

Stakeholder-orientierter Software Test für Geschäftsanwendungen

Alexander Klaus

Fraunhofer Institut für Experimentelles Software Engineering,
67663 Kaiserslautern, Fraunhofer-Platz 1, Alexander.Klaus@iese.fraunhofer.de

1. Einleitung und Motivation

Geschäftsanwendungen werden immer wichtiger für Unternehmen. Basierend auf Geschäftsprozessen sollen die Systeme Mitarbeiter unterstützen und die Arbeitsabläufe effizienter gestalten. Die verschiedenen Anwender stellen unterschiedlichste Anforderungen aus mehreren Perspektiven an diese Systeme, einhergehend mit verschiedensten Zielen, die die Stakeholder mit den Systemen erreichen wollen. Diese Anforderungen müssen beim Test berücksichtigt werden.

Basierend auf einem Verfahren für das Requirements Engineering (RE) [1] wird daher eine Teststrategie vorgestellt, die die Priorisierung der Tests auf Basis der Ziele und Aufgaben der Stakeholder vornimmt und parallele Teilprozesse berücksichtigt.

2. Basis: Requirements Engineering

Ausgangsbasis ist das Konzept TORE (Task-oriented Requirements Engineering) [1, 2]. Dieses basiert auf typischen im RE zu treffenden Entscheidungen, welche abstrakt beginnen und stufenweise verfeinert und konkretisiert werden (Abbildung 1).

Das RE beginnt mit der Aufnahme der Stakeholder, deren Zielen (Geschäftsziele und Ziele der Benutzer) und den zur Erreichung notwendigen Aufgaben. Letztere werden als Geschäftsprozesse erfasst. Mit zunehmender Konkretisierung werden diese in Teilprozesse, Systemfunktionen und Interaktionen verfeinert und die beabsichtigte Nutzung beschrieben.

Als Notationen werden häufig Aktivitätsdiagramme, EPKs (Ereignisgesteuerte Prozessketten), Use Cases und Zielbäume genutzt. Grundsätzlich ist der Ansatz aber unabhängig von speziellen Notationen. [1, 2]

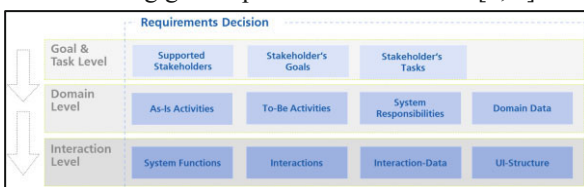


Abbildung 1: Entscheidungen in TORE (aus [1])

3. Teststrategie

Die Strategie zum Testen baut auf den oben genannten Punkten auf. Ebenso wie TORE zwingt sie Anwendern keine bestimmte Notation auf.

Die Strategie kann frühzeitig parallel zum RE eingesetzt werden. Der Fokus liegt auf der Priorisierung und dem Testen auf Prozessebene. Abbildung 2 zeigt ein Beispiel für eine Anwendung, die Schwerpunkte sind markiert.

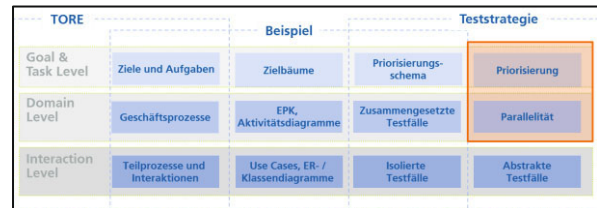


Abbildung 2: TORE und die Teststrategie

3.1. Priorisierung von Tests

Die Ziele der Stakeholder können für die Priorisierung genutzt werden. Einerseits können die Stakeholder selbst unterschiedlich priorisiert werden, etwa in Abhängigkeit davon, wie kritisch es ist, wenn ein Stakeholder die Ziele nicht erreichen kann. Andererseits können die Aufgaben der Stakeholder priorisiert werden, in Abhängigkeit davon, zu wie vielen Zielen der Stakeholder sie beitragen.

Da die Aufgaben über die Geschäfts- und Teilprozesse bis zu einzelnen Interaktionen verfeinert werden, und TORE Verfolgbarkeit gewährleistet, kann auch die Priorisierung bis zu diesen detailliert werden. So können zum einen die Teilprozesse in einem Geschäftsprozess unterschiedlich stark priorisiert werden, zum anderen auch die einzelnen Interaktionen innerhalb eines Teilprozesses.

3.2. Tests auf Prozessebene – Parallelität

Auf Basis der Interaktionen können Testfälle abgeleitet werden, etwa über Use Cases [3, 5]. Über die übergeordneten Teil- und Geschäftsprozesse können die Interaktionen (bzw. Testfälle) zu einem größeren Kontext zusammengesetzt werden. Dabei muss oft Expertenwissen einfließen.

Aufgrund der begrenzenden Vor- und Nachbedingungen der Interaktionen können diese isoliert geprüft und die üblichen Abdeckungsmaße verwendet werden, etwa Zweigabdeckung. Für danach folgende Interaktionen wird ein Stellvertreter gewählt, das heißt, ein Testfall, der die Nachbedingung auslöst, die mit der Vorbedingung einer anderen Interaktion übereinstimmt. Nach diesen Schritten kann die

folgende Interaktion wie eine isolierte getestet werden, mit dem Unterschied, dass eben diese Schritte für die Herstellung der Vorbedingung nötig sind. Dabei können auch mehrere Interaktionen durchlaufen werden, bis die jeweilige Vorbedingung hergestellt ist. Dadurch sinkt die Zahl der Testfälle im Vergleich zu einer vollständigen Kombination der Interaktionen, aber es kann nicht immer jeder durch bestimmte Kombinationen auftretende Fehler entdeckt werden.

