

Dynamische Ermittlung der Softwarequalität zur Unterstützung der Testfallpriorisierung mit einem agentenbasierten Ansatz

Yang Yang
Logica Deutschland GmbH & CO. KG
yang.yang@logica.com

Christoph Malz, Peter Göhner
IAS, Universität Stuttgart
christoph.malz@ias.uni-stuttgart.de
peter.goehner@ias.uni-stuttgart.de

Lothar Beller, Thomas Kerler
TEP, ZF Friedrichshafen AG
lothar.beller@zf.com
thomas.kerler@zf.com

1 Einleitung

Angesichts der kürzeren Entwicklungszeiten und steigender Anforderungen an die Software stehen Softwareentwickler der Herausforderung gegenüber, qualitativ hochwertige Softwareprodukte zu liefern und dabei die Kundenzufriedenheit beizubehalten oder sogar zu steigern. Infolgedessen müssen bestimmte Qualitätssicherungsmaßnahmen für den Entwicklungsprozess übernommen werden. Eine wichtige Maßnahme davon ist der Softwaretest, der in Verbindung mit der Erfassung weiterer Daten aus der Softwareentwicklung sicherstellt, dass die integrierte Software alle Qualitätsanforderungen erfüllt. Dabei ist der Testmanager für das Management der Testaktivitäten, -ressourcen und die Bewertung des Testobjekts verantwortlich.

In Zeiten der hochkomplexen Systeme steht im Softwaretest eine große Menge von Testfällen und daraus resultierender Daten zur Verfügung. Angesichts der Verteilung und Größe von Softwareprojekten und besonders der Notwendigkeit von Testzyklen aufgrund von Softwareänderungen ist eine effektive Auswahl der Testfälle sehr schwierig. Häufig fällt es dem Testmanager bei seiner Tätigkeit schwer zu beurteilen, wann der Softwaretest ausreichend ist. Infolgedessen ist es notwendig, eine richtige Testfallpriorisierung zu gewährleisten. Dabei spielt eine verlässliche Aussage über die aktuelle Qualität der Software eine entscheidende Rolle.

Im Folgenden wird ein Konzept zur Unterstützung des Testmanagers bei seiner Tätigkeit zur Bewertung des Testobjekts sowie zur Priorisierung der Testaktivitäten vorgestellt. Als Grundlage dient dabei ein agentenbasiertes Testmanagementsystem (ATMS) [MaJa10]. Dieses System ist in der Lage, eine während der Produktentwicklungsphasen dynamisch ermittelte Softwarequalität zu ermitteln, und diesbezüglich dem Testmanager Vorschläge für die Priorisierung der Testfälle zur Verfügung zu stellen.

2 Softwaretest bei der ZF Friedrichshafen AG

Das hier vorgestellte Konzept bezieht sich auf eine Zusammenarbeit zwischen dem Institut für Automatisierungstechnik und Softwaretechnik der Universität Stuttgart und der Abteilung TEP der ZF Friedrichshafen AG. Beim Softwaretest in dieser Zusammenarbeit handelt es sich um den Software- und Systemfunktionstest von Steuergeräten und Steuergeräte-Prototypen innerhalb der Entwicklung

von PKW-Automatikgetrieben. Dabei wird der Softwaretest in der funktionalen Sicht durchgeführt. Getestet werden verschiedene Funktionsthemen im Fahrzeug-Getriebe-System. Das Testmanagement orientiert sich auf die Testaktivitäten dieser Funktionsthemen. Diese Funktionsthemen werden im Testmanagement auch als Testthemen bezeichnet. Ein Testthema ist hier nichts anders als eine Menge von Testfällen, die bezogen auf einer bestimmten Funktionalität des Fahrzeug-Getriebe-Systems durchzuführen sind.

Allerdings besteht hier auch der Zusammenhang zwischen der funktionalen Sicht und der Softwaresicht. Ein Funktionsthema besteht aus einem oder mehreren Softwaremodulen (siehe Abbildung 1).

