mehr zum thema:
toop.gi-ev.de

WIRTSCHAFTLICHKEITS-BERECHNUNG FÜR MBT: WANN SICH MODELLBASIERTES TESTEN LOHNT

Modellbasiertes Testen verspricht potenziell eine höhere Effizienz und Effektivität im Testprozess. Ob im eigenen Kontext der Einsatz wirtschaftlich ist, ist jedoch häufig unklar. Dieser Beitrag analysiert systematisch Kosten- und Nutzenfaktoren und stellt ein Verfahren zur Abschätzung der Wirtschaftlichkeit des modellbasierten Testens vor. Anhand eines Beispiels wird der Ablauf veranschaulicht.

Das *modellbasierte Testen (MBT)* beschäftigt die Testing-Community seit Jahren ungebremst: Zahlreiche akademische und industrielle Arbeitsgruppen, Workshops und Konferenzen sind diesem Thema gewidmet und entsprechende Werkzeuge werden bereits erfolgreich in Industrieprojekten eingesetzt (vgl. z. B. Teil III MBT im Praxiseinsatz von [Roß10]).

In früheren OBJEKTspektrum-Artikeln haben die Mitglieder des Arbeitskreises TOOP/MBT der Gesellschaft für Informatik (AK TOOP/MBT) eine Einführung zum Thema MBT gegeben (vgl. [Gül10]) und über Fallstudien zum MBT berichtet (vgl. [Wei11]). Allerdings bestehen bei Projekt- und Testmanagern oft Zweifel an der Wirtschaftlichkeit des modellbasierten Testens in den eigenen Projekten. Insbesondere die Einführungskosten und der vorab nicht messbare Nutzen neuer Technologien im eigenen Anwendungskontext sind Gründe dafür. Obwohl Forschungseinrichtungen und Industrie regelmäßig über positive Erfahrungen berichten, fehlen Projekt- und Testmanagern hier konkrete Hilfsmittel für die Wirtschaftlichkeitsbetrachtung des modellbasierten Testens.

Zu diesem Zweck untersuchte der AK TOOP/MBT die Kosten- und Nutzenfaktoren für die Einführung und Nutzung des modellbasierten Testens in gängigen Szenarien und erstellte ein Rahmenwerk zum Schätzen der Wirtschaftlichkeit von MBT durch den *Return on Investment (ROI)*. Es hat sich gezeigt, dass es aufgrund der Vielfalt der Einflussfaktoren keine einfache ROI-Formel geben kann, die alle Faktoren enthält. In diesem Artikel schränken wir die Freiheitsgrade etwas ein und stellen somit ein schlankes Vorgehen sowie einen Katalog der Kosten- und Nutzenfaktoren zur Verfügung, mit dem eine individuelle Analyse der Wirtschaftlichkeit des modellbasierten Testens durchgeführt werden kann.

In diesem Artikel geben wir zunächst einen kurzen Überblick über das modellbasierte Testen und seine möglichen Ausprägungen in der Praxis. Danach stellen wir unseren Ansatz zur Abschätzungen der Wirtschaftlichkeit von MBT vor. Die dazu benötigten Kosten- und Nutzenfaktoren werden im darauf folgenden Abschnitt besprochen. Zum Abschluss demonstrieren wir anhand eines konkreten Projekts das Verfahren zur Abschätzung der Wirtschaftlichkeit.

Modellbasiertes Testen

Maßnahmen zur Qualitätssicherung von Software sind ein signifikanter Kostenfaktor der Softwareentwicklung, der typischerweise 25 bis 50 % der Gesamtkosten verschlingt – in sicherheitskritischen Domänen auch mehr. Eine höhere Effizienz ist daher von enormer wirtschaftlicher Bedeutung. MBT kann dies erfüllen (vgl. [Wei11]), wenn es in passenden Situationen auf korrekte Weise eingesetzt wird. Anderenfalls ist das Risiko hoch, dass MBT erfolglos eingesetzt und auf Jahre hinaus als unbrauchbar abgestempelt wird. Deshalb führt dieser Artikel in MBT sowie dessen Einsatzszenarien ein und baut darauf eine Kennzahlenanalyse zur Wirtschaftlichkeit auf.

Beim MBT werden für die Aktivitäten im Testprozess (formale) Modelle genutzt. Aus einem Modell generiert ein Testfall-Generator nach vorgegebenen Selektionskriterien Testfälle. Durch die Prägnanz der Modelle und ihre automatisierte Verarbeitbarkeit verbessern sich die Effizienz bei der

die autoren

David Faragó (David.Farago@kit.edu) ist wissenschaftlicher Mitarbeiter am Karlsruher Institut für Technologie und promoviert im Bereich formaler Methoden für modellbasiertes Testen.

Arne-Michael Törsel (Arne-Michael.Toersel@adesso.de) arbeitet bei der adesso AG im Bereich Softwareentwicklung und -test und promoviert an der Universität Rostock.

Dr. Michael Mlynarski (michael.mlynarski@qualityminds.de) ist Geschäftsführer und Senior Test Consultant bei der QualityMinds GmbH. Seine Forschungsbereiche sind modellbasiertes und agiles Testen.

Dr. Stephan Weißleder (Stephan.Weissleder@fokus.fraunhofer.de) arbeitet beim Fraunhofer-Institut FOKUS als Forschungsgruppenleiter im Bereich Qualitätssicherung und Test.

Baris Güldali (Baris.Gueldali@s-lab.upb.de) ist Researcher im s-lab – Software Quality Lab der Universität Paderborn mit den Forschungsbereichen Testautomatisierung, modellbasiertes Testen und Requirements-Engineering.

