

## □ Masud Fazal-Baqaie

[E-Mail: mfazal-baqaie@s-lab.upb.de]  
ist Mitglied im Leitungsgremium der GI-Fachgruppe „Vorgehensmodelle für die betriebliche Anwendungsentwicklung“. Seine Forschungsinteressen umfassen die Entwicklung von projekt- und organisationspezifischen Softwareentwicklungsmethoden.

## □ Dr. Stefan Sauer

[E-Mail: sauer@s-lab.upb.de]  
ist Geschäftsführer des s-lab. Die systematische Entwicklung modellbasierter Software-Entwicklungsmethoden ist ein Schwerpunktthema seiner Forschung, über das er auch promovierte.

## □ Baris Güldali

[E-Mail: bguldali@s-lab.upb.de]  
ist aktives Mitglied des Arbeitskreises TOOP/MBT in der Gesellschaft für Informatik (GI). Seine Arbeitsschwerpunkte sind modellbasiertes Testen und Testautomatisierung.

## □ Michael Spijkerman

[E-Mail: mspijkerman@s-lab.upb.de]  
hat den Schwerpunkt seiner Forschungsinteressen auf die Spezifikation von Verwaltungssoftware im Bereich der Server-Systementwicklung und die Abbildung der daraus entstehenden Entwicklungstätigkeiten in einer situationsgerechten Entwicklungsmethodik gelegt.

## □ Markus Luckey

[E-Mail: mluckey@s-lab.upb.de]  
hat für das s-lab mehrere Methodenprojekte im Requirements Engineering und in der Systemspezifikation durchgeführt. Seine Forschungsinteressen umfassen die Spezifikation und Entwicklung von selbst-adaptiven Systemen und deren Qualitätssicherung.

*Alle Autoren sind Researcher im s-lab – Software Quality Lab der Universität Paderborn.*

# Maßgeschneidert und werkzeugunterstützt

## Entwickeln angepasster Requirements Engineering-Methoden

Grundlage für eine erfolgreiche Softwareentwicklung sind gute Anforderungen. Die Vorteile eines systematischen Requirements Engineerings (RE) zur Erhebung und Verwaltung von Anforderungen sind gemeinhin bekannt. Ebenso, dass verschiedenste RE-Techniken genutzt werden können. Doch wie wählt eine Organisation die passenden RE-Techniken aus und kombiniert sie zu einer durchgängigen RE-Methode, die auf die eigenen Bedürfnisse zugeschnitten ist und durch geeignete Werkzeuge unterstützt wird? Dieser Artikel beschreibt die Schritte bei der Entwicklung einer organisationspezifischen RE-Methode und – am Beispiel des Enterprise Architect – die Anpassung eines RE-Werkzeugs. Wir erläutern unsere Methodik mittels eines Fallbeispiels auf der Grundlage eines unserer Projekte.

### Einleitung

Aktuelle Untersuchungen bestätigen, dass statistisch gesehen die meisten Fehler in einem Softwareentwicklungsprojekt während der Anforderungsdefinition und während des Entwurfs entstehen [HP07]. Allerdings werden diese Fehler meist erst in einer späten Entwicklungsphase, z. B. bei den Abnahmetests entdeckt. Die Behebung solcher spät entdeckten Fehler kann je nach Anwendungsdomäne nicht nur sehr viel Geld und Zeit kosten, sondern auch Marken- und Imageschaden verursachen [SQS12].

Mit anderen Worten, es sinken die Fehlerfolgekosten, wenn die Fehler früher im Entwicklungsprozess entdeckt werden (vgl. [SuL10], [SQS12]). Fehler, die während der Anforderungsdefinition entstehen, können durch konstruktive Qualitätssicherungsmaßnahmen mit einer systematischen Requirements Engineering-Methode (RE-Methode) reduziert, entdeckt und behoben werden.

