Erfahrungsbasierte Verbesserung der Dokumentation von Anforderungen auf Basis von heuristischem Feedback

Eric Knauss
eric.knauss@inf.uni-hannover.de
FG Software Engineering, Leibniz Universität Hannover

Zusammenfassung

Komplexe Systeme führen zu immer komplexerer Anforderungsdokumentation. Um diese in ausreichender Qualität zu erstellen, ist ein hohes Maß an Erfahrung in der entsprechenden Domäne sowie im Requirements Engineering (RE) allgemein nötig. Für Software erstellende Organisationen ist der Aufbau dieser Erfahrung herausfordernd. Dieser Beitrag skizziert ein *Lernmodell* für den systematischen Aufbau von RE-Erfahrung, basierend auf *heuristischen Kritiken*.

1 Motivation: RE braucht Erfahrung

Trotz aller Errungenschaften im Bereich des Software Engineering haben Organisationen immer noch Schwierigkeiten, funktionierende Software zu liefern, die die Kundenwünsche in ausreichender Qualität erfüllt. Dies ist zu einem beträchtlichen Teil auf mangelhaftes Requirements Engineering zurückzuführen: Wenn wichtige Aspekte der Anforderungen nie geklärt oder sogar missverstanden werden, führen offene Fragen zu hohem Mehraufwand und schlimmstenfalls zu einem für den Kunden unbrauchbaren Produkt. Immer komplexere Softwaresysteme sowie kürzere Releasezyklen und Time-to-Market führen zu einem enormen Zeitdruck, der dieses Problem noch weiter verschärft. Erfahrung ist einer der wichtigsten Erfolgsfaktoren, um in diesem Kontext verlässlich gute Ergebnisse zu erzielen. Ein systematischer Ansatz um neue Erfahrung zu erlernen erscheint daher nötig.

2 Problemstellung: Erfahrungsaufbau

Mitarbeiter mit ausreichender Erfahrung in der Dokumentation von Anforderungen sind rar. Zudem müssen diese Mitarbeiter häufig auch mit den fachlichen Aspekten der zu erstellenden Software vertraut sein. Um dennoch eine ausreichende Kompetenz bei der Dokumentation aufzubauen, greifen viele Organisationen auf externe Berater zurück oder versuchen, ihre Prozesse weiter zu entwickeln. Häufig reicht dies jedoch nicht um in einer bestimmten Domäne verlässlich Anforderungsdokumentation in angemessener Qualität zu erstellen.

3 Ansatz: Heuristische Kritiken

Dieser Beitrag präsentiert einen systematischen Ansatz, wie Organisationen vorhandene Erfahrungen mit einem etablierten Artefakt- und Vorgehensmodell besser nutzen und ausbauen können. Mit Hilfe von *heuristischen Kritiken* kann ein Rechner auf Basis von Erfahrungen und Best Practices Verbesserungspotential in Anforderungsdokumenten aufzeigen. Dies ist ein mächtiges Konzept [6]:

- Mit Hilfe von heuristische Kritiken können Erfahrungen einer Organisation bei der Dokumentation von Anforderungen mit eingebracht werden.
- Autoren von Anforderungsdokumentation können diese Erfahrungen um eigene Erfahrungen erweitern – und sind bereit dazu.
- Heuristische Kritiken helfen, bessere Anforderungsdokumentation zu erstellen.

4 Konzept: Lernmodell auf Basis heuristischer Kritiken

Bisherige Ansätze zur automatischen Analyse natürlichsprachlicher Anforderungen überprüfen eher generische Aspekte (wie z.B. [1, 8, 7, 2]). Sie erlauben zudem in der Regel nicht, eigene, domänenspezifische Erfahrungen hinzuzufügen.

Dieser Beitrag präsentiert ein Lernmodell (Abbildung 1), in dem vorhandene Erfahrungen mit Hilfe automatischer Überprüfung natürlichsprachlicher Anforderungen aktiviert werden. Dazu prüft das System auf Basis einer heuristischen Regel, ob eine entsprechend kodierte Erfahrung der Organisation *wiederverwendbar* ist. Falls ja, wird der Benutzer pro-aktiv darauf aufmerksam gemacht, wenn er z.B. eine Best-Practice der Organisation verletzt und eine Anforderung passiv formuliert.

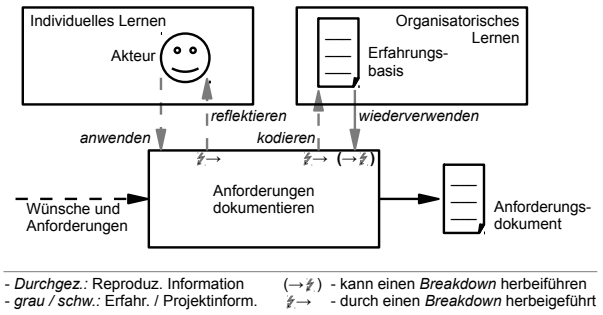

Abbildung 1: Informationsflüsse des individuellen und organisatorischen Lernens durch heuristisches Feedback beim Dokumentieren von Anforderungen.

Der Benutzer kann diese Meldung ignorieren (wie beispielsweise in Abbildung 2). Andernfalls wird er durch diese konstruktive Unterbrechung (=Breakdown) in die Lage versetzt, über die Situation zu *reflektieren* und so zu einem Erkenntnisgewinn und neuer Erfahrung zu kommen.

