

# Der Umgang mit Requirements-Engineering an wissenschaftlichen Bibliotheken

Susanne Blumesberger\*

*Zielsetzung* — Ziel der Arbeit war es zu zeigen, dass wissenschaftliche Bibliotheken von einem professionellen Umgang mit Anforderungen in IT-Projekten profitieren. In diesem Zusammenhang sollte auch untersucht werden, inwieweit Requirements-Engineering in wissenschaftlichen Bibliotheken in Österreich verbreitet ist.

*Forschungsmethoden* — Neben einem Literaturreview zum Requirements-Engineering wurde eine explorative Umfrage mit BibliothekarInnen und IT-ExpertInnen durchgeführt.

*Ergebnisse* — Die Untersuchung zeigt, dass Requirements Engineering in wissenschaftlichen Bibliotheken noch nicht in ausreichender Weise angewendet wird, obwohl bereits intensiv mit IT-Abteilungen zusammengearbeitet wird.

*Schlussfolgerungen* — Wissenschaftliche Bibliotheken sollten einem professionellen Requirements-Engineering mehr Aufmerksamkeit schenken, um auf lange Sicht Zeit und Kosten zu sparen. BibliotheksmitarbeiterInnen sollte der Zugang zu diesem Themenfeld mittels Schulungen ermöglicht werden, generell scheint eine stärkere Bewusstseins-schaffung in diesem Bereich wichtig.

*Schlagwörter* — IT-Projektmanagement, Requirements-Engineering, wissenschaftliche Bibliothek

## Dealing with requirements engineering at academic libraries

*Objective* — The aim of the work was to show that academic libraries benefit from a professional handling of requirements in IT projects. In this context, it should also be examined to what extent requirements engineering is widespread in academic libraries in Austria.

*Methods* — In addition to a literature review on requirements engineering, an explorative survey was conducted with librarians and IT experts.

*Results* — The research shows that requirements engineering is not yet sufficiently used in academic libraries, although there is already intensive collaboration between IT departments and libraries.

*Conclusion* — Academic libraries should pay more attention to professional requirements engineering in order to save time and money in the long term. Library staff should be given access to this area through training, and overall awareness raising in this area seems important.

*Keywords* — IT project management, requirements engineering, academic library

Diesem Beitrag liegt folgende Abschlussarbeit zugrunde / This article is based upon the following thesis:  
Blumesberger, Susanne: Der Umgang mit Requirements-Engineering an wissenschaftlichen Bibliotheken. Master-Thesis (ULG), Universität Wien. Universitätslehrgang Library and Information Studies (MSc) 2018.  
Der Volltext der Abschlussarbeit ist unter <https://othes.univie.ac.at/46030> abrufbar.

\* Susanne Blumesberger | [susanne.blumesberger@univie.ac.at](mailto:susanne.blumesberger@univie.ac.at) | <http://www.blumesberger.at> | ORCID: 0000-0001-9018-623X



## 1 Einführung

Der theoretische Teil der Arbeit umfasst die Darstellung des aktuellen Stands zum Requirements-Engineering, die Klärung der Begriffe Requirements-Engineering (RE) und Requirements-Management (RM), die Aufgaben und die Einsatzmöglichkeiten von RE, sowie Ausbildungsmöglichkeiten. Der empirische Teil umfasst die Umfrage über den Einsatz von RE in wissenschaftlichen Bibliotheken, sowie die Präsentation und die Interpretation der Ergebnisse.

Die wissenschaftliche Literatur zum Thema RE bzw. RM ist noch recht überschaubar. Hier ein kurzer chronologischer Überblick: »Requirements Engineering« von Helmut Partsch (1991) hat sich zum Ziel gesetzt, Basiswissen zu vermitteln und vor allem Interesse für das Thema zu wecken. Obwohl das Werk nicht mehr auf dem neuesten Stand der Technik ist, handelt es sich doch um eine gut lesbare und verständlich aufbereitete Einführung in das Thema. Dirk Heche und Oliver Ollech (2006) geben in ihrem Beitrag »Projektbegleitendes Anforderungsmanagement: unnötige Kosten und Ärger vermeiden« einen knappen aber informativen Ersteinblick in das Thema und fassen die Argumente, warum man mit Anforderungen sorgsam umgehen sollte, übersichtlich zusammen. Chris Rupp und Christof Geißel (2007) beschäftigen sich in ihrem übersichtlichen Beitrag »Nichtfunktionale Anforderungen strukturiert erfassen und wieder verwenden« mit dem Umgang von