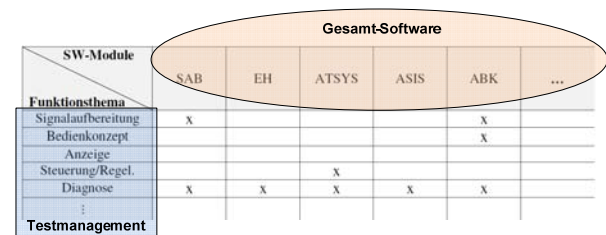


Abbildung 1: Mapping zwischen der funktionalen Sicht und der Softwaresicht

2.1 Qualitätsbewertung der Testthemen

Ziel dieser Arbeit ist es, die Effizienz und die Wirksamkeit der Testaktivitäten durch dynamische Ermittlung der Softwarequalität des Produktes während des Entwicklungsprozesses zu erhöhen. Softwarequalität im Sinne der Arbeit ist in erster Linie die Softwarequalität einzelner Testthemen und des Weiteren die Softwarequalität des Fahrzeug-Getriebe-Systems.

Die eigentliche Bewertung der Qualität geschieht anhand eines Qualitätsmodells. Die Qualitätsbewertung dieser Arbeit basiert auf dem Qualitätsmodell in [ISO / IEC 9126]. Dieses Modell betrachtet sowohl die funktionalen als auch die nichtfunktionalen Kriterien der Software. Die Testaktivitäten der ZF Friedrichshafen AG fokussieren auf die funktionalen Aspekte (Funktionalität, Zuverlässigkeit) der Software. Darüber hinaus soll die Änderbarkeit der Software mitberücksichtigt werden, da wir den Rückschluss ziehen, dass in Software mit schlechter Änderbarkeit mehr Fehler bei Softwareänderungen zu erwarten sind. Daher soll in dieser Arbeit die Softwarequalität in Hinsicht auf die Funktionalität, die Zuverlässigkeit sowie die Änderbarkeit bewertet werden.

2.2 Priorisierung der Testthemen

Kernaufgabe der Priorisierung ist die Optimierung des Testaufwandes und die Erhöhung der Produktqualität. Großes Potenzial liegt unter anderem in der Bestimmung der optimalen Reihenfolge zu testender Testthemen. Zur Planung der Testaktivitäten ist für den Testmanager in der ersten Linie von Interesse, die Priorität der Testthemen zu bestimmen.

Zur Bestimmung der Priorität einzelner Testthemen müssen zuerst alle relevanten Einflussfaktoren aufgestellt werden, die zur Steigerung der Systemqualität die wesentlichen Rollen spielen. Nach der Diskussion mit dem Experten bei der ZF Friedrichshafen AG wurden 6 relevante Einflussfaktoren auf die Priorisierung abgeleitet. In Tabelle 1 sind alle Einflussfaktoren inklusive einer Beschreibung aufgeführt, die zur Priorisierung von Testthemen herangezogen werden.

Einflussfaktor	Beschreibung
Qualität	Die Qualität eines Testthemas zeigt die aktuelle funktionale Qualität des Testthemas in Bezug auf die Funktionalität, Zuverlässigkeit sowie die Änderbarkeit des Testthemas.
Sicherheitsrelevanz	Die Sicherheitsrelevanz eines Testthemas ist der mittlere Wert der Sicherheitsrelevanz aller Funktionalitäten, die das Testthema beinhaltet.
Implementierungskomplexität	Die Implementierungskomplexität eines Testthemas vertritt sowohl die Programmgröße als auch die Programmkomplexität des Testthemas aus der Softwaresicht.
Änderungshäufigkeit	Die Änderungshäufigkeit eines Testthemas basiert auf der Anzahl der Änderungen des Testthemas sowie den Zeitraum, in dem diese Änderungen vorgelegt wurden.
Fehleranfälligkeit	Die Fehleranfälligkeit eines Testthemas stellt die Fehlerauftrittswahrscheinlichkeit des Testthemas dar. Dabei werden die Implementierungskomplexität, die Änderungshäufigkeit sowie die White-Box-Testabdeckung des Testthemas berücksichtigt.
Vertrauensgrad	Der Vertrauensgrad eines Testthemas zeigt sowohl die Testvollständigkeit als auch die Fehlerdichte des Testthemas.

