ehemaliger Professor an der Hochschule Bremen und Autor des C++-Standardwerks »Der C++-Programmierer«.

chenden C++-Programmtexten und den zugehörigen Testfällen dargestellt. Dabei sind die Programmbeispiele so gehalten, dass sie auch ohne C++-Kenntnisse verständlich sind.

Auf beide Bücher wird in diesem Buch immer wieder verwiesen ([Linz 24], [Spillner 16]), um den Leserinnen und Lesern weiterführende Informationen zu bieten. Vielleicht sind wir mit den vielen Verweisen etwas über das Ziel hinausgeschossen, dafür bitten wir um Nachsicht – aber ein wenig Werbung für die eigene Arbeit wird hoffentlich noch erlaubt sein, oder?

Basiswissen

Von Anfang an haben wir den ersten Teil unseres Buchtitels »Basiswissen« ernst genommen und bewusst keine Themen behandelt, die sich in der Praxis erst noch »beweisen müssen«. Auch »Spezialdisziplinen« im Testen, wie beispielsweise der Test von Webapplikationen oder der Test von eingebetteten Systemen, gehören für uns nicht zu den Grundlagen. Hier verweisen wir auf die entsprechende aktuelle Literatur zu diesen und anderen Themen.

Was hat sich geändert?

Aber auch Grundlagen unterliegen dem Wandel und sind aktuell zu halten. Ebenso haben die etablierten agilen Praktiken mit Bezug zum Softwaretest ihren Weg in das Buch gefunden. In der vorliegenden 7. Auflage wurden die Inhalte umfassend überarbeitet, ergänzt und aktualisiert.

Folgende Themen wurden neu aufgenommen oder ausführlicher behandelt:

- Whole-Team-Ansatz
- Test-First-Ansatz
- Shift-Left
- Retrospektive
- Testen im Kontext von DevOps
- Abnahmetestgetriebene Entwicklung (ATDD)
- User Stories
- Akzeptanzkriterien
- Iterations- und Releaseplanung bei agilen Projekten
- Testpyramide und Testquadrant

Exkurse sind nicht Teil des Lehrplans.

Bei der Überarbeitung des ISTQB®-Lehrplans wurden einige Themen auf höhere Ausbildungsstufen verschoben oder ganz weggelassen und sind somit nicht mehr Bestandteil des »Foundation Level«-Lehrplans. Wir haben diesen Schritt nicht strikt umgesetzt, sondern uns entschieden, die für die Praxis wichtigen Teile im Buch zu belassen. Diese sind als Exkurse farblich hervorgehoben (blaue Schrift). Wer das Buch nur zur Prüfungsvorbereitung nutzt, kann die Exkurse einfach überspringen.

Standard-Nachschlagewerk

Aus vielen Gesprächen mit Leserinnen und Lesern wissen wir, dass unser Buch als Nachschlagewerk für die tägliche (Test-)Arbeit genutzt wird. Deshalb haben wir neben den Inhalten des Lehrplans weitere grundlegende Testverfahren – als Exkurse – aufgenommen (z.B. kombinatorisches Testen, »Pairwise Testing«).

Das durchgehende Fallbeispiel und das Literaturverzeichnis wurden aktualisiert. Das Verzeichnis der Normen und Standards wurde ebenfalls überarbeitet. Die Angaben zu den Internetseiten (URLs) sind auf den aktuellsten Stand gebracht worden.

Webseite

Um die Leserinnen und Leser über zukünftige Aktualisierungen des Lehrplans und des Glossars zu informieren, betreiben wir die Internetseite »Softwaretest Knowledge« [URL: Softwaretest Knowledge]. Auf dieser Seite werden auch notwendige Korrekturen zum Buchtext aufgeführt. Neben Übungsaufgaben zu den einzelnen Buchkapiteln findet sich dort auch eine Cross-Referenz-Tabelle, in der zu jedem Lernziel des Lehrplans die entsprechenden Abschnitte des Buches aufgeführt sind, in denen das Lernziel ausführlich behandelt wird.

Ebenfalls sind dort das Vorwort zur 1. Auflage des Buches und die Geleitworte von Dorothy Graham, David Parnas und Martin Pol zu finden.

Danksagung

Erfolg hat meist viele Väter und Mütter – so auch hier. Allen Kolleginnen und Kollegen des GTB und des ISTQB® sei an dieser Stelle herzlich gedankt. Ohne ihr Engagement hätte das »Certified Tester«-Ausbildungsschema nicht den geschilderten Erfolg und die weltweite Akzeptanz erreicht. Ebenso möchten wir uns für die vielen Anmerkungen und Rezensionen unserer Leserinnen und Leser bedanken, die uns für unsere Arbeit am Buch sehr motiviert und zur Qualitätssteigerung beigetragen haben. Unserer Lektorin Christa Preisendanz und dem gesamten dpunkt.team danken wir für die langjährige und sehr gute Zusammenarbeit.

Wir wünschen allen Leserinnen und Lesern gutes Gelingen bei der Umsetzung der im Buch beschriebenen Testansätze in die Praxis und – falls das Buch als Grundlage für die Vorbereitung auf die Prüfung zum »Certified Tester – Foundation Level« dient – viel Erfolg bei der Beantwortung der Prüfungsfragen.

Andreas Spillner und *Tilo Linz*
Bremen, Möhrendorf
März 2024

1 Einleitung

Software ist allgegenwärtig! Es gibt kaum noch Geräte, Maschinen oder Anlagen, in denen die Steuerung nicht über Software bzw. Softwareanteile realisiert wird. So sind im Automobil wesentliche Funktionen wie die Motor- oder Getriebesteuerung seit Langem durch Software realisiert. Hinzu kommen immer intelligentere softwarebasierte Fahrerassistenzsysteme, vom Bremsassistenten über die automatische Einparkhilfe oder Spurassistenten bis zum vollständig autonom fahrenden Fahrzeug. Systeme mit künstlicher Intelligenz finden zurzeit eine rasend schnelle Verbreitung und wirken sich bereits heute in ganz vielen Bereichen unseres Lebens aus. Die Software – und besonders deren Qualität – trägt somit ganz entscheidend nicht nur zum Funktionieren unserer Welt bei, sondern definiert zunehmend auch unsere Sicherheit.

Ebenso ist der reibungslose Geschäftsablauf in Firmen und Organisationen heute weitgehend von der Zuverlässigkeit der Softwaresysteme abhängig, die zur Abwicklung der Geschäftsprozesse oder einzelner Aufgaben eingesetzt werden. Software entscheidet damit auch über die künftige Wettbewerbsfähigkeit der Unternehmen. Wie schnell beispielsweise ein Versicherungskonzern ein neues Produkt oder auch nur einen neuen Tarif am Markt einführen kann, ist heutzutage davon abhängig, wie schnell die konzerneigenen IT-Systeme entsprechend angepasst oder ausgebaut werden können.

Hohe Abhängigkeit vom reibungslosen Funktionieren der Software

In beiden Bereichen (technische und kommerzielle Softwaresysteme) ist die Qualität der Software damit zum entscheidenden Faktor für den Erfolg von Produkten und Unternehmen geworden.

Die meisten Unternehmen haben diese hohe Abhängigkeit von Software – sowohl vom Funktionieren der vorhandenen als auch von der schnellen Verfügbarkeit neuer oder besserer Software – erkannt. Sie investieren daher in ihre Softwareentwicklungskompetenz und in eine verbesserte Qualität ihrer Softwaresysteme. Ein wichtiges Mittel, dies zu erreichen, ist das systematische Prüfen und Testen der entwickelten Software. Teilweise haben sehr umfassende und rigide Verfahren Einzug in die Praxis der Softwareentwicklung gefunden. In vielen Projekten ist aber weiterhin ein erheblicher Bedarf an Wissensvermittlung zu Prüf- und Testverfahren und deren Leistungsfähigkeit und Nutzen erforderlich.

Grundlagenwissen zum strukturierten Prüfen und Testen

Mit diesem Buch stellen wir Grundlagenwissen bereit, das bei entsprechender Umsetzung zu einem strukturierten, systematischen Vorgehen beim Prüfen und Testen führt und somit zur Qualitätsverbesserung der Software beiträgt. Der Inhalt des Buches ist so abgefasst, dass kein Vorwissen im Bereich der Softwarequalitätssicherung vorausgesetzt wird. Das Buch ist als Lehrbuch konzipiert und auch zum Selbststudium geeignet. Ein durchgängiges Fallbeispiel hilft, jedes dargestellte Thema und seine praktische Umsetzung schnell zu verstehen.

Ansprechen möchten wir Softwaretester[1] in allen Unternehmen und Organisationen, die ihre Softwaretestkenntnisse auf eine fundierte Grundlage stellen wollen, Programmierer und Entwickler sowie Mitglieder in agilen Teams, die Testaufgaben übernommen haben bzw. übernehmen werden, aber auch Softwaremanager, die in den Projekten über Verbesserungsmaßnahmen und Budgets entscheiden. Auch Quereinsteiger in entwicklungsnahen IT-Berufen und Mitarbeiter in Fachabteilungen, die an der Abnahme, Einführung oder Weiterentwicklung von IT-Anwendungen beteiligt sind, werden Hilfestellung für ihre tägliche Arbeit finden.

Das lebenslange Lernen ist besonders im IT-Bereich unverzichtbar. Auch zum Thema Softwaretest werden Weiterbildungsmaßnahmen von vielen Firmen und Trainern angeboten. Ebenso werden an immer mehr Hochschulen Lehrveranstaltungen zu diesem Thema durchgeführt. Das Buch soll Lernende und Lehrende im gleichen Maße ansprechen.

Zertifizierungsprogramm für Softwaretester

Der weltweite Standard für die Aus- und Weiterbildung im Bereich Software-Qualitätssicherung und Softwaretest ist heute das »ISTQB® Certified Tester«-Schema des International Software Testing Qualifications Board (ISTQB®). Das ISTQB® [URL: ISTQB] koordiniert die nationalen Initiativen und sorgt für die Einheitlichkeit und Vergleichbarkeit der Lehr- und Prüfungsinhalte unter den beteiligten Ländern. Die nationalen Testing Boards sind zuständig für die Herausgabe und Pflege landessprachlicher Lehrpläne und für die Definition und Durchführung von Prüfungen in ihren jeweiligen Ländern. Sie überprüfen die im jeweiligen Land angebotenen Kurse nach definierten Qualitätskriterien und sprechen Akkreditierungen der Trainingsanbieter aus. Die Testing Boards gewährleisten damit einen hohen Qualitätsstandard der Kurse, und die Kursteilnehmer erhalten mit bestandener Prüfung einen international anerkannten Qualifikationsnachweis. Die entsprechenden Gremien im deutschsprachigen Raum sind das Austrian Testing Board [URL: ATB], das German Testing Board [URL: GTB] und das Swiss Testing Board [URL: CHTB]. In diesen Gremien sind Trainings-

1. Wir verwenden im Buch überwiegend die männliche Form und wollen damit Frauen sowie alle anderen Geschlechter selbstverständlich nicht ausschließen bzw. ausgrenzen.

anbieter, Testexperten aus Industrie- und Beratungsunternehmen sowie Hochschullehrende organisiert. Wichtige Kompetenz bringen weiterhin Vertreter verschiedener Fachverbände ein. So arbeiten im GTB u. a. Mitglieder der Fachgruppe TAV (Test, Analyse und Verifikation von Software) [URL: GI TAV] der Gesellschaft für Informatik e.V. (GI e.V.) mit.

Dreistufiges Qualifizierungsschema

Das »Certified Tester«-Ausbildungsschema besteht aus den Ausbildungsstufen »Foundation«, »Advanced« und »Expert«. Es wird ergänzt um Module für die Arbeit in agilen Teams sowie Spezialistenmodule. Details dazu finden sich auf der Webseite des ISTQB® [URL: ISTQB] und des GTB [URL: GTB]. Die Grundlagen zum Softwaretest – sowohl bei klassischer als auch bei agiler Entwicklung – sind im Lehrplan zum »Foundation Level« beschrieben. Darauf aufbauend kann das Zertifikat zum »Advanced Level« [URL: GTB Lehrpläne] erworben werden, um vertiefte Kenntnisse im Prüfen und Testen nachzuweisen. Ergänzend bietet das Schema »Specialist Module«, die spezielle oder domänenspezifische Methoden, Techniken und Prozesse vermitteln. Beispiele sind »Sicherheitstester«, »Usability Testing« und »Certified Tester AI Testing«. Die dritte Stufe, das »Expert Level«-Zertifikat, richtet sich an erfahrene, professionelle Softwaretester und umfasst die Module »Improving the Test Process« und »Test Management«.

Der Inhalt des Buches deckt den Stoff des Zertifikats »Foundation Level« ab. Das prüfungsrelevante Fachwissen kann im Selbststudium erworben oder nach bzw. parallel zu einer Teilnahme an einem Kurs vertieft werden.

Kapitelübersicht

Die Themen des Buches und somit auch die grobe Struktur der Inhalte der Kurse zum Erwerb des »Foundation Certificate« sind im Folgenden beschrieben.

Grundlagen des Softwaretestens

In Kapitel 2 werden die Grundlagen des Softwaretestens erörtert. Neben der Motivation, wann, mit welchen Zielen und wie intensiv getestet werden soll, wird das Konzept eines grundlegenden Testprozesses beschrieben. Es wird auch auf erforderliche Kompetenzen beim Testen eingegangen.

Testen im Softwarelebenszyklus

Kapitel 3 stellt in der Softwareentwicklung gebräuchliche Lebenszyklusmodelle (sequenziell, iterativ-inkrementell, agil) kurz vor und erläutert, welche Rolle das Testen im jeweiligen Modell spielt. Wichtige Elemente agiler Vorgehensweisen (Backlogs, Whole-Team-Ansatz, User Stories, Continuous Integration etc.) werden erläutert und es wird aufgezeigt, welchen Einfluss sie auf die Gestaltung der Testaktivitäten haben. Die verschiedenen Teststufen und Testarten werden ausführlich erklärt und auf die Unterschiede beim funktionalen und nicht funktionalen Test eingegangen. Das Thema Regressionstest wird ebenfalls angesprochen. Wichtige Ansätze zur Verbesserung und Automatisierung

der Softwareentwicklung und des Testprozesses (CI/CD, frühes Testen/Shift-Left, testgetriebene Entwicklung, DevOps) werden kompakt vorgestellt und erläutert.