Dr. Christian Brandes (Christian.Brandes@imbus.de) arbeitet als Principal Consultant und Trainer für die imbus AG. Seine Schwerpunktthemen sind agiles Testen, Testautomatisierung und modellbasiertes Testen.

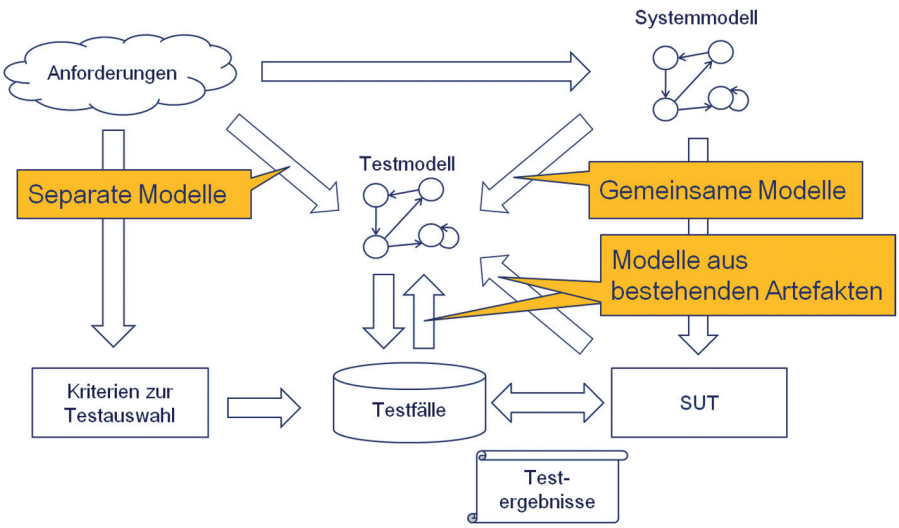

Abb. 1: Unterschiedliche Szenarien für den Einsatz von MBT (basiert auf [Pre05]).

Erstellung und Wartung von Testfällen, die Wiederverwendbarkeit von Artefakten sowie die erreichbare Testabdeckung und die Rückverfolgbarkeit zwischen den Anforderungen, Modellen, Testfällen und Testergebnissen (vgl. [Roß10]). Widersprüche und Ungenauigkeiten in den Anforderungen werden potenziell früher erkannt, da die Erstellung der Modelle die Validierung der Anforderungen unterstützt. Modelle sind zudem ein Hilfsmittel zur Kommunikation zwischen den Stakeholdern, um Unklarheiten und Testanforderungen zu diskutieren. Modellbasierte Testverfahren sind sowohl für den Komponententest, den Integrationstest als auch für den Systemtest einsetzbar.

Um Kosten und Nutzen des MBT zu analysieren, ist es zunächst notwendig, den grundsätzlichen Ablauf des modellbasierten Testprozesses speziell unter dieser Fragestellung zu klären: „Woher kommen die zu verwendenden Modelle eigentlich?" Diese Fragestellung haben wir in Form von so genannten MBT-Szenarien in Abbildung 1 dargestellt. Dabei kann zwischen folgenden Vorgehensweisen unterschieden werden (vgl. [Pre05]):

- gemeinsame System- und Testmodelle
- separate Testmodelle
- abgeleitete Modelle aus bestehenden Artefakten (Testfälle, Programmcode)

Die erste Möglichkeit besteht darin, vorhandene Systemmodelle (Modelle der Entwicklung) in Testmodelle zu überführen – das heißt, nicht benötigte Informationen aus den Modellen zu entfernen und für den Test benötigte Informationen hinzuzufügen. Dieses Vorgehen wird daher *als gemeinsame Nutzung von Systemmodellen* bezeichnet. Vorteilhaft ist hier der geringere Aufwand im Vergleich zum separat erstellten Modell. Als Nachteil ist jedoch festzuhalten, dass Fehler im Systemmodell, aus dem gegebenenfalls fehlerhafte Software erstellt wurde, nun auch im Testmodell enthalten sein können, womit diese nicht im Test gefunden werden.

Eine weitere Möglichkeit besteht darin, die Modelle zusätzlich im Entwicklungsprozess aus den Anforderungen bzw. Spezifikationen zu erstellen. Der Vorteil liegt hier in der Unabhängigkeit auf Modellebene, was einen effektiven Test erst ermöglicht, der Nachteil liegt im hohen Modellierungsaufwand. Dieses Vorgehen ist als *separate Modellierung des Testmodells* bekannt.
In der Praxis sind Mischformen aus beiden erläuterten Szenarien denkbar, in denen Aspekte des vorhandenen Entwicklermodells (zum Beispiel Informationen zur Struktur der Anwendung) um separate Aspekte (zum Beispiel des erwarteten Anwendungsverhaltens) zum Testmodell ergänzt werden.

Schließlich ist es als prinzipielles drittes Szenario auch möglich, *die Modelle (anfangs) aus bestehenden Artefakten wie der Software oder existierenden Testfällen zu erzeugen*. Der Aufwand ist hier – je nach Automatisierungsgrad – moderat bis sehr gering. Zu bedenken ist jedoch, dass die Grundlage dabei nicht die Anforderungen selbst, sondern die bereits einmal interpretierten und umgesetzten Anforderungen sind. Somit werden falsch verstandene Anforderungen nicht vom Test erfasst und der Test ist eventuell deutlich weniger aussagefähig.