Im Sprachgebrauch eines Requirements Engineers kann ein Fehler in Anforderungen *nicht eindeutige* oder *fehlende* Anforderungen bedeuten. Der IEEE-Standard 830-1998 [IEEE98] definiert

mehrere Qualitätseigenschaften für Anforderungsspezifikationen, deren Verletzung zu einem Fehler im Entwicklungsprojekt führen kann. Die Einhaltung dieser Qualitätseigenschaften ist nicht nur damit verbunden, wie die Anforderungen fachlich und technisch erhoben werden, sondern auch damit, ob und wie diese Anforderungen dokumentiert, geprüft und verwaltet werden. Dementsprechend wirkt sich die Güte der verwendeten Requirements Engineering-Methode auf den gesamten Softwareentwicklungsprozess aus.

In der Praxis kommt es vor, dass der Softwareentwicklungsprozess und damit auch das Requirements Engineering (RE) und die dabei eingesetzten Werkzeuge historisch gewachsene Größen sind. Die Anpassung der RE-Methode an neue Rahmenbedingungen der Anwendungsdomäne, des Unternehmens oder der Projekte wird oftmals vernachlässigt.

Die existierende Literatur bietet eine reichhaltige Quelle an allgemeinen Grundsätzen und Best Practices zum Thema Requirements Engineering. Dass eine gute RE-Methode die Kernbereiche des Requirements Engineerings *Erhebung, Abstimmung, Dokumentation, Prüfung*

und *Verwaltung* abdecken sollte (vgl. [Rup09], [PuR09]), ist unstrittig.

Die sinnvolle Auswahl und Zusammenstellung von konkreten RE-Techniken für diese Kernbereiche ist allerdings situations- und unternehmensabhängig, da z. B. vorhandenes Mitarbeiter-Know-how, genutzte Werkzeuge und bestehende Organisationsstrukturen berücksichtigt werden müssen. Daraus folgt, dass für jede Organisation eine individuell zugeschnittene Lösung gefunden werden sollte.

In diesem Artikel beschreiben wir die Entwicklung einer organisationspezifischen RE-Methode, da wir dieses Thema in der Literatur nicht ausreichend beschrieben sehen. Wir stellen dazu ein systematisches und in der Praxis erprobtes Vorgehen vor, das im s-lab – Software Quality Lab der Universität Paderborn entwickelt wurde.

### Ansatz iterativer Methodenverbesserung

Bei der Entwicklung bzw. Verbesserung einer organisationspezifischen RE-Methode verfolgen wir insbesondere das Ziel, die Qualität der Anforderungsdefinition zu steigern, indem wir die Art, wie Anforderungen erhoben, abgestimmt, dokumen-

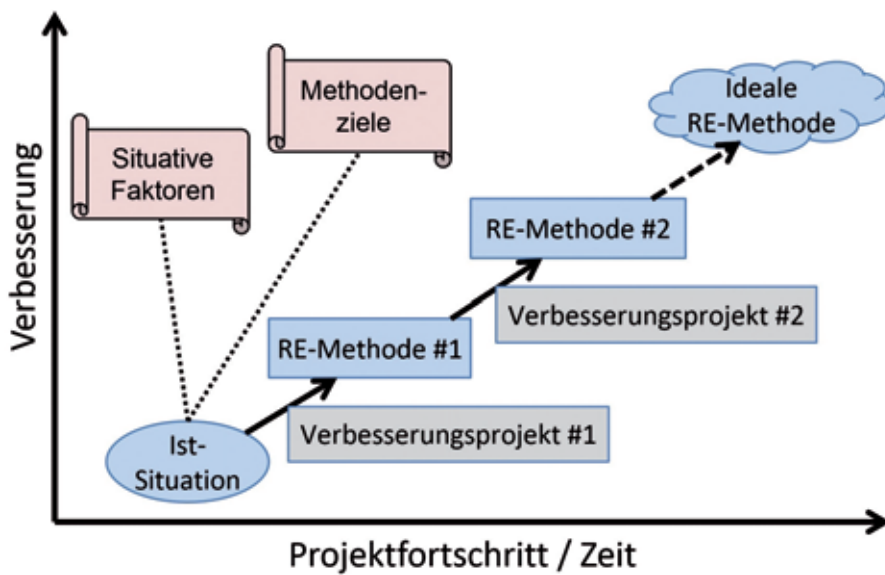


Abb. 1: Iterative Methodenverbesserung

tiert, geprüft und verwaltet werden, verbessern. Daneben soll die RE-Methode ökonomisch sinnvoll und zukunftssicher sein.