Statischer Test

Statische Verfahren, d.h. Verfahren, bei denen das Testobjekt nicht zur Ausführung kommt, werden in Kapitel 4 vorgestellt. Reviews und statische Tests werden bereits in vielen Unternehmen mit gutem Erfolg angewendet. Die unterschiedlichen Vorgehensweisen werden ausführlich beschrieben.

Dynamischer Test

Das Kapitel 5 behandelt den Test im engeren Sinne. Die Klassifizierung der dynamischen Testverfahren in »Blackbox«- und »Whitebox«-Verfahren wird erörtert. Zu jeder Klasse werden unterschiedliche Testverfahren bzw. -methoden an Beispielen ausführlich erklärt. Auf die sinnvolle Verwendung des erfahrungsbasierten bzw. intuitiven Tests, nämlich in Ergänzung zu den anderen Verfahren, wird am Ende des Kapitels eingegangen.

Testmanagement

Welche Organisationsformen, Rollen und Aufgaben beim Testmanagement zu berücksichtigen sind und welche Anforderungen an die Qualifikation der Mitarbeiter bestehen, wird in Kapitel 6 diskutiert. Welche Elemente zu einer Teststrategie gehören und wie diese durch eine fundierte Testplanung, Teststeuerung und Testüberwachung umgesetzt wird, wird ausführlich erklärt. Wichtige Verfahren zur Schätzung von Testaufwand und Testkosten werden vorgestellt und es wird erläutert, welchen Beitrag das Testen leistet, um Risiken zu mindern, und wie risikobasiertes Testen funktioniert. Abschließend werden Anforderungen an das Fehlermanagement und Konfigurationsmanagement sowie das Thema Wirtschaftlichkeit des Testens besprochen.

Testwerkzeuge

Testen von Software ist ohne Werkzeugunterstützung sehr arbeits- und zeitintensiv. Im letzten Kapitel des Buches (Kap. 7) werden unterschiedliche Klassen von Werkzeugen zur Testunterstützung vorgestellt und Hinweise zur Werkzeugauswahl und Werkzeugeinführung gegeben.

Fallbeispiel »VirtualShowRoom – VSR-II«

Die in diesem Buch vorgestellten Vorgehensweisen beim Testen von Software werden größtenteils anhand eines durchgängigen Fallbeispiels veranschaulicht. Das folgende Szenario liegt diesem Beispiel zugrunde:

Ein Automobilkonzern betreibt seit mehr als zehn Jahren ein elektronisches Verkaufssystem, genannt VirtualShowRoom (VSR). Dieses Softwaresystem ist weltweit bei allen Autohändlern der Marke installiert und in Betrieb:

- Ein Kunde, der ein Fahrzeug erwerben möchte, kann unterstützt durch einen Verkäufer oder selbstständig sein Wunschfahrzeug am Bildschirm konfigurieren (Modellauswahl, Farbe, Ausstattung usw.).

Das System zeigt mögliche Modelle und Ausstattungsvarianten an und ermittelt zu jeder Auswahl sofort den jeweiligen Listenpreis. Diese Funktionalität wird vom Teilsystem *DreamCar* realisiert.

- Hat sich der Kunde für ein Fahrzeug entschieden, kann er am Bildschirm mit *EasyFinance* die für ihn optimale Finanzierung kalkulieren, mit *JustInTime* das Fahrzeug online bestellen und mittels *NoRisk* auch die passende Versicherung abschließen. Das Teilsystem *FactBook* schließlich verwaltet sämtliche Kundeninformationen und Vertragsdaten.

Der Konzernbereich Marketing und Vertrieb hat entschieden, dass das System modernisiert werden soll und die folgenden Projektziele definiert:

- VSR ist ein klassisches Client-Server-System. Das neue System VSR-II soll ein webbasiertes System sein, das auf beliebigen Geräten (Desktop, Tablet, Smartphone) über Browser genutzt werden kann.
- Jedes der bisherigen Teilsysteme *DreamCar*, *EasyFinance*, *FactBook*, *JustInTime, NoRisk* wird auf die neue Technologie portiert und in diesem Zuge auch (in unterschiedlichem Umfang) funktional erweitert.
- Als neues System ist das Teilsystem *ConnectedCar* anzubinden. Dieses System ermittelt und verwaltet Statusinformationen aller verkauften Fahrzeuge und gibt dem Fahrer aber auch dem Händler oder Servicepartner Informationen über anstehende Wartungs- und Reparaturarbeiten. Außerdem bietet es dem Fahrer verschiedene buchbare Services (wie Helpdesk, Notruf etc.). Auch Softwareupdates für das Fahrzeug können »Over the air« eingespielt und freigeschaltet werden.
- Jedes der fünf alten Teilsysteme wird von einem eigenen Entwicklungsteam separat portiert und weiterentwickelt. Ein weiteres Team kümmert sich um die Neuentwicklung von *ConnectedCar*. Insgesamt sind rund 60 Entwickler und weitere Mitarbeiter aus den jeweils betroffenen konzerninternen Fachabteilungen an dem Projekt beteiligt sowie externe Softwarefirmen.
- Die Teams arbeiten agil nach Scrum. Im Rahmen der agilen Entwicklung soll jedes Teilsystem während der Iterationen getestet werden. Die Auslieferung des Systems erfolgt in Inkrementen.
- Um einen komplizierten mehrfachen Abgleich von Daten zwischen Altsystem und Neusystem zu vermeiden, ist vorgesehen, dass VSR-II erst dann erstmalig produktiv in Betrieb geht, wenn die Funktionalität des alten VSR erreicht ist.

Im Rahmen des Projekts und des agilen Vorgehens werden die meisten Projektmitarbeiter in unterschiedlichem Umfang mit Testarbeiten konfrontiert oder betraut werden. Das Grundlagenwissen über Testtechniken und Vorgehensweisen, das sie dazu benötigen, wird in diesem Buch vermittelt. Abbildung 1–1 zeigt das neue System VSR-II in der Übersicht.

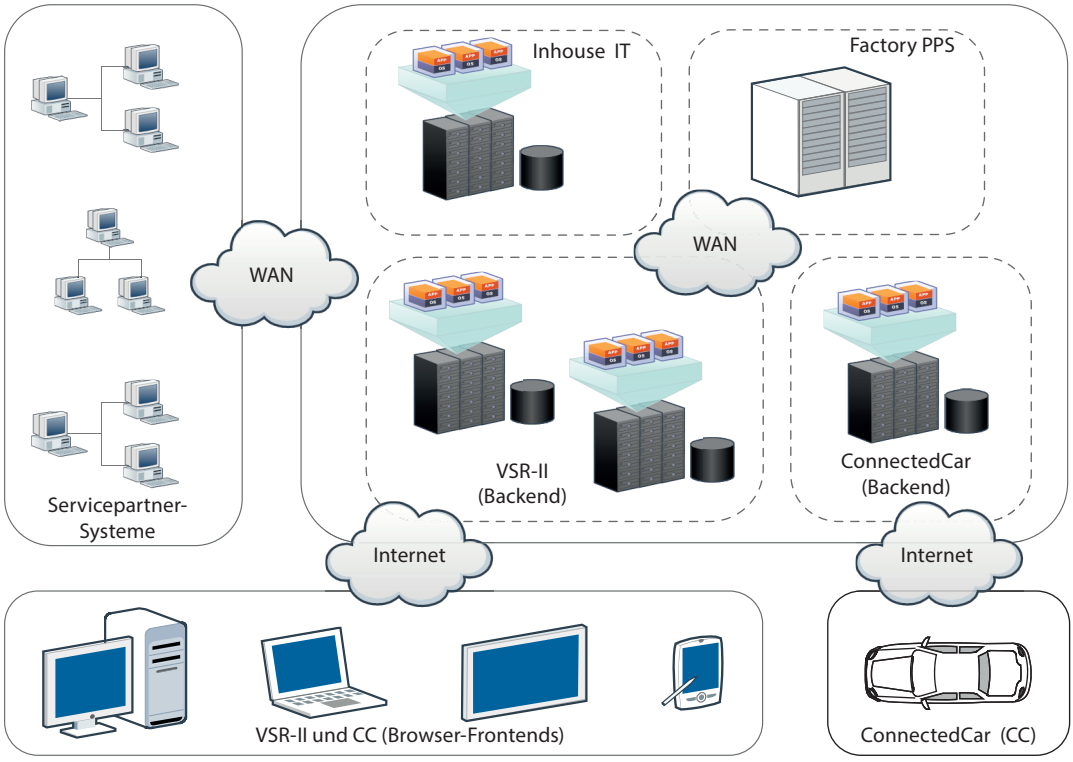

Abb. 1–1 *VSR-II in der Übersicht*

Hinweise zu Lehrstoff und Prüfung

Im Anhang werden wichtige Hinweise zum Lehrstoff und zur Prüfung zum Certified Tester gegeben. Weitere Anhänge des Buches beinhalten ein Glossar und das Quellenverzeichnis. Textpassagen, die über den Stoff des Lehrplans hinausgehen, sind als »**Exkurs**« gekennzeichnet.

WWW-Seite zum Buch

Unter [URL: Softwaretest Knowledge] finden sich neben Übungsaufgaben zu den einzelnen Buchkapiteln weitere und aktuelle Informationen zum Buch wie auch zu anderen Büchern der Autoren, die das Certified-Tester-Ausbildungsschema ergänzen.

4 Statischer Test

Statische Prüfungen und Analysen von Arbeitsergebnissen (Dokumente und Programmtext) tragen sehr wirksam zur Qualitätssteigerung bei. Das allgemeine Vorgehen wird ebenso beschrieben wie der Prozess, der einzuhalten ist, mit seinen Aktivitäten und zu besetzenden Rollen. Vier in der Praxis verbreitete Vorgehensweisen werden erörtert. Erfolgsfaktoren und Vorteile werden in diesem Kapitel vorgestellt. Am Ende werden die Unterschiede zwischen statischen und dynamische Tests dargelegt.

Häufig unterschätzte Prüfmethode

Eine oft unterschätzte Prüfmethode ist der sogenannte statische Test, der häufig auch als statische Analyse oder statische Prüfung bezeichnet wird und sowohl werkzeuggestützt als auch manuell durchgeführt werden kann. Während beim dynamischen Test (s. Kap. 5) das Testobjekt ein ausführbares Programm ist, das mit Testdaten versehen und auf einem Rechner ausgeführt wird, kann der statische Test auf jede Art von Arbeitsergebnis bzw. Dokument, das relevant für die Erstellung der Software ist, angewendet werden. Mithilfe der statischen Analyse können bereits vor dem dynamischen Testen Probleme aufgedeckt werden. Da keine Testfälle spezifiziert und ausgeführt werden, ist die statische Analyse mit weniger Aufwand verbunden – vor allem beim Einsatz von Werkzeugen.

Die Analyse kann in Form einer intensiven Betrachtung durch eine oder mehrere Personen als »strukturierte Gruppenprüfung« erfolgen – dieses wird allgemein als Review (s. u.) bezeichnet – oder durch entsprechende Analysewerkzeuge. Die werkzeuggestützte Analyse wird in den Abschnitten 4.6 und 7.1.3 kurz vorgestellt.

Grundidee Prävention

Grundidee der Reviews ist die Prävention: Fehler(zustände)[1] und Abweichungen (Anomalien) sollen erkannt werden, noch bevor sie im weiteren Verlauf der Softwareentwicklung negativ zum Tragen kommen. Alle wichtigen Arbeitsergebnisse sollen qualitätsgesichert werden, bevor mit ihnen weitergearbeitet wird. Damit soll verhindert werden, dass fehlerhafte Zwischenergebnisse, Festlegungen oder Aussagen Eingang in die weiteren Entwicklungsschritte finden und dann dort aufwendige Nachbesserungen erforderlich machen.

1. Die Schreibweise soll sowohl Fehlerzustände im Programmtext als auch Fehler in anderen Arbeitsergebnissen umfassen.

Neben den Zielen der Verbesserung der Qualität und der Aufdeckung von Fehler(zustände)n gehört auch die Bewertung von Merkmalen wie Lesbarkeit, Vollständigkeit, Korrektheit, Testbarkeit und Konsistenz zu den Ergebnissen der Reviews. Aus den Merkmalen ergibt sich eine Bewertung über die Wartbarkeit. Reviews können sowohl zur Verifizierung als auch zur Validierung herangezogen werden.

Reviews sind in vielen Projekten integrierter Bestandteil des Entwicklungsprozesses, d.h., Dokumenten- und Codereviews werden regelmäßig eingeplant und durchgeführt. Bei sicherheitskritischen Projekten wie beispielsweise in der Luft- und Raumfahrt oder im medizinischen Bereich sind Reviews besonders wichtig, um die angestrebte hohe Qualität zu gewährleisten. Ebenso können Aspekte der IT-Sicherheit mittels Reviews beurteilt werden.

4.1 Was kann analysiert und geprüft werden?

Vielfältige Einsatzmöglichkeiten

Während der Entwicklung von Software entsteht neben dem Programmcode eine ganze Reihe von Arbeitsergebnissen. Meist dient jedes Dokument als Grundlage für die weiteren Aktivitäten der Softwareentwicklung und somit ist die Qualität jedes einzelnen Dokuments wichtig für die Gesamtqualität der erstellten Software.

Arbeitsergebnisse, die durch Reviews geprüft werden können, sind jede Art von Spezifikation, einschließlich Geschäftsanforderungen, funktionale und nicht funktionale Anforderungen und Sicherheitsanforderungen. Fehler in Spezifikationen sollen gefunden und behoben werden, bevor diese in Programmtext umgesetzt werden.

Bei agiler Entwicklung sind beispielsweise Epics [URL: Epic] und User Stories sowie Product-Backlog-Elemente und Test-Chartas Gegenstand der Reviews. Auch kann die Zusammenarbeit von Vertretern des Fachbereichs, Entwicklern und Testern beim Example Mapping[2], beim gemeinsamen Schreiben von User Stories und bei der Verfeinerung (Refinement) des Backlogs durch Reviews unterstützt werden. Es wird geprüft, ob die User Stories und die zugehörigen Arbeitsergebnisse definierten Kriterien entsprechen, z.B. der Definition of Ready. Reviews stellen somit sicher, dass die User Stories vollständig und verständlich sind und testbare Abnahmekriterien enthalten.

