

Anforderungsmanagement: Von Open Source Projekten lernen

Methodische Spezifikation der Anforderungen scheitert in der Praxis oft am unterschiedlichen Wissens- und Interessensstand von Fachabteilung und IT. Die Fachabteilung will ihre aktuellen Anforderungen umgesetzt wissen, die IT will abnehmbare Spezifikationen um sich rechtlich abzusichern. Die interne Zerrissenheit der IT in Plan-Build-Run trägt nur wenig zur Kundenzufriedenheit bei. Open Source Projekte weisen dagegen eine hohe Kundenzufriedenheit und einen hohen Zielerreichungsgrad auf, obwohl in den meisten Fällen kein methodisches Vorgehen namhaft gemacht werden kann. Es stellt sich die Frage, ob Open Source Projekte generell einen besseren Ansatz zur Anforderungsspezifikation verfolgen, wie solche Ansätze aussehen können und ob sie in konventionellen IT-Projekten anwendbar sind.

Definition Anforderungsmanagement:

Prozess um die Anforderungen an Softwaresysteme (in weiterer Folge IT-Systeme) zu erheben, zu dokumentieren und zu warten (=langfristig aktuell zu halten).

Seit einigen Jahren beweisen Open Source Projekte ihre nachhaltige erfolgreiche Entwicklung. In einigen Bereichen sind sie bereits führend, in andern Bereichen holen sie stark auf. Anhand von 3 Hypothesen soll geklärt werden, ob und wie Anforderungsmanagement in Open Source Projekten zum Erfolg beiträgt:

H1: Anforderungsmanagement wird nicht benötigt (ist redundanter Overhead)

H2: Word (ev. Excel) reicht als Tool vollkommen aus

H3: Open Source Projekte sind effizienter als konventionelle Projekte

Motivation

Formal spezifizierte Anforderungsdokumente werden nicht unmittelbar von der bedarfstragenden Fachabteilung oder der genehmigenden Geschäftsführung benötigt.

Sie dienen der IT als Planungs- und Kontrollinstrument sowie als Hilfestellung bei Divergenzen.

Warum / Wofür

Anforderungsspezifikationen dienen folgendem Zweck:

- Aufwands-, Kosten- und Zeitplanung
- Festlegen von Skills und Kompetenzabgleich
- Toolauswahl
- Kommerzielle Rahmenbedingungen

Wer

Primäre Nutzer sind:

- Interne IT-Abteilungen
- Sachverständige und Mediatoren (im Streitfall)

Wie / Wieviel

- Standardsoftware (gering) → Individualentwicklung (umfassend)
- Unkritisch (informell) → Kritisch, lebensnotwendig (Formal)
- Kurzlebig (rudimentär) → Langfristig (detailliert)

Anforderungsspezifikationen sind (entgegen der Lehrmeinung) ungeeignet:

- Zur Absicherung des Projekterfolgs
- Als Vorgabe für die Umsetzung und Bauplan der Lösungspartner
- Als Vorgabe für Standardsoftware

Anforderungsmanagement und Projektarten

In **wasserfallartigen Projekten** werden die Anforderungen zu Beginn erhoben und danach eingefroren. Komplexere Projekte laufen damit entweder Gefahr, am zukünftigen **Bedarf vorbei** zu produzieren oder durch Change Requests den ursprünglichen Zeit- und Aufwandsplan zu sprengen.

Iterative Vorgehensmodelle verschränken Spezifikation und Umsetzung. **Designmängeln** werden erst spät erkannt. Damit steigt die Gefahr der Entwicklung unwartbarer **Legacyanwendungen** mit der Agilität des Vorgehens.

Agile Methoden stellen frühzeitig funktionsfähige Lösungen bereit. Projektintern setzt man auf methodische Hilfestellungen wie Testdriven Development, Continuous Integration und Refactoring. **Designmängel** sind damit nicht vermeidbar aber **behebbar**.

In **Open Source Projekten** ist fachliche und technische Kompetenz oft nicht getrennt („Every project starts with a developers itch“ gilt auch heute noch). **Interaktion** mit den Anwendern und **häufige Tests** sind **systemimmanent** („Release early, release often“) und verifizieren rasch und oft die Anforderungen. **Transparenz** der internen Konstruktionsprinzipien führen zu einem **klarerem, stabilerem Design** und **wartbarem Code**. Durch Kopieren kann von anderen Projekten abgeleitet und gelernt werden.

