Dr. Michael Felderer

(michael.felderer@uibk.ac.at)
ist wissenschaftlicher Mitarbeiter am Institut für Informatik der Universität Innsbruck. Seine Arbeitsschwerpunkte in Forschung und Lehre sind Testen von Software, Requirements Engineering sowie modellbasierte Softwareentwicklung. Neben seiner wissenschaftlichen Tätigkeit ist Dr. Felderer Geschäftsführer und Senior Berater für die QE LaB Business Services GmbH.

Prof. Dr. Ina Schieferdecker

(ina.schieferdecker@fokus.fraunhofer.de)
beschäftigt sich mit Fragen der modellbasierten Softwareentwicklung, der Analyse, des Testens und der Bewertung softwareintensiver Systeme und der Automatisierung und Optimierung von Prozessen zur Software-(Weiter-) Entwicklung und Qualitätssicherung. An der Freien Universität Berlin leitet sie das Fachgebiet Modellbasierte Entwicklung und Qualitätssicherung software-basierter Systeme und ist am Fraunhofer Institut FOKUS, Berlin beim System Quality Center aktiv.

Handreichung zur Methodenauswahl
Eine Taxonomie risikobasierter Softwaretests

In der Regel muss Software mit limitierten Ressourcen und unter hohem Zeitdruck getestet werden. Trotz dieser erschwerten Bedingungen besteht die Notwendigkeit sicherzustellen, dass die Software den an sie gestellten Anforderungen entspricht. Darüber hinaus müssen mögliche Fehler offengelegt werden, welche die missionskritischen Funktionen der Software gefährden. Dies kann nur mir Softwaretests geleistet werden. Risikobasiertes Testen (engl.: risk-based testing) fußt dabei auf dem Prinzip, dass Risiken bewertet und diese Bewertung kontinuierlich aktualisiert wird.

Einleitung

Beim Risikobasierten Testen (kurz RBT) bestimmt die Risiko(neu)bewertung alle Phasen des Testverlaufes, was eine Optimierung der Testbemühungen und eine Minimierung der Risiken, denen softwarebasierte Systeme ausgesetzt sind, zur Folge hat. In Wissenschaft und Industrie wurden bereits mehrere risikobasierte Testverfahren entwickelt, da diese Methode einen besonders hohen Stellenwert und insbesondere eine große praktische Relevanz besitzt. Dieser Artikel präsentiert eine Taxonomie risikobasierter Testverfahren und bietet einen Rahmen, um die verschiedenen Ansätze verstehen, kategorisieren, bewerten und vergleichen zu können. Diese Taxonomie soll zudem die gezielte Auswahl passender Verfahren erleichtern und eine Anpassung an spezielle Erfordernisse ermöglichen. Die Taxonomie risikobasierter Softwaretests wurde auf Basis der Analyse von zahlreichen Publikationen zum risikobasierten Testen erstellt.

Das Testen sicherheits- oder geschäftskritischer Software ist in erster Linie mit einem Problem verbunden: Es muss festgelegt werden, welche Tests geeignet sind, die grundlegenden Funktionen der Software zu gewährleisten und gleichzeitig die Softwareschäden offenzulegen, welche die missionskritischen Funktionen der Software gefährden. Doch auch das Testen von „normaler“, weniger sicherheits- oder geschäftskritischer Software unterliegt einem vergleichbaren Problem: In der Regel wird Software unter hohem Druck getestet, der den knappen zur Verfügung stehenden Ressourcen und den engen Zeitvorgaben geschuldet ist. Dementsprechend müssen die Testbemühungen stark fokussiert werden.

Dieses Problem – also die Entscheidung für welche Testmethode und welchen Umfang – kann durch die Verwendung risikobasierter Testansätze beseitigt werden. Risikobasierte Testansätze betrachten die Risiken des Softwareproduktes als Leitfaktoren, die alle Phasen des Testvorganges bestimmen, nämlich Testplanung, Testentwurf, Implementierung, Ausführung und Auswertung [GerTh02, SchGrSch12, FeRa14]. Risikobasiertes Testen ist ein pragmatischer, in der Industrie bereits weit verbreiteter Ansatz, der dazu beiträgt, das Kernproblem der begrenzten Ressourcen beim Testen von missionskritischer Software zu beseitigen. Risikobasierte Tests fußen auf dem einfachen Grundprinzip, die Testaktivitäten auf jene Szenarien zu konzentrieren, welche die kritischsten Situationen für Softwaresysteme herbeiführen könnten.