Fehler(zustände) und Unstimmigkeiten in den Anforderungen, die aufgedeckt werden können, sind unter anderem Inkonsistenzen, Mehrdeutigkeiten, Widersprüche, Lücken, Ungenauigkeiten und Redundanzen sowie auch Rechtschreibfehler.

2. Example Mapping ist eine Technik, um ein gemeinsames Verständnis zum Umfang und Inhalt von User Stories zu erhalten.

Während des Entwurfs von Softwaresystemen entstehen Architektur- und Entwurfsspezifikationen (z. B. Klassendiagramme). Diese Modelle können ebenso mittels Reviews geprüft werden wie der Programmcode. Auch der Code von Webseiten kann Gegenstand der Untersuchungen sein.

Für alle Dokumente, Festlegungen und Hilfsmittel, die beim Testen entstehen (z. B. Testkonzepte, Testfälle, Testabläufe und Testskripte, sowie Abnahme- bzw. Akzeptanzkriterien), ist eine Prüfung ebenso sinnvoll. Weitere Dokumente bzw. Arbeitsergebnisse, die untersucht werden können, sind Verträge, Projektpläne, Zeit- und Budgetpläne sowie Benutzeranleitungen.

Nicht statisch prüfbar

Sind Dokumente für Personen schwer oder gar nicht zu interpretieren, eignen sie sich in der Regel auch nicht für die Durchführung eines Reviews. Auch rechtliche Gründe können entgegenstehen, z. B. ausführbarer Code von Drittanbietern, der nicht analysiert werden darf.

Darüber hinaus können die Ergebnisse der Reviews dazu dienen, auch den Entwicklungsprozess selbst zu verbessern. Treten Fehler häufig bei bestimmten Arbeitsschritten auf, ist der Entwicklungsprozess bei diesen Arbeitsschritten zu untersuchen und ggf. zu optimieren, meist begleitet durch Schulungsmaßnahmen der Teammitglieder.

4.2 Vorgehen beim Review

Menschliche Analyse- und Denkfähigkeit

Die menschliche Analyse- und Denkfähigkeit wird genutzt, um komplexe Sachverhalte zu prüfen und zu bewerten. Dies erfolgt durch intensives Lesen und Nachvollziehen der untersuchten Dokumente. Voraussetzung für den Erfolg der Prüfung ist, dass die beteiligten Personen das untersuchte Dokument verstehen und die enthaltenen Aussagen und Festlegungen inhaltlich nachvollziehen können. Oft sind Reviews die einzige Möglichkeit, die Semantik der Dokumente und Arbeitsergebnisse zu überprüfen.

Es gibt verschiedene Vorgehensweisen, die sich durch die Intensität und die benötigten Ressourcen (Personen und Zeit) und die verfolgten Ziele unterscheiden. Im Weiteren werden die unterschiedlichen Verfahren zu Reviews[3] näher erläutert.

3. Die Bezeichnungen der Verfahren richten sich nach den Begriffen im ISTQB®-Lehrplan und der ISO-Norm 20246 [ISO/IEC 20246]. Ausführliche Beschreibungen von Reviews sind in [Rösler 13] zu finden.

»Review« hat viele unterschiedliche Bedeutungen.

Der Begriff »Review[4]« wird für Unterschiedliches verwendet: sowohl als Bezeichnung für ein Vorgehen bei der statischen Analyse von Arbeitsergebnissen als auch als gebräuchlicher Oberbegriff für alle verschiedenen statischen Prüfverfahren, die von Personen durchgeführt werden. In diesem Sinne wurde der Begriff »Review« bisher verwendet. Ein weiterer Begriff, in meist analoger Bedeutung verwendet, ist Inspektion. Unter »Inspektion« wird jedoch häufig eine formale Durchführung eines statischen Tests (mit der Sammlung von Metriken/Daten) nach definierten Regeln verstanden.

Unterschiedliches Vorgehen bei Reviews

Reviews können von informell bis hin zu sehr formal durchgeführt werden. Bei den informellen Reviews wird kein definierter Prozess befolgt und auch das zu erzielende und zu dokumentierende Ergebnis der Untersuchung ist nicht festgelegt. Bei den formalen Reviews sind die am Review beteiligten Personen (s. Abschnitt 4.3.3) ebenso festgelegt wie die zu dokumentierenden Ergebnisse. Der Prozess zur Durchführung der formalen Reviews ist ebenfalls definiert (s. Abschnitt 4.3.1). Welche Ausprägung des Prozesses zur Durchführung eines Reviews genutzt werden soll bzw. kann, hängt von folgenden Faktoren ab:

- Softwareentwicklungslebenszyklus-Modell – Welches Modell wird genutzt und welche »Stellen« und Arbeitsergebnisse im Modell eignen sich für die Durchführung von Reviews?
- Reife des Entwicklungsprozesses – Wie hoch ist der Reifegrad des Entwicklungsprozesses und wie hoch ist damit die Qualität der Dokumente, die geprüft werden sollen?
- Komplexität des zu überprüfenden Dokuments – Welche Arbeitsergebnisse weisen eine hohe Komplexität auf und sind einem – eher formalen – Review zu unterziehen?
- Gesetzliche oder regulatorische Anforderungen und/oder die Notwendigkeit eines Prüfnachweises (»Audit-Trail«) – Welche Vorschriften sind einzuhalten und welche Maßnahmen zur Qualitätssicherung – somit auch Reviews – sind nachzuweisen?

Unterschiedliche Ziele

Auch die gewünschten resultierenden Ziele eines Reviews sind entscheidend für die Wahl des Reviewprozesses bzw. der Reviewart (s. Abschnitt 4.4). Steht das Finden von Fehler(zustände)n oder steht das gemeinsame Verständnis des untersuchten Arbeitsergebnisses im Mittelpunkt? Sollen neue Teammitglieder in ein bestimmtes Dokument eingearbeitet werden

4. Kritische Besprechung eines [künstlerischen] Produkts o.Ä. (Definition von *www.duden.de*).

und damit sich erforderliches Wissen aneignen? Soll ein Review durchgeführt werden, um nach einer Diskussion zu einer Konsensentscheidung zu kommen? Je nach Beantwortung der Fragen ist die passende Reviewart auszuwählen und durchzuführen.

4.3 Der Reviewprozess

In der Norm ISO/IEC 20246 [ISO/IEC 20246] wird ein generischer Reviewprozess definiert, der einen strukturierten, aber flexiblen Rahmen bietet. Der Reviewprozess kann auf eine bestimmte Situation zugeschnitten werden – von formal bis informell. Bei einem formalen Review sind mehrere der unten beschriebenen Aufgaben für die verschiedenen Aktivitäten erforderlich.

Sind Arbeitsergebnisse sehr umfangreich und können daher nicht in einem einzigen Review vollständig geprüft werden, so kann der Reviewprozess mehrfach durchgeführt werden, um ein Reviewergebnis für das gesamte Arbeitsergebnis zu erhalten.

Aktivitäten im Reviewprozess

Der ISTQB®-Reviewprozess zur Prüfung von Arbeitsergebnissen umfasst die Planung, den Reviewbeginn, das individuelle Review durch die teilnehmenden Personen (Gutachter, Reviewer, Inspektoren), die Kommunikation und Analyse sowie die Behebung der aufgedeckten Fehler(zustände) und die Berichterstattung (s. Abb. 4–1).

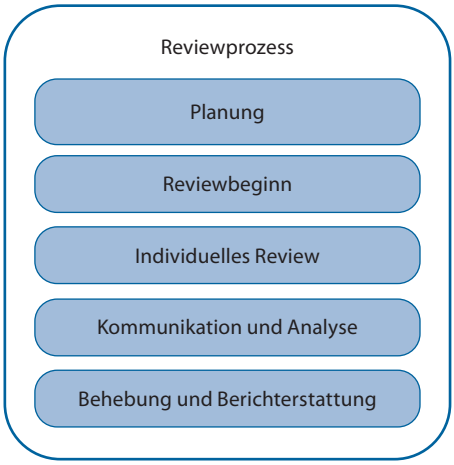

Abb. 4–1 *Aktivitäten im Reviewprozess*

4.3.1 Aktivitäten im Reviewprozess

Die oben aufgeführten Hauptaktivitäten des Reviewprozesses werden im Folgenden beschrieben.

Planung

Reviewart und Ziele festlegen

In der Planung ist vom Management – genauer von der Projektleitung – zu entscheiden, welche Dokumente oder Teile von Dokumenten welcher Art von Review (s. Abschnitt 4.4) unterzogen werden sollen. Mit der Entscheidung für eine Reviewart sind die zu besetzenden Rollen (s. Abschnitt 4.3.3), die durchzuführenden Aktivitäten und ggf. die Nutzung von Checklisten verbunden. Festzulegen ist ferner, welche Qualitätsmerkmale bewertet werden sollen. Der zu veranschlagende Aufwand und der benötigte Zeitrahmen für die einzelnen Reviews sind bei der Planung einzubeziehen. Der Manager bzw. die Projektleitung wählt die fachlich geeigneten Personen aus, teilt die Rollen zu und stellt ein Reviewteam zusammen. In Kooperation mit dem Autor des zu untersuchenden Arbeitsergebnisses vergewissert man sich, dass dieses einen reviewfähigen Status hat, d.h. die Arbeiten am Dokument einen gewissen Abschluss gefunden und eine ausreichende Vollständigkeit erreicht haben.

Bei formalen Reviews sind Eingangskriterien und die entsprechenden Endekriterien (oder auch Austrittskriterien) in der Planungsphase zu definieren. Sind Eingangskriterien vorhanden, dann ist die Einhaltung der Kriterien zu prüfen, bevor das Review weiter durchgeführt wird.

Unterschiedliche Sichtweisen erhöhen den Erfolg.

Ein Review ist meist erfolgreicher, wenn das untersuchte Dokument von unterschiedlichen Standpunkten aus betrachtet wird oder nur bestimmte Aspekte (von den einzelnen Personen) untersucht werden. Die dabei zu berücksichtigenden Sichtweisen bzw. Aspekte sind bei der Planung festzulegen.

Ebenso kann entschieden werden, dass nicht das gesamte Arbeitsergebnis einem Review unterzogen wird. Es können nur solche Teile ausgewählt werden, bei denen Fehler(zustände) ein hohes Risiko beinhalten, oder nur beispielhafte Teile, um daraus auf die Qualität des gesamten Dokuments zu schließen.

Findet eine Vorbesprechung statt, sind Ort und Zeit festzulegen.

Initiierung des Reviews

Die Initiierung des Reviews (oder auch Reviewbeginn[5], Einführung, Vorbesprechung, Initialisierung, Kick-off) dient zur Versorgung der am Review beteiligten Personen mit allen benötigten Informationen (physisch oder elektronisch). Dies kann in Form einer schriftlichen Einladung erfolgen, oder ein erstes Treffen des Reviewteams wird durchgeführt, um

über Bedeutung, Sinn und Zweck sowie über die Ziele des durchzuführenden Reviews zu informieren. Sind die beteiligten Personen mit dem Umfeld des zu prüfenden Dokuments wenig vertraut, kann auf dem Treffen neben der kurzen Vorstellung des Prüfdokuments auch seine Einbettung in das Anwendungsgebiet dargestellt werden.

Prüfgrundlagen sind notwendig.

Neben dem Arbeitsergebnis, das einem Review unterzogen werden soll, müssen den beteiligten Personen weitere Unterlagen zur Verfügung stehen. Dazu gehören die Dokumente, die herangezogen werden müssen, um zu entscheiden, ob eine Abweichung, ein Fehler(zustand) oder eine korrekte Beschreibung des Sachverhalts vorliegt. Dies sind die Dokumente (z. B. Use Cases, Designdokumente, Richtlinien oder Standards), gegen die geprüft wird. Diese Dokumente werden auch als Basisdokumente (»Baseline«) bezeichnet. Darüber hinaus sind Prüfkriterien (z. B. in Form von Checklisten) sehr sinnvoll, die ein strukturiertes Vorgehen unterstützen. Sind Vorlagen zur Protokollierung der Befunde einzusetzen, dann sind diese zu verteilen. Weiter sollen sich alle Teilnehmer am Review über ihre Rollen und ihre Verantwortlichkeiten im Klaren sein.

Bei formalen Reviews ist die Einhaltung der Eintrittskriterien zu prüfen. Bei deren Nichterfüllung ist das Review abzubrechen. Dadurch wird Zeit gespart, die sonst durch Prüfung von »unreifen« Arbeitsergebnissen verloren geht.

Individuelles Review (individuelle Vorbereitung)

Intensive Auseinandersetzung mit dem Prüfobjekt

Die beteiligten Personen des Reviewteams haben sich individuell auf das Review vorzubereiten. Nur bei einer guten Vorbereitung aller Personen ist ein erfolgreiches Review überhaupt möglich.

Die Reviewer oder Gutachter – die Reviewteilnehmer, die das Dokument einer intensiven Prüfung unterziehen – setzen sich intensiv mit dem zu prüfenden Arbeitsergebnis auseinander und verwenden dabei die zur Verfügung gestellten Dokumente als Prüfgrundlage. Erkannte potenzielle Fehler(zustände), Empfehlungen, Fragen oder Kommentare werden notiert.

Für die individuelle Vorbereitung gibt es eine Reihe von unterschiedlichen Verfahren, die im nächsten Abschnitt 4.3.2 (als Exkurs) näher erörtert werden.

---

5. Reviewbeginn ist der im Lehrplan (Version Sept. 2023) verwendete Begriff. Wir finden diesen Begriff nicht sehr passend, da er leicht mit dem Beginn einer Reviewsitzung verwechselt werden kann.

Kommunikation und Analyse (Reviewsitzung)

Zusammentragen der Befunde

Nach der individuellen Vorbereitung werden die Ergebnisse zusammengetragen und diskutiert. Dies kann bei einer Reviewsitzung erfolgen oder auch auf andere Weise, z. B. in einem firmeninternen Onlineforum. Die von den einzelnen Reviewern identifizierten potenziellen Abweichungen und Fehler(zustände) werden besprochen und näher analysiert. Zuständigkeiten für die Behebung und ggf. erneute Kontrolle (z. B. ein Folgereview) werden ebenso festgelegt wie deren Status (s. Abschnitt 6.4.4).

Bei der Planung wurde festgelegt, welche Qualitätsmerkmale untersucht werden sollen. In der Befundanalyse wird nun jedes Merkmal bewertet und das Ergebnis dokumentiert.