Einfache Werkzeuge zum Verteilen von Lösungen bieten **Kommunikationshilfen** zwischen Anwendern und Entwicklern. Diese reicht von erweiterten Anforderungen, Testberichten und Nutzerfeedback über Codebeiträge zu Fehlerbehebungen. **Metrikssysteme** ermöglichen eine Bewertung der Projektqualität. Entwicklung und Betrieb sind (zumindest anfänglich) in einer Hand. Das führt zu **frühzeitig festgelegtem Systemdesign**.

Aber

Open Source hat auch Schwächen:

- Keine kommerzielle und juristische Führungsfunktionen
- Passende Lizenzwahl (Auswirkungen auf zukünftige Geschäftsmodelle) schwierig
- Kommerzielle Problemstellungen sind keine technischen Herausforderungen

Diese Punkte sich aber nicht unmittelbar auf das Anforderungsmanagement aus.

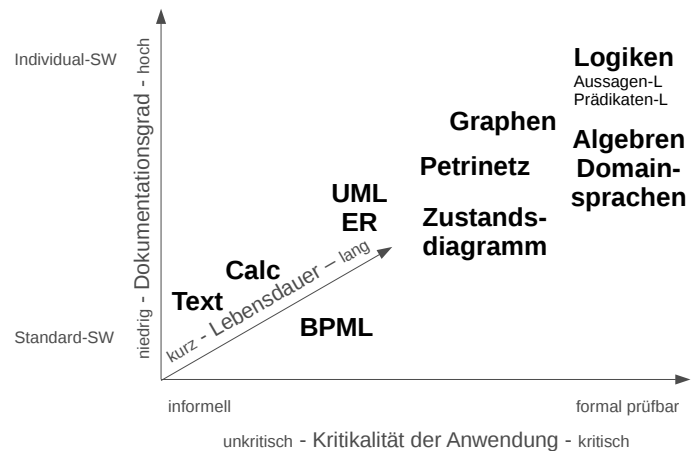
Wieviel Anforderungsmanagement braucht ein Projekt?

Aus Sicht des Fachbereichs ist jede formale Spezifikation zu viel. „Die Prozesse sind klar, die Zusatzanforderungen marginal.“

Je komplexer die Systeme, desto umfassender die Erhebungsmethoden und desto größer wird erfahrungsgemäß der **Akzeptanzunterschied** zwischen Fachbereich und IT. Mit steigender Komplexität der Anwendung steigen auch die Anforderungen an die Ersteller (Entwickler) und Begutachter (Fachanwender).

Traditionell werden Anforderungen in **freiformatigen** Textdokument erfasst. Diese unterstützen **keine Formalprüfung** auf Konsistenz, Korrektheit, Klarheit und Widerspruchsfreiheit.

Hochkomplexe Werkzeuge mit hohem Lernaufwand wie die Rational Suite oder Abacus sind **nicht zwingend notwendig**. Komplexe Open Source Projekte beweisen, dass mit kombinierten webbasierten Lösungen Dokumente unterschiedlichen Formalisierungsgrades bereitgestellt werden können (bugtraq, launchpad, github, sourceforge etc.).



Erfahrungen aus Open Source Projekten

Open Source Projekte spezifizieren Anforderungen in **kurzen Workshops** mit hohem persönlichen Interaktionsanteil. Die Dokumentation erfolgt in mehrfach genutzten webbasierten Werkzeugen, welche im Entwicklungsprozess weitergenutzt werden.

Traditionell werden Anforderungen in MUSS, SOLL und KANN unterteilt, um eine Priorisierung vorzunehmen. Die Verteilung weist oft eine Überhang beim MUSS auf. Open Source Projekte unterteilen die Anforderungen nach dem **Änderungsgrad** in STATISCH, DYNAMISCH und UNBEKANNT. Daraus lassen sich Rückschlüsse auf die benötigte Qualität, Flexibilität, Erweiterbarkeit ziehen und das Systemdesign ableiten.

Gemeinsam mit **versionierten Roadmaps** werden hypertrophe Lösungen vermieden, die Vorteile iterativer Entwicklungskonzepte genutzt und die Systeme bleiben einfach und beherrschbar.

Schlussfolgerungen

H1 ist falsch: Anforderungsspezifikationen werden benötigt. Ihr Hauptnutzen liegt in der Definition einer Baseline sowie als Beweismittel.

H2 ist falsch: Textverarbeitung und Tabellenkalkulation sind unzureichende Werkzeuge. Damit kann man Anforderungen nicht gewichtet vernetzen, kategorisieren und priorisieren. Roadmaps und Testkonzepte sind daraus nicht ableitbar.