Auf Prozessebene ergibt sich das Problem von parallel ablaufenden Teilprozessen, die sich an einem oder mehreren Punkten gegenseitig beeinflussen. Aufgrund der Parallelität können die einzelnen Schritte der Testfälle in unterschiedlicher Reihenfolge ausgeführt werden, was sich auf die Ergebnisse auswirken kann. Manuell kann oft nicht jede Kombination getestet werden. Automatisiert könnte man hier Test Runner für jeden parallelen Teilprozess verwenden, die von einem „Meta-Test Runner“ gesteuert werden. Dieser bestimmt, wann welcher Test Runner einen Testschritt ausführt, so dass über diese eine Veränderung der Reihenfolgen erzeugt wird. Damit variiert die Verschachtelung der Schritte der parallelen Teilprozesse. Die Testfälle bleiben gleich, nur die Abarbeitung der Schritte untereinander variiert.

4. Fokus der Strategie und Diskussion

Der Fokus der Strategie ist der Systemtest. Es wird Expertenwissen benötigt, doch eine Unterstützung durch Werkzeuge oder eine Teilautomatisierung ist angestrebt. Die Strategie ist unabhängig von Notationen, bietet eine schnelle Möglichkeit für die Priorisierung und geht auf parallele Teilprozesse ein.

Nebenbei werden die Dokumente des RE geprüft, etwa auf Inkonsistenzen. Voraussetzung ist aber die repräsentative Erfassung der Ziele der Stakeholder und die Detaillierung der Geschäftsprozesse bis hinunter zu den Interaktionen. Wichtig sind zudem die Verfolgbarkeit und die Vor- und Nachbedingungen der Interaktionen.

5. Verwandte Arbeiten

Viele Ansätze beruhen auf dem RE, und es existiert ein Ansatz, der auf TORE basiert, allerdings mit Fokus auf der Hierarchie der Entscheidungen [4]. Verschiedene Beispiele für die anforderungsbasierte Testfallableitung können etwa in [5] gefunden werden. Ansätze zum prozessbasierten Test finden sich in [6] und [7].

[8] stellt einen Ansatz zur Priorisierung anhand von Zielen von Stakeholdern vor, aber mit Fokus auf Produktlinien. Die Ziele sind hier auf die Domäne und nicht auf die Anwendung bezogen.

6. Praktische Anwendung und Ausblick

Die in diesem Artikel vorgestellte Strategie wurde zum Test einer Anwendung für Publikationsprozesse verwendet. Weitere Arbeiten sollen neue Erkenntnisse bringen, etwa für parallele Teilprozesse höherer Komplexität oder mit Schleifen.

Zudem muss geprüft werden, ob eine Priorisierung anhand der Ziele der Stakeholder ausreichend für komplexe Systeme ist. Hinsichtlich des Verbindens der isolierten Testfälle zu prozessübergreifenden sollen Regeln und Hilfsmittel entwickelt werden. Eine Werkzeugunterstützung wird insbesondere im Hinblick auf den erwähnten Meta-Test Runner geprüft.

Referenzen

- [1] S. Adam, J. Doerr, M. Eisenbarth, A. Gross: "Using Task-oriented Requirements Engineering in Different Domains – Experiences with Application in Research and Industry", 17th IEEE International Requirements Engineering Conference, 2009, pp. 267-272.
- [2] B. Paech, K. Kohler: "Task-driven Requirements in object-oriented development". In: J. Leite, J. Doorn: "Perspectives on Requirements Engineering", Kluwer Academic Publishers, 2003.
- [3] A. Cockburn: "Writing Effective Use Cases", Boston, Addison-Wesley, 2001.
- [4] L. Borner, T. Illes, B. Paech: "The Testing Process - A Decision Based Approach". In: Proc. of the Second International Conference on Software Engineering Advances (ICSEA 2007), IEEE CS.
- [5] M. J. Escalona, J. J. Gutierrez, M. Mejías, G. Aragón, I. Ramos, J. Torres, F. J. Domínguez: "An overview on test generation from functional requirements", J. Syst. Softw. 84, 8 (August 2011), pp. 1379-1393.
- [6] A. Jimenez-Ramirez, R. M. Gasca, A. J. Varela-Vaca: "Contract-based test generation for data flow of business processes using constraint programming," Fifth International Conference on Research Challenges in Information Science (RCIS), 2011, pp. 1-12.
- [7] S. Debricon, F. Bouquet, B. Legeard: "From Business Processes to Integration Testing", In O. Zendra, editor, 5èmes journées sur l'Ingénierie Dirigée par les Modèles, vol. 1, March 2009.
- [8] A. Ensan, E. Bagheri, M. Asadi, D. Gasevic, Y. Biletskiy: "Goal-Oriented Test Case Selection and Prioritization for Product Line Feature Models," Eighth International Conference on Information Technology: New Generations (ITNG), 2011, pp. 291-298.