Zur Bestimmung des geeigneten MBT-Szenarios für ein konkretes Projekt ist eine detaillierte Auseinandersetzung unter mehreren Aspekten notwendig, wie zum Beispiel:

- Handelt es sich um eine Neuentwicklung oder sind schon Artefakte vorhanden? Welche Testanforderungen gibt es?
- Werden Systemmodelle genutzt?
- Welche szenariospezifischen Kosten- und Nutzenfaktoren (siehe unten) schlagen in dem Projekt besonders zu Buche?

Eine weiterführende Darstellung der Grundlagen findet sich in [Gül10], eine ausführliche Abhandlung in [Utt07] oder [Roß10].

Die Wirtschaftlichkeit von MBT-Projekten abschätzen

In diesem Abschnitt beschreiben wir unser Vorgehen zur Berechnung der Wirtschaftlichkeit von MBT. Eine der populärsten und aussagekräftigsten Kennzahlen dafür ist der ROI, auch Kapital- oder Anlagenrendite genannt, der die Einnahmen (Nutzen) zum eingesetzten Kapital (Kosten) in Verhältnis setzt:

ROI = (Nutzen - Kosten) / Kosten

Zur Anwendung dieser Formel müssen die Kosten- und Nutzenfaktoren aus dem folgenden Abschnitt kontextspezifisch beziffert werden. Grundvoraussetzung sollte sein, dass der Nutzen höher als die Kosten ist (ROI > 0). Im Allgemeinen kann man aber zwischen einer negativen, neutralen und profitablen ROI unterscheiden und diese als Entscheidungsgrundlage nehmen. Konkrete Zahlen zur ROI findet man in der Literatur zu *Software Process Improvement (SPI)* in [Sol04]. Dort wurden Fallstudien untersucht, die über ROI-Werte der SPI-Anwendung berichten. Diese variierten zwischen 1,5 und 19, als Durchschnittswert für ROI wurde 7 ermittelt. Das bedeutet, dass jeder investierte Euro sieben Euro Gewinn bringt (vgl. [Sol04]). Ein allgemeiner Schwellwert für den ROI, ab dem in-

Testphase	Initiale Kostenfaktoren	Laufende Kostenfaktoren
Planung und Steuerung	• Prüfung existierender Ressourcen und Know-hows • Schulungskosten für Methoden und Werkzeuge • Anbindung an das Anforderungsmanagement, das Testmanagement sowie gegebenenfalls an die Continuous-Integration-Umgebung umsetzen (Beschaffung oder eigene Integrationslösung) • Von Beratern begleitetes Pilotprojekt	• Analyse der Testbasis bzgl. der MBT-Testbarkeit • Planung zu erstellender/anzureichernder/abzuleitender Modelle (je nach MBT-Szenario) und zu testender Aspekte • Fortwährende Prüfung der Modellqualität
Analyse und Design	• Lizenzierung von Werkzeugen für: o Modellierung o Modellvalidierung o Modellverwaltung o Testfallableitung • Initialer Modellierungsaufwand bzw. Aufwand zur Beschaffung der Modelle (siehe MBT-Szenarien)	• Modellierung • Modell-Refactoring • Modellvalidierung und -verifikation
Realisierung und Durchführung	• Lizenzierung/Schaffung von Werkzeugen für die Überführung von aus Modellen abgeleiteten (abstrakten) Testfällen in Testskripte für die Testautomatisierung	• Auswahl geeigneter Testfall-Selektionskriterien entsprechend den Zielen der Planung • Generierung und Konkretisierung der Testfälle, eventuell ist eine Anreicherung mit Testdaten notwendig • Transformation in das Format des Automatisierungswerkzeugs, falls automatisierte Testdurchführung erfolgen soll
Auswertung und Bericht	• Automatismen für Bericht einrichten (speziell Mapping-Fehler auf Modell)	• Rückverfolgung aufgetretener Fehler im Modell, um betroffene Anforderungen zu identifizieren • Dokumentation der Testendekriterien (insbesondere Testabdeckung) auf der Code-, Modell- und Anforderungsebene • Definition weiterer nötiger Testaktivitäten anhand der Modelle • Zusammenführung von MBT-Tests und „konventionellen" Tests in uniformen Bericht
Abschluss	• Archivierungsmöglichkeiten entwerfen	• Archivierung der Modelle und Testkonfigurationen • Versionsmanagement der Modelle • Dokumentation des gewonnenen Modellierungs-Know-hows (projektspezifisch und allgemein)

Tabelle 1: *Kostenfaktoren von MBT.*

vestiert werden sollte, kann jedoch nicht angegeben werden, da er kontextspezifisch ist.

Die oben genannte ROI-Formel ist die einfachste und populärste. In [Jef10] werden realistischere ROI-Formeln aufgeführt, die jedoch auch um einiges komplexer sind. Sie werden hier nicht verwendet, um sich auf die Kosten- und Nutzenfaktoren konzentrieren zu können. Nach Bezifferung der einzelnen Faktoren steht dem Leser auch die Anwendung komplexerer ROI-Formeln statt der obigen frei, mit denen die Steuerung besser möglich ist, das heisst, dass der ROI fortlaufend während des Projekts gemessen werden kann.