Wegen der Bedeutung und hohen Praktikabilität risikobasierter Tests haben sowohl die Wissenschaft, beispielsweise [ChPrSi02, StMe07, Zimm09, BaiKenYu12, YooCh11, WeKrSch12, FeHaiBrMo12, ZeFeBr12], als auch die Industrie, beispielsweise [Bach99, RoStGa00, Aml00, GerTh02, Redm05, Veen12], unterschiedliche Ansätze entwickelt. Seit kurzem werden Risiken sogar im internationalen Standard zu Testtechniken, -prozessen und -dokumentation ISO/IEC/IEEE 29119 Software Testing [ISO29119] als ein wesentlicher Bestandteil des Testplanungsprozesses berücksichtigt. Angesichts der großen Anzahl risikobasierter Testansätze und ihrer stetig steigenden Verbreitung in industriellen Testprozessen scheinen eine Kategorisierung und Bewertung sowie Unterstützung beim Vergleich und der Auswahl der unterschiedlichen risikobasierten Testansätze allerdings dringend geboten.

In diesem Artikel soll eine solche Handreichung geboten werden. Sie beinhaltet eine Taxonomie der unterschiedlichen risikobasierten Testmethoden und bietet einen Rahmen, um die unterschiedlichen Ansätze

verstehen, kategorisieren, bewerten und vergleichen zu können – mit dem Ziel, die angemessenste Testmethode auszuwählen und entsprechend der individuellen Anforderungen anzupassen. Im Allgemeinen legt eine Taxonomie (Klassifikation oder Klassifizierungsschema) eine Hierarchie von Klassen (Kategorien, Dimensionen, Kriterien oder Charakteristika) mit dem Ziel fest, Dinge oder Konzepte zu kategorisieren beziehungsweise einzuordnen. Sie entspricht also einer Baumstruktur, in der die Blätter konkreten Werten einzelner Fälle entsprechen. Die vorliegende Taxonomie geht von einer Berücksichtigung von Risiken in allen Phasen des Testprozesses aus und beinhaltet die Oberkategorien Risikotreiber (mit den Unterkategorien Funktionalität, funktionale Sicherheit, IT-Sicherheit), Risikobewertung (mit den Unterkategorien Typ des Risikogegenstandes, Risikofaktoren, Risikoabschätzung und Automatisierungsgrad) sowie risikobasierte Testprozesse (mit den Unterkategorien risikobasierte Testplanung, Testentwurf, Implementierung, Ausführung und Bewertung). Zur Erstellung dieser Taxonomie wurden zahlreiche Ansätze aus Forschung und Industrie analysiert.

Grundlegende Konzepte von risikobasierten Tests

Testen ist nichts anderes als die Bewertung von Software durch Beobachtung ihrer Ausführung. Das ausgeführte softwarebasierte System wird „System Under Test“ (kurz SUT) genannt. Risikobasiertes Testen ist ein Testansatz, in dem die Risiken eines Softwareproduktes als Leitfaden betrachtet werden, auf dessen Basis in allen Phasen des Testprozesses Entscheidungen getroffen werden. Ein Risiko wird definiert als ein Faktor, der zu einem gewissen Zeitpunkt negative Auswirkungen haben könnte, und wird normalerweise durch seine Eintrittswahrscheinlichkeit und die Tragweite seiner Auswirkungen ausgedrückt. In Bezug auf Softwaretests wird die Eintrittswahrscheinlichkeit davon bestimmt, mit welcher Wahrscheinlichkeit eine einem gewissen Risiko zugeschriebene Fehlfunktion auftritt. Die Auswirkung wird bestimmt durch die Kosten oder die Tragweite, die eine Fehlfunktion im laufenden Betrieb zur Folge hätte. Der sich daraus ergebende Risikowert oder Risikoexposition werden bestimmten Risikogegenständen zugeschrieben. Im Kontext von Softwaretests wird der Risikogegenstand als ein Asset verstanden (also ein Gut mit einem bestimmten Wert), der einem Test unterzogen wird – also beispielsweise eine Anforderung, Komponente oder ein Ausfallszenario.

RBT ist ein testbasierter Ansatz des Risikomanagements, der nur dann zufriedenstellende Ergebnisse liefern kann, wenn ein Testvorgang ausgeführt und eine ausreichende Risikobewertung in diesen Prozess integriert wird. Ein Testprozess besteht aus den Kernaktivitäten Testplanung, Testentwurf, Testimplementierung, Testausführung und Testbewertung. Im Folgenden sollen die einzelnen Bestandteile des Testprozesses und die damit verbundenen Konzepte näher erläutert werden.