Vom Reviewteam ist eine Empfehlung über die Annahme des Arbeitsergebnisses abzugeben:

- akzeptieren (ohne Änderungen) bzw. akzeptieren (mit geringfügigen Änderungen),
- Überarbeitung erforderlich, da umfangreiche Änderungen notwendig,
- nicht akzeptieren.

Exkurs: Empfehlungen für die Durchführung einer Reviewsitzung

Findet zur Diskussion eine Reviewsitzung statt, können folgende Empfehlungen gegeben werden, die sich in der Praxis bewährt haben:

- Die Reviewsitzung wird auf zwei Stunden beschränkt. Falls erforderlich wird eine weitere Sitzung frühestens am nächsten Tag einberufen.
- Der Moderator (s. u.) hat das Recht, eine Sitzung abzusagen oder abzubrechen, wenn einer oder mehrere Reviewer nicht erschienen oder ungenügend vorbereitet sind.
- Das Resultat (also das zu prüfende Dokument, das Arbeitsergebnis) und nicht der Autor steht zur Diskussion:
  - Die Reviewer müssen auf ihre Ausdrucksweise achten.
  - Der Autor darf weder sich noch das Resultat verteidigen müssen (d. h., der Autor wird keinen »Angriffen« ausgesetzt, die ihn zur »Verteidigung« zwingen; eine Rechtfertigung seiner Entscheidungen wird teilweise als legitim und hilfreich angesehen).
- Der Moderator darf nicht gleichzeitig als Reviewer fungieren.
- Allgemeine Stilfragen (außerhalb der Prüfkriterien) dürfen nicht diskutiert werden.
- Die Entwicklung und Diskussion von Lösungen ist nicht Aufgabe des Reviewteams (Ausnahmen s. u.) und soll unterbleiben.
- Jeder Reviewer muss Gelegenheit haben, seine Befunde angemessen präsentieren zu können.
- Der Konsens der Reviewer zu einem Befund ist anzustreben und zu protokollieren.
- Befunde sind nicht in Form von Korrekturanweisungen an den Autor zu formulieren. Vorschläge für mögliche Korrekturen oder Verbesserungen werden allerdings teilweise durchaus als sinnvoll und hilfreich für die Qualitätsverbesserung angesehen.

- Die einzelnen Befunde sind zu gewichten[6] als
  - kritischer Fehler (Arbeitsergebnis ist für den vorgesehenen Zweck unbrauchbar, Fehler(zustand) muss vor der Freigabe behoben werden),
  - Hauptfehler (Nutzbarkeit des Arbeitsergebnisses ist beeinträchtigt, Fehler(zustand) vor der Freigabe beheben),
  - Nebenfehler (geringfügige Abweichung, z.B. Rechtschreibfehler im Ausdruck, beeinträchtigt den Nutzen kaum),
  - gut (fehlerfrei, bei Überarbeitung nicht ändern).

Behebung und Berichterstattung

Fehler(zustände) korrigieren

Die abschließende Aktivität im Reviewprozess ist die Erstellung von Berichten und die Behebung der aufgedeckten Fehler(zustände) bzw. Unstimmigkeiten. Oft beinhaltet das Protokoll der Reviewsitzung alle erforderlichen Informationen, sodass keine zusätzlichen Fehlerberichte angefertigt werden müssen, die jeden Fehler(zustand) einzeln dokumentieren, der eine Änderung des untersuchten Arbeitsergebnisses erfordert. Im Regelfall wird der Autor die Fehler(zustände) in seinem Arbeitsergebnis auf Grundlage der Reviewergebnisse beseitigen und eine Überarbeitung vornehmen.

Neben den Fehlerberichten können gefundene Fehler(zustände) auch direkt an die zuständige Person oder das zuständige Team übermittelt werden. Hierbei ist kommunikatives Geschick erforderlich, da sich kaum eine Person über die Mitteilung von Fehlern in den eigenen Arbeitsergebnissen freut (s. Abschnitt 2.4).

Formale Reviews verlangen mehr.

Bei formalen Reviews ist der aktuelle Status der Fehler(zustände) bzw. der dazugehörigen Fehlermeldung festzustellen (s. Abschnitt 6.4.4). Dabei ist ein Übergang von einem Status zu einem Folgestatus ggf. nur mit der Zustimmung des jeweiligen Reviewers möglich. Darüber hinaus ist das erfolgreiche Erreichen der Endekriterien/Austrittskriterien zu prüfen.

Eine umfassende Auswertung der jeweiligen Reviewsitzungen und deren Ergebnisse soll bei formalen Reviews dazu genutzt werden, den Reviewprozess zu verbessern und die verwendeten Dokumente (Richtlinien und Checklisten) den jeweiligen Gegebenheiten anzupassen und aktuell zu halten. Das Erfassen und die Auswertung von Daten (Metriken) sind hierzu notwendig.

Die Reviewergebnisse können in Abhängigkeit von der Reviewart und dem Grad der Formalität stark variieren, dies wird in Abschnitt 4.4 beschrieben. Zunächst werden die unterschiedlichen Vorgehensweisen beim individuellen Review erörtert und dann die Rollen und Verantwortlichkeiten bei den formaleren Reviews.

---

6. Siehe hierzu auch Abschnitt 6.4.3. Fehler der Klasse 2 und 3 können als Hauptfehler angesehen werden und die der Klasse 4 und 5 als Nebenfehler.

Exkurs

4.3.2 Unterschiedliche Vorgehensweisen beim individuellen Review

Die grundlegende Vorgehensweise beim Reviewprozess sieht eine individuelle Vorbereitung – ein individuelles Review – vor. Für diese Vorbereitung gibt es eine Reihe von Verfahren[7], die genutzt werden können, um Fehler(zustände) aufzudecken. Diese Verfahren können für alle beschriebenen Reviewarten (s. Abschnitt 4.4) zur Anwendung kommen. Die Wirksamkeit der Verfahren kann je nach Reviewart unterschiedlich sein. Nachfolgend sind Beispiele für verschiedene individuelle Vorgehensweisen aufgeführt.

Ad-hoc-Vorgehen

Keine Vorgaben

Wie der Name bereits nahelegt, sind keine Vorgaben gegeben, wie die individuelle Vorbereitung beim Ad-hoc-Review durchzuführen ist. Der Reviewer liest das Arbeitsergebnis meist sequenziell und erkennt Probleme, Befunde oder Fehler(zustände) und dokumentiert diese. Die Art der Dokumentation ist ebenso wenig vorgegeben. Ad-hoc-Reviews werden oft durchgeführt, da sie kaum bzw. keinerlei Vorbereitung erfordern. Da kein definiertes Vorgehen vorgeschrieben ist, ist das Ad-hoc-Review besonders stark von den Fähigkeiten des einzelnen Reviewers abhängig. Prüfen mehrere Reviewer ein Arbeitsergebnis, kommt es häufig vor, dass viele Probleme doppelt von den verschiedenen Reviewern gemeldet werden.

Vorgehen basiert auf Checklisten[8]

Nutzung von Checklisten

Zur Steigerung der Effektivität der individuellen Reviews können Checklisten eingesetzt werden, die ein strukturiertes Lesen unterstützen. Sie helfen bei der Aufdeckung von Unstimmigkeiten und Fehler(zuständen). Es kann festgelegt werden, welche Checkliste(n) von welchem Reviewer genutzt werden soll(en). Eine Review-Checkliste besteht aus einem Satz von Fragen, die auf potenziellen Fehler(zustände)n basieren. Die Fragen ergeben sich aus der Erfahrung, z. B. aus früheren Projekten. Die Fragen der Checkliste können sich auch auf einzuhaltende Formalitäten oder Standards beziehen, wie im folgenden Beispiel gezeigt.

---

7. Die hier beschriebenen Verfahren können auch als eigenständiges Review gesehen werden, wenn auf einen Reviewprozess ganz verzichtet wird.
8. Nicht mit dem in Abschnitt 5.3 beschriebenen erfahrungsbasierten Testverfahren zu verwechseln, das ebenfalls Checklisten verwendet, die aber zur Erstellung von Testfällen genutzt werden.

Beispiel: Checklisten

Mögliche Checklistenpunkte für ein Review eines Anforderungsdokuments:

- Liegt das Anforderungsdokument in einer standardisierten Struktur vor?
- Existiert ein einleitendes Kapitel, in dem beschrieben wird, wie das Anforderungsdokument zu benutzen ist?
- Gibt es eine zusammenfassende Beschreibung des Projekts, für das das Anforderungsdokument erstellt wurde?
- Ist ein Glossar zur Unterstützung des einheitlichen Verständnisses von Fachbegriffen vorhanden?
- …

Beispiele für weitere Checklisten sind zu finden unter [URL: Reviewtechnik].

Im Beispiel dient die Checkliste zur Prüfung eines Anforderungsdokuments. Für jede Art von Arbeitsergebnis ist das Anlegen und Verwenden einer spezifischen Checkliste sinnvoll, z.B. eine Checkliste für ein Codereview. Checklisten sind regelmäßig zu aktualisieren. Fragen zu neu erkannten Problemen sind aufzunehmen und Fragen zu veralteten Problemen sind zu streichen. Eine Checkliste soll keine allzu lange Liste sein und sich auf die wesentlichen Fragen beschränken.

Der Hauptvorteil des checklistenbasierten Reviews liegt in der systematischen Abdeckung typischer Fehlerarten. Ein Nachteil wird darin gesehen, dass die Checkliste das Review zu stark fokussiert und Fehler(zustände) »neben« den Checklistenfragen unerkannt bleiben. Es ist daher darauf zu achten, dass auch nach Fehler(zustände)n »außerhalb« der Checkliste gesucht wird.

Vorgehen unter Nutzung von Szenarien und »Probeläufen« (»Dry Runs«)[9]

Individuelles »Durchspielen« der späteren Nutzung

In einem szenariobasierten Review nutzen die Reviewer eine vorgegebene, strukturierte Richtlinie, aus der hervorgeht, wie sie das Arbeitsergebnis durchlesen oder »durchlaufen« sollen. Stehen Anwendungsfälle (»Use Cases«) zur Verfügung, gestaltet sich das individuelle, szenariobasierte Review recht einfach, indem die Anwendungsfälle »nachgespielt« werden. So kann beispielsweise im Projekt VSR-II nachvollzogen werden, ob die Anwendungsfälle »Fahrzeug zusammenstellen« und »Sondermodell auswählen« korrekt umgesetzt wurden (s.a. Abschnitt 5.1.6).

Solche Szenarien können auch auf Fehlerklassen beruhen, wenn Programmtext einem individuellen Review unterzogen werden soll. So kann ein Reviewer beim Codereview sich auf die Prüfung von Programmtext zur Fehlerbehandlung konzentrieren, ein anderer auf die Einhaltung von Feldgrenzen. Szenarien helfen dem Reviewer, bestimmte Fehlerarten identifizieren zu können, und der Reviewer ist bei diesem Vorgehen meist erfolgreicher als beim »Abarbeiten« einzelner Checklisteneinträge.

Eine ausschließliche Konzentration auf die Szenarien soll jedoch nicht erfolgen, um andere Fehlerarten (z.B. fehlende Merkmale) nicht zu übersehen.

---

9. Sehr ähnlich dem Walkthrough (s. Abschnitt 4.4), wobei dort das »Durchspielen« in der Gruppe erfolgt.

Rollenbasiertes und perspektivisches Vorgehen

Unterschiedliche Rollen und Perspektiven einnehmen

Nutzung von Fachwissen

Beim rollenbasierten Review hat der Reviewer den Auftrag, das Reviewobjekt aus der Sicht einer bestimmten fachlichen Rolle zu prüfen. Dazu muss er sich in die jeweilige Rolle ausreichend hineinversetzen können oder selbst in der betreffenden Rolle tätig sein. Grundidee des rollenbasierten Vorgehens beim Review ist die Nutzung des Fachwissens der jeweiligen Rollen. Dieses muss von den Personen, die die Rollen einnehmen, »mitgebracht« werden. Es ist beispielsweise sehr sinnvoll, Tester an den Reviews der Anforderungen zu beteiligen. So wird frühzeitig geprüft, ob die Anforderungen ausreichend detailliert und präzise formuliert sind, um daraus Testbedingungen und Testfälle abzuleiten.

Weitere typische Rollen sind die von Stakeholdern, wie z. B. Endanwendern oder den späteren Betreibern der Systeme in einer Organisation. Kann der Endanwender oder der Betreiber nicht am Review teilnehmen – was die Regel sein wird –, sind die Rollen von den am Review teilnehmenden Personen einzunehmen. Dabei kann die Rolle »Endanwender« auch weiter präzisiert werden, z. B. in erfahrener oder unerfahrener Nutzer. Wird das Softwaresystem in einer Organisation eingesetzt, kann beispielsweise die Rolle des Administrators (Benutzeradministrator, Systemadministrator) eingenommen werden.

Beispiel: Rollenbasiertes Review

Ein Anforderungsdokument soll einem Review unterzogen werden. Unter anderem wird Herrn Mustermann die Rolle des Designers zugeteilt. Idealerweise kennt sich Herr Mustermann mit dem Design von Softwaresystemen aus und hat diese Rolle auch im Projekt inne. Herr Mustermann wird die Anforderungen dahingehend analysieren, welche Änderungen an der Systemarchitektur notwendig werden, damit die Anforderung umgesetzt werden kann. Frau Musterfrau ist Testerin und so ist auch ihre Rolle im Review. Ihr Augenmerk beim Review wird beispielsweise sein, ob bei jeder Anforderung eindeutig entscheidbar ist, ob sie (nach Implementierung und Test) erfüllt ist. Die Anforderung »Das System soll zügiges Arbeiten ermöglichen« wird von Frau Musterfrau bemängelt, da diese Anforderung nicht quantifiziert ist. Es fehlen konkrete Zeitangaben, die gemessen werden können, und es fehlen Randbedingungen und Voraussetzungen, in welchen Situationen welche Zeiten als ausreichend angesehen werden. Andere Rollen sind entsprechend einzunehmen und werden ebenfalls die Anforderungen kritisch hinterfragen.

Unterschiedliche Standpunkte

Alle unterschiedlichen Rollen weisen so auf mögliche Fehler, Unstimmigkeiten und Lücken hin. Beim perspektivischen Lesen, das dem rollenbasierten Review ähnelt, nehmen die Reviewer unterschiedliche Standpunkte der Stakeholder ein. Damit sollen unterschiedliche Aspekte (je nach Sichtweise) berücksichtigt werden. Auch sollen so die (durchaus subjektiven) Interessen der Stakeholder einbezogen werden. Das folgende Beispiel illustriert solche unterschiedlichen Interessen beim Review des Projektzeitplans für die anstehende Iteration.