Zur Abschätzung des ROI schlagen wir in diesem Beitrag einen mehrstufigen Ansatz vor (siehe Abbildung 2). Zunächst ist das zutreffende bzw. geeignete Szenario für den MBT-Einsatz zu ermitteln. Das ist wichtig, weil es szenarioabhängig Abweichungen bei bestimmten Kosten und Nutzenfaktoren gibt. Im nächsten Schritt werden anhand eines Katalogs von Kosten- und Nutzenfaktoren diejenigen Faktoren ermittelt, die im eigenen Kontext relevant sind. Schließlich werden die individuell gewichteten Faktoren in die vorgeschlagene ROI-Formel eingesetzt. Damit wird ein Indikator geliefert, der einen Anhaltspunkt zur Wirtschaftlichkeit des MBT im eigenen Kontext liefert und der Möglichkeiten aufzeigt, den ROI durch eine Einflussnahme auf einzelne Faktoren zu verbessern. Wenn der ROI-Wert negativ ausfällt, sollte eine Investition in MBT vermieden werden. Wenn er positiv ist, kann ein Schwellwert festgelegt werden (x in Abbildung 2), ab dem sich die Investition in MBT lohnt. Diesen Ansatz demonstrieren wir zum Abschluss dieses Artikels anhand eines Rechenbeispiels, das sich an einem erfolgreichen Projekt mit MBT-Einsatz orientiert.

Kosten und Nutzen des modellbasierten Testens

In diesem Abschnitt analysieren wir die potenziellen Kosten- und Nutzenfaktoren als Einflussgrößen auf den ROI des modellbasierten Testens. Die Faktoren können gut anhand des fundamentalen Testprozesses (vgl. [Spi05]) in die einzelnen Testprozessphasen aufgeschlüsselt werden.

Testphase	Nutzenfaktoren
Planung und Steuerung	• Die Testbasis kann durch die Modellierung besser analysiert werden. Dadurch lassen sich die Testanforderungen besser beurteilen. • Kommunikation und Diskussion der Testaktivitäten werden erleichtert.
Analyse und Design	• Modellierung ersetzt die Erstellung abstrakter Testfälle, Aufwandsersparnis durch Automatisierung der Testfallerstellung mittels Generatoreinsatz. • Auswirkungsanalyse zur Abschätzung der Testanforderungen bei Anwendungsänderungen wird vereinfacht. • Modellierung dient gleichzeitig als Review der Testbasis und ermöglicht frühe Fehlerfindung. • Transparentes und standardisiertes Vorgehen zur Testfallerstellung senkt die Pflegeaufwände und vereinfacht den Nachweis von Testaktivitäten. • Gegebenenfalls Wiederverwendung bestehender Modelle möglich (Investitionsschutz).
Realisierung und Durchführung	• Automatisierte Umsetzung abstrakter in konkrete Testfälle mit entsprechender Werkzeugunterstützung möglich. • Erspart Implementierungsaufwand von Testskripten. • Trennung der fachlichen und technischen Sicht. • Frühzeitige Analyse der Überdeckung möglich. • Erhöhte Fehlerfindungswahrscheinlichkeit (insbesondere durch nicht offensichtliche Kombinationen in den generierten Testschritten und Testdaten).
Auswertung und Bericht	• Testabdeckung jederzeit ermittelbar.
Abschluss	• Verringerte Menge zu archivierender Artefakte.

Tabelle 2: Nutzenfaktoren von MBT.

Da Kosten und Nutzen sehr projektspezifisch sind, lassen sich diese nicht pauschal angeben. Auch ist es für viele Faktoren schwierig, die Auswirkungen zu quantifizieren. Hier können Metriken aus früheren Projekten hilfreich sein, um einen Anhaltspunkt für Schätzungen zu liefern. Vor – und auch nach – der Einführung von MBT ist es deshalb sinnvoll, die eigenen Projekte mit geeigneten Metriken zu dokumentieren. Im Folgenden stellen wir die gängigen Kostenfaktoren vor und ordnen diese den Phasen im fundamentalen Testprozess zu.

Kosten von MBT

Die Kosten der Einführung und Nutzung von MBT lassen sich in *initiale* und *laufende* Faktoren unterteilen (siehe Tabelle 1).

Nutzen von MBT

Die Kostenfaktoren sind zwar für die Einschätzung der nötigen Investitionen von entscheidender Bedeutung. Allerdings sollte der erwartete Nutzen die geschätzten Kosten nach einer gewissen Zeitspanne überwiegen (daher ROI). Die Betrachtung der Nutzenseite ergibt schnell, dass in der MBT-Literatur eine Fülle von Aspekten aufgelistet wird. In [Roß10] werden beispielsweise unter der Überschrift „Was MBT leisten kann“ ca. zehn Aspekte aufgelistet, die von einem MBT-Einsatz positiv beeinflusst werden. Eine Analyse der Aspekte ergibt, dass sich diese gut in zwei Hauptkategorien einordnen lassen.

Nutzenkategorie 1: Geringerer Aufwand zur Erstellung und Pflege von Testfällen

Die Phase „Analyse und Entwurf“ im fundamentalen Testprozess ist diejenige, auf die typischerweise der größte Aufwand entfällt (laut [Koo07] etwa im Systemtest mehr als die Hälfte des Gesamtaufwands, nämlich 54 %). Folglich ist dies nicht nur die Phase mit dem größten Einsparpotenzial, sondern genau diejenige, die von MBT adressiert wird. Werden Testfälle in Modellform spezifiziert und daraus ausführbare Testskripte generiert, wird spätestens der Generierungsschritt zu einem Kostensenkungsfaktor (dem der Aufwand der Modellerstellung gegenübersteht). Dieser Effekt tritt in späteren Testzyklen – also in der Pflegephase, etwa nach Änderungen an den Anforderungen – noch deutlicher zu Tage: Nach den nötigen Änderungen am Testmodell wird lediglich die Testgenerierung wiederholt, um die nötigen Anpassungen an den Testfällen vorzunehmen.