Gemäß [ISTQB] versteht sich die Testplanung als die Handlung, mit der ein Testplan entwickelt oder aktualisiert wird. Ein Testplan ist ein Dokument, in dem Umfang, Herangehensweise, Ressourcen und Zeitplan einer angestrebten Testaktivität beschrieben werden. Er definiert unter anderem das Ziel des Tests, die zu testenden Eigenschaften, die Testentwurftechniken und die Testendekriterien, die dem Test zugrunde gelegt werden, sowie die Grundlagen dieser Entscheidungen. Die Testziele entsprechen den Gründen oder dem Zweck, zu dem ein Test entworfen und ausgeführt wird. Der Grund für einen Test ist entweder die Überprüfung des Funktionsverhaltens des Systems oder seiner nicht-funktionalen Eigenschaften. Das funktionale Testen beschäftigt sich mit der Bewertung des Funktionsverhaltens eines SUT, während sich das nicht-funktionale Testen auf nicht-funktionale Anforderungen konzentriert, wie funktionale Sicherheit, IT-Sicherheit, Verlässlichkeit und Leistung. Getestet werden können Komponenten, Integrationen oder ganze Systeme. Im Rahmen der Komponententests (auch Modul- oder Unittest genannt) wird die kleinste testbare Einheit, etwa eine Klasse, isoliert getestet. In Integrationstests werden Komponenten miteinander kombiniert und diese werden als Subsystem getestet, welches aber noch kein vollständiges System darstellt. Ein Systemtest prüft dagegen ein vollständiges System inklusive aller Subsysteme. Der Regressionstest ist das wiederholte Testen einzelner Systeme oder Komponenten. Er dient dazu sicherzustellen, dass Modifikationen in bereits getesteten Systemen keine neuen, unerwünschten Effekte nach sich ziehen und dass das System und seine Komponenten nach wie vor die an sie gestellten Anforderungen erfüllen. Testendekriterien sind jene Bedingungen, die erfüllt werden müssen, um den Test offiziell für beendet zu erklären. Auf ihrer Grundlage werden Testberichte erstellt und es wird festgelegt, wann der Test beendet wird. Überdeckungskriterien in Verbindung mit getesteten Feature-Typen und den angewandten Testentwurfstechniken sind dabei typische Testendekriterien. Nach der Erstellung des Testplans beginnt die Testkontrolle – eine fortlaufende Handlung, in der der tatsächlich ausgeführte Prozess mit dem Testplan verglichen wird und die häufig konkrete Maßnahmen zur Folge hat.

Während der Testentwurfsphase werden die allgemeinen, im Testplan definierten Testziele auf greifbare Testbedingungen und abstrakte Testfälle übertragen. In der Testimplementierung werden dann die abstrakten Testfälle durchführbar gemacht, beispielsweise durch das Anlegen einer Testumgebung und das Sammeln von Testdaten, Bereitstellung einer Protokollierungsunterstützung und das Schreiben von Testskripten, die für die automatisierte Ausführung von Testfällen notwendig sind. In der Testausführungsphase werden die Testfälle dann durchgeführt und alle relevanten Ausführungsdaten werden protokolliert und überwacht. Schließlich werden in der Testauswertungsphase die Testendekriterien bewertet und die protokollierten Testergebnisse in einem Testbericht zusammengefasst.

Risikomanagement umfasst die Kernaktivitäten Risikoerkennung, Risikoanalyse, Risikobewältigung und Risikoüberwachung. In der Risikoerkennungsphase werden die Risikogegenstände identifiziert. In der Phase der Risikoanalyse werden Eintrittswahrscheinlichkeiten und Auswirkungen der Risikogegenstände sowie die entsprechende Risikoexposition geschätzt. Auf Grundlage der Risikoexpositionswerte erfolgen die Priorisierung der einzelnen Risikogegenstände sowie die Einteilung in unterschiedliche Risikogruppen entsprechend der Höhe des Risikos.

In der Risikobewältigung werden dann Maßnahmen bestimmt und implementiert, die am Ende eine zufriedenstellende Situation herbeiführen. In der Risikoüberwachungsphase werden die Risiken über einen bestimmten Zeitraum beobachtet und Statusberichte erstellt. Darüber hinaus werden die Auswirkungen der implementierten Maßnahmen ausgewertet. Gemeinhin werden die Phasen der Risikoerkennung und Risikoanalyse unter dem Begriff Risiko-