Ökonomisch sinnvoll heißt, dass die Umsetzung unternehmensspezifischer, langfristiger Methodenziele auf Grundlage der bestehenden RE-Methode anhand verfügbarer Ressourcen und geschätzter Wirkung priorisiert wird. Zukunftssicher heißt, dass die Entwicklung als iterativer Prozess ausgelegt ist, bei dem jede Iteration die RE-Methode entlang unternehmensspezifischer, langfristiger Methodenziele verbessert.

Unsere RE-Methode wird mit einem systematischen Entwicklungsprozess erstellt, ähnlich wie bei der strukturierten Erstellung von Software, d.h. Artefakte unseres Vorgehens werden explizit dokumentiert und systematisch ineinander überführt. Damit lässt sich auch im Nachhinein der Entstehungsprozess der verbesserten RE-Methode nachvollziehen.

Abbildung 1 veranschaulicht den iterativen Ansatz der Methodenverbesserung. Während einer Voranalyse wird die Ist-Situation der aktuellen RE-Methode bestimmt, indem diese mit situativen Faktoren (z. B. Organisationsstruktur, Projektgröße oder Mitarbeiterkenntnisse) [BWBM08] charakterisiert wird. Des Weiteren werden auf Grundlage der bestehenden Probleme langfristige Methodenziele (z. B. Rückverfolgbarkeit aller Entwicklungsartefakte zu ihrer Quelle oder realistische Aufwandschätzung für geänderte Anforderungen) für eine ideale organisationsspezifische RE-Methode skizziert. Mithilfe von aufeinander aufbauenden

Verbesserungsprojekten wird dann die RE-Methode schrittweise verbessert. Ähnlich wie aus der iterativen Softwareentwicklung bekannt, fließen die Erfahrungen aus den vergangenen Iterationen in die folgenden Iterationen ein und das hohe Risiko wird gegenüber einer radikalen „Big-Bang“-Umstellung reduziert.

Der Prozess eines einzelnen Verbesserungsprojektes ist in Abbildung 2 illustriert. Die Phasen *Analyse*, *Konzeption*, *Realisierung* und *Einführung* werden in jedem Verbesserungsprojekt durchgeführt und damit mehrfach durchlaufen. Während der Analyse wird eine Untermenge von bisher unerfüllten Methodenzielen gewählt und zu Methodenanforderungen verfeinert, die in dem aktuellen Verbesserungsprojekt umzusetzen sind.

In der Konzeption wird ein Methodenkonzept erstellt, das diese Methodenanforderungen erfüllt. Analog zu einer Soft-

warespezifikation definiert das Methodenkonzept die für die RE-Methode notwendigen RE-Techniken und ihre Zusammenhänge. Bei der Konzeption und bei der Realisierung müssen die situativen Faktoren aus der Voranalyse berücksichtigt werden.

In der Realisierung werden die RE-Techniken mit Leben gefüllt, d.h. einerseits werden unterstützende Werkzeuge implementiert oder existierende Werkzeuge angepasst und andererseits Verfahrensanweisungen für die Anwendung der RE-Techniken erstellt. Durch die Realisierung wird die RE-Methode anwendbar.

In der Einführung wird die RE-Methode sukzessive im Unternehmen ausgerollt. Nach der Einführung sollten die RE-Methode evaluiert und die Erkenntnisse bei der Definition von Methodenanforderungen in der nächsten Analyse-Phase berücksichtigt werden. Bei jeder Phase eines Verbesserungsprojektes greifen wir auf einen Grundstock an Hilfsmitteln zurück, den wir uns anhand der Fachliteratur (vgl. z. B. [Coc00], [IIB09], [Rup09], [RuR10], [SuS97]) und mit unseren Projektpartnern in den vergangenen Projekten erarbeitet haben und ständig erweitern.