Nutzenkategorie 2: Qualitätssteigerung durch frühzeitige Fehlerfindung

Unter diese Kategorie fallen vor allem zwei Faktoren: Erstens werden während der Erstellung von Testmodellen frühzeitig Mängel und Fehler an textbasierten Anforderungsspezifikationen aufgedeckt und korrigiert, was bekanntermaßen enorm kostensenkend wirkt. Dieser Vorteil ergibt sich natürlich nur dann, wenn das Testmodell separat aus den Anforderungen erstellt wird (Szenario „separate Modellierung des Testmodells”). Zweitens entstehen durch den MBT-Einsatz typischerweise mehr Testfälle als durch manuelles Testdesign, die auch unerwartete Kombinationen testen. Damit steigt die Fehlerfindungswahrscheinlichkeit. Es sei allerdings bemerkt, dass das hier positiv formulierte Problem der „Testfall-Explosion“, das MBT mit sich bringen kann, meistens dazu führt, dass MBT seinen vollen Nutzen erst im Kontext einer reifen Testautomatisierung entfalten kann.

Aufgeschlüsselt auf den fundamentalen Testprozess, lassen sich die Nutzenfaktoren von MBT im Einzelnen so einordnen, wie in Tabelle 2 dargestellt.

Kosten und Nutzen in einzelnen MBT-Szenarien

Der Ursprung der verwendeten Modelle hat einen entscheidenden Einfluss darauf, welche der Kosten- und Nutzenfaktoren aus dem letzten Abschnitt dominieren bzw. gar nicht relevant sind. Zum Aufstellen der

eigenen ROI-Formel sind also nicht alle Faktoren aus den oben aufgeführten Tabellen in jedem Projekt zu berücksichtigen. Deshalb diskutieren wir im Folgenden die Auswirkungen der gängigsten MBT-Szenarien (vgl. [Pre05]) auf die vorgestellten Kosten- und Nutzenfaktoren. Wie bereits erwähnt, ist eine Quantifizierung vieler Faktoren oft nur durch eine Schätzung möglich, deren Güte unter anderem auf Erfahrungswerten aus Vergleichsprojekten beruht.

Gemeinsames Modell
Aus einem gemeinsamen Modell werden Code und Testfälle erstellt. Dieses Szenario wirkt sich folgendermaßen auf die *Kosten* aus:

- Es entstehen nur einmalige Kosten für die Werkzeug-Infrastruktur zur Modellverwaltung und -erstellung, da die gleichen Werkzeuge für System- und Testmodelle verwendet werden können.
- Genauso entstehen nur einmalige Kosten für die Schulung der Mitarbeiter.
- Ein Modellierungsaufwand sowie ein Aufwand zur Modellpflege entfallen, gegebenenfalls sind Anreicherungen bzw. Bereinigungen notwendig.

Das Szenario wirkt sich folgendermaßen auf den *Nutzen* aus:

- Ein Review der Testbasis ist durch die Modellierung für die Entwicklung bereits erfolgt. Hier gibt es keinen weiteren Nutzengewinn für eine frühe Fehlerfindung.
- Die für funktionales Testen nötige Unabhängigkeit zwischen System- und Testmodellen (siehe oben) ist nicht vorhanden. Ein Nutzen der Fehlerfindung in der Anwendung entfällt. Ermöglicht wird stattdessen die Verifikation eines Code-Generators und der Annahmen über die Softwareumgebung – dies ist ein szenariospezifischer Nutzen. Es wird angenommen, dass das Modell valide ist, und werkzeuggestützt verifiziert wird, damit ist die Anwendung „korrekt", da sie aus dem korrekten Modell durch den verifizierten Codegenerator erstellt wurde und nicht funktional getestet werden muss.

Separates Modell
Aus den Anforderungen wird manuell ein Modell erzeugt, ein Selektionskriterium zur Testgenerierung gewählt sowie die Spezifikation für den Code erstellt.

Dieses Szenario wirkt sich folgendermaßen auf die *Kosten* aus:

- Die Werkzeugkosten sind nur dann niedrig, falls die Werkzeug-Infrastruktur für die Codespezifikation für das Modell zur Testgenerierung wiederverwendet werden kann.
- Die Schulungskosten sind nur dann niedrig, falls das Modell zur Testgenerierung Ähnlichkeiten mit der Codespezifikation aufweist.
- Durch die separaten Modelle für den Test entstehen hohe Kosten für Modellerstellung, -validierung und -verwaltung. Gibt es eine Wiederverwendung zwischen den Modellen und der Codespezifikation, so können diese Kosten zwar gesenkt werden, aber die Unabhängigkeit und damit Effektivität der Testgenerierung sinkt zwangsweise (vgl. Abschnitt über Mischformen von Szenarien).

Das Szenario wirkt sich folgendermaßen auf den *Nutzen* aus:

- Durch die hohe Unabhängigkeit zwischen System- und Testmodellen ist eine effektive Testgenerierung möglich.
- Durch die Entkopplung der Modelle können unterschiedliche Teams beauftragt werden.
- Modelle können abstrakt sein, es entfällt alles, was für die Testgenerierung nicht notwendig ist. Dadurch können die Übersichtlichkeit und Testabdeckung erhöht werden. Wird noch mehr abstrahiert, verschieben sich die Kosten in die Konkretisierung der Testfälle bei der Testadapter-Entwicklung.

Modelle aus den bestehenden Artefakten
Für die Erstellung des Testmodells werden bestehende Artefakte wie Testfälle oder Programmcode verwendet.