Beispiel: Perspektivisches Review

Aus Sicht der Verkaufsabteilung muss ein neues Feature im Anforderungsdokument eine hohe Priorität erhalten, da dieses Feature von einem der Großkunden dringend gewünscht ist und daher schnellstmöglich umgesetzt werden soll. Aus der Perspektive der Testerin hat dieses Feature große Ähnlichkeiten zu anderen bereits umgesetzten Features und somit sind keine gravierenden Probleme bei der Umsetzung und beim Test zu erwarten. Der Wunsch nach kurzfristiger Umsetzung wird daher befürwortet.

Aus Testsicht ist jedoch ein anderes Feature viel kritischer, da hier erstmalig die Nutzung von künstlicher Intelligenz (KI) vorgesehen ist und im Projekt bislang noch keinerlei Erfahrungen mit dieser Technologie vorliegen. Daher geht die Testerin davon aus, dass neben einem hohen Implementierungs- auch ein sehr hoher Testaufwand erforderlich sein wird. Für dieses Feature muss aus der Testperspektive daher deutlich mehr Aufwand eingeplant werden.

Kernidee beim perspektivischen Vorgehen ist, dass jeder Reviewer aus einer unterschiedlichen Perspektive auf das Reviewobjekt schaut und damit verbunden andere Szenarien durchdenkt oder »durchspielt«. Die erzielten Erkenntnisse werden nicht nur zur Bewertung des untersuchten Dokuments herangezogen, sondern können auch genutzt werden, um Dringlichkeit und Umfang von Korrekturmaßnahmen abzustimmen. Jeder Reviewer »spielt« dabei ein anderes Szenario[10] durch und nutzt die Ergebnisse zur Einschätzung und Bewertung des untersuchten Dokuments. Die Nutzung unterschiedlicher Perspektiven der Stakeholder führt zu mehr Tiefe im individuellen Review und durch die verteilten Rollen (und auch die unterschiedlichen Perspektiven) zu weniger Doppelnennungen von Problemen bzw. Fehler(zustände)n. Die Nutzung von Checklisten erhöht auch hier die Effektivität des Reviews.

Empirische Studien (s. [Sauer 00], [Shull 00]) sehen das perspektivische Review als effektivstes Vorgehen zur Prüfung von Anforderungen und technischen Beschreibungen. Ein wesentlicher Erfolgsfaktor ist die risikoorientierte Einbeziehung und Abwägung verschiedener Sichtweisen der Stakeholder.

4.3.3 Rollen und Verantwortlichkeiten im Reviewprozess

Einzunehmende Rollen[11] und deren Verantwortlichkeiten im Reviewprozess gehen aus der Beschreibung des allgemeinen Vorgehens bereits grob hervor. Im Folgenden werden die Rollen und ihre Aufgaben bei einem typischen formalen Review genauer beschrieben.

Nicht jede Rolle muss auch von einer Person ausgefüllt werden, es gibt Überschneidungen bei den Verantwortlichkeiten, die ein Zusammenlegen von Rollen auf eine Person rechtfertigen (z. B. Moderator und Reviewleiter). Es kommt aber auf das Projektumfeld und die zu erreichen-

10. Hier gibt es Querbezüge zum oben beschriebenen szenariobasierten Review.
11. Die ISO-Norm 20246 [ISO/IEC 20246] führt weitere, detailliertere Rollen auf.

den Qualitätskriterien an, wann ein Zusammenlegen sinnvoll ist. Je nach Reviewart (s. Abschnitt 4.4) können auch die einzelnen Aktivitäten variieren, die mit jeder Rolle verknüpft sind.

Management[12]

Management

Das Management sorgt für die Auswahl der zu prüfenden Arbeitsergebnisse und die heranzuziehenden Dokumente und entscheidet über die durchzuführende Reviewart. Es ist verantwortlich für die Planung der Reviews und stellt die notwendigen Ressourcen (Personen, Budget und Zeit) zur Verfügung. Eine weitere Aufgabe des Managements ist die Überwachung der Kosteneffizienz und das Treffen von steuernden Entscheidungen im Fall von unzureichenden Ergebnissen des Reviewprozesses.

Exkurs

Vertreter des Managements (Projektleitung) sollen, wenn sie zum Autor in einer Vorgesetzten- oder ähnlichen Beziehung stehen, nicht an den Reviewsitzungen teilnehmen, da eine Bewertung der Qualifikation des Autors (und nicht eine Bewertung des Arbeitsergebnisses) durch das Management nicht auszuschließen ist und eine »freie« Diskussion unter den Teilnehmenden dann möglicherweise eingeschränkt wird. Ein weiterer Grund kann sein, dass Managern unter Umständen das aktuelle detaillierte IT-Fachwissen fehlt und sie in dieser Hinsicht weniger inhaltliche Diskussionsbeiträge auf der Sitzung beisteuern können. Für Reviews von Projektplänen und ähnlichen managementlastigen Dokumenten trifft dieses natürlich nicht zu. Hier ist Managementwissen bzw. Projektleitungswissen gefragt.

Reviewleiter

Reviewleiter

Der Reviewleiter trägt die Gesamtverantwortung für das Review, d.h., Planung, Vorbereitung, Durchführung und Überarbeitung sowie Nachbereitung sollen so erfolgen, dass die Ziele des Reviews erreicht werden. Er entscheidet, welche Personen am Review beteiligt sind, und organisiert Zeitpunkt und Räumlichkeiten, wenn eine Sitzung stattfindet.

Reviewmoderator (Moderator oder Facilitator)

Moderator

Der Moderator hat die Aufgabe, den erfolgreichen Ablauf der Reviewsitzung sicherzustellen. Der Erfolg eines Reviews – oder genauer einer Reviewsitzung – hängt oft vom Moderator ab. Er muss ein sehr guter Sitzungsleiter sein, d.h. sehr organisiert und gleichzeitig diplomatisch vorgehen bei der Vermittlung von gegensätzlichen Standpunkten und bei der Beendigung von unnützen Diskussionen, ohne dabei Beteiligte zu verletzen oder zu kränken. Gleichzeitig muss er über eine gewisse Durch-

12. Hier ist das Projektmanagement (die Projektleitung) gemeint.

setzungsfähigkeit verfügen und auf »Untertöne« achten. Er darf nicht voreingenommen sein und keine eigene Meinung zum untersuchten Arbeitsergebnis äußern. Er sorgt ggf. für die Sammlung der Daten (Metriken) und achtet auf die Erstellung des Protokolls.

Autor

Autor

Der Autor ist der Ersteller des Dokuments bzw. des Arbeitsergebnisses, das einem Review unterzogen wird. Sind mehrere Personen an der Erstellung beteiligt gewesen, ist ein Hauptverantwortlicher zu benennen, der die Rolle des Autors übernimmt. Er ist verantwortlich dafür, dass die Eintrittskriterien vor dem Review erfüllt sind, d.h., dass sich das Dokument in einem »reviewfähigen« Zustand befindet. In aller Regel behebt der Autor die beim Review aufgedeckten Fehler(zu-stände) in seinem Arbeitsergebnis, was zur Erfüllung der Austrittskriterien führt. Wichtig ist, dass der Autor die geäußerte Kritik nicht als Kritik an seiner Person auffasst, sondern dass es ausschließlich um die Verbesserung der Qualität des Arbeitsergebnisses geht.

Reviewer

Reviewer

Die Reviewer, auch Gutachter oder Inspektoren genannt, sind mehrere (meist nicht mehr als fünf) Fachexperten, die nach der entsprechenden Vorbereitung am Review teilnehmen. Reviewer arbeiten in der Regel im Projekt, dessen Dokumente einem Review unterzogen werden. Es können aber auch Stakeholder sein, die Interesse an den Arbeitsergebnissen haben, und Personen mit spezifischen technischen oder fachlichen Kenntnissen.

Aufgabe der Reviewer ist das Identifizieren und Beschreiben der problematischen Stellen im Arbeitsergebnis. Von Vorteil ist, wenn sie dabei verschiedene Sichtweisen oder Perspektiven (s.o.) einnehmen, wie beispielsweise die Sichtweise von Testern, Programmierern, der späteren Nutzer und Betreiber, von Businessanalysten und Experten für Gebrauchstauglichkeit (»Usability«). Dabei sollen selbstverständlich nur die Sichten berücksichtigt werden, die einen direkten Bezug zum Arbeitsergebnis haben.

Um die Effizienz des Reviews zu erhöhen, sollen sich einzelne Reviewer um spezielle Aspekte kümmern. So kann beispielsweise ein Reviewer die Konformität mit einem einzuhaltenden Standard und ein anderer die Einhaltung der Syntax prüfen. Die Verteilung der Zuständigkeiten ist bei der Planung vorzunehmen.

Die Reviewer haben darauf zu achten, dass die als gut befundenen Teile der Arbeitsergebnisse auch so benannt werden. Mangelhafte Teile des Arbeitsergebnisses sind als solche zu kennzeichnen und die Fehler(zustände) sind so zu dokumentieren, dass die Nachvollziehbarkeit und damit deren einfache Beseitigung gegeben ist.

Protokollant

Protokollant

Der Protokollant hat die vom Reviewteam diskutierten und gefundenen Unklarheiten (z. B. noch zu lösende Probleme, neue Anomalien, durchzuführende Arbeiten, zu entscheidende Fragestellungen und Ablehnungen) zu dokumentieren. Er muss knapp und präzise formulieren und während der Sitzung meist nicht ausreichend formulierte Diskussionsbeiträge verfälschungsfrei aufschreiben können.

Durch die steigende Anzahl an Werkzeugen zur Unterstützung des Reviewprozesses wird die Rolle des Protokollanten zunehmend obsolet, insbesondere können Fehler(zustände), offene Punkte und Entscheidungen von jedem Reviewer direkt im Werkzeug vermerkt werden und es bedarf keiner zusätzlichen Erstellung eines Protokolls.

Exkurs: Autor und Protokollant in einer Person

Aus pragmatischen Gründen wird in der Praxis allerdings sehr oft der Autor selbst das Protokoll führen. Er weiß genau, was im Einzelnen und wie präzise und ausführlich die Anmerkungen der Reviewer zu protokollieren sind, um ausreichend Information für die später von ihm durchzuführenden Änderungen zu haben.

4.4 Reviewarten

Exkurs: Managementreview

Zwei Gruppen von Reviews lassen sich anhand des untersuchten Arbeitsergebnisses unterscheiden:

- Reviews, die sich auf Dokumente beziehen, die während des Entwicklungsprozesses erstellt werden, und
- Reviews, die den Projektablauf oder den Entwicklungsprozess als solches analysieren.

Die zweite Gruppe wird als Management-, Projekt- oder Prozessreview bezeichnet. Ziel von diesen Reviews ist die Analyse des Entwicklungsprozesses an sich. Es werden beispielsweise die Einhaltung von Vorgaben und Plänen, die Umsetzung der erforderlichen Arbeitsaufgaben oder die Effektivität der Verbesserungen bzw. Veränderungen im Prozess untersucht.

- Prüfobjekt ist das Projekt als Ganzes und die Feststellung seines aktuellen »Zustands«. Der Projektzustand wird hinsichtlich technischer, wirtschaftlicher, zeitlicher und managementmäßiger Aspekte bewertet.

- Managementreviews werden oft beim Erreichen eines Meilensteins der Projektplanung durchgeführt, wenn eine Hauptphase im Softwareentwicklungsprozess abgeschlossen werden soll, oder als Post-mortem-Analyse, um aus dem beendeten Projekt zu lernen.
- In Projekten, die nach agilen Vorgehensweisen arbeiten, kann dies in Form von »Retrospektive-Meetings« stattfinden. Diese werden in der Regel nach jedem Sprint durchgeführt und das Team tauscht sich beim Meeting aus und sammelt Ideen, was (im nächsten Sprint) verbessert werden kann.

Hauptziel: Fehler(zustände) aufdecken

Im Folgenden wird die erste Gruppe der Reviews weiter präzisiert, deren Hauptziel es ist, Fehler(zustände) aufzudecken, also Dokumente zu prüfen. Die Auswahl der Reviewart richtet sich nach den Bedürfnissen des Projekts, den verfügbaren Ressourcen, der Art des zu prüfenden Arbeitsergebnisses und den möglichen Risiken, dem Geschäftsbereich und auch nach der Unternehmenskultur sowie weiteren Auswahlkriterien.

Es können folgende Ausprägungen der Reviews unterschieden werden: informelles Review, Walkthrough, technisches Review und Inspektion. Alle Reviews können als sogenannte »Peer-Reviews« durchgeführt werden, d.h. unter der Beteiligung von Kollegen auf der gleichen oder einer ähnlichen Ebene der Unternehmenshierarchie.

Ein einzelnes Arbeitsergebnis kann auch durch mehr als eine Reviewart geprüft werden. So kann beispielsweise ein informelles Review vor einem technischen Review durchgeführt werden, um mit dem informellen Review nachzuweisen, dass das Arbeitsergebnis einen Bearbeitungsstand erreicht hat, der mit einem technischen Review geprüft werden kann und sich somit der Aufwand für das technische Review lohnt.

Informelles Review

Keine großen Vorgaben

Das informelle Review ist eine abgeschwächte Version eines Reviews, folgt aber mehr oder weniger dem Reviewprozess (s. Abschnitt 4.3), allerdings in stark vereinfachter und nicht formaler – also nicht vorgeschriebener – Form. Hauptzweck des informellen Reviews ist die Erkennung potenzieller Fehler(zustände) bzw. Anomalien und kurzfristiges Feedback an den Autor. Darüber hinaus können neue Ideen vorgestellt oder Lösungen entwickelt werden. Auch kleinere Probleme können im Rahmen dieses Reviews schnell gelöst und beseitigt werden.