**Fallbeispiel**

Wir möchten das im vorherigen Abschnitt vorgestellte allgemeine Vorgehen mit aufeinander aufbauenden Beispielen aus einem s-lab-Projekt konkretisieren. Bevor die iterativen Verbesserungsprojekte gestartet wurden, wurde in einer Voranalyse die Ist-Situation in Form von situativen Faktoren und unternehmensspezifischen, langfristigen Methodenzielen erarbeitet. Dafür wurden mehrere Workshops und Einzelinter-

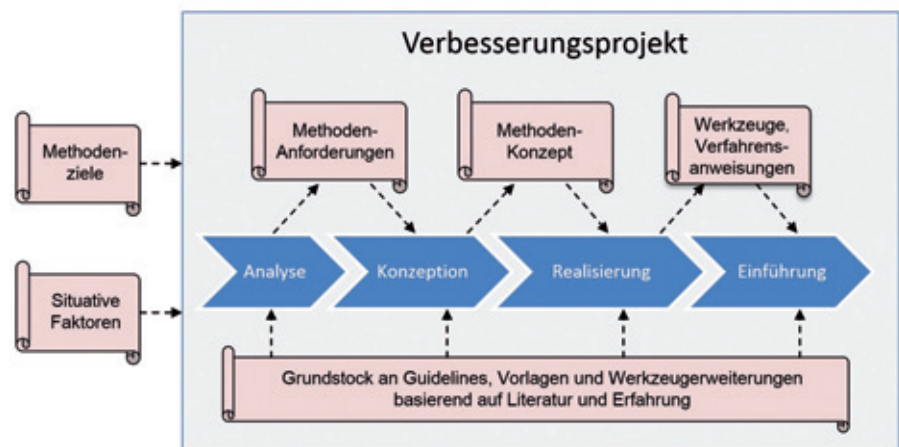


Abb. 2: Phasen eines Methodenverbesserungsprojektes

views durchgeführt, in denen den Projektbeteiligten ein Überblick über den Stand der Technik von RE-Methoden gegeben und ihre Meinung zu den Problemen in bisherigen RE-Projekten festgehalten wurde.

Wichtig ist, bei der Befragung alle RE-Kernbereiche *Erhebung, Abstimmung, Dokumentation, Prüfung und Verwaltung* abzudecken, um einen vollständigen Überblick zu gewinnen, auch wenn einzelne Kernbereiche in dem Unternehmen nur schwach ausgeprägt sind. Die **Tabellen 1-3** zeigen konkrete Beispiele für die erfassten situativen Faktoren, erkannten Probleme und langfristigen Methodenziele, wobei Letztere durch die erkannten Probleme motiviert sind.

**Situative Faktoren**

- Visuelle Sprachen: Grundkenntnisse der Mitarbeiter in UML vorhanden.
- Organisationsstruktur: Entwicklung wird verteilt durchgeführt. Off-Shore-Firmen sind beteiligt.
- Eingesetzte Anforderungswerkzeuge:
  - o zur Verwaltung: MS Excel, Enterprise Architect
  - o zur Dokumentation: MS Word, MS Visio, Enterprise Architect

Tab. 1: Situative Faktoren

**Probleme**

- Tester können die Anforderungen, gegen die getestet werden soll, nicht in einen größeren Kontext stellen, z. B. wegen fehlender Begründungen, fehlender Verfeinerungsbeziehungen (Traces) oder unvollständigen Beschreibungen. Daher kann nicht effizient bzw. effektiv getestet werden.
- Es herrscht keine Einigkeit darüber, was die Mindeststandards einer Anforderungsspezifikation sind.
- Anforderungen sind instabil.

Tab. 2: Probleme

**Methodenziele**

- Alle Entwicklungsartefakte sind zu ihrer Quelle zurückverfolgbar.
- Anforderungen sind eindeutig und werden von Spezifikateuren, Entwicklern und Testern gleich interpretiert.

Tab. 3: Methodenziele

Basierend auf den Erkenntnissen aus der Voranalyse konnten die Iterationen für Verbesserungsprojekte starten. Im Folgenden möchten wir über die Phasen der ersten Iteration (vgl. **Abbildungen 1 und 2**) berichten und erläutern, wie die Artefakte aus der Voranalyse systematisch behandelt wurden und am Ende eine erste Version der verbesserten RE-Methode entstand.