Dieses Szenario wirkt sich folgendermaßen auf die *Kosten* aus:

- Werkzeuge für das (automatisierte) Re-Engineering müssen eingekauft oder selbst entwickelt werden.
- Die Korrektheit und Vollständigkeit des Testmodells gegenüber den Anforderungen müssen geprüft werden, z. B. durch manuelle Reviews.

Dieses Szenario wirkt sich folgendermaßen auf den *Nutzen* aus:

- Da die Testmodelle auf der Basis von vorhandenen Artefakten entstehen, wird existierendes Know-how wiederverwendet.
- Da die Testmodelle direkt aus der Implementierung oder den vorhandenen Testfällen entstehen, ist es offen-

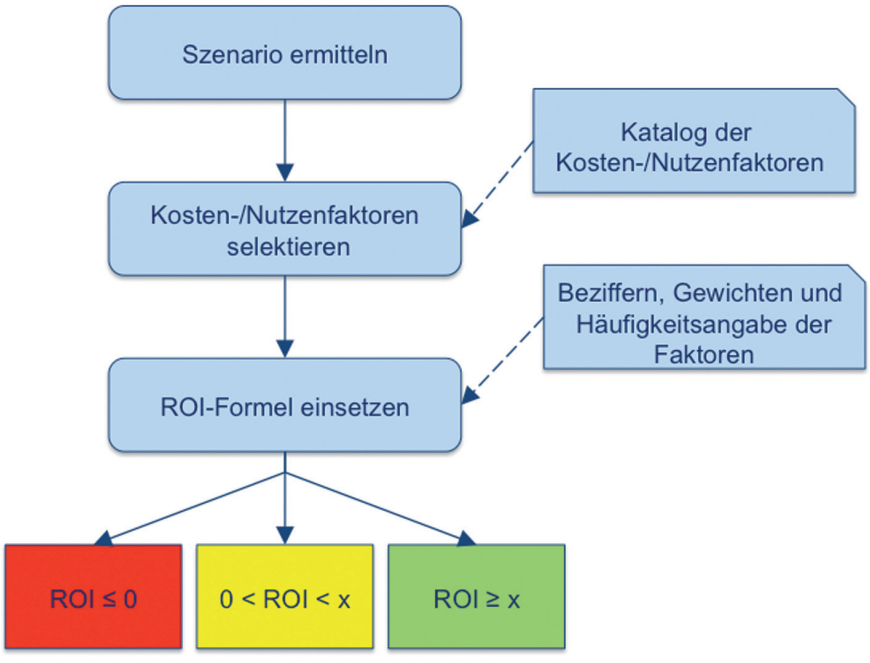

Abb. 2: Vorgehen zur ROI-Abschätzung für MBT-Projekte.

Testphase	Kostenfaktoren	Nutzenfaktoren
Planung und Steuerung	• Vernachlässigbare Kosten für die Anbindung, da Spec Explorer als kostenfreies Plug-in für VisualStudio (VS) angeboten wird. • Moderate Kosten zur Prüfung existierender Ressourcen und zur Testplanung, da gegebenenfalls viele informelle Spezifikationen und Anforderungen vorhanden sind. • Relativ geringe Kosten für Schulungen, denn SpecExp ist sehr anwendungsfreundlich und PQAP ermöglicht ein systematisches Training-On-The-Job durch Senior-Reviewer. • Relativ geringe Kosten für die fortlaufende Prüfung der Modellqualität, da Spec Explorer dies durch spezielle Sichten (Views und Slicing, vgl. [Gri11]) vereinfacht, und PQAP die Steuerung effizient durch Senior- Reviewer organisiert.	• Testaktivitäten konnten dank separater Testmodelle gut koordiniert werden. Insbesondere war Outsourcing zu externen, günstigen Testteams möglich. • Mit PQAP war eine effiziente Steuerung möglich.
Analyse und Design	• Moderate laufende Kosten zur Modellierung, da in SpecExp die Testmodelle abstrakt gehalten werden können. • Geringe Kosten für Modell-Refactoring, da Testmodelle in C# erstellt werden, d. h. ein automatisiertes Refactoring von VS ist nutzbar. • Dank Sichten in SpecExp haben Modellvalidierung und -verifikation relativ geringe Kosten. • Vernachlässigbare Lizenzkosten, falls VS schon eingesetzt wird, andernfalls moderat.	• Modellierung ersetzt manuelle Erstellung abstrakter Testfälle und ermöglicht eine frühe Fehlerfindung. • Prägnante Modelle erleichtern den Pflegeaufwand der Tests, insbesondere, wenn sich Anforderungen ändern.
Realisierung und Durchführung	• Relativ geringe Kosten für Testfallselektionskriterien: SpecExp bietet Zustands-, Kanten- und Interaktionsüberdeckung sowie verschiedene Sichten. • Vernachlässigbare Kosten zur Generierung abstrakter Testfälle. • Übersetzung von abstrakten zu konkreten Testfällen wird im Testadapter durchgeführt. Das ist auch in SpecExp recht aufwändig, sodass hohe Kosten entstehen, falls nicht schon vorhandene Testadapter wiederverwendet werden können.	• Automatisierte, frühe Überdeckung erhöht Fehlerfindungswahrscheinlichkeit bzw. Vertrauen in Korrektheit. • Einfache aber exakte Rückverfolgbarkeit.
Auswertung und Bericht	Geringe Kosten für Bericht, da SpecExp die Protokollierung und Rückverfolgbarkeit gut umsetzen kann.	Automatisierung ermöglicht einen ausführlichen, exakten Bericht, inklusive Nachweis von Testaktivitäten, insbesondere mittels Werten zum Überdeckungsgrad.
Abschluss	Geringe Kosten für die Archivierung: Da die Testmodelle in C# geschrieben sind, können die Werkzeuge zur Quelltext-Verwaltung für die Modelle wiederverwendet werden.	Abstrakte Testmodelle verringern die Menge zu archivierender Artefakte.