Tabelle 1: Einflussfaktoren auf die Priorisierung von Testthemen

2.3 Herausforderungen beim Softwaretest

Im Laufe des Softwaretests bei der ZF Friedrichshafen AG werden bereits zahlreiche Daten für eine Qualitätsbewertung bzw. eine Priorisierung erzeugt und in Datenbanken gespeichert. Diese Daten stehen somit zur Verfügung, aber angesichts der Anzahl und Komplexität der Daten sowie der Zeitanprüche im Softwaretest ist es nicht möglich für einen Testmanager, alle diese Daten auszuwerten. In diesem Beitrag wird ein agentenbasierter Ansatz vorgestellt, bei dem Softwareagenten diese Auswertung übernehmen und anhand der Daten relevante Informationen für die Softwarequalitätsbewertung sowie die Testthemenpriorisierung ableiten und damit die Bewertung und Priorisierung durchführen.

3 Konzept des ATMS

3.1 Softwareagenten

Das Konzept in dieser Arbeit basiert auf dem Paradigma der agentenorientierten Softwareentwicklung. Bei der agentenorientierten Softwareentwicklung wird ein System als eine Menge

von autonomen Agenten betrachtet, die selbstständig innerhalb ihres Entscheidungsrahmens handeln und dabei vorgegebene Ziele verfolgen. Agenten können miteinander flexibel interagieren und durch Verhandlungen kooperieren, um ihre individuellen Ziele zu erreichen [UWG03].

Das Verhalten eines Agenten wird prinzipiell durch die *Ziele* bestimmt, die der Agent erreichen soll. Um eigene Entscheidungen bezüglich der auszuführenden Aktionen treffen zu können, muss ein Softwareagent über alle benötigten Informationen über und aus seiner Umgebung verfügen. Diese werden im *Umgebungsmodell* beschrieben. Die möglichen internen Aktionen des Softwareagenten sowie die möglichen Interaktionen mit seiner Umgebung, die ein Softwareagent ausführen kann, werden durch seine *Fähigkeiten* festgelegt. Das *Verhalten* eines Softwareagenten besteht darin, die Informationen aus seiner Umgebung zu erfassen, diese hinsichtlich der vorgegebenen Ziele zu evaluieren, sowie die geeigneten Aktionen und Interaktionen auf Basis der Fähigkeit auszuwählen und anschließend auszuführen [WGU03]. Abbildung 2 stellt die grundlegende Struktur eines Agenten dar.

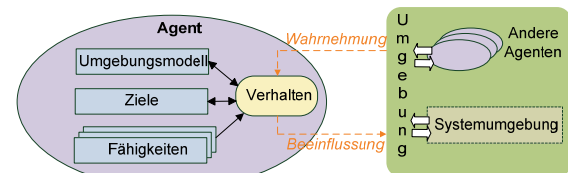


Abbildung 2: Grundlegende Struktur eines einzelnen Agenten

3.2 Agentenbasiertes Testmanagementsystem

Die Hauptaufgabe des agentenbasierten Testmanagementsystems (ATMS) ist die Unterstützung des Testmanagers durch Automatisierung der Bewertung der Testthemen sowie der Priorisierung der Testthemen. Dazu müssen verschiedene Daten der Testthemen, der entsprechenden Softwaremodulen und des Gesamtsystems erfasst und verarbeitet werden. Diese Aufgabe wird im ATMS durch die Softwareagenten übernommen.

3.2.1 Agententypen des ATMS

Folgende Agententypen, die für die Ausführung der Aufgaben des ATMS benötigt werden, wurden herausgearbeitet (siehe Abbildung 3):

- **System-Agent:** Das zu testende Softwarefunktionssystem wird durch einen System-Agenten repräsentiert. Dabei übernimmt der System-Agent globale Aufgaben für das Softwarefunktionssystem.
- **Module-Agent:** Ein Softwarefunktionssystem besteht aus mehreren Testthemen. Hinter jedem Testthema stecken die zugehörigen Softwaremodule, über die viele relevante Informationen in der Softwaresicht vorhanden sind.