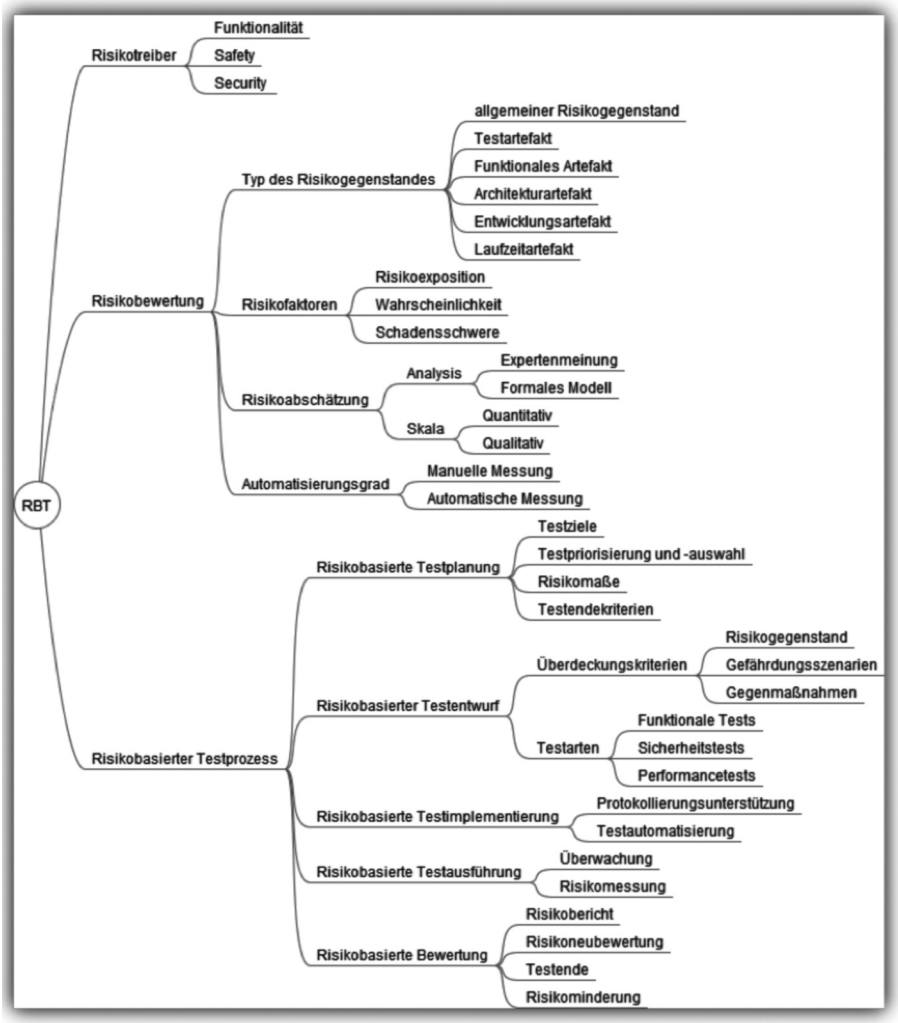

Abb. 1: Taxonomie für das risikobasierte Testen

bewertung zusammengefasst, während die Risikobewältigung und Risikoüberwachung als Risikokontrolle gelten.

Taxonomie für risikobasiertes Testen

Die Taxonomie risikobasierter Softwaretests (siehe Abbildung 1) besteht aus den Oberkategorien

- Risikotreiber,
- Risikobewertung sowie
- risikobasierte Testprozesse,

und ist abgestimmt auf die Berücksichtigung von Risiken in allen Phasen des Testprozesses. In diesem Abschnitt werden kurz die Oberkategorien und ihre Unterkategorien erläutert. Weiterführende Informationen werden in [FeSch14] gegeben.

Risikotreiber

Wie [ISTQB] erläutert, sind Risiken die Ergebnisse von ‚Hazards' (dt.: Gefährdung, Gefahr, Gefahrenmoment, Risiko, Wagnis, Zufall). In softwarebasierten Systemen entsprechen Hazards Schwachstellen und Fehlfunktionen der Software, die sicherheits- oder geschäftsrelevante Probleme hervorrufen können. Daher muss getestet werden, dass ein softwarebasiertes System

- zuverlässig ist. Es muss also in der Lage sein, die festgelegten Dienste und Leistungen zu erbringen.
- verfügbar ist, also die gestellten Anforderungen zum gewünschten Zeitpunkt erfüllt.
- funktional sicher ist, also ohne kritische Fehler operiert.
- die IT-Sicherheit gewährleistet ist, das System also vor absichtlichen oder unabsichtlichen Angriffen geschützt ist.
- robust ist und zeitnah mit unerwarteten Vorfällen umgehen kann.

Die unterschiedlichen Testmethoden (funktionale Tests, Sicherheitstests und Performancetests) unterscheiden sich voneinander, daher bestimmen die *Risikotreiber*, welche Testmethode angemessen ist und zur Anwendung kommt. Somit erscheinen die Risikotreiber auch an erster Stelle der Taxonomie, denn diese sind das erste differenzierende Kriterium bei risikobasierten Testansätzen. Wir begreifen *Funktionalität*, *funktionale Sicherheit* (engl.: Safety) und *IT-Sicherheit* (engl.: Security) als die wichtigsten Risikotreiber von Software. Zusammengenommen ergeben sich aus diesen Risikotreibern andere Aspekte wie Verlässlichkeit, Verfügbarkeit und Belastbarkeit von softwarebasierten Systemen.