nicht-funktionalen Anforderungen, also all jenen Anforderungen, die die Qualität des Produktes bestimmen bzw. alles, was das System benutzerfreundlich macht. »Anforderungsmanagement in sieben Tagen. Der Weg vom Wunsch zur Konzeption« von Thomas Niebisch (2013) ist für alle EinsteigerInnen sehr zu empfehlen, da im Bereich Anforderungsmanagement die einzelnen Schritte, aber auch mögliche Probleme und Hindernisse aufgezeigt werden. Sehr zu empfehlen ist das bereits in 6. Auflage erschienene Handbuch »Requirements-Engineering und -Management. Aus der Praxis von klassisch bis agil« von Chris Rupp & die SOPHISTen (2014). Es führt in klar definierten Schritten durch den gesamten Prozess des RE und zeigt anhand von praxisnahen Beispielen aus dem Bibliotheksbetrieb anschaulich und humorvoll, wie man mit RE umgehen kann. Sehr hilfreich sind auch die Zusatzmaterialien im Internet. Im Beitrag von Matthias Kulke (2014) »Wie Sie die natürliche Sprache bändigen. Die SOPHIST-Satzschablone: Funktionale Anforderungen präzise formulieren« wird übersichtlich dargestellt, warum Satzschablonen sinnvoll sind und wie sie optimal gestaltet werden. Für einen kompakten aber fundierten Überblick ist das nun schon in vierter Auflage erschienene Buch von Klaus Pohl und Chris Rupp (2015) »Basiswissen Requirements Engineering. Aus- und Weiterbildung zum »Certified Professional for Requirements Engineering«, ein idealer Einstieg.

## 2 Was bedeutet Requirements-Engineering?

*Das, was Sie als Requirements-Engineer erstellen ist [...] ein Modell der Realität, das den Lösungsmodellraum aufspannt, indem sich später der Architekt austoben kann.*

(Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 24)

Nach Patig und Dibbern (2008–2018) umfasst »Requirements-Engineering [...] das Ermitteln, Analysieren, Spezifizieren und Validieren aller Eigenschaften und Rahmenbedingungen eines Softwaresystems, die über seinen gesamten Lebenszyklus ge-

wünscht werden bzw. relevant sind.« Moser (2007, S. 11) definiert RE als »das Aufstellen von Anforderungen. Darunter fallen die Bereiche Anforderungserhebung (Elicitation), Analyse, Dokumentation bzw. Kommunikation, Prüfung (Validation) und Abstimmung (Negotiation). Unter RE wird der Weg von der Projektidee bis zu einem vollständigen Satz von Anforderungen verstanden.« Pohl und Rupp (2015, S. 4) verstehen unter RE »den systematischen und disziplinierten Ansatz zur Spezifikation und zum Management von Anforderungen mit folgenden Zielen:

1. Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform

- zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
- Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren sowie die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht«.

Unter RM »wird der Prozess zum Managen der Anforderung verstanden, es umfasst Maßnahmen, welche die Anforderungsanalyse und die weitere Verwendung der Anforderungen unterstützen.« (Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 368). Demnach ist RM ein Teilgebiet von RE das mit vielen Aufgaben verbunden ist. Die Verwaltung von Anforderungen umfasst mehrere Techniken, wie die Attributierung, die Priorisierung, die Verfolgbarkeit, die Versionierung, das Änderungsmanagement und die Messung von Anforderungen (siehe auch Pohl und Rupp 2015, 119 ff.).

Anforderungen, auch als »Forderungen«, »Ziele«, »Eigenschaften«, »Features« oder fälschlicherweise als »Wünsche« bezeichnet, kann man auch folgendermaßen definieren:

- eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird;
- eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen;

- eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2) (IEEE 1998, übers. von Pohl und Rupp 2015, S. 3).

Die Projekte, die in den letzten Jahren beispielsweise rund um das Forschungsdatenmanagement in wissenschaftlichen Bibliotheken durchgeführt wurden, basieren auf Entwicklungen von Technologien. Beim Aufbau von Repositorien muss klar sein, welche Anforderungen die Forschenden an das jeweilige System stellen, um die Vorgaben der Fördergeber erfüllen zu können. RE unterstützt die Umsetzung dieser Anforderungen.