H3 ist bedingt richtig: Die Methodik und Toolunterstützung zur Anforderungsdefinition kann von Open Source Projekten bei einfachen und unkritischen Projekten übernommen werden. Je komplexer und kritischer die Lösung wird, desto mehr muss man auf formale Methoden zurückgreifen.

Anforderungsmanagement konzentriert sich auf die zu schaffende Lösung aber nicht auf die zugrundeliegenden Probleme. Open Source Projekte betrachten auch die Problemstellung.

Tools

Open Source Tools zur Unterstützung des Anforderungsmanagements:

| | |
|--|---|
| Arbiter | http://arbiter.sourceforge.net/ |
| BOUML | http://bouml.free.fr/ |
| Concordion | http://www.concordion.org/ |
| Concordion.Net | http://code.google.com/p/concordion-net/ |
| Efficient | http://efficient.citi.tudor.lu/ |
| FaMa-FW | http://www.isa.us.es/fama/ |
| Frame UML | http://sourceforge.net/projects/frameuml/ |
| Green | http://green.sourceforge.net/ |
| iConcur Axiom 2011 | http://marketplace.eclipse.org/content/iconcur-axiom |
| JavaRequirementsTracer | http://reqtracer.sourceforge.net/ |
| JFeature | http://www.technobuff.net/webapp/product/showProduct.do?name=jfeature |
| JNarrate | https://bitbucket.org/testingreflections/jnarrate/wiki/Home |
| Modelio | http://www.modelio.org/ |
| Open Source Requirements Management Tool | http://sourceforge.net/projects/osrmt/ |
| OpenOME | https://se.cs.toronto.edu/trac/ome/wiki |
| Open Requirements Management Framework | http://www.eclipse.org/proposals/ormf/ |
| Oryx Editor | http://code.google.com/p/oryx-editor/ |
| pyconcordion | http://code.google.com/p/pyconcordion/ |
| Quick Sequence Diagram Editor | http://sdedit.sourceforge.net/ |
| Rally | http://www.rallydev.com/ |
| Requirement Heap | http://sourceforge.net/projects/reqheap/ |
| RE-Tools | http://www.utdallas.edu/~supakkul/tools/RE-Tools/ |
| RTH - Requirements and Testing Hub | http://requirementsandtestinghub.wordpress.com/ |
| RTH Turbo | http://code.google.com/p/rth-turbo/ |
| Ruby Concordion | http://ruby-concordion.rubyforge.org/ |
| S.P.L.O.T. | http://www.splot-research.org/ |
| Sculptor | http://fornax-platform.org/cp/x/aAQ |
| SLAM Software Lifecycle Artefact Manager | http://sourceforge.net/projects/slamreq/ |
| specs | http://code.google.com/p/specs/ |
| Spectacular | http://code.google.com/p/spectacular/ |
| Stylebase for Eclipse | http://stylebase.tigris.org/ |
| TextUML Toolkit | http://abstratt.com/ |
| Umbrello UML Modeller | http://uml.sourceforge.net/index.php |
| Umple | http://cruise.site.uottawa.ca/umple/ |
| UML Almighty | http://www.uml-almighty.com/ |
| UNICASE | http://teambuegge.informatik.tu-muenchen.de/groups/unicase/ |
| Use Case Maker | http://use-case-maker.sourceforge.net/ |

20131115, Wolf Rogner, corporate quality consulting GmbH, FH Technikum Wien

(wolf.rogner@cqc.at, <http://www.cqc.at>, rognernw@technikum-wien.at, <http://www.technikum-wien.at>, w.rogner@rsb.at, <http://www.rsb.at>, +43 664 88446815, +43 681 20126764)

Wolf Rogner: Nach unterschiedlichsten Führungsfunktionen – von der Softwareentwicklung bis zum RZ-Betrieb – beschäftigt sich Hr. Rogner heute mit Themen rund um Open Source, offene Innovationen und IT Strategien. Neben seiner Beratungstätigkeit unterrichtet er zu offenen Systemen an der FH Technikum Wien. Als allgemein beedeter, gerichtlich zertifizierter Sachverständiger für Informationstechnologie erstellt er Gerichts- und Privatgutachten.

corporate quality consulting GmbH. gehört zu Deutschlands führenden IT- und Innovationsberatungen. Das 2005 gegründete Unternehmen zählt Topunternehmen der deutschen und internationalen Wirtschaft, Finanz und Politik zu seinen Beratungskunden.

FH Technikum Wien ist Österreichs größte technische Fachhochschule. Jährlich werden über 3.000 Studenten ausgebildet. Das Technikum Wien ist die erste Universität, welche Open Souce nicht nur intern nutzte, sondern auch im Ausbildungsweg konsequent anbietet.