Risikobewertung

Das zweite differenzierende Kriterium von RBT-Ansätzen ist die Art und Weise, in der Risiken bestimmt werden. Gemäß [STTT14] ist die *Risikobewertung* der Prozess des Identifizierens und kontinuierlichen Analysierens dieser identifizierten Risiken. Ermittelt wird die Höhe des Risikos – in der Regel durch die Bestimmung der Eintrittswahrscheinlichkeit und der möglichen Auswirkungen beziehungsweise Schadensschwere. Bei der Risikobewertung sind viele unterschiedliche Aspekte zu berücksichtigen, sodass eine weitergehende Differenzierung erfolgen muss, beispielsweise über den *Typ des Risikogegenstandes*, auf den sich ein gewisses Risiko bezieht, die *Faktoren*, die die Risiken beeinflussen, die *Risikoabschätzungstechnik*, mit der das Risiko eingeschätzt und/oder bewertet wird, sowie über den *Automatisierungsgrad* der Risikobewertung.

Der *Typ des Risikogegenstandes* bestimmt diejenigen Elemente, auf welche sich Risikoexposition und Tests beziehen [RaMo13]. Der Typ des Risikogegenstandes wird zudem von Teststufen bestimmt. Beispielsweise werden Funktions- und Architekturartefakte häufig für Systemtests genutzt und generische Risiken für Sicherheitstests.

Die *Risikofaktoren* quantifizieren erkannte Risiken [SouGuVe10]. Der Wert des quantifizierten Verlustpotenzials wird *Risikoexposition* genannt. Sie berechnet sich aus der *Eintrittswahrscheinlichkeit* multipliziert mit dem möglichen *Verlustpotenzial*, also der Schadensschwere oder Auswirkung. Die Risikoexposition berücksichtigt in der Regel Aspekte wie Haftungsprobleme, Vermögensverluste oder -schäden und Verlagerungen bei den

Anforderungen an das Produkt. RBT-Ansätze können zudem die spezifischen Aspekte der Eintrittswahrscheinlichkeit berücksichtigen, beispielsweise zur Festlegung der Testpriorisierung, zur Testauswahl oder zum spezifischen Aspekt des *impact ratings*, mit dem bestimmt wird, wie hoch der benötigte Testaufwand war, mit dem Gegenmaßnahmen in der Software analysiert wurden.

Die Technik der *Risikoabschätzung* bestimmt die Grundlage, auf der die Risikoexposition geschätzt wird. Sie kann auf *Expertenmeinungen* oder *formalen Modellen* basieren. Der grundlegende Unterschied zwischen einer modell- und einer expertenbasierten Schätzung ist der Schritt der Quantifizierung, also dem letzten Schritt, in dem der Input in die Risikoabschätzung übertragen wird. Formale Risikoabschätzungsmodelle basieren auf mechanischen Quantifizierungsschritten wie Formeln oder Testmodellen. In der expertenbasierten Methode beruht der Quantifizierungsschritt dagegen auf dem Ermessen des Experten.

Die Risikobewertung kann auch mithilfe von *automatisierten Methoden* und Werkzeugen unterstützt werden. So können risikoorientierte Maße manuell oder automatisch gemessen werden. Die *manuelle Messung* wird häufig von zusätzlichen Richtlinien und erklärenden Hinweisen unterstützt, während die *automatische Messung* häufig mit statischen Analysetools durchgeführt wird. Ein anderes Beispiel für die automatisierte Risikobewertung ist die Ableitung von Risikoexpositionen aus formalen Risikomodellen (siehe hierzu beispielsweise [FeRa14]).

Risikobasierter Testprozess

RBT folgt dem fundamentalen Testprozess [ISTQB] oder Abwandlungen davon auf der Basis von festgelegten und beschriebenen Risiken. Alle in einem Testprozess ausgeübten Handlungen und alle Testphasen werden beeinflusst von der Risikoperspektive, welche beim RBT eingenommen wird.

In der *Testplanung* werden Rahmen, Ansatz, Ressourcen und Zeitplan intendierter Testaktivitäten bestimmt oder aktualisiert. Unter anderem werden Testziele, Testpriorisierung und Testauswahl, Risikomaße und Testendekriterien, die einen risikobasierten Test beeinflussen, festgelegt.