Sie werden durch einen Module-Agenten vertreten, der alle relevanten Informationen des Testthemas in der Softwaresicht auswertet.

- **Testthema-Agent:** Ein Testthema wird in der funktionalen Sicht gegen die Anforderungen sowie die Änderungen getestet. In das Testthema fließen viele relevante Informationen während des Testverlaufs ein. Ein Testthema-Agent vertritt das Testthema mit allen relevanten Informationen in der funktionalen Sicht.
- **Master-Agent:** Für jedes Testthema existiert ein Master-Agent. Der Master-Agent holt die Informationen bei den Module-Agenten sowie den Testthemen-Agenten ein, und übernimmt die Aufgaben der Qualitätsbewertung sowie der Priorisierung von Testthemen.

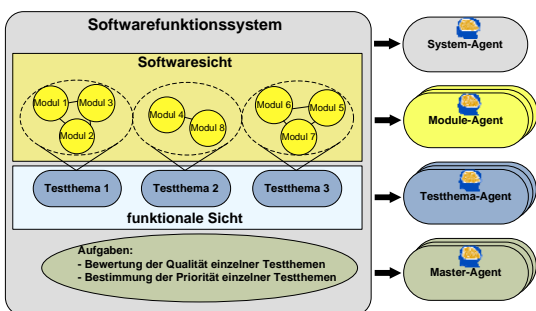


Abbildung 3: Agententypen des ATMS

3.2.2 Arbeitsweise des ATMS

Abbildung 4 zeigt die Übersicht über die Interaktionen der Agenten. Jedes Testthema im Softwarefunktionssystem wird durch einen Testthema-Agenten, einen Module-Agenten und einen Master-Agenten vertreten. Darüber hinaus bestehen Verbindungen zwischen den Softwareagenten und den Datenbanken, die bereits bei der ZF Friedrichshafen AG existieren.

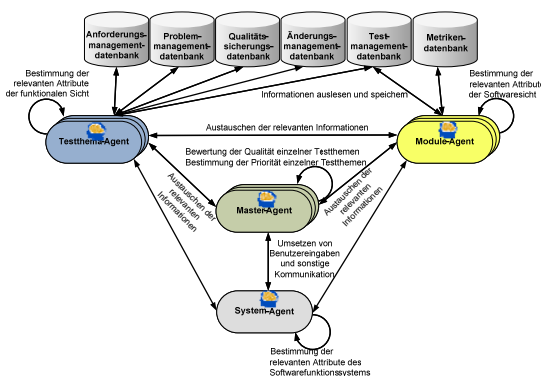


Abbildung 4: Interaktionen der Agenten

Der Module-Agent holt aus den Datenbanken die Informationen des Testthemas in der Softwaresicht, bestimmt die relevanten Attribute des Testthemas in der Softwaresicht und gibt diese ermittelten Attribute gleichzeitig an den Testthema-Agenten sowie den Master-Agenten weiter. Der Testthema-Agent holt aus

den Datenbanken die dynamischen Informationen des Testthemas in der funktionalen Sicht während des Testverlaufs, erhält die Informationen in der Softwaresicht von den Module-Agenten, bestimmt die relevanten Attribute des Testthemas in der funktionalen Sicht und gibt die ermittelten Attribute an den Master-Agenten weiter. Der Master-Agent erhält die relevanten Attribute des Testthemas in der Softwaresicht durch den Module-Agenten sowie in der funktionalen Sicht durch den Testthema-Agenten und übernimmt die Aufgaben, die Qualität des Testthemas zu bewerten sowie die Priorität des Testthemas zu bestimmen. Die ermittelte Qualität sowie Priorität einzelner Testthemen werden an den System-Agenten weitergegeben. Der System-Agent vertritt das gesamte Softwarefunktionssystem und übernimmt die globalen Aufgaben für das System, und bildet gleichzeitig eine Schnittstelle zur Kommunikation mit dem Benutzer. Die Benutzerentscheidungen wie die Gewichtung einzelner Qualitätsmerkmale sowie die Gewichtung einzelner Einflussfaktoren auf die Priorisierung können durch ihn den anderen Agenten mitgeteilt werden.