Tabelle 3: Kosten- und Nutzenfaktoren für das Beispiel.

sichtlich, welche Anforderungen und Teile der Implementierung schon überdeckt sind und welche von neuen Testfällen noch überdeckt werden müssen.

Rechenbeispiel

In diesem Abschnitt veranschaulichen wir, wie die drei Schritte des Ablaufdiagramms aus Abbildung 2 prinzipiell durchgeführt werden, um einen ROI-Wert zu erhalten. Um die bisher bewusst abstrakt gehaltenen Faktoren zu konkretisieren, verwenden wir in diesem Abschnitt nicht beliebige Werte aus einer fiktiven Situation, sondern orientieren uns grob an einem der bekanntesten

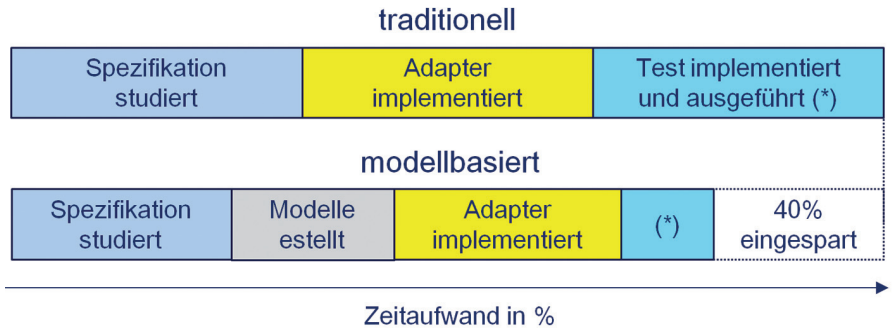

Abb. 3: Vergleich des Zeitaufwands zwischen traditionellem Test und MBT im PDP.

und größten Beispiele für MBT-Projekte, ohne dabei zu sehr ins Detail zu gehen: dem *Protocol Documentation Program (PDP)* von Microsoft, das in [Gri11] beschrieben und in [Wei11] zusammengefasst ist.

Schritt 1: Szenario ermitteln
Im PDP wurde mittels Systemtest die Konformität einer Suite von Protokollspezifikationen zu den entsprechenden Protokollimplementierungen getestet. Da die Implementierungen schon vorhanden waren, eignet sich das Szenario mit separatem Testmodell am besten. Durch vorhandenen Legacy-Code und fehlende Modelle ist diese Situation häufig in Projekten gegeben, insbesondere bei der Einführung von MBT.

Schritt 2: Kosten- und Nutzenfaktoren selektieren
Da PDP sehr erfolgreich war, setzen wir die gleichen Mittel ein (in [Gri11] beschrieben): den *Protocol Documentation Quality Assurance Process (PQAP)* und das Werkzeug *Spec Explorer (SpecExp)* von Microsoft. Wie diese Wahl und andere Aspekte des PDP sich auf die Kosten- und Nutzenfaktoren auswirken, ist in Tabelle 3 dargestellt.

Schritt 3: In ROI-Formel einsetzen
Da die konkreten Kosten- und Nutzenfaktoren des PDP nicht gemessen bzw. publik gemacht wurden, müssen wir hier schätzen, orientieren uns aber dabei an den bekannten Größenverhältnissen des Projektaufwands von PDP (siehe Abbildung 3 und [Sto11]). Um nicht nur abstrakte Gewichte, sondern konkrete Geldwerte nennen zu können, gehen wir von einem ähnlichen Gesamtaufwand aus (250 Personenjahre innerhalb von zwei Kalenderjahren) und legen einen Satz von 100€ pro Personenstunde zu Grunde. Im Allgemeinen ist man für eine ROI-Berechnung vor dem Projektstart in einer ähnlichen Situation: Man hat keine Details, sondern muss schätzen und orientiert sich dabei an nur groben Informationen. In Tabelle 4 haben wir die Kosten- und Nutzenfaktoren aus Schritt 2 grob beziffert.

Damit ergibt sich ein ROI von *(110 - 40) / 40 = 1,75*. Folglich ist das Projekt gewinnbringend und sollte finanziert werden, falls das Gesamtbudget nicht durch noch rentablere Projekte aufgebraucht wird (siehe unterer Teil von Abbildung 2).

Um die Rentabilität zu untermauern, sind in Abbildung 3 die Werte, die Microsoft gemessen und herausgegeben hat, aufgeführt (siehe [Sto11]). Dieser Vergleich wurde ermöglicht, indem bei gleicher Anforderungsüberdeckung einige Protokolle traditionell getestet wurden und andere mittels MBT.