Beim *Testentwurf* geht es darum, die auf der Risikobewertung eines gewissen Produktes oder des entwickelnden Projektes basierenden Testaktivitäten und Testanstrengungen zu bündeln. Einfach ausgedrückt: Gibt es ein hohes Risiko, wird eingehend getestet. Gibt es kein Risiko, wird wenig getestet. Der Grund, einen Test zu entwerfen und ihn zu vollziehen, also ein Testziel festzulegen, ist entsprechend mit dem zu testenden Risikogegenstand verbunden, mit den Gefährdungsszenarien eines Risikogegenstandes und mit den Gegenmaßnahmen, die erarbeitet wurden, um den Risikogegenstand zu sichern.

Zur Optimierung der Testkosten und/oder der Qualität und der Fehlererkennung gibt es – in der Praxis weithin genutzte – Techniken, um Tests und Testkombinationen zu priorisieren, auszuwählen und auf ein Mindestmaß zu beschränken. Diese Techniken werden angewendet, um festzulegen, welche Tests geeignet sind, um die vorher festgelegten, risikobezogenen Testziele zu erfüllen, und kategorisieren die Risiken auf einer Skala von ‚nicht vertretbar hoch' bis ‚As Low As Reasonably Practicable'(ALARP)[1]).

Maße für RBT werden genutzt, um unterschiedliche Aspekte des Testens zu quantifizieren, wie zum Beispiel den minimalen Testaufwand, die aufgrund der Anzahl der gefundenen Fehler erforderlichen Extratests, die Qualität der Tests und des Testprozesses. Sie dienen der Verwaltung des RBT-Prozesses und dazu, ihn hinsichtlich Zeit, Anstrengungen und Qualität zu optimieren [Aml00].

Spezifische RBT-*Testendekriterien* [Aml00] erweitern klassische Kriterien um überdeckungsbezogene Kriterien für die Risiken und solche, die sich auf das Restrisiko des Produktes beziehen – also alle Risikogegenstände, die jeweiligen Gefahrenszenarien und/oder abgedeckte Gegenmaßnahmen.

Für den *Testentwurf* nutzt RBT Methoden, die auf Risikoartefakte und auf die Risikotreiber Funktionalität, funktionale Sicherheit und IT-Sicherheit zugeschnitten sind.

Die klassischen, codeorientierten und modellbasierten *Überdeckungskriterien* wie Pfadüberdeckung, zustandsorientierte Überdeckungskriterien wie Modified Condition/Decision Coverage, anforderungsorientierte Überdeckungskriterien wie Anforderungs- oder Anwendungsfallüberdeckung werden bei RBT durch spezielle Überdeckungskriterien erweitert, die einzelne oder alle Werte, Gefährdungsszenarien und Gegenmaßnahmen abdecken. Während die Überdeckung des Risikogegenstandes eher der anforderungsorientierten Überdeckung zugerechnet wird, kann der Überdeckung der Gefährdungsszenarien und der Gegenmaßnahmen eher mit codeorientierten, modellbasierten und/oder zustandsorientierten Überdekkungskriterien begegnet werden.

Wie verschiedene Computer-Notfall-Teams (engl.: computer emergency response teams) wie das GovCERT-UK berichten, sind Softwarefehler noch immer eine der wichtigsten, wenn nicht die wichtigste Quelle für Zwischenfälle in softwarebasierten Systemen. Daher ist beim RBT das funktionale Testen eine der wichtigsten *Testarten*, um die Zuverlässigkeit und Sicherheit solcher Systeme zu untersuchen. Bei Sicherheits- und Belastbarkeitsanalysen spielen darüber hinaus Sicherheitstests wie Penetrationstests, Fuzz-Tests und/oder Zufallstests (engl.: randomized testing) eine wichtige Rolle bei RBT. Außerdem werden Leistungs- und Skalierungstests zur Analyse der Verfügbarkeit und Belastbarkeit durchgeführt, die Normallast-, Höchstlast- und Überlastszenarien adressieren.

Die *Testimplementierung* umfasst Aufgaben wie das Erstellen einer Testumgebung und von Testdaten, das Bereitstellen einer Protokollierungsunterstützung und das Erstellen automatisierter Testskripts zur automatisierten Ausführung der Testfälle. Die Risikoaspekte betreffen hier in erster Linie die Bereitstellung der *Protokollierungsunterstützung* und die *Testautomatisierung*.

In der Testausführungsphase wird das risikobasierte Testen unterstützt von der *Überwachung* und der *Messung* der Risikomaße.