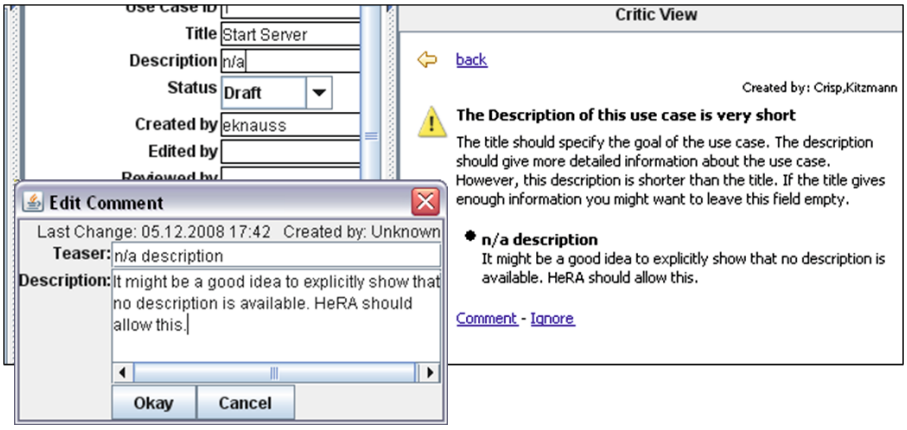

Abbildung 2: Ein erfahrungsbasiertes Requirements Engineering Werkzeug (hier: HeRA) gibt eine heuristische Kritik. Der Nutzer entschließt sich, diese Kritik zu ignorieren und gibt dazu Feedback.

Es ist davon auszugehen, dass der Benutzer bei der Dokumentation der Anforderungen seine eigene Erfahrung *anwendet*. Diese kann durchaus von den Vorschlägen der heuristischen Kritiken abweichen. So könnte im Beispiel der Benutzer zwar prinzipiell zustimmen, dass Anforderungen nicht passiv formuliert werden sollten. Falls es sich bei der von der automatischen Überprüfung hervorgehobenen Passage jedoch um eine Bedingung handelt, könnte der Benutzer zu dem Schluss kommen, dass hier Passiv vertretbar und sogar nützlich ist.

Diese neue Erfahrung (Passiv ist bei Bedingungen erlaubt) kann der Benutzer der heuristischen Kritik hinzufügen. Er kann aber auch die heuristische Regel anpassen (= *kodieren*), so dass Bedingungen von der Prüfung auf Passiv in Zukunft ausgenommen sind.

Auf diese Weise kann das Erfahrungswissen der Organisation und der individuellen Anforderungsautoren wachsen.

5 Vor- und weiterführende Arbeiten

Die in diesem Beitrag vorgestellten Ergebnisse gehen auf meine Dissertation zurück [6]. Ein Vorläufer des skizzierten Lernmodells ist als Poster auf der Requirements Engineering Konferenz präsentiert worden [5].

Zur Evaluation ist das Lernmodell in einem Forschungsprototypen integriert worden. Der Heuristic Requirements Assistant (HeRA) erlaubt die Dokumentation von Anforderungen und Use-Cases und unterstützt dabei mit heuristischer Kritik. HeRA ist zuerst auf dem Fachgruppentreffen 2006 [3] vorgestellt worden und dann in einer deutlich erweiterten Fassung auf der ICSE 2009 [4].

Literatur

[1] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami. The Linguistic Approach to the Natural Language Requirements Quality: Benefit of the use of an Automatic Tool. In *SEW '01: Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop*, pages 97–105, Washington, DC, USA, 2001. IEEE Computer Society.

[2] Benedict Gleich, Oliver Creighton, and Leonid Kof. Ambiguity Detection: Towards a Tool Explaining Ambiguity Sources. In R. Wieringa and A. Persson, editors, *Proceedings of Requirements Engineering: Foundation for Software Quality (REFSQ)*, volume 6182 of *LNCS*, pages 218–232, Essen, Germany, 2010. Springer.

[3] Eric Knauss. Einsatz computergestützter Kritiken für Anforderungen. *GI Softwaretechnik-Trends*, 27(1):27–28, Februar 2007. Beitrag zum Jahrestreffen 2006 der GI-Fachgruppe Requirements Engineering.

[4] Eric Knauss, Daniel Lübke, and Sebastian Meyer. Feedback-Driven Requirements Engineering: The Heuristic Requirements Assistant. In *Proceedings of the 31st International Conference on Software Engineering (ICSE '09)*, pages 587 – 590, Vancouver, Canada, May 2009.

[5] Eric Knauss, Kurt Schneider, and Kai Stapel. Learning to Write Better Requirements through Heuristic Critiques. In *Proceedings of the 17th IEEE Requirementes Engineering Conference (RE '09)*, pages 387–388, Atlanta, USA, 2009. IEEE Computer Society. Poster.

[6] Eric Werner Knauss. *Verbesserung der Dokumentation von Anforderungen auf Basis von Erfahrungen und Heuristiken.* Cuvillier Verlag, Göttingen, Germany, 2010. Phd Thesis.

[7] Ralf Melchisedech. *Verwaltung und Prüfung natürlichsprachlicher Spezifikationen.* PhD thesis, Fakultät Informatik, Universität Stuttgart, Stuttgart, 2000.

[8] William M. Wilson, Linda H. Rosenberg, and Lawrence E. Hyatt. Automated analysis of requirement specifications. In *Proceedings of the 19th International Conference on Software Engineering (ICSE '97)*, pages 161–171, New York, NY, USA, 1997. ACM.