Testphase	Kostenfaktoren	Nutzenfaktoren
Planung und Steuerung	• Werkzeuge und deren Anbindung: 0 € • Schulungen: 1,5 M € • Senior-Reviewer für fortlaufende Prüfung: 1,5 M € • Erstellung des Testplans und PQAP-Anpassung: 3 M €	• durch Outsourcing: 10 M € • effiziente Steuerung mit PQAP: 4 M €
Analyse und Design	• für Modellerstellung: 9 M € • für Modellvalidierung: 2 M € • für Modellverwaltung: 1 M €	• frühe Fehlerfindung: 10 M € • geringerer Pflegeaufwand: 10 M €
Realisierung und Durchführung	• Testfallselektionskriterien: 2 M € • Generierung abstrakter Testfälle: 0 € • Testadapter: 12 M € • Durchführung: 6 M €	• wie bei klassischem Testen: 50 M € • höhere Testabdeckung: 10 M € • Einfache aber exakte Rückverfolgbarkeit: 4 M € • durch Zeitersparnis wegen effektiverer Testgenerierung: 10 M €
Auswertung und Bericht	automatisierten Bericht erstellen und auswerten: 1 M €	durch Automatisierung der Berichterstellung: 1 M €
Abschluss	Archivierungskosten: 1 M €	durch geringeren Archivierungsaufwand: 1 M €
Summe	40 M €	110 M €

Tabelle 4: Beziffern und Schätzen der Faktoren.

Fazit

Wir haben gezeigt, wie ein Vorgehen zur Wirtschaftlichkeitsanalyse einer MBT-Einführung aussehen kann. Wünschenswert ist eine schlanke, aber aussagekräftige Kennzahlenanalyse, welche die Investitionen in MBT nachweisbar rechtfertigt oder ablehnt und bei einer Einführung von Anfang an mit Kennzahlen zur Wirtschaftlichkeit und zur Zielerreichung unterlegt.

Die Schwierigkeit einer solchen Wirtschaftlichkeitsbetrachtung liegt häufig in der Komplexität der so genannten ROI-Formeln. Da es keine allgemeine Formel geben kann, die mit fixen Zahlen alle Situationen mit allen erdenklichen Aspekten abdeckt, empfehlen wir eine projektspezifische zu definieren.

Das Beispielszenario hat einen ersten Eindruck vermittelt, wie eine solche Wirtschaftlichkeitsanalyse aussehen kann und wie dabei ein ROI-Wert entsteht. Es soll dazu inspirieren, solide Analysen durchzuführen, um dann gegebenenfalls MBT nutzbringend einzusetzen. Empfehlenswert ist die fortwährende Analyse des ROI im laufenden Projekt.

Einladung und Dank

Haben Sie für Ihren (potenziellen) Einsatz von MBT eine Wirtschaftlichkeitsanalyse oder gar fortlaufende ROI-Messungen durchgeführt? Dann steuern Sie diese doch auf [MBT] bei. Je mehr Fallstudien gesammelt werden, desto mehr profitiert jeder in der Testing-Community. Diese wiederum besteht aus einigen MBT-Experten, die Ihnen gerne beratend zur Seite stehen.

Wenn Sie sich für die Nutzung des vorgestellten Berechnungsverfahrens für die Wirtschaftlichkeit von MBT oder weiterführende Aspekte des modellbasierten Testens interessieren, laden wir Sie herzlich ein, über die Homepage des Arbeitskreises TOOP/MBT (http://toop.gi-ev.de) Kontakt mit uns aufzunehmen und an den regelmäßigen Arbeitskreistreffen teilzunehmen.

Wir bedanken uns herzlich bei Wolfgang Grieskamp, Nico Kicillof und den Teilnehmern des Arbeitskreises TOOP/MBT der GI (besonders bei Prof. Dr. Mario Winter sowie Florian Prester) für ihre ausführlichen Reviews und Anregungen zu Vorversionen dieses Artikels.

Literatur & Links

[Gri11] W. Grieskamp, N. Kicillof, K. Stobie, V. Braberman, Model-based Quality Assurance of Protocol Documentation: Tools and Methodology. in: STVR, Vol. 21, Issue 1, 2011

[Gül10] B. Güldali, S. Jungmayr, M. Mlynarski, S. Neumann, M. Winter: Starthilfe für modellbasiertes Testen, in: OBJEKTspektrum 3/2010

[Jef10] M. Jeffery: Return on Investment Analysis, in: H. Bidgoli (Hrsg.), The Handbook of Technology Management: Volume 1, John Wiley & Sons 2010

[Koo07] T. Koomen, L. van der Aalst, B. Broekman, M. Vroon, TMap Next, dpunkt.verlag 2007

[MBT] Model-Based Testing Community (MBT), siehe: model-based-testing.info

[Pre05] A. Pretschner, J. Philips, Methodological Issues in Model-Based Testing, in: M. Broy, et.al. (Eds.), Model-Based Testing of Reactive Systems, Springer-Verlag 2005

[Roß10] T. Roßner, C. Brandes, H. Götz, M. Winter, Basiswissen Modellbasierter Test, dpunkt.verlag 2010

[Sol04] R. v. Solingen, Measuring the ROI of Software Process Improvement, IEEE Software Mai/Juni 2004

[Spi05] A. Spillner, T. Linz, Basiswissen Softwaretest (3. überarbeitete Auflage), dpunkt.verlag 2005

[Sto11] K. Stobie Doyenz, Model-Based Testing in Industry. Keynote zur MBTUC, 2011, siehe: model-based-testing.de/mbtuc11/presentations/Keynote_Stobie_Microsoft_MBTUC2011.pdf

[Utt07] M. Utting, B. Legeard, Practical Model-Based Testing: A Tools Approach, Morgan Kaufmann 2007

[Wei11] S. Weißleder, B. Güldali, M. Mlynarski, A.-M. Törsel, D. Faragó, F. Prester, M. Winter: Modellbasiertes Testen: Hype oder Realität?, in: OBJEKTspektrum 6/2011