Die *Testbewertung* umfasst Entscheidungen auf Basis der zuvor festgelegten Testendekriterien und der protokollierten Testergebnisse, die zudem in einem Risikobericht zusammengefasst werden. Hierbei werden Risiken nötigenfalls neubewertet. Das Testende erfordert eine konkrete Entscheidung, ob und wann der Testvorgang zu beenden ist [RaMo13], es kann allerdings auch erneute Risikomini-

[1]) *Das ALARP-Prinzip (dt.: so niedrig, wie vernünftigerweise praktikabel) wird in der Regel für sicherheits- und missionskritische Systeme genutzt. Es besagt, dass das eingegangene Restrisiko so weit reduziert werden soll, wie es praktikabel ist.*

Abb. 2: Klassifikation von RBT-Methoden in der Taxonomie

mierungsmaßnahmen zur Folge haben. Die Risikominderung umfasst Maßnahmen zur Minimierung der Eintrittswahrscheinlichkeit oder der Auswirkungen eines Risikos. Im Rahmen des risikobasierten Testens kann es notwendig sein, zusätzliche Maßnahmen zur Minimierung der Eintrittswahrscheinlichkeit oder der Auswirkungen eines Risikos zu implementieren, wenn sich im Verlauf des Tests zeigt, dass die angenommenen Risiken von den tatsächlichen Testergebnissen und den Testendekriterien, die in einem Testbericht aufgeführt werden, abweichen.

Arbeiten mit der Taxonomie

Exemplarisch erläutern wir nun an ausgewählten Beispielen die Arbeit mit der Taxonomie. Dazu werden wesentliche Arbeiten zum risikobasierten Testen nach der Taxonomie in Abbildung 2 eingeordnet.

Drei davon werden beispielhaft erklärt: [ChPrSi02] definiert eine spezifikationsbasierte Auswahl von Regressionstests mit Risikoanalyse. Jeder Testfall entspricht einem Weg durch ein Aktivitätsdiagramm (seine Elemente repräsentieren Anforderungsattribute) und hat eine zugeordnete Wahrscheinlichkeit, Kosten und Schadensschwere. Die Testauswahl umfasst die Schritte (1) Bewertung der Kosten, (2) Herleitung der Wahrscheinlichkeit, (3) Berechnung der Risiken für jeden Testfall sowie (4) die Auswahl der Sicherheitstests. Das Risikopotenzial der Testfälle, die rund um Szenarien gruppiert werden, wird summiert, bis keine Zeit beziehungsweise Ressourcen zur Verfügung stehen.

[StMePo08] definiert einen modellbasierten Ansatz für sicherheitskritische Systeme, bei dem Risiken mit Hilfe des Modells der Faktor-Kriterien-Metriken gemessen werden und an UML-Use-Cases und Aktivitätsdiagramme annotiert werden. Von diesen werden die Tests abgeleitet.

Auch [KlHuE11] definiert einen modellbasierten Ansatz für sicherheitskritische Systeme. Er nutzt eine Fehlerbaumanalyse (engl.: Fault Tree Analysis) während der Herleitung der Automaten, die die Testmodelle repräsentieren. Von diesen werden Testfälle abgeleitet, ausgewählt und priorisiert entsprechend der bestimmten Risiken und der Ereignisse, die diese verursachen können.

Die weiteren in Abbildung 2 dargestellten Arbeiten können aufgrund der Begrenzungen des Artikels nicht erläutert werden. Weiterführende aktuelle Arbeiten zu RBT finden sich zudem auch in [STTT14].

Fazit

In diesem Artikel wurde eine Taxonomie für das risikobasierte Softwaretesten vorgestellt. Sie geht von der Annahme aus, dass Risiken in allen Phasen des Testprozesses berücksichtigt werden und basiert auf drei Oberkategorien: Risikotreiber, Risikobewertung und risikobasierter Testprozess. Risikotreiber sind in erster Linie Funktionalität, funktionale Sicherheit und IT-Sicherheit. Risikobewertung umfasst die Unterkategorien Typ des Risikogegenstandes, Risikofaktoren, Risikoabschätzung und Automatisierungsgrad. Die Kategorie ‚risikobasierter Testprozess' berücksichtigt die geschätzten Risiken und nutzt diese zur Steuerung der Testaktivitäten. Die Unterkategorien sind: risikobasierte Testplanung, Testentwurf, Testimplementierung, Testausführung, und Testbewertung.

Die vorliegende Taxonomie bietet ein Gerüst, um unterschiedliche risikobasierte Testansätze verstehen, kategorisieren, bewerten und vergleichen zu können. Sie soll die Auswahl eines Ansatzes erleichtern und die Anpassung an spezifische Bedürfnisse vereinfachen.

Danksagung

Dieser Artikel wurde in Teilen von den Wissenschaftsprojekten MOBSTECO (FWF P 26194-N15), QE LaB – Lebendige Modelle für offene Systeme (*Living Models for Open Systems*; FFG 822740), ITEA2 DIAMONDS (Entwicklung und industrielle Anwendung von Technologien für Multi-Domain-Sicherheitstests) und EU RASEN (Kompositionelle Risikoanalyse und IT-Sicherheitstesten für vernetzte Systeme) finanziert.