RE hängt eng mit dem Projektmanagement zusammen, ist aber auch zugleich verbunden mit der Idee der agilen Softwareentwicklung. Dabei stehen möglichst häufige Rückkoppelungsprozesse und zyklisches (iteratives) Vorgehen auf allen Ebenen im Vordergrund – sowohl bei der Programmierung als auch beim Management. Im Unterschied zur klassischen Vorgehensweise wird bei der agilen Softwareentwicklung das neue System nicht in allen Einzelheiten bis zum Projektende genau durchgeplant und dann in einem einzigen langen Durchgang entwickelt, sondern es wird die Tatsache berücksichtigt, dass sich Anforderungen während der Projektlaufzeit immer wieder ändern bzw. zu Projektbeginn noch gar nicht vollständig bekannt sein können. Bei der agilen Softwareentwicklung wechseln sich daher kurze Planungs- und Entwicklungsphasen ab. Durch ständige Prüfungen sind Fehlentwicklungen rasch erkennbar und es kann unmittelbar korrigierend eingegriffen werden (siehe Beck et al. 2001). RE ergänzt das klassische Projektmanagement ideal, indem es sich intensiv mit Anforderungen beschäftigt.

### 3 Worum geht es bei Requirements-Engineering genau?

Kurz zusammengefasst kann man hier sechs Punkte nennen:

- Formulierung von klaren und unmissverständlichen Anforderungen,
- Prüfung der Sinnhaftigkeit und Durchführbarkeit von Anforderungen,
- Verwaltung der Anforderungen,
- Begleitung des Prozesses von der Wunschäußerung bis zur Umsetzung,

- Schaffung einer gemeinsamen sprachlichen Basis für alle Beteiligten sowie
- unmissverständliche Kommunikation mit den AuftraggeberInnen, Usern und TechnikerInnen.

Um Anforderungen effektiv umsetzen zu können, ist es wichtig, dass der Kommunikationsfluss nie abreißt. Jegliche Kommunikationslücke birgt die Gefahr, dass sich Widersprüche einschleichen oder Doppelungen entstehen. Der Requirements-Engineer muss Katalysator zwischen AuftraggeberInnen,

TechnikerInnen und den einzelnen Stakeholdern sein. Die unterschiedlichen Wünsche der KundInnen müssen erkannt und extrahiert werden, eine

gemeinsam verwaltete und verwendete Terminologie mittels Glossar ist unverzichtbar.

## 4 Welche Schritte sind nötig um zu einem guten Ergebnis zu kommen?

### 1 Alle betroffenen Personen einbeziehen

Es ist vor allem wichtig, Betroffene, Stakeholder, Dokumente und schon vorhandene Systeme auszuforschen und in den Prozess miteinzubeziehen, um Widerstände zu vermeiden. Feedback muss gesammelt, bewertet und berücksichtigt werden. Eine allen zugängliche Stakeholder-Tabelle, in der die einzelnen Rollen genau definiert sind, ist dabei hilfreich (siehe Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 399). Stakeholder sind die zukünftigen User, Qualitätsmanager, Projektverantwortliche, Sicherheitsverantwortliche, andere Projekte und Personen aus Support oder Marketing. Dabei sollen auch benachbarte Bereiche berücksichtigt werden. Der Informationsfluss muss durch regelmäßige Besprechungen in Gang gehalten werden. Anhand vorhandener Dokumente werden Vorerfahrungen genutzt, um keine Fehler zu reproduzieren oder Arbeiten doppelt zu erledigen.

### 2 Die Grenzen des Systems definieren

Vor Arbeitsbeginn muss klargestellt werden, wo die Grenzen des Systems liegen und damit auch die Anforderungen enden. Damit wird der eigene Aufgabenbereich abgesteckt und eventuelle Lücken sichtbar gemacht. Zugleich wird auch der Systemkontext definiert, das ist jener »Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen des betrachteten Systems relevant ist.« (Pohl und Rupp 2015, S. 13). Dazu zählen Hardwaresysteme, Stakeholder, Organisationen, Prozesse, Ereignisse sowie Dokumente, wie zum Beispiel Standards oder Gesetze, die mit dem zu analysierenden System in irgendeiner Form in Verbindung stehen und eine Beziehung dazu haben.

### 3 Ist-Situationen erheben, bewerten und Ziele ableiten

Um neue Anforderungen umsetzen zu können ist es zunächst einmal wichtig, den Ist-Zustand genau zu analysieren und beispielsweise zu prüfen, welche

Funktionalitäten fehlen bzw. wie zu entwickelnde Funktionen in das Gesamtkonzept passen.

### 4 Ziele benennen und quantifizierbare Angaben daraus ableiten

Ziele müssen möglichst genau und eindeutig und ohne Interpretationsspielraum beschrieben werden. Daraus werden kleinere und überschaubare Ziele abgeleitet, die überprüft werden können. Diese müssen klar definiert, nachvollziehbar, mit einem Stakeholder verknüpfbar und gut kommunizierbar sein.

