

Interdisziplinäres Requirements-Engineering

Editorial



Kommunikation²

Im Requirements-Engineering ist viel Kommunikation erforderlich. Dies gehört zum Basiswissen eines jeden, der schon einmal spezifiziert hat.

Die Kommunikation ist in Projekten, die nur eine Disziplin umfassen (wie z.B. ein reines Softwareprojekt) oftmals schon recht schwierig (der Auftraggeber/Produktverantwortliche hat oft zu wenig Zeit, die Requirements werden nur als Texte formuliert und sind oft unklar, etc.). Außerdem gibt es auch in unidisziplinären Projekten meist eine große Zahl von Stakeholdern, die bei der Spezifikation von Requirements mitreden bzw. eingebunden werden sollten. Diese Stakeholder durch eine gute Kommunikation und ein passendes Umfeld zu synchronisieren und auf den gemeinsamen Weg zu bringen gestaltet sich in vielen Fällen als recht schwierig und aufwändig.

Im interdisziplinären Umfeld potenzieren sich diese Kommunikationsaufwände.

Es gibt vielfach mehr Stakeholder, da ja übergreifend alle Disziplinen berücksichtigt werden müssen. Manch einer verliert in so einem Umfeld den Überblick und die Projekte werden sehr schwer steuerbar. Damit dies nicht passiert, sollte man sich im interdisziplinären Umfeld zeitgerecht vor dem Projektstart überlegen, welche Vorgehen, Methoden und Techniken des Requirements-Engineerings angewendet werden sollen, damit das Projekt ein Erfolg wird. Dieser Knowledge Letter bietet einige Denkanstöße und Hilfestellungen dafür.

Viel Spaß beim Lesen und anregende Diskussionen bei der Einführung und Anwendung des interdisziplinären Requirements-Engineerings.

Dipl.-Ing. Johannes Bergmann

Berater, Trainer, Ziviltechniker für Informatik

Um mit der digitalen Transformation und deren speziellen Ausprägungen wie z.B. Industrie 4.0 oder Internet of Things erfolgreich zu sein, müssen heute Projekte gestartet werden, die im Gegensatz zu früher oft isoliert durchgeführten Systementwicklungen eine übergreifende Sicht auf Business-Prozesse und die betroffenen Fachdomänen berücksichtigen.

Der Artikel stellt dar, wie Requirements-Spezifikationen erstellt und gehandhabt werden sollten, wenn in einem interdisziplinären Projekt verschiedene Fachbereiche (z.B. Software, Mechanik, Elektronik, Pneumatik, Produktion, Vertrieb, Marketing, etc.) zusammenarbeiten und doch ein sinnvolles Ganzes als Produkt daraus entstehen soll.

Einleitung

Gerade in einem interdisziplinären Projekt, in dem verschiedene Fachbereiche zusammenarbeiten, um ein sinnvolles Ganzes als Produkt daraus entstehen zu lassen, ist gutes Requirements-Engineering einer der wichtigsten Erfolgsfaktoren.

Es gilt hier, nicht nur innerhalb des Fachbereichs klar zu verstehen, was der Kunde wünscht und wie es umgesetzt werden kann, sondern es müssen auch Sichtweisen aus den anderen Fachbereichen einfließen und berücksichtigt werden.

Es sind zusätzliche Abhängigkeiten und Schnittstellen übergreifend zu beachten. Durch effektive Zusammenarbeit zwischen den Fachbereichen lassen sich oft bessere Lösungen kostengünstiger entwickeln. Wichtig dabei ist jedoch, dass die Fachbereiche voneinander wissen, sich entsprechend abstimmen und regelmäßig „synchronisieren“. Nur wenn alle im Gleichklang rudern, geht es zügig voran! Dabei ist vor allem in den hardware-lastigen Fachbereichen ein Umdenken gefragt, da auch hier immer stärker die agilen Methoden zur Anwendung kommen.

Wenn man das Thema interdisziplinäres Requirements-Engineering genauer betrachten möchte, sollte man sich unter anderem folgende Fragen stellen:

- Welche Requirements-Spezifikationstechniken sind im interdisziplinären Umfeld sinnvoll anwendbar?
- Welche Requirements-Artefakte können/sollen von welchen Fachbereichen erstellt und verwaltet werden und wie können sie in einen sinnvollen Zusammenhang gebracht werden?
- Wie kann sichergestellt werden, dass der Überblick bei hunderten oder tausenden von Requirements auch fachbereichsübergreifend nicht verloren geht?
- Wann und mit welchen Techniken kann/soll man die Details in den einzelnen Fachbereichen spezifizieren und inwieweit sollen die Fachbereiche übergreifend die Details kennen?
- Wer von den zusammenarbeitenden Bereichen kommuniziert wie mit dem Kunden?
- Wie kann heute auch in hardware-lastigen Fachbereichen agil und „synchron“ zur SW-Entwicklung vorgegangen werden?

Nachfolgend werden in diesem Zusammenhang mögliche Wege der Zusammenarbeit aufgezeigt und Techniken im gemeinsamen Umgang mit einzelnen Requirements sowie auch mit der gesamten Spezifikation dargestellt.

Motivation

Früher gab es im Engineering sehr oft nur ein einziges System mit klaren Aufgaben und ohne komplexe Schnittstellen.



Abb. 1 – Systemkomplexität früher

In diesen Situationen ist eine isoliert durchgeführte Systementwicklung durchaus möglich und wurde von vielen Engineering Unternehmen auch so umgesetzt.