Meist wird ein informelles Review durch den Autor initiiert. Die Planung beschränkt sich auf die Auswahl der Reviewer und die Festlegung des Abgabetermins der Ergebnisse. Der Erfolg und Nutzen des informellen Reviews ist vom jeweiligen Reviewer abhängig, besonders von seinen Kenntnissen und seiner Motivation. Die Verwendung von Checklisten ist möglich. Auf eine Sitzung oder einen Austausch der Ergebnisse zwischen den Reviewern wird oft verzichtet. In diesem Fall ist das informelle Re-

Autor-Leser-Feedbackzyklus

view ein einfacher Autor-Leser-Feedbackzyklus. Das informelle Review ist dann ein bloßes Gegenlesen des Arbeitsergebnisses durch ein oder mehrere Kollegen (Buddy-Check). Gegenseitiges Lernen und der Austausch unter Kollegen sind willkommene »Nebeneffekte«. Eine schriftliche Rückmeldung mit einer Liste der Anmerkungen oder das korrigierte und mit Kommentaren versehene Exemplar des Arbeitsergebnisses genügen meistens. Vorgehensweisen wie »Pair Programming« (paarweises Programmieren), »Buddy Testing« (gegenseitiger Test unter Kollegen) und »Code Swaps« (Austausch von Programmtext) können als informelle Reviews angesehen werden. Das informelle Review hat wegen des geringeren Aufwands eine hohe Akzeptanz und weite Verbreitung in der Praxis.

Walkthrough

»Durchspielen« des Dokuments

Neben dem Erkennen von potenziellen Fehler(zustände)n kann beim Walkthrough die Konformität mit einzuhaltenden Standards und den umzusetzenden Spezifikationen bewertet werden. Ziele sind die Beurteilung der Qualität und der Aufbau von Vertrauen in das untersuchte Arbeitsergebnis. Ebenso können Verbesserungen sowie alternative Umsetzungen diskutiert werden. Ideenaustausch über Verfahren oder Stilvariationen sind weitere »Nebeneffekte« und können als zusätzliche Qualifikation der Teilnehmer gesehen werden, deren Befähigung zur Verbesserung und zur Aufdeckung von Anomalien dadurch gefördert wird. Die Ergebnisse des Walkthrough sollen nach Möglichkeit im Konsens erzielt werden.

Schwerpunkt des Walkthrough bildet die Sitzung, die üblicherweise vom Autor des Arbeitsergebnisses geleitet wird. Die Ergebnisse sind aufzuschreiben. Formale Protokolle oder zusammenfassende Berichte der Reviewergebnisse sind nicht zwingend anzufertigen. Die Verwendung von Checklisten ist ebenso optional. Die individuelle Vorbereitung hat im Vergleich zum technischen Review und zur Inspektion (s. u.) den geringsten Umfang, es kann sogar ganz auf sie verzichtet werden.

In der Reviewsitzung werden typische Benutzungssituationen, auch Szenarien genannt, ablauforientiert durchgespielt. Simulationen oder Probeläufe von Programmteilen »im Trockenen« (»Dry Runs«) sind ebenfalls möglich. Es können auch einzelne Testfälle nachgespielt werden. Die Reviewer versuchen durch spontane Fragen, mögliche Fehler(zustände) und Probleme aufzudecken.

Für die Nacharbeit ist der Autor verantwortlich, eine Kontrolle findet meist nicht statt.

Exkurs

Das Vorgehen ist für kleine Entwicklungsteams von bis zu fünf Personen geeignet und verursacht wenig Aufwand, da Vor- und Nachbereitungen von geringem Umfang bzw. nicht zwingend erforderlich sind. Es lässt sich bei der Prüfung von »unkritischen« Dokumenten einsetzen. In der Praxis existiert eine breite Spannweite vom informellen bis zum formalen Walkthrough.

Da der Autor in der Regel auch die Sitzungsleitung innehat, besteht die Möglichkeit, dass er den Verlauf der Sitzung stark beeinflusst. Dies kann sich nachteilig auf das Ergebnis auswirken, wenn der Autor die Diskussion der kritischen Stellen des Arbeitsergebnisses nicht intensiv genug oder gar nicht durchführen lässt.

Technisches Review

Diskussion von Alternativen unter den Fachexperten erwünscht

Beim technischen Review[13] steht neben der Fehlerfindung die Konsensbildung im Mittelpunkt. Eine »gemeinsame Sicht auf die Dinge« verbunden mit einer Entscheidungsfindung in Bezug auf ein (technisches) Problem soll erreicht werden. Die Bewertung der Qualität und der Aufbau von Vertrauen in das Arbeitsergebnis sind auch hier die weiteren Ziele.

Beim technischen Review sind die Generierung neuer Ideen sowie Überlegungen zu alternativen Umsetzungen durchaus erwünscht. Fachliche Probleme können hier von den Fachexperten gelöst werden. Um die Diskussion darüber fundiert führen zu können, sollen fachliche Kollegen des Autors als Reviewer teilnehmen. Ergänzt werden kann das Reviewteam durch fachliche Experten aus anderen Fachdisziplinen, um einer gewissen »Blindheit« innerhalb der eigenen Disziplin vorzubeugen.

Die individuelle Vorbereitung der Reviewer ist beim technischen Review zwingend erforderlich. Checklisten können genutzt werden. Die anschließende Reviewsitzung ist von einem geschulten Moderator zu leiten. Der Autor soll diese Aufgabe nicht übernehmen. Die Diskussionen unter den Reviewern (die ja stattfinden soll und muss, zur Herstellung der »gemeinsamen Sicht auf die Dinge«) darf nicht ausarten, sondern soll dem Autor alternative Umsetzungen aufzeigen und ihn motivieren und befähigen, seine zukünftigen Arbeitsergebnisse durch die Abwägung von Alternativen zu verbessern. Die Durchführung einer Reviewsitzung ist allerdings nicht zwingend, der Austausch kann in Ausnahmefällen auch anderweitig erfolgen (z. B. in einem Forum im Intranet).

Ein Aufschreiben der Ergebnisse ist notwendig, soll aber nicht vom Autor vorgenommen werden. Üblicherweise werden eine Liste der Beschreibungen von potenziellen Fehler(zustände)n und ein zusammenfassender Bericht der Reviewergebnisse angefertigt.

13. Fachliches Review wäre eine allgemeinere und präzisere Bezeichnung, da hier Fachexperten über fachliche Fragen (die nicht ausschließlich technischer Art sein müssen) diskutieren.

Auch beim technischen Review sind in der Praxis sehr unterschiedliche Ausprägungen anzutreffen: von einem rein informellen Ablauf bis zu einem strikten, formal festgelegten Vorgehen (mit definierten Eintritts- und Austrittskriterien der einzelnen Prüfschritte) und der verbindlichen Nutzung von Berichtsvorlagen.

Exkurs

Eine auf die wichtigen Punkte fokussierte Reviewsitzung kann dadurch erreicht werden, dass alle gefundenen Fehler(zustände) und Anmerkungen der Reviewer zu den einzelnen Prüfkriterien schriftlich festgehalten werden und dem Moderator vorab übergeben werden. Der Moderator priorisiert diese nach vermuteter Wichtigkeit und in der Sitzung werden dann nur die wesentlichen und unterschiedlich gesehenen Anmerkungen diskutiert.

Beim technischen Review ist das Ergebnis von allen beteiligten Personen zu tragen. Ist dies nicht erreichbar und um die Diskussion nicht unnütz in die Länge zu ziehen, können Sondervoten zu Protokoll gegeben werden.

Es ist nicht Aufgabe der am technischen Review Beteiligten, über die Konsequenzen des Ergebnisses zu entscheiden, dies ist Aufgabe des Managements.

Inspektion

Formaler Ablauf festgelegt

Eine Inspektion ist die formalste Reviewart und folgt einem definierten Ablauf[14]. Jede beteiligte Person, die entweder aus dem direkten Fachkollegenkreis kommt oder Experte einer für das Arbeitsergebnis relevanten Fachrichtung ist, hat eine vorgeschriebene Rolle auszufüllen. Der Ablauf ist durch Regeln definiert und es werden zu den einzelnen Aspekten Prüfkriterien von den Reviewern verwendet. Für die einzelnen Prüfschritte sind Eintritts- und Austrittskriterien definiert.

Neben der Bestimmung der Qualität des Arbeitsergebnisses sowie dem Schaffen von Vertrauen in das Arbeitsergebnis ist das Hauptziel der Inspektion die maximale Aufdeckung von Unklarheiten und Fehler(zuständen)n. Konsens aller Beteiligten ist auch bei der Inspektion anzustreben. Durch die Erkenntnisse, die der Autor (und die Teilnehmenden) bei der Inspektion gewinnt, werden zukünftig ähnliche Fehler(zustände) verhindert und – wie beim technischen Review – die Arbeitsergebnisse zukünftig verbessert.

Ein weiteres Ziel ist die Steigerung der Qualität des Softwareentwicklungsprozesses (s. u.). Die konkreten Ziele der Inspektion werden bei der Planung festgelegt und es wird eine begrenzte Anzahl von Aspekten behandelt, auf die sich die Reviewer vorbereiten.

---

14. Sie ist daher hier auch ausführlicher als die drei anderen Reviewarten beschrieben. Der Ablauf einer Reviewsitzung wird beispielhaft erörtert.

Das Arbeitsergebnis wird vor der Inspektion formal geprüft, ob es reviewfähig ist und ob die Eintrittskriterien erfüllt sind. Die individuelle Vorbereitung der Reviewer erfolgt nach Vorschriften oder Standards unter der Verwendung von Checklisten.

Möglicher Ablauf einer Sitzung

Eine Sitzung kann wie folgt durchgeführt werden: Sie wird von einem geschulten Moderator (und nicht vom Autor) geleitet und beginnt neben der Vorstellung der beteiligten Personen und ihrer Rollen mit einer kurzen Einführung zur Thematik des Arbeitsergebnisses. Der Moderator fragt alle Reviewer, ob sie ausreichend auf das Treffen vorbereitet sind. Dazu können beispielsweise die für die Inspektion ausgewählten Checklisten genutzt werden, um sicherzustellen, dass alle Fragen der Checklisten beantwortet wurden und somit alle Reviewer ausreichend vorbereitet sind. Darüber hinaus kann nachgefragt werden, wie viel Zeit die Reviewer für ihre Vorbereitung verwendet und wie viele Fehler(zustände) und Unstimmigkeiten sie gefunden haben.

Unstimmigkeiten genereller Art, die das gesamte Arbeitsergebnis betreffen, werden zuerst diskutiert und protokolliert.

Ein Reviewer trägt in einer straffen und möglichst logischen Weise die Inhalte des Arbeitsergebnisses vor. Wenn es sinnvoll erscheint, werden auch Passagen – allerdings nicht vom Autor – vorgelesen. Die Reviewer stellen dabei ihre Fragen, wobei die ausgewählten Aspekte der Inspektion intensiv diskutiert werden. Der Autor, dem keine der Rollen Reviewleiter, Moderator und Protokollant zugeteilt werden darf, beantwortet an ihn gerichtete Fragen. Sind Autor und Reviewer uneins über einen Fehler(zustand), so wird am Ende der Sitzung darüber entschieden.

Wenn die Diskussion abschweift, ist es die Aufgabe des Moderators, einzugreifen. Er muss auch dafür sorgen, dass alle ausgewählten Aspekte und das gesamte Dokument berücksichtigt und die Fehler(zustände) und Unstimmigkeiten ausreichend protokolliert werden.

Am Ende der Sitzung werden alle aufgezeichneten Fehler(zustände) vorgestellt und von allen Beteiligten auf Vollständigkeit geprüft. Eine kurze Diskussion über die zurückgestellten Fehler(zustände) schließt sich an, ohne dabei Lösungsmöglichkeiten zu diskutieren. Gibt es keine Einigung über die Einschätzung (Fehler(zustand) ja oder nein), wird dies im Protokoll vermerkt. Neben der Liste der Beschreibungen von potenziellen Fehler(zustände)n wird ein zusammenfassender Bericht der Reviewergebnisse erstellt.

Eine abschließende umfassende Bewertung beendet die Inspektion und legt fest, ob eine Überarbeitung des Arbeitsergebnisses erforderlich ist. Die Überarbeitung und Nachbereitung sind bei der Inspektion formal geregelt.

Zusätzliche Bewertung der Entwicklungs- und Reviewprozesse

Bei einer Inspektion werden auch Metriken (Daten) gesammelt, die zur Qualitätsbeurteilung des Entwicklungs- und Inspektionsprozesses selbst herangezogen werden. Die Inspektion dient somit neben der Beurteilung der untersuchten Dokumente auch zur Optimierung des Entwicklungsprozesses. Die ermittelten Metriken werden dahingehend analysiert, die Ursachen für Schwachstellen im Entwicklungsprozess zu finden. Nach einer Verbesserung des Prozesses wird der Erfolg der Änderung kontrolliert, indem die gesammelten Metriken vor der Änderung mit den aktuellen Metriken verglichen werden. Dies trifft auch für den Reviewprozess selbst zu.

Exkurs

Häufig wird diese Reviewart auch als Design- oder Code-/Softwareinspektion bezeichnet. Der Name weist damit auf die Dokumente hin, die einer Inspektion unterzogen werden. Es können aber alle Dokumente einer Inspektion unterzogen werden, wenn formale Prüfkriterien existieren.

Kriterien zur Auswahl

Auswahl der Reviewart

Wann welche Reviewart eingesetzt werden soll, hängt stark von den verfolgten Zielen, der geforderten Qualität und dem einzusetzenden Aufwand ab. Das Projektumfeld ist entscheidend, direkte Empfehlungen können nicht gegeben werden. Es ist im Einzelfall zu klären, welche Reviewart in der jeweiligen Projektsituation angemessen ist. Nachfolgend sind einige Fragen und Kriterien zur Erleichterung der Auswahl der Reviewart aufgeführt:

- Die Form, in der das Ergebnis des Reviews vorliegen soll, kann zur Auswahl herangezogen werden. Ist eine ausführliche Dokumentation erforderlich oder reicht eine undokumentierte Umsetzung der Prüfergebnisse?
- Lässt sich die Terminkoordination einfach oder schwierig vornehmen? Fünf bis sieben Fachkräfte für einen oder mehrere Termine zu verpflichten, kann sehr aufwendig sein.
- Ist die Berücksichtigung von Fachwissen aus mehreren Gebieten erforderlich?
- Wie umfangreich muss das Fachwissen der Reviewer über das zu prüfende Arbeitsergebnis sein?
- Wie hoch ist das Engagement der vorgesehenen Reviewer, d. h. deren Motivation, die zeitaufwendige Reviewarbeit konzentriert zu leisten?
- Steht der Vorbereitungsaufwand in einem angemessenen Verhältnis zum erwarteten Ergebnis?
- Wie hoch ist der Grad der Formalisierung des Arbeitsergebnisses? Lassen sich werkzeuggestützte Analysen vorab durchführen?
- Wie groß ist die Unterstützung durch das Management? Wird bei zunehmendem Zeitdruck im Projekt die Durchführung von Reviews eingeschränkt oder gar ganz gestrichen?