3.3 Eingesetzte Methoden

Die Softwareagenten des ATMS sind in der Lage, die identifizierten relevanten Informationen aus ihrer Umgebung zu erfassen und diese hinsichtlich der Qualitätsbewertung sowie der Prioritätsbestimmung zu evaluieren. Dazu benötigen die Softwareagenten das entsprechende Wissen, um erfasste Daten bezüglich der Qualitätsbewertung sowie der Prioritätsbestimmung auszuwerten. Zur Modellierung des Wissens von Softwareagenten sind geeignete Methoden einzusetzen.

Bei der Bewertung der Qualität sowie der Priorisierung von Testthemen sind viele relevanten Metriken zu berücksichtigen. Es ist überaus aufwendig, das Zusammenspiel dieser Metriken durch eine mathematische Beschreibung zu definieren. Häufig ist der Zusammenhang zwischen den Metriken und den zu interpretierenden Aspekten mathematisch nicht darstellbar. Darüber hinaus stehen häufig nur unscharfe Regeln zur Qualitätsbewertung sowie zur Prioritätsbestimmung zur Verfügung. Zum Lösen dieser Probleme kommt die Fuzzy-Logik [Lippe06] zum Einsatz.

Hinter jedem Softwareagenten steckt eine Menge von Fuzzy-Inferenz-Systemen (FIS) (siehe Abbildung 5). Diese ermöglichen es, das verbale Expertenwissen zur Qualitätsbewertung sowie zur Prioritätsbestimmung zu modellieren.

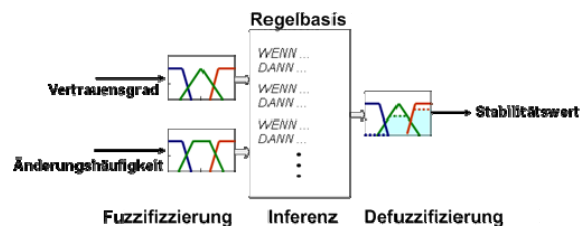


Abbildung 5: Fuzzy-Inferenz-System

In der Regelbasis eines FIS stehen die aufgestellten Regeln zur Qualitätsbewertung bzw. zur Prioritätsbestimmung zur Verfügung. Die identifizierten Daten werden als Eingangsgrößen des FIS zuerst durch die Zugehörigkeitsfunktionen fuzzifiziert. Diese fuzzifizierten Größen fließen in die Regelbasis ein. Diese werden dann in einem Defuzzifikationsvorgang bearbeitet. Als Ausgang des FIS wird eine konkrete Zahl zur Beschreibung eines qualitativen Merkmals geliefert.

Die quantifizierten Qualitätsmerkmale besitzen bei der Qualitätsbewertung unterschiedliche Gewichte. So ist z. B. die Funktionalität eines Testthemas etwas wichtiger als ihre Änderbarkeit. Darüber hinaus haben die Einflussfaktoren unterschiedliche Gewichte bei der Prioritätsbestimmung. Zur Festlegung der Gewichte der Qualitätsmerkmale sowie der Gewichte der Einflussfaktoren auf die Priorisierung wurde der von Thomas Saaty 1980 entwickelte Analytic Hierarchy Process (AHP) [LFRM09] verwendet. Mit diesem Prozess werden relevante Faktoren bezüglich ihrer Einflüsse systematisch gewichtet.

Mit Hilfe von Fuzzy-Logik und der AHP-Methode bietet das ATMS die Möglichkeit, einen generierten Qualitätswert für jedes Testthema sowie für das gesamte Softwarefunktionssystem zu liefern, sowie eine dynamische anpassbare Priorisierung der Testthemen zu schaffen.