**Analyse**

Im Gespräch mit Entwicklern und Testern stellten wir fest, dass das in der Voranalyse definierte Problem bzgl. der Instabilität von Anforderungen darauf zurückging, dass den Testern oft der Kontext der vorliegenden Anforderungen unklar war, da die Beziehung zu anderen Anforderungen nicht dokumentiert wurde. Auch fehlten oft bestimmte Attribute, um sie eindeutig zuzuordnen (z. B. Autor und Version von Anforderungen).

Für die Entwickler waren vor allem textuelle Anforderungen oft mehrdeutig und zu unspezifisch. So mussten Anforderungen, bei denen dies offensichtlich war, nachträglich präzisiert werden, teilweise wurden diese Mängel aber zu spät erkannt. **Tabelle 4** zeigt beispielhaft Methodenanforderungen, die aus den Methodenzielen abgeleitet wurden.

**Methodenanforderungen**

- Rückverfolgbarkeit: Der Tester soll eine Übersicht der Zusammenhänge der Anforderungen in der Testbasis erhalten können.
  - o Dazu soll jede Anforderung einen eindeutigen Bezeichner tragen.
  - o Dazu soll jede Anforderung mit in Beziehung stehenden Anforderungen verknüpft werden.
- Zentrale Verwaltung: Alle Anforderungen eines Projektes sollen in einer zentralen Anforderungsdatenbank verwaltet werden.

Tab. 4: Methodenanforderungen

**Konzeption**

In der Konzeption wurde das Methodenkonzept der zukünftigen RE-Methode definiert, das die ausgewählten Methodenanforderungen unter Berücksichtigung der situativen Faktoren erfüllen sollte. Dazu wurden die Artefakttypen, Rollen, Aktivitäten und der Prozess der RE-Methode aktualisiert. Da die Entwicklung verteilt stattfindet, ist es wichtig, dass die Anforderungen vor der Weiterverwendung formal geprüft werden. In **Tabelle 5** illustrieren wir *nur auszugsweise* einen geänderten Artefakttyp, eine neue Aktivität und deren Nutzung im Prozess.

**Realisierung**

In der Realisierung wurden auf Grundlage des Methodenkonzeptes die Werkzeugunterstützung geschaffen und Verfahrensanweisungen für die Anwender der RE-Methode erstellt.

Wie bereits mit den situativen Faktoren festgehalten wurde, war das Werkzeug

**Methodenkonzept**

Artefakttypen:

- Requirement
  - o Sprache: Strukturierte natürliche Sprache
  - o Attribute:
    - ID (Pflichtfeld): Eindeutiger Bezeichner entsprechend dem Namensschema
    - Autor (Pflichtfeld): Name des Verantwortlichen für die Anforderung
    - Basiert auf: Liste von Requirement-IDs, Use-Case-IDs
    - Status (Pflichtfeld): ungeprüft, mangelbehaftet oder abgenommen
    - Prüfer: Name des Verantwortlichen für die letzte Prüfung

Aktivitäten:

- Requirement-Prüfung
  - o Kurzbeschreibung: In der Requirement-Prüfung wird jedes betreffende Requirement auf formale und inhaltliche Qualität geprüft.
  - o Schritte:
    - Prüfe, ob das Attribut „ID“ dem Namensschema entspricht
    - Prüfe, ob das Attribut „Basiert auf“ IDs von Requirements und Use Cases enthält, auf denen das betreffende Requirement basiert

Prozess:

- Vorgänger: Aktivität „Requirements-Spezifikation“
- Nachfolger:
  - o Falls NICHT alle Requirements den Status „abgenommen“ haben: Aktivität „Requirements-Spezifikation“
  - o Sonst: Meilenstein „Quality Gate Anforderungsspezifikation“

Tab. 5: Methodenkonzept

Enterprise Architect (EA) von Sparx Systems bereits teilweise im Unternehmen im Einsatz. Der EA ist zwar in erster Linie ein grafisches UML-Modellierungswerkzeug, unterstützt allerdings viele Aktivitäten aus den RE-Kernbereichen *Dokumentation, Prüfung und Verwaltung*. Wie in **Abbildung 3** dargestellt, lassen sich zum Beispiel natürlichsprachliche Anforderungen als Modellelemente in einer zentralen Datenbank hierarchisch organisieren (oben rechts) und in einer Tabellenansicht betrachten (oben links). Deshalb wurden diese Funktionen zur Erfüllung der Methodenanforderung bzgl. der zentralen Verwaltung eingesetzt.