4.5 Erfolgsfaktoren, Vorteile und Grenzen

Qualitätssteigerung und Kostensenkung

Reviews sind ein effizientes Mittel zur Sicherung der Qualität der untersuchten Arbeitsergebnisse. Idealerweise sind sie direkt nach der Fertigstellung der Arbeitsergebnisse durchzuführen, um Fehler(zustände) und Unstimmigkeiten kurzfristig festzustellen und frühes Feedback zu liefern. Reviews und die werkzeuggestützte statische Analyse (s. Abschnitt 4.6) erfüllen darüber hinaus den Grundsatz, dass frühes Testen Zeit und Kosten spart.

Die Beseitigung der Fehler(zustände) führt zu einer verbesserten Qualität der Dokumente und wirkt sich positiv auf den gesamten Entwicklungsprozess aus, da die Entwicklung mit Dokumenten fortgesetzt wird, die weniger oder sogar keine Fehler(zustände) mehr enthalten. Zusätzlich identifizieren statische Tests Fehler(zustände), die durch dynamische Tests nur schwer oder auch gar nicht zu finden sind (s. Abschnitt 4.7).

Das frühe Finden von Fehler(zustände)n durch Reviews und deren unmittelbar anschließendes Beseitigen ist in aller Regel kostengünstiger als das spätere Erkennen von Fehler(zustände)n in ablauffähigen Programmen durch dynamisches Testen. Dies trifft besonders dann zu, wenn (frühere) Versionen der Software bereits ausgeliefert wurden bzw. im Kundeneinsatz sind.

Der statische Test ist im Vergleich zum dynamischen Test (s. Kap. 5) auch deshalb kostengünstiger, weil die Fehlerkorrektur im untersuchten Dokument direkt vorgenommen wird und eine erneute Prüfung der Korrektur meist[15] nicht erfolgt. Eine beim dynamischen Test aufgedeckte Fehlerwirkung wird in aller Regel nach deren Beseitigung zur Ausführung von Fehlernach- und/oder Regressionstests führen. Entwicklungskosten und -zeit werden somit durch Reviews eingespart.

Allerdings ist diese pauschale Aussage zur Kostenreduktion nicht (immer) richtig bzw. zutreffend: Wenn eine werkzeuggestützte statische Codeanalyse z.B. ein Memory Leak (Fehlerzustand beim Speicherzugriff) findet, dann kann dessen Beseitigung extrem aufwendig sein. Es ist zwischen dem Aufwand bzw. den Kosten für die statische Analyse und dem Aufwand bzw. den Kosten für die Korrekturarbeiten zu unterscheiden. Ist nur ein Text in einem Dokument zu korrigieren, sind die Kosten fast zu vernachlässigen. Wie aufwendig eine Korrektur ist, hängt also vom zu beseitigenden Problem ab.

Der dynamische Test kann u.U. auch mit geringerem Umfang geplant werden, da nach einem durchgeführten Codereview weniger Fehlerzustände im Testobjekt (Programmcode) erwartet werden und das Risiko, weitere übersehen zu haben – dank Review –, als geringer eingestuft wird.

---

15. Bei gravierenden Fehler(zustände)n oder umfangreichen Korrekturen kann ein erneutes Review zur Prüfung der Änderungen anberaumt und durchgeführt werden.

Kostenreduzierung während der Lebenszeit des Systems

Durch die geringere Zahl an Fehler(zustände)n und Ungenauigkeiten in den geprüften und korrigierten Dokumenten ist auch eine Kostenreduzierung während der Lebenszeit des Softwaresystems zu erwarten. Beispielsweise können durch Reviews Ungenauigkeiten der Kundenwünsche in den Anforderungen aufgedeckt und geklärt werden. Absehbare Änderungswünsche nach der Inbetriebnahme des Softwaresystems können somit im Vorfeld verhindert werden. Es ist auch eine reduzierte Fehlerhäufigkeit im Einsatz des Systems zu erwarten. Hinzu kommt, dass sich die Entwicklungsproduktivität erhöht, da beispielsweise mit verbesserten Entwürfen gearbeitet wird und/oder der Programmcode wartungsfreundlicher ist, d.h. leicht zu verstehen und leicht zu ändern ist.

Verbesserte Kommunikation im Team

Neben Qualitätssteigerung und Kostensenkung haben Reviews positive Auswirkungen auf die Zusammenarbeit im Team:

- Da die Überprüfungen im Team durchgeführt werden, ergibt sich daraus ein Wissensaustausch unter den beteiligten Personen. Das wird zur Verbesserung der Arbeitsmethoden der einzelnen Personen führen und somit die Qualität der nachfolgenden Arbeitsergebnisse verbessern.
- Da mehrere Personen an einem Review beteiligt sind, ist eine klare und verständliche Darstellung der Sachverhalte notwendig. Oft bringt der Zwang zu einer klaren Darlegung den Autor bereits zu Einsichten, die er vorher nicht hatte.
- Das gesamte Team fühlt sich verantwortlich für die Qualität des untersuchten Dokuments und es entsteht ein einheitliches Verständnis über den Dokumentinhalt.
- Durch die Beteiligung der Stakeholder in Reviews von Anforderungsdokumenten kann sichergestellt werden, dass deren Anforderungen richtig verstanden wurden. So wird frühzeitig ein gemeinsames Verständnis zwischen allen Beteiligten geschaffen.

Um den Erfolg von Reviews zu steigern bzw. überhaupt zu gewährleisten, sind sowohl organisatorische als auch personenbezogene Faktoren zu beachten, die im Folgenden aufgeführt sind.

Organisatorische Erfolgsfaktoren

Organisatorische Faktoren

- Das Management bzw. die Projektleitung unterstützt den Reviewprozess, indem ausreichend Ressourcen im Softwareentwicklungsprozess für die Reviews der Arbeitsergebnisse eingeplant werden.
- Reviews sind Bestandteil der Unternehmenskultur, um das Lernen und die Verbesserung der Prozesse zu fördern.

- Formelle Reviews werden mit angemessener Frist geplant und für jedes Review sind klare Ziele und überprüfbare Endekriterien bei der Planung festzulegen.
- Den Teilnehmern an einem Review steht genügend Zeit zur individuellen Vorbereitung zur Verfügung.
- In Abhängigkeit von den vereinbarten Zielen, der Art und dem Niveau des untersuchten Arbeitsergebnisses und in Abhängigkeit des Kenntnisstands der beteiligten Personen ist unter den verschiedenen Reviewarten die »passende« auszuwählen. Prinzipiell sind aber alle verwendeten Vorgehen, wie z.B. checklistenbasiertes oder rollenbasiertes Review, für eine effektive Fehlererkennung im zu überprüfenden Arbeitsergebnis geeignet.
- Das Feedback aus den Reviews wird an die Stakeholder und Autoren der analysierten Dokumente gegeben, damit diese die Möglichkeit bekommen, das Produkt, aber darüber hinaus auch ihre Aktivitäten bzw. ihre Prozesse zu verbessern.
- Die verwendeten Checklisten decken die wichtigsten Risiken ab und sind auf dem neuesten Stand.
- Umfangreiche Dokumente werden nicht als Ganzes einem Review unterzogen, sondern in kleineren Teilen überprüft, sodass frühzeitiges und häufiges Feedback zu eventuellen Fehler(zustände)n an die jeweiligen Autoren gegeben werden kann. Der Reviewprozess ist dann mehrfach durchzuführen, um das gesamte Dokument zu analysieren.

Personenbezogene Erfolgsfaktoren

Personenbezogene Faktoren

- Für Reviews sind die »passenden« Personen auszuwählen, um die vereinbarten Ziele zu erreichen, z.B. eignen sich Personen mit unterschiedlichen Fähigkeiten oder Perspektiven, die das Dokument für ihre weiteren Tätigkeiten als Arbeitsgrundlage nutzen werden und sich daher sowieso in das Dokument einarbeiten müssen. Es ist daher sehr sinnvoll, Tester als Gutachter bei den Reviews einzusetzen, da die zu prüfenden Dokumente in aller Regel als Testbasis zur Erstellung der Testfälle genutzt werden. Die Tester sind dann frühzeitig mit den Dokumenten vertraut und die Testfälle können zeitnah spezifiziert werden. Durch die »Testsicht« auf die Dokumente kommen auch weitere Aspekte der Qualitätssicherung hinzu, z.B. die Prüfung auf Testbarkeit.
- Ist ein Moderator beim Review vorgesehen, kommt diesem eine besondere Bedeutung zu. Eine schlechte Moderation kann den Erfolg von Reviews schmälern, da z.B. zu viel Zeit für eher unbedeutende Fehler(zustände) »vergeudet« wird.

- Reviews dienen zur Qualitätssicherung der untersuchten Arbeitsergebnisse. Das Aufzeigen von Fehler(zustände)n, Ungenauigkeiten oder Abweichungen ist daher erwünscht bzw. gefordert. Die Mitteilungen der Reviewer darüber müssen aber wertfrei und objektiv erfolgen. Jedes Review ist in einer vertrauensvollen Atmosphäre durchzuführen. Die Teilnehmer sollen Gesten oder Körpersprache vermeiden, die Langeweile, Frust oder Feindseligkeit gegenüber anderen Teilnehmern signalisieren. Der Autor muss sich sicher sein, dass die Reviewergebnisse nicht zur Evaluation seiner Leistung, z. B. beim nächsten Gehaltsgespräch, herangezogen werden. Es soll erreicht werden, dass der Autor des untersuchten Dokuments das Review als eine positive Erfahrung empfindet. Alle Teilnehmer sollen das Review als wertvoll genutzte Zeit ansehen.
- Die Teilnehmer nehmen sich ausreichend Zeit und Aufmerksamkeit für Details. Die Reviews werden so durchgeführt, dass die Reviewer die Konzentration nicht verlieren. Es sollen keine großen Dokumente am Stück überprüft werden (s. o.) und auch eine zeitliche Begrenzung ist sinnvoll.
- Schulungsmaßnahmen sind für die am Review beteiligten Personen vorab anzubieten, dies gilt verstärkt für die mehr formaleren Reviewarten, wie die Inspektion.
- Ein wichtiger Aspekt für die Teilnehmer ist ein ständiges Lernen aus durchgeführten Reviews – eine Kultur des Lernens entsteht – und dies fördert auch die Verbesserung des Reviewprozesses und der einzelnen unterschiedlichen Vorgehen.

Exkurs: Gründe für ein Scheitern

- Scheitern Reviews an der fehlenden Vorbereitung, so liegt es oft an der terminlich/zeitlich ungünstigen Auswahl der Reviewer.
- Ist fehlende Einsicht der Reviewer in die Bedeutung der Reviews und den hohen Wirkungsgrad zur Qualitätssteigerung die Ursache für das Scheitern, dann kann die Nützlichkeit von Reviews untermauert werden, indem entsprechendes Zahlenmaterial (aus dem Projekt oder aus anderen Projekten des Unternehmens) vorgelegt wird, das die Wirksamkeit von Reviews bekräftigt.
- Ein Scheitern eines Reviews kann auch an der fehlenden oder unzureichenden Dokumentation liegen. Es muss stets vorab geprüft werden, ob alle benötigten Dokumente neben dem zu untersuchenden Arbeitsergebnis in ausreichender Beschreibungstiefe vorhanden sind. Nur wenn das der Fall ist, ist ein Review durchzuführen.

4.6 Werkzeuggestützte statische Analyse

Das Ziel der statischen Analyse ist, ähnlich dem der Reviews, die Aufdeckung vorhandener Fehler oder fehlerträchtiger Stellen in einem Dokument, allerdings wird die Analyse durch Werkzeuge vorgenommen.

Statische Analyse und Review

Statische Analyse und Review stehen in einem engen Zusammenhang. Wird vor dem Review eines Dokuments eine statische Analyse durchgeführt, kann bereits eine Anzahl von Fehlern und Unstimmigkeiten nachgewiesen werden, und die Menge der im Review zu berücksichtigenden Aspekte ist erheblich geringer. Beispielsweise lassen sich Rechtschreibfehler in einem Dokument vor dem eigentlichen Review durch ein Rechtschreibprüfprogramm finden. Die Reviewer können sich dann auf fachlich-inhaltliche Aspekte konzentrieren. Da die statischen Analysen werkzeuggestützt durchgeführt werden, ist der Aufwand wesentlich geringer als bei einem Review.

Ermittlung von Metriken

Die Bezeichnung »statische Analyse« weist darauf hin, dass diese Form der Prüfung auch keine Ausführung der Prüfobjekte (eines Programms) beinhaltet. Ein Ziel der Analyse ist auch die Ermittlung von Messgrößen oder Metriken, um eine Qualitätsbewertung durchführen und somit Qualität messen und nachweisen zu können.

Formale Dokumentstruktur

Das zu analysierende Dokument muss allerdings nach einem vorgegebenen Formalismus aufgebaut sein, also einer formalen Struktur unterliegen, um durch ein Werkzeug überprüft werden zu können. Dokumente mit einer formalen Struktur sind neben dem Programmcode beispielsweise config-files oder auch XML/HTML-Files.

KI-basierte Werkzeuge

Für KI-basierte Textanalysatoren trifft die Einschränkung auf eine formale Dokumentstruktur allerdings nicht zu, sie sind in der Lage, auch natürlichsprachige Texte (z.B. Anforderungsdokumente) zu analysieren. Die Werkzeuge können genutzt werden, um aus dem Text Metriken zu berechnen (z.B. die Anzahl und Komplexität der Sätze) oder auch Ähnlichkeiten von Sätzen oder Gruppierung von Textpassagen nach »ähnlichen Themen« zu ermitteln. Die so gewonnenen Erkenntnisse können zur Bewertung und Verbesserung von Anforderungsdokumenten verwenden werden – also eine Art von automatischer Analyse von Anforderungsdokumenten und als Assistent für den menschlichen Reviewer.

Sicherheitslücken

Statische Analysen können zum Aufdecken von Sicherheitslücken herangezogen werden. Viele Sicherheitsprobleme treten dadurch auf, dass bestimmte fehleranfällige Programmkonstrukte verwendet oder notwendige Überprüfungen nicht durchgeführt werden. Das fehlende Abfangen von Speicherüberläufen oder keine Überprüfung der Einhaltung von Datenbeschränkungen in der Eingabe sind Beispiele hierfür. Diese

Mängel können Analysewerkzeuge aufdecken, da sie meist einem bestimmten »Muster« unterliegen, das von den Werkzeugen gesucht und analysiert werden kann.