4 Prototyps des ATMS

Zur Realisierung des Konzepts wurde ein Prototyp des agentenbasierten Testmanagementsystems erstellt, das in der Lage ist, die Qualität einzelner Testthemen, die Priorität einzelner Testthemen sowie die Qualität des gesamten Softwarefunktionssystems zu ermitteln.

4.1 Werkzeugunterstützung

Der Prototyp wurde in Java SE entwickelt. Zur Entwicklung eines Agentensystems wurden Bibliotheken mittels JADE (Java Agent Development Framework) zur Verfügung gestellt. JADE ist ein in Java implementiertes Software-Framework. Es vereinfacht die Implementierung von Agentensystemen und ermöglicht die Entwicklung der agentenbasierten Software nach den FIPA (Foundation for Intelligent Physical Agents) Spezifikationen. Darüber hinaus wurde die Open Source Bibliothek jFuzzylogic verwendet, um Fuzzy-Inferenz-Systeme innerhalb von Agenten zu erstellen. Zum Exportieren der Ergebnisse wurde die Open Source Bibliothek Apache POI eingesetzt, die Java-APIs zum Lesen und Schreiben von Daten im Datenformat von Microsoft Office bereitstellt.

4.2 Evaluierung des Prototyps

Für die erste Evaluierung des Prototyps wurden historische Daten aus abgeschlossenen Projekten bei der ZF Friedrichshafen AG verwendet. Anhand dieser ersten Evaluierung wurden bereits viel versprechende

Ergebnisse erzielt. Die ausgewerteten Ergebnisse haben gezeigt, dass mithilfe des ATMS viele Testthemen in folgenden Testzyklen ausgeschlossen werden konnten. Dies wurde dadurch ermöglicht, dass die Softwareagenten den Hinweis geben konnten, dass diese Testthemen über eine hohe Qualität verfügen und stabil sind. Infolgedessen wurden diese Testthemen von den Softwareagenten niedrig priorisiert.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Konzept basierend auf Softwareagenten zur Bewertung der Softwarequalität und zur Priorisierung von Testthemen vorgestellt. Die verlässliche Aussage über die aktuelle Qualität der Software erleichtert es, eine richtige Testfallpriorisierung beim Softwaretest zu gewährleisten. Dadurch lassen sich die Testaktivitäten auf die kritischen Stellen fokussieren. Unnötiges Testen kann dadurch verhindert werden.

Der realisierte Prototyp stellte ein erstes Bild eines solchen intelligenten Testmanagementwerkzeugs dar und liefert viel versprechende erste Evaluierungsergebnisse. Die Evaluierung weiterer Daten bei der ZF Friedrichshafen AG wird in weiteren Arbeiten folgen.

6 Literatur

- [Lippe06] W.-M. Lippe: *Soft-Computing*, Springer-Verlag Berlin Heidelberg, 2006.
- [LFRM09] H. Q. Le, N. Friedl, J. Ramsbrock und M. Martinus: *Fokussierung auf den Testprozess - Methoden zur Testfall-Priorisierung*, ATZ-Elektronik, Nr. 3, Mai 2009.
- [MaJa10] C. Malz und N. Jazdi: *Agent-based Test Management for Software System Test*, IEEE International Conference on Automation, Quality and Testing, Robotics, Mai 2010.
- [UWG03] P. de A. Urbano, T. Wagner und P. Göhner: *Softwareagenten - Einführung und Überblick über eine alternative Art der Softwareentwicklung, Teil II: Agenten in der Automatisierungstechnik: Modellierung eines Anwendungsbeispiels*, atp – Automatisierungstechnische Praxis, Vol. 45 (2003), Heft 11, 2003.
- [WGU03] T. Wagner, P. Göhner und P. de A. Urbano: *Softwareagenten - Einführung und Überblick über eine alternative Art der Softwareentwicklung, Teil I: Agentenorientierte Softwareentwicklung*, atp – Automatisierungstechnische Praxis, Vol. 45 (2003), Heft 10, 2003.