Heute gibt es komplexe und spezielle Ausprägungen und Anforderungen an das Engineering wie z.B. Industrie 4.0 oder Internet of Things. Die Systeme können nicht mehr nur isoliert voneinander betrachtet werden, sondern müssen als Netzwerk von Systemen gesehen werden!

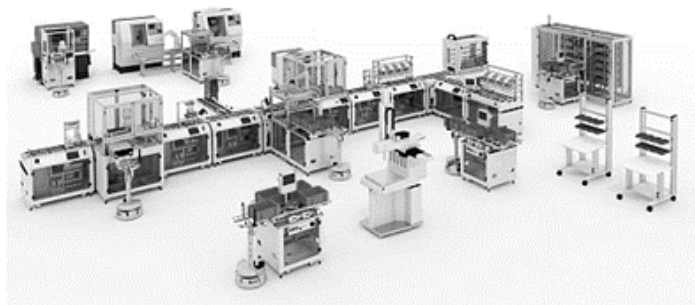


Abb. 2 – Systemkomplexität heute

Konsequenterweise ist daher auch beim Requirements-Engineering eine übergreifende Sicht auf die Business-Prozesse und betroffenen Fachdomänen unbedingt erforderlich!

Unterschiedliche Sichten

Es gibt durch die höhere Systemkomplexität und die unterschiedlichen Fachdomänen natürlich unterschiedliche Sichten und teilweise auch Widersprüche zwischen den Sichten zu berücksichtigen.

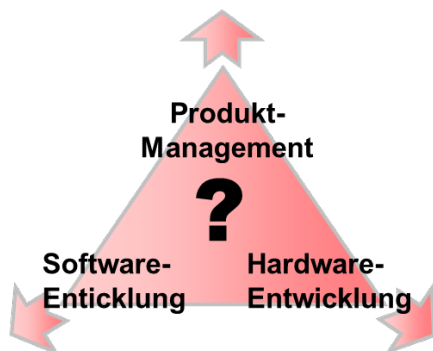


Abb. 3 – unterschiedliche Sichten

Vor allem bezüglich Umsetzungsdauer, Flexibilität, Vorgehensmodell, Qualität, etc. können sich Differenzen ergeben. So kann z.B. das Produktmanagement einen Fokus auf viele Features und schnelle Time-to-Market haben und „natürlich“ beste Qualität wollen.

Der Fokus in der SW-Entwicklung ist manchmal – vor allem in frühen Requirements-Phasen – noch recht unklar. Ganz generell erwartet man sich meist sehr schnelle Zyklen und Reaktionszeiten und die Entwickler wünschen sich eine prototypische und agile Arbeitsweise.

Die Hardwareentwicklung dagegen benötigt meist längere Zyklen, hat besondere Restriktionen bezüglich Austauschbarkeit von Komponenten und physikalischen Gegebenheiten und muss ev. von Beginn an klarer definiert sein.

Gutes Requirements-Engineering hilft, diese Widersprüche zu reduzieren.

Unterschiedliche Vorgehensmodelle

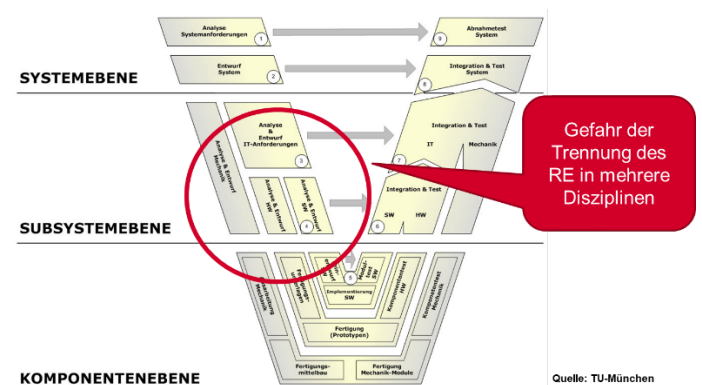


Abb. 4 – Interdisziplinäres RE im V-Modell

Im V-Modell ist bereits auf der obersten Ebene (Systemebene) ein interdisziplinärer Startpunkt vorgesehen.

Die Gefahr lauert dann auf den unteren Ebenen, wenn das Engineering in die einzelnen Disziplinen getrennt wird.

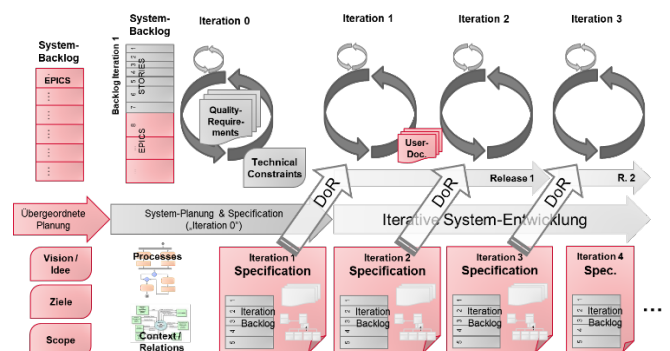


Abb. 5 – Interdisziplinäres RE im agilen Umfeld

In der agilen Entwicklung ist die Gefahr einer Trennung weniger gegeben, sofern funktionsübergreifende interdisziplinäre Teams

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.



Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter <http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/>

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Letters von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!



Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungsergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com