Zur Erfüllung der Methodenanforderung bzgl. der Rückverfolgbarkeit nutzen wir die EA-Funktion, beliebige Modellelemente in einer Matrix-Ansicht (unten links in **Abbildung 3**) oder in Diagrammen (unten rechts) miteinander zu verknüpfen. Des Weiteren nutzen wir die EA-Erweiterungsmöglichkeiten durch *Skripte* und *Plug-ins*,

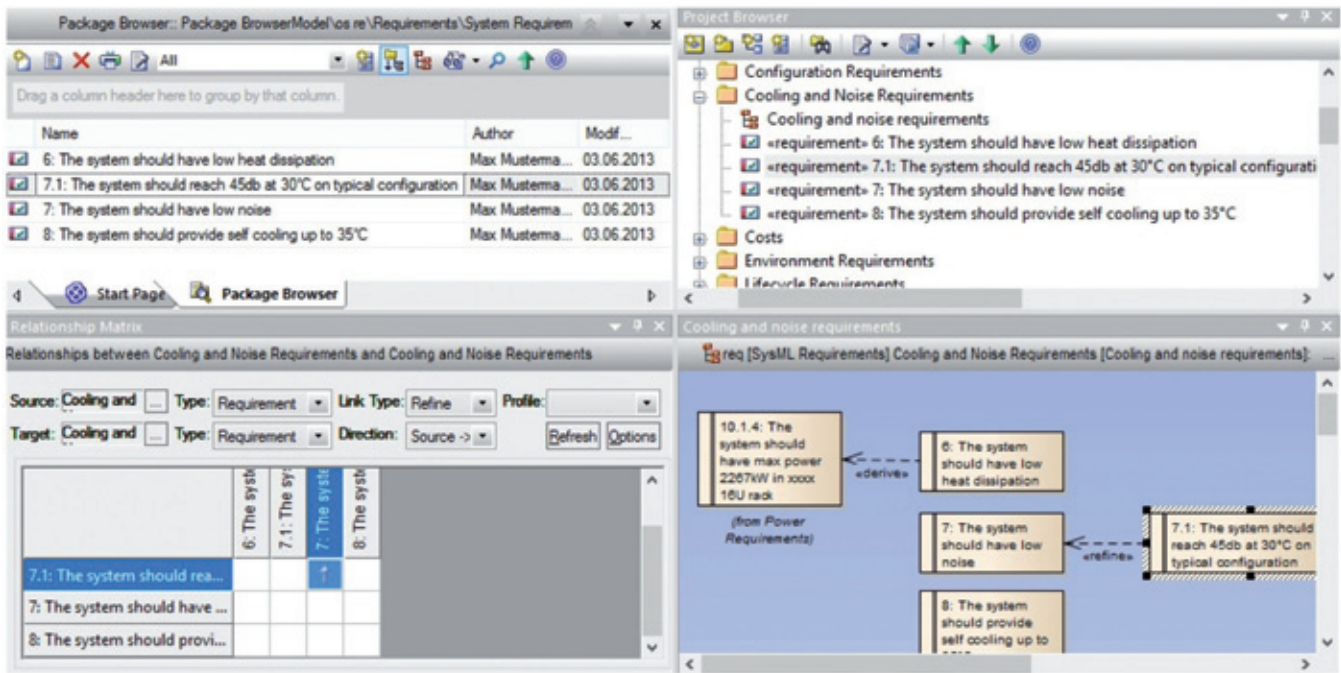


Abb. 3: Verschiedene Sichten auf textuelle Anforderungselemente im EA

um weitere für die Rückverfolgbarkeit relevante Techniken, wie z. B. Abhängigkeitsanalyse, umzusetzen:

- Ein direkt aus dem EA aufrufbares Skript meldet Requirements, die keinerlei Beziehungen zu anderen Requirements besitzen. Es meldet auch Requirements, deren Pflichtattribute leer sind. Diese Funktion beugt vor allem Flüchtigkeitsfehlern direkt nach der Einführung der RE-Methode vor.
- Ein direkt aus dem EA aufrufbares Skript generiert IDs für Requirements, die dem Namensschema des Methodenkonzepts entsprechen.
- Ein direkt aus dem EA aufrufbares Plug-in erlaubt es, eine Gruppe von Requirements auszuwählen, für die in Beziehung stehende Requirements angezeigt werden sollen. Es erzeugt dann ein Diagramm, das diese Abhängigkeiten visualisiert (siehe Abbildung 4).

Neben der Werkzeugunterstützung erstellen wir auch Verfahrensanweisungen, die beispielsweise beschreiben, wie Artefakte vom Typ „Requirement“ im Werkzeug erstellt werden sollen oder wie man prüft, ob die „Basiert auf“-Beziehungen zu anderen Requirements korrekt sind. Durch die zusätzliche Standardisierung mit dem neuen Methodenkonzept konnten diese Verfahrensanweisungen, insbesondere die

Prüfanweisungen viel detaillierter beschrieben werden als im Vergleich zur vorherigen Ist-Situation. Das Methodenkonzept und die Verfahrensanweisungen wurden später im Intranet als Methodenhandbuch zur Verfügung gestellt.

**Einführung**

Während der Einführung wurde die neue RE-Methode sukzessive im Unternehmen geschult und ausgerollt. Es handelte sich um eine mehrtägige Schulung mit theoretischen und praktischen Anteilen vor gemischtem Publikum (also Mitarbeitern verschiedener Bereiche und Ebenen). Um die RE-Methode schnell ins Unternehmen zu tragen, wurden ausgewählte Mitarbeiter zu zukünftigen unternehmensinternen Trainern fortgebildet. Es wurden außerdem Pilotprojekte definiert und die frühen

Phasen über Vor-Ort-Kick-offs und Reviews sowie telefonischen Support begleitet.

**Fazit und gewonnene Erkenntnisse**

In diesem Artikel haben wir unser systematisches, iteratives Vorgehen für die Entwicklung von organisationsspezifischen RE-Methoden vorgestellt.

Die Erhebung von organisationsspezifischen situativen Faktoren erleichtert die Konzeption von individuellen Maßnahmen, z.B. die Berücksichtigung von Mitarbeiterkenntnissen und vorhandenen Werkzeugen. Vor allem die Mitarbeiter, die die RE-Methode regelmäßig einsetzen, schätzen es, wenn sich die Änderungen gut in ihren bisherigen Arbeitsalltag integrieren und sie sich dadurch nicht überfordert fühlen.

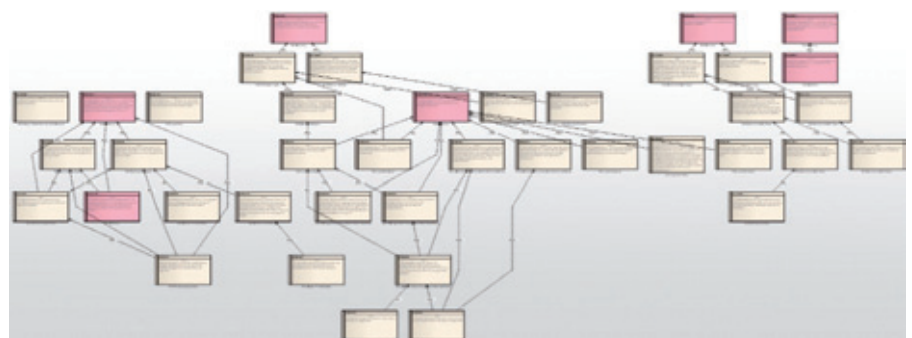


Abb. 4: Ergebnisdiagramm einer Abhängigkeitsanalyse aus einem EA-Plug-in

Dennoch stoßen wir bei unserem Vorgehen in der Praxis auch auf Widerstände von einzelnen Mitarbeitern und Abteilungen, die mit Veränderungen nicht einverstanden sind oder denen Änderungen zu weit gehen. Mit der Erhebung der situativen Faktoren können wir hier darauf verweisen, dass wir individuelle Umstände – soweit möglich – berücksichtigt und uns um den bestmöglichen Interessenausgleich bemüht haben.

