Dr. Ernst Sikora

(ernst.sikora@astech-auto.de)
ist Teamleiter für die Entwicklung vorausschauender Sicherheitsfunktionen bei der Automotive Safety Technologies GmbH. Er gestaltet den Produkt- und Systementwicklungsprozess unter Verwendung vielfältiger Methoden. Sein Werkzeugkasten beinhaltet neben einem Hammer sowohl klassische wie auch agile Techniken, textuelle wie auch modellbasierte Spezifikation, Reviews ebenso wie Prototyping und Simulation

objektspektrum themenspecial: Requirements Engineering 2015

Harmlos oder schädlich? Requirements Engineering-Mythen

Kennen Sie die „Myth Busters" [myt]? Die Myth Busters überprüfen moderne Mythen mittels aufwendiger und spektakulärer Experimente. Am Ende wird jeder Mythos als widerlegt, als plausibel oder als bestätigt eingestuft. Auch im Requirements Engineering gibt es Mythen. Wirklich überprüft oder gar widerlegt werden sie leider nur selten.

Der Mythos der perfekten Anforderungsspezifikation

Als Mythen werden im Systems und Software Engineering weit verbreitete, jedoch zweifelhafte oder überholte Ansichten und Lehrmeinungen bezeichnet. Auch von Requirements Engineering-Mythen wird berichtet, wie zum Beispiel dem Mythos der „perfekten Anforderungsspezifikation zu Projektbeginn" (vgl. [Mul96] und [Hou04]).

Warum ist dies ein Mythos? Ganz einfach: Entwickeln bedeutet, durch einen Prozess etwas Neues, Fortschrittlicheres zu konstruieren. Werden die Anforderungen nicht weiterentwickelt, dann wird einfach nur ein bereits bekanntes und gut verstandenes System noch einmal gebaut, also ein Imitat eines bestehenden Produkts. Soll hingegen etwas Neues entwickelt werden, gelangen die Requirements-Ingenieure im Verlauf der Entwicklung zu neuen Erkenntnissen, die sie in die Anforderungen einfließen lassen.

Was ist das Problem an Mythen? Ist eine perfekte Anforderungsspezifikation zu Projektbeginn nicht doch ein erstrebenswertes Ziel, auch wenn es letztlich nicht erreicht wird? Einiges spricht dagegen:

- Bereits bei dem Versuch, eine Anforderungsspezifikation von vorne herein perfekt zu machen, kann viel Zeit vergeudet werden, in der kein erlebbares Produkt entsteht. Erst viel später zeigt sich, wieviel Wahrheit in dem Ausspruch „I will know it when I see it" [Boe00] steckt.
- Eine als perfekt angesehene Spezifikation wird im Entwicklungsverlauf nicht mehr kritisch überprüft. Ob die spezifizierten Anforderungen auch die richtigen sind, damit das spätere Produkt seinen Zweck zufriedenstellend erfüllt, wird nicht genügend hinterfragt.
- Neue Erkenntnisse, die im Projektverlauf gewonnen werden, münden in zeitraubenden Schuldzuweisungs- und Rechtfertigungs-Zyklen, warum denn die Spezifikation nicht gleich von Anfang an richtig geschrieben worden ist.
- Die Architektur und die Tests orientieren sich sehr eng an den vorliegenden Anforderungen, denn Flexibilität und Erweiterbarkeit im Vorgriff auf neue und geänderte Anforderungen sind bei einer perfekten Spezifikation ja nicht nötig.

- Aufwand und Ressourcen für die Umsetzung von Anforderungsänderungen werden nicht eingeplant und stehen dem Projekt dann auch nicht zur Verfügung.

All dies führt dazu, dass Kunden und Nutzer ein deutlich schlechteres Produkt bekommen, als es möglich gewesen wäre.

Wenn Veränderung die einzige Konstante ist...

... und Sie auch nicht an die perfekte Anforderungsspezifikation zu Projektbeginn glauben, dann benötigen Sie ein leistungsfähiges Änderungsmanagement. **Ralf Klimpke** beschreibt in seinem Beitrag wie Anforderungsänderungen mittels Change Sets verwaltet werden können. In einem Change Set sind dabei jeweils die Änderungen enthalten, die zu einem bestimmten Änderungsantrag gehören. Change Sets bieten somit die Möglichkeit, zusammenhängende Änderungen zu erfassen und über einen geordneten Review- und Freigabeprozess in das Anforderungsdokument einfließen zu lassen.

Die Legende der lösungsneutralen Anforderungen

Sehr hartnäckig und ebenso schädlich ist der Mythos, dass alle Anforderungen unabhängig von einer Lösung formuliert sein müssen. Wie **Dr. Kim Lauenroth** in seinem Beitrag ausführt, gehört es zu den Aufgaben eines Requirements-Ingenieurs, die Eigenschaften der gewünschten Lösung zu definieren. Dazu benötigt der Requirements-Ingenieur nicht zuletzt eine Vorstellung davon, was die Lösung ist. Wenn diese Vorstellung fehlt oder bewusst ausgeblendet wird, führt dies zu sehr abstrakten Anforderungen, die für die weitere Entwicklung wertlos sind.

