

Review-Modelle zur Unterstützung in der funktionszentrierten Entwicklung eingebetteter Systeme

Marian Daun,¹ Thorsten Weyer,² Klaus Pohl³

Abstract: Dieser Vortrag berichtet von dem Beitrag *Improving manual reviews in function-centered engineering of embedded systems using a dedicated review model* [DWP19], der in der Fachzeitschrift *Software and Systems Modeling* veröffentlicht wurde. Im Rahmen des Beitrags wurde ein dezidiertes Review-Modell vorgeschlagen, um die Validierung der interaktionsbasierten Verhaltensanforderungen und des funktionalen Entwurfs in einem gemeinsamen Qualitätssicherungsschritt zu unterstützen. Neben der automatisierten Erzeugung des Review-Modells wurde mithilfe kontrollierter Experimente die Vorteilhaftigkeit des vorgeschlagenen Review-Modells nachgewiesen.

Keywords: Reviews; Eingebettete Systeme; Requirements Engineering; Manuelle Validierung

1 Einleitung und Problemstellung

In der modellbasierten Entwicklung eingebetteter Systeme werden Modelle häufig durch Inaugenscheinnahme qualitätsgesichert, um sicherzustellen, dass das zu entwickelnde System die Erwartungen der diversen Stakeholder erfüllt. Hierzu werden in der Regel manuelle Reviews oder Inspektionssitzungen durchgeführt. Änderungen der Stakeholderintentionen führen zu einer Überarbeitung der Entwicklungsartefakte, der Anforderungen und des Entwurfs. In der industriellen Praxis werden Änderungen der Stakeholderintentionen häufig nicht unmittelbar aufgedeckt und in alle Entwicklungsartefakte eingearbeitet [DHW14]. Hierdurch entstehen Inkonsistenzen zwischen verschiedenen Entwicklungsartefakten, die im Laufe der Zeit nicht einfach aufzulösen sind, da es schwer nachzuvollziehen ist, welches der inkonsistenten Entwicklungsartefakte den aktuell korrekten Stand zeigt und welches veraltet ist, sofern nicht beide veraltet sind. Daher wird ein dezidiertes Review-Modell vorgeschlagen, das durch Modelltransformationen automatisiert erzeugt werden kann und den Entwicklern Inkonsistenzen zwischen verschiedenen Artefakten aufbereitet anzeigt.

2 Review-Modell

Um die manuelle Qualitätssicherung zu unterstützen, wird ein Review-Modell erzeugt, das verschiedene Ursprungsartefakte zusammenfasst und Unterschiede hervorhebt. Für

¹ Universität Duisburg-Essen, paluno - The Ruhr Institute for Software Technology, marian.daun@uni-due.de

² Universität Koblenz-Landau, Institut für Softwaretechnik, thorstenweyer@uni-koblenz.de

³ Universität Duisburg-Essen, paluno - The Ruhr Institute for Software Technology, klaus.pohl@uni-due.de

die funktionszentrierte Entwicklung eingebetteter Systeme wurde dies für die Behavioral Requirements und das Functional Design gezeigt. Zur Unterstützung des Reviewers werden drei unterschiedliche Diagrammtypen unterschieden:

- *Refinement Diagrams* zeigen Sachverhalte, die in beiden Artefakten konsistent dargestellt sind. Da in der Regel das Functional Design eine konkretere Spezifikation des Systems darstellt als die Behavioral Requirements, werden hierbei auch Verfeinerungsbeziehungen automatisiert identifiziert und berücksichtigt.
- *Unrefinable Diagrams* zeigen Sachverhalte, die nur in den Behavioral Requirements enthalten sind und nicht im Functional Design berücksichtigt wurden.
- *Unspecified Diagrams* zeigen Sachverhalte, die im Functional Design enthalten sind, aber nicht auf die Behavioral Requirements zurückgeführt werden können.

3 Evaluation

Zur Evaluation wurden kontrollierte Experimente entworfen [Da15, Da19]. Mit diesen konnte die Vorteilhaftigkeit des Review-Modells für die Qualitätssicherung nachgewiesen werden. Insbesondere zeigten sich eine verbesserte Effektivität und Effizienz sowie eine sicherere Entscheidungsfindung. Daneben wurde das Review-Modell von den Teilnehmern mehrheitlich als vorteilhaft angesehen. Darüber hinaus erlaubt der Vergleich mehrerer Experimente, Aussagen über die Wirkung des Review-Modells zu treffen. Es liegt nahe, dass sowohl die gewählte Notation des Review-Modells als auch die Synthese zweier Artefakte in einem einzelnen Modell einen positiven Effekt auf den Review haben.

Literaturverzeichnis

- [Da15] Daun, Marian; Salmon, Andrea; Weyer, Thorsten; Pohl, Klaus: The Impact of Students' Skills and Experiences on Empirical Results: A Controlled Experiment with Undergraduate and Graduate Students. In: Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering. EASE '15, ACM, New York, NY, USA, S. 29:1–29:6, 2015.
- [Da19] Daun, M.; Brings, J.; Krajinski, L.; Weyer, T.: On the Benefits of using Dedicated Models in Validation Processes for Behavioral Specifications. In: 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP). S. 44–53, May 2019.
- [DHW14] Daun, M.; Höflinger, J.; Weyer, T.: Function-centered engineering of embedded systems: Evaluating industry needs and possible solutions. In: 2014 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE). S. 1–9, April 2014.
- [DWP19] Daun, Marian; Weyer, Thorsten; Pohl, Klaus: Improving manual reviews in function-centered engineering of embedded systems using a dedicated review model. Software and Systems Modeling, 18(6):3421–3459, 2019.