Einschränkungen der statischen Analyse

Mit der statischen Analyse lassen sich allerdings nicht alle Fehler und Unstimmigkeiten nachweisen. Es gibt Fehler oder präziser Fehlerzustände, die erst bei der Ausführung, also zur Laufzeit des Programms, zur Wirkung kommen und vorher nicht ermittelbar sind. Wird beispielsweise bei einer Division der Wert des Divisors in einer Variablen gehalten, so kann diese Variable zur Laufzeit den Wert null annehmen, was zu einer Fehlerwirkung führt. Statisch ist dieser Fehlerzustand meist nicht erkennbar, es sei denn, der Variablen wird eine Konstante mit dem Wert null zugewiesen. Alle möglichen Abläufe des Testobjekts können vom Werkzeug analysiert werden und es kann auf solche möglichen Fehlerstellen, wenn sie denn vorhanden sind, hingewiesen werden.

Es gibt aber auch Unstimmigkeiten oder fehlerträchtige Stellen im Programm, die durch dynamisches Testen nur schwer nachweisbar sind. So lässt sich die Nichteinhaltung von Programmierstandards oder die Verwendung von verbotenen fehleranfälligen Programmkonstrukten nur mit der statischen Analyse (oder mit Reviews) nachweisen.

Beispiele

Der Compiler ist wohl das gebräuchlichste Analysewerkzeug und deckt Fehlerzustände in der Syntax des Programmcodes auf. Compiler bieten aber meist noch weitere Untersuchungen und Informationsaufbereitungen an. Der Verstoß gegen Standards und andere Konventionen kann ebenfalls mithilfe von sprachspezifischen Analysewerkzeugen ermittelt werden. Zum Nachweis von Anomalien im Daten- und Kontrollfluss der Programmtexte (s. Abschnitt 7.1.3) stehen ebenfalls Werkzeuge zur Verfügung. Hilfreiche Aussagen zum Ablauf und zur Datenverwendung werden ermittelt, die oft auf fehlerträchtige Stellen hinweisen.

4.7 Unterschiede zwischen statischen und dynamischen Tests

Verschiedene Arten von Fehlern werden gefunden.

Statischer und dynamischer Test können die gleichen Ziele verfolgen (s. Abschnitt 2.1.2), ergänzen sich aber gegenseitig, indem sie verschiedene Arten von Fehlern finden. Statische Tests entdecken Fehler(zustände) in den in der Regel nicht ausführbaren Arbeitsergebnissen bzw. Dokumenten direkt. Dynamische Tests weisen Fehlerwirkungen nach, und zwar im ausführbaren Programmcode und nicht in anderen Dokumenten (von ausführbaren Modellen abgesehen). Die nachgewiesenen Fehlerwirkungen sind auf Fehlerzustände zurückzuführen. Fehlerwirkungen kön-

nen auch lange unerkannt bleiben, z.B. wenn das Programmstück, das den Fehlerzustand enthält, nur höchst selten zur Ausführung kommt. Entsprechende aufdeckende Testfälle sind bei dynamischen Tests oft schwierig zu spezifizieren, was mit einem hohen Aufwand verbunden ist. Statische Tests können einen solchen Fehlerzustand meist mit geringerem Aufwand im Programmcode nachweisen.

»Innere« Qualität wird geprüft.

Das extern sichtbare Verhalten des Testobjekts wird bei dynamischen Tests überprüft. Statische Tests legen den Fokus auf die Verbesserung der Beschaffenheit und der internen Qualität der analysierten Arbeitsergebnisse. So kann der statische Test zur Messung von den Qualitätsmerkmalen genutzt werden, die nicht von der Ausführung des Programms abhängen. Hierzu zählt beispielsweise die Wartbarkeit des Programms. Beim dynamischen Test können Qualitätsmerkmale gemessen werden, die von der Ausführung des Programms abhängen (z.B. Laufzeit und Performanz).

Typische Fehler, die erkannt werden

In der folgenden Auflistung sind typische Fehler(zustände) bzw. Ungenauigkeiten aufgeführt, die durch Reviews einfach und kostengünstig zu finden und damit frühzeitig zu beheben sind:

- **Anforderungsfehler**
  Es können Inkonsistenzen, Mehrdeutigkeiten, Widersprüche, Lücken, Ungenauigkeiten und Redundanzen in den Anforderungsdokumenten nachgewiesen werden.
- **Entwurfsfehler**
  In Architekturdokumenten, in denen die internen Schnittstellen spezifiziert sind, können Fehler – z.B. schlechte Modularität – nachgewiesen werden. So kann die Abhängigkeit (Kopplung) zwischen einzelnen Systemteilen (Komponenten, Module) sehr hoch sein, was sowohl das Verständnis der einzelnen abhängigen Teile als auch deren separates Testen erschwert oder sogar verhindert, da der Aufwand, eine entsprechende Testumgebung zu schaffen, unverhältnismäßig hoch ist. Ebenso kann der innere Zusammenhang (Kohäsion) eines Systemteils (Komponente, Modul) analysiert werden. Ein Systemteil mit starker Kohäsion ist für genau eine einzige wohldefinierte Aufgabe zuständig. Eine mehr oder weniger lose Sammlung von Aufgaben in einer Komponente deutet auf eine geringe Kohäsion hin und hat ähnlich negative Auswirkungen wie eine hohe Kopplung. Auch Algorithmen und Datenbankstrukturen, die ineffizient entworfen wurden, können erkannt werden. Beim Datenbankentwurf kann mithilfe von Reviews eine ineffiziente Datenbankstruktur aufgedeckt werden.

- **Programmierfehler**
  Prinzipiell können alle Fehlerzustände im Programmcode durch Codereviews erkannt werden, wenn qualifizierte Reviewer und ausreichend Zeit zur Verfügung stehen. So kann die Nutzung von Variablen, die keinen definierten Wert haben, und die Deklaration von Variablen, die im Programm nie genutzt werden, erkannt werden. Ebenso lässt sich unerreichbarer und doppelter (duplizierter) Code feststellen. Allerdings kann diese Art von Fehler(zustände)n auch von Compilern oder statischen Analysewerkzeugen erkannt werden, was weniger zeitaufwendig und somit kostengünstiger ist.
- **Abweichungen von Standards**
  Standards und Richtlinien tragen zu besserer Qualität bei. Eine mangelnde Einhaltung von beispielsweise Programmierrichtlinien führt zum Gegenteil – zu schlechter Qualität. Ein willkommener Nebeneffekt: Wenn klar ist, dass die Einhaltung von Programmierrichtlinien kontrolliert wird, ist die Motivation, diese auch zu befolgen, weit höher, als wenn keine Prüfung des Programmcodes vorgenommen wird.
- **Fehlerhafte Schnittstellenspezifikationen**
  Die Schnittstellen zwischen Systemteilen sind im Hinblick auf Namen und Parameter (Anzahl, Datentyp und Reihenfolge) zu prüfen, denn nicht alle Unstimmigkeiten werden bei der Integration der Systemteile auf dem Rechner erkannt.

Beispiel: Unzureichende Schnittstellenspezifikation

Ein bekanntes und dokumentiertes Beispiel für unzureichende Spezifikation von Schnittstellen ist die NASA-Sonde Mars Climate Orbiter [URL: Mars Climate Orbiter]. Die Programmierer verwendeten unterschiedliche Maßsysteme (metrisches und angloamerikanisches), was zum Verlust der Sonde führte. Bei einem formalen Review der Schnittstellenspezifikation, wie der Inspektion, wäre die fehlende Festlegung auf das Maßsystem mit einer gewissen Wahrscheinlichkeit erkannt worden.

- **Schwachstellen in der Zugriffssicherheit**[16]
  Neben den dynamischen Tests auf Sicherheit können viele solcher Schwachstellen auch statisch ermittelt werden. Hierzu gehört beispielsweise die Anfälligkeit bei Pufferüberlauf (Buffer-Overflow), wenn die Einhaltung von Grenzen nicht explizit geprüft wird, und die Möglichkeit des direkten Manipulierens der Eingabedaten oder von SQL-Abfragen.

16. Werkzeuggestützte Analysen liefern hier sicherlich verlässlichere Ergebnisse (s. Kap. 7).

- **Lücken oder Ungenauigkeiten bei Verfolgbarkeit oder Überdeckungsgrad**
  Auf die Bedeutung der Verfolgbarkeit und des Überdeckungsgrades wurde in Abschnitt 2.3.8 hingewiesen. Lücken bei der Verfolgbarkeit machen diese wertlos, da beispielsweise kein Zusammenhang mehr zwischen den Anforderungen und den Testfällen hergestellt werden kann. Ungenauigkeiten bei den Abnahmekriterien und dem Grad der nachzuweisenden Überdeckung machen diese ebenfalls unbrauchbar und können zu falscher Einschätzung oder falschen Ergebnissen führen. Reviewergebnisse können z. B. auf fehlende Tests für ein zu erfüllendes Abnahmekriterium hinweisen.

Wartbarkeit

Softwaresysteme haben oft eine erstaunlich lange Lebenszeit, weshalb der Wartbarkeit der Systeme eine bedeutende Rolle zukommt. Die meisten Schwächen in Bezug auf Wartbarkeit können durch statische Tests gefunden werden. Hierzu gehören: unsachgemäße Modularisierung (Kopplung und Kohäsion, s. o.), schlechte Wiederverwendbarkeit von Komponenten, schwer verständlicher und schwer änderbarer Programmiercode (kein »Clean Code« [Martin 09]), der dadurch ein hohes Risiko birgt, dass bei notwendigen Änderungen neue Fehlerzustände eingebaut werden.

4.8 Zusammenfassung

- Jedes Arbeitsergebnis, also jedes Dokument, kann einem Review unterzogen werden. Voraussetzung ist, dass die Teilnehmer am Review wissen, wie das Dokument zu lesen und zu verstehen ist.
- Reviews sind statische Tests, bei denen das Arbeitsergebnis – im Gegensatz zum dynamischen Test – nicht auf einem Rechner zur Ausführung kommt. Daher können Reviews frühzeitig und auf alle Arbeitsergebnisse, die bei der Softwareerstellung genutzt oder erstellt werden, angewendet werden.
- Mehrere Augenpaare sehen mehr als ein Augenpaar – auch in der Softwareentwicklung. Reviews zur Kontrolle und Qualitätssteigerung nehmen diesen Grundsatz auf. Dokumente werden von mehreren Personen begutachtet und die Ergebnisse in einer Sitzung diskutiert und protokolliert.
- Der Reviewprozess besteht aus den folgenden Aktivitäten: Planung, Reviewbeginn (Reviewinitiierung), individuelles Review (d.h. individuelle Vorbereitung), Kommunikation und Analyse (Diskussion der Befunde – meist in einer Reviewsitzung), Behebung und Berichterstattung.

- Um Fehler(zustände) im Rahmen der individuellen Vorbereitung besser erkennen zu können, gibt es eine Reihe von Vorgehensweisen, die angewendet werden können:
  - **Ad hoc**
    Ohne Vorgaben
  - **Checklistenbasiert**
    Fragen aus Checklisten helfen beim Review
  - **Szenarien und Probeläufe** (»Dry Runs«)
    »Durchspielen« von Szenarien
  - **Rollenbasiert**
    Einnehmen von Rollen beim Review
  - **Perspektivisch**
    Unterschiedliche Sichten auf das Arbeitsergebnis
- Rollen beim Review und damit zu verteilende Aufgaben sind: Manager, Reviewleiter, Moderator, Autor, Reviewer (Gutachter) und Protokollant.
- Es gibt eine ganze Reihe von unterschiedlichen Ausprägungen oder Arten von Reviews, von denen in diesem Kapitel vier näher beschrieben sind. Leider gibt es keine einheitliche, durchgängige Begriffsfestlegung zu Reviews in den verschiedenen Standards und Normen.
  - Das informelle Review unterliegt keinem formalisierten Prozess und auch die Form der Ergebnisdokumentation ist nicht vorgeschrieben. Wegen des geringeren Aufwands hat diese Form des Reviews eine hohe Akzeptanz und weite Verbreitung in der Praxis gefunden.
  - Der Walkthrough ist eine informelle Vorgehensweise, bei der der Autor sein Dokument den Gutachtern in der Sitzung präsentiert. Die Vorbereitung hat einen geringen Umfang. Der Walkthrough ist besonders für kleine Entwicklungsteams geeignet, zum Diskutieren von Alternativen und zur Verbreitung von Wissen im Team.
  - Beim technischen Review ist der Ablauf genauer festgelegt und eine Vorbereitung der Reviewer zwingend erforderlich. Reviewer sollen fachliche Kollegen des Autors sein, damit alternative Umsetzungen diskutiert werden können.
  - Die Inspektion ist die formalste Reviewart. Eine Vorbereitung erfolgt anhand von Checklisten, für die einzelnen Prüfschritte sind Eintritts- und Austrittskriterien definiert. Die Sitzung wird von einem geschulten Moderator geleitet. Ziel von Inspektionen ist neben der Qualitätskontrolle des Arbeitsergebnisses auch die Verbesserung des Entwicklungs- und Reviewprozesses.

- Allgemein unterliegen Reviews firmenspezifischen Ausprägungen. Sie werden auf die jeweiligen Bedürfnisse und Erfordernisse zugeschnitten, wodurch ihr Wirkungsgrad erhöht wird. Wichtig ist, dass eine kooperative Zusammenarbeit zwischen den an der Softwareentwicklung beteiligten Personen etabliert ist.
- Unter Beachtung von Erfolgsfaktoren, die in organisatorische und personenbezogene unterteilt werden können, und der Vermeidung von möglichen Schwierigkeiten sind Reviews ein sehr effizientes Werkzeug zur Qualitätssteigerung in der Softwareentwicklung.
- Neben den Reviews lässt sich auch werkzeuggestützt eine ganze Reihe von Analysen an Dokumenten durchführen. Die Untersuchungen werden unter dem Begriff statische Analysen zusammengefasst und ohne Ausführung des Prüfobjekts durchgeführt.
- Reviews finden typischerweise andere Fehler(zustände) als das dynamische Testen.

Inhaltsübersicht

Inhaltsverzeichnis