Wie isst man einen Elefanten?

Nicht nur der Glaube an Mythen wirkt sich negativ auf das Ergebnis der Entwicklung aus. Auch der Versuch, das ganze Produkt oder zu große Teile davon in einem Schritt zu realisieren, ist schädlich. Um dem zu entgehen, zeigt **Matthias Bohlen** in seinem Beitrag auf, wie komplexe User Stories in kleinere – und damit in kürzeren Inkrementen realisierbare – Teile zerlegt werden können. Der Autor stellt eine Reihe von Techniken vor, mit denen die Zerlegung systematisch und reproduzierbar gelingt.

Das Halteproblem des Requirements Engineering

Eine vollständige Anforderungsspezifikation für ein komplexes, reales System zu erreichen, ist nach der Erfahrung zahlreicher Entwickler ein Mythos. Ebenso wie in der theoretischen Informatik das Halteproblem nicht entscheidbar ist [Weg05], fällt im Requirements Engineering die Feststellung, wann die Anforderungsspezifikation fertig ist, ebenfalls nicht leicht. **Johannes Bergsmann** befasst sich in seinem Beitrag eingehend mit den Kriterien, die herangezogen werden können, um das Halteproblem des Requirements Engineering in den Griff zu bekommen und die Anforderungen für einen Sprint freizugeben.

Durch den Nebel der nichtfunktionalen Anforderungen

Nichtfunktionale Anforderungen (NFA) sind nicht weit von einem Mythos entfernt, da sie eine Art Sammelbecken für Anforderungen bilden, die nicht als Funktionen, Daten oder Verhalten deklariert werden können. Das schädliche an dem Begriff der NFA ist, dass er suggeriert, alle nichtfunktionalen Anforderungen könnten auf dieselbe Art und Weise bearbeitet werden. Der Irrtum könnte kaum größer sein, da z. B. Zeitanforderungen ganz anders zu behandeln sind als Sicherheitsanforderungen.

Glücklicherweise lässt sich der Nebel lichten, wenn sich Requirements-Ingenieure mit spezifischen Unterarten nichtfunktionaler Anforderungen befassen. **Prof. Dr. Christof Ebert** und **Dr. Eduard Metzker** geben einen Überblick über die notwendigen Requirements Engineering-Aktivitäten für Safety-Anforderungen sowie für Security-Anforderungen.

Sie zeigen, wie sich diese Aktivitäten in den Requirements Engineering-Prozess für funktionale Anforderungen einfügen. Anhand des Beispiels eines Fahrerassistenzsystems zeigen die Autoren die Vorteile einer Werkzeugunterstützung auf, die die funktionalen Systemaspekte mit Safety- und Security-Aspekten integriert.

Das Entflechten der Sprachverwirrung

Der Mythos, formale Spezifikationssprachen seien das Allheilmittel für sämtliche Gebrechen des Requirements Engineering, hat nur wenig praktische Bedeutung erlangt. Doch auch das Festhalten an natürlicher Sprache zum Spezifizieren von Anforderungen bringt Defizite mit sich.

Jörn Koch und **Sebastian Middeke** stellen in ihrem Beitrag einen Ansatz vor, der die Eindeutigkeit von Anforderungen in agilen Projekten erhöht. Vor Beginn eines Sprints werden exemplarische Testfälle definiert, die einerseits als Anforderungen – ähnlich zu Use Cases oder Szenarien – sowie am Ende des Sprints als Abnahmekriterien dienen. Die Autoren berichten von den positiven Effekten dieses Vorgehens und gehen auch auf etwaige Vorbehalte ein.

Domänenspezifische Sprachen (engl. domain-specific language, DSL) verheißen Lesbarkeit und die Vorteile einer teilweisen Formalisierung zu vereinen. **Christoph Ballhause** und **Jörg Leuser** berichten in ihrem Beitrag von ihren Erfahrungen bei der Verwendung einer solchen Sprache nebst der zugehörigen Werkzeugumgebung. Die positiven Erfahrungen mit Features wie z. B. Autovervollständigung und automatisierter Referenzierung führen die Autoren zu dem Plan, eine vollständige DSL für die typischen Anforderungsartefakte in ihrem Unternehmen zu entwickeln. ■

Viel Vergnügen beim Lesen dieses Themenspecials!

Ernst Sikora

Dr. Ernst Sikora

Literatur

[myt] http://www.discovery.com/tv-shows/mythbusters/

[Mul96] Geoff Mullery: The Perfect Requirement Myth. Requirements Engineering, Vol 1(2), Springer, 1996, 132–134.

[Hou04] Frank Houdek and Matthias Weber: Future Trends in Automotive Requirements Engineering. INCOSE International Symposium, Vol. 14(1), 2004, 1830–1843.

[Boe00] Barry Boehm: Requirements that Handle IKIWISI, COTS, and Rapid Change. IEEE Computer, Vol. 33(7), 2000, 99–102.

[Weg05] Ingo Wegener: Theoretische Informatik - eine algorithmenorientierte Einführung. 3. Auflage, Teubner, 2005.