Wir verfolgen ein systematisches Vorgehen, um im Vorfeld Probleme während des „Betriebs“, also der Durchführung eines konkreten Softwareentwicklungsprojektes, zu vermeiden. Die dokumentierten Artefakte verringern den Aufwand für Verbesserungs-

projekte folgender Iterationen, da diese direkt auf die Artefakte aufsetzen können.

Allerdings erwarten einige Stakeholder, dass bekannte Standardlösungen schnell und einfach im Unternehmen etabliert werden können und zweifeln die Vorteile eines systematischen und evtl. aufwendigeren Vorgehens bei der Methodenentwicklung an. Um den Aufwand gering zu halten, versuchen wir, möglichst immer auf unseren Grundstock an Fragen, Vorlagen und Werkzeugbibliotheken zurückzugreifen.

Das iterative Vorgehen hilft dabei, die mit den Methodenzielen ausgedrückten tiefgreifenden Veränderungen schrittweise zu planen und durchzuführen. Dadurch kann auf unerwartete Auswirkungen von

Veränderungen an der RE-Methode reagiert werden.

Insgesamt hat sich unser Vorgehen bewährt. Nach Aussagen der Mitarbeiter hat sich die Situation in den Unternehmen an vielen Stellen verbessert. Eine Frage, die immer wieder aufgeworfen wird, ist allerdings die Quantifizierung der Verbesserungen. Prinzipiell ließe sich diese über die dedizierte Erhebung und Betrachtung von Key-Performance-Indikatoren (KPI) im Entwicklungsprozess feststellen. Dies wird allerdings von unserem Vorgehen noch nicht explizit unterstützt. Außerdem erweist sich der Vorher/Nachher-Vergleich in der Praxis oft als schwierig, da entsprechende Werte für die vergangenen Projekte selten vorliegen. ■

## Referenzen

- [BWBM08]** W. Bekkers, I. van de Weerd, S. Brinkkemper und A. Mahieu. The Influence of Situational Factors in Software Product Management: An Empirical Study. In Proceedings of the 2008 Second International Workshop on Software Product Management, Seiten 41–48. IEEE Computer Society, 2008.
- [Coc00]** Alistair Cockburn: Writing Effective Use Cases. Addison-Wesley, 2000.
- [HP07]** HP White Paper: Reducing risk through requirements-driven quality management: An end-to-end approach. HP, 2007. [http://www.hp.com/hpinfo/newsroom/press\\_kits/2008/swuniversevienna/docs/requirements-driven\\_quality\\_management.pdf](http://www.hp.com/hpinfo/newsroom/press_kits/2008/swuniversevienna/docs/requirements-driven_quality_management.pdf)
- [IEEE98]** IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830. IEEE, 1998.
- [IIB09]** International Institute of Business Analysis: A Guide to the Business Analysis Body of Knowledge (BABOK GUIDE). International Institute of Business Analysis, 2009.
- [PuR09]** Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering. dpunkt.verlag, 2009.
- [RuR10]** Suzanne Robertson, James Robertson: Mastering the requirements process. Addison-Wesley, 2010.
- [Rup09]** Chris Rupp, die SOPHISTen: Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis. Carl Hanser Verlag, 2009.
- [SQS12]** SQS Group: SQS PractiQ: Early Error Detection by Test Case Specification. September 2012 [http://www.sqs.com/de/de/Download/eed\\_de.pdf](http://www.sqs.com/de/de/Download/eed_de.pdf)
- [SuS97]** Ian Sommerville, Pete Sawyer: Requirements engineering: a good practice guide. John Wiley & Sons, 1997.
- [SuL10]** Andreas Spillner, Tilo Linz: Basiswissen Softwaretest, Ausgabe 4. dpunkt.verlag, 2010.