Literatur & Links

[Aml00] S. Amland, Risk-based testing: Risk analysis fundamentals and metrics for software testing including a financial application case study, in: Journal of Systems and Software 53(3) (2000), 287-295

[Bach99] J. Bach, Heuristic risk-based testing, in: Software Testing and Quality Engineering Magazine 11, 1999

[BaiKenYu12] X. Bai, R. S. Kenett, W. Yu, Risk assessment and adaptive group testing of semantic web services, in: International Journal of Software Engineering and Knowledge Engineering 22(05) (2012), 595-620

[ChPrSi02] Y. Chen, R. L. Probert, D. P. Sims, Specification-based regression test selection with risk analysis, in: Proceedings of the 2002 conference of the Centre for Advanced Studies on Collaborative research, IBM Press (2002)

[FeHaiBrMo12] M. Felderer, C. Haisjackl, R. Breu, J. Motz, Integrating manual and automatic risk assessment for risk-based testing. Software Quality, in: Process Automation in Software Development (2012), 159-180

[FeRa13] M. Felderer, R. Ramler, Experiences and challenges of introducing risk-based testing in an industrial project, in: Software Quality. Increasing Value in Software and Systems Development. Springer (2013), 10-29

[FeRa14] M. Felderer, R. Ramler, Integrating risk-based testing in industrial test processes, in: Software Quality Journal 22(3) (2014), 543-575

[FeSch14] M. Felderer, I. Schieferdecker, A taxonomy of risk-based testing, in [STTT14], International Journal on Software Tools for Technology Transfer, doi: 10.1007/s10009-014-0332-3

[GerTh02] P. Gerrard, N. Thompson, Risk-based e-business testing, Artech House Publishers, 2002

[ISO29119] ISO: ISO/IEC/IEEE 29119 Software Testing (2013), verfügbar über http://www.softwaretestingstandard.org/

[ISTQB] ISTQB: Standard glossary of terms used in software testing. version 2.2. Tech. rep., ISTQB (2012)

[KlHuE11] J. Kloos, T. Hussain, R. Eschbach, Risk-based testing of safety-critical embedded systems driven by fault tree analysis, in: ICSTW 2011, IEEE (2011), 26-33

[RaMo13] M. Ray, D. P. Mohapatra, Risk analysis: a guiding force in the improvement of testing, in: IET Software 7(1) (2013), 29-46

[Redm05] F. Redmill, Theory and practice of risk-based testing, in: Software Testing, Verification and Reliability 15(1) (2005), 3-20

[RoStGa00] L. Rosenberg, R. Stapko, A. Gallo, Risk-based object oriented testing, in: Proc. of 13th International Software/Internet Quality Week-QW 2 (2000)

[SchGrSch12] I. Schieferdecker, J. Grossmann, M. Schneider, Model-based security testing, in: Proceedings 7th Workshop on Model-Based Testing, 2012

[SouGuVe10] E. Souza, C. Gusmao, J. Venancio, Risk-based testing: A case study, in: Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, IEEE (2010), 1032-1037

[StMe07] H. Stallbaum, A. Metzger, Employing requirements metrics for automating early risk assessment, in: Proc. Of MeReP07, Palma de Mallorca, Spain, (2007), 1-12

[StMePo08] H. Stallbaum, A. Metzger, K. Pohl, An automated technique for risk-based test case generation and prioritization, in: Proceedings of the 3rd international workshop on Automation of software test, ACM (2008), 67-70

[STTT14] M. Felderer, I. Schieferdecker (eds.), Special Section on Risk-Based Testing of International Journal on Software Tools for Technology Transfer (STTT), Springer-Verlag, 2014

[Veen12] E. van Veenendaal, Practical Risk-Based Testing – The PRISMA Approach, UTN Publishers, 2012

[WeKrSch12] M. F. Wendland, M. Kranz, I. Schieferdecker, A systematic approach to risk-based testing using risk-annotated requirements models, in: ICSEA 2012, 636-642

[YooCh11] H. Yoon, B. Choi, A test case prioritization based on degree of risk exposure and its empirical study, in: International Journal of Software Engineering and Knowledge Engineering 21(02) (2011), 191-209

[Zech11] P. Zech, Risk-based security testing in cloud computing environments, in: ICST 2011, IEEE (2011), 411-414

[ZeFeBr12] P. Zech, M. Felderer, R. Breu, Towards risk-driven security testing of service centric systems, in: QSIC (2012), 140-143

[Zimm09] F. Zimmermann, R. Eschbach, J. Kloos, T. Bauer et al., Risk-based statistical testing: A refinement-based approach to the reliability analysis of safety-critical systems, in: EWDC 2009