### 5 Anforderungen ermitteln

Für die Ermittlung von Anforderungen gibt es keine universelle Technik. Sind die Anforderungen wenig detailliert, kann man mit Befragungs- und Beobachtungstechniken arbeiten. Bei sehr detaillierten Anforderungen bietet es sich eher an, Dokumentationen zu nutzen bzw. sich bereits existierende Systeme anzusehen. Die Kombination mehrerer Techniken vermindert die Risiken und gleicht die Schwächen der einzelnen Techniken aus.

Bei den Beobachtungstechniken unterscheidet man die Feldbeobachtung, bei der die Stakeholder eher passiv bleiben und die Teilnahme an der Arbeit der Stakeholder, bei der diese den Beobachtenden auch Informationen geben. Bei artefaktbasierten Techniken bzw. dokumentenzentrierten Techniken verwendet man Dokumente wieder, die Informationen über abzulösende Systeme bieten. So werden keine Funktionalitäten vergessen. Mittels der so genannten Systemarchäologie lassen sich Funktionen eines alten Systems, die nicht mehr bewusst sind, wieder erkennbar machen, indem der Code analysiert und geprüft wird, welche Funktionalitäten das System aufwies. Diese Verfahrenstechnik ist aufwändig, führt aber zu detaillierten Anforderungen und garantiert als einzige Technik, dass sämtliche Funktionalitäten eines alten Systems in ein neues System übernommen werden können. Bei den szenariobasierten Techniken werden Storyboards oder Simulation unterschieden bzw. Use Cases erstellt

oder Geschäftsprozessmodellierung betrieben. Storyboards bzw. Simulationen eignen sich vor allem für die Prüfung von Akzeptanz bzw. für das Auffinden noch unbekannter Aspekte. Die darzustellenden Abläufe sind bekannt und werden eventuell mit Bildschirmmasken oder Skizzen vorgestellt. Ein großer Vorteil dabei ist die allgemeine Verständlichkeit, ein Nachteil ist der große Aufwand.

## 6 Anforderungen priorisieren

Die Frage nach Prioritäten kann sehr heikel in einem Projekt sein, da unterschiedliche Stakeholder dies oft unterschiedlich einschätzen. Beim Priorisieren werden zunächst die Ziele und die Randbedingungen der Priorisierung festgehalten, z.B. wann welche Stakeholdergruppe verfügbar ist oder wann auf welche Ressourcen zugegriffen werden kann. Vom Ziel der Priorisierung ausgehend, werden die Priorisierungskriterien festgelegt. Dazu zählen u.a. die Kosten für die Umsetzung, Risiken, Schaden bei nicht erfolgter Umsetzung, die Wichtigkeit und die Zeitdauer für die Umsetzung (Pohl und Rupp 2015, S. 126). Danach wird eine geeignete Priorisierungstechnik gewählt bzw. eine Kombination aus unterschiedlich aufwändigen Techniken. Bei den meisten Projekten werden Ad-hoc-Priorisierungstechniken angewendet.

## 7 Anforderungen mit Attributen versehen

Oft sind in dieser Phase die Anforderungen allgemein und unspezifisch. Deshalb ist es unerlässlich, sie mit weiteren Attributen zu versehen. Die Bedeutung der Attribute muss klar sein, außerdem sollte festgelegt werden, welche Schritte bei der Änderung eines Attributs zu unternehmen sind. Während des gesamten Lebenszyklus eines Systems müssen Informationen über Anforderungen festgehalten werden. Neben der permanenten ID ist auch ein eindeutiger Name für die Anforderung, Autor und Quelle sowie eine für die Anforderung verantwortliche Person unerlässlich. Ein Beispiel für ein Attribut ist der Status. Hier wird festgehalten, ob die Anforderung bereits ausgearbeitet oder getestet ist, ob sie von den Stakeholdern akzeptiert wurde oder sogar schon freigegeben ist.

## 8 Qualitätssicherung

Qualitätssicherung als Teilbereich des RE betrifft die Qualität von Anforderungen und die Anforderungs-

dokumente. Es muss so früh wie möglich festgelegt werden welche Qualitätsziele erreicht werden müssen. Sobald Anforderungen stabil sind, sollten sie getestet werden. Pohl und Rupp (2015, S. 97) setzten drei Hauptziele bei der Überprüfung von Anforderungen fest:

- Inhalt: Wurden alle relevanten Anforderungen ermittelt und im erforderlichen Detaillierungsgrad erfasst?
- Dokumentation: wurden die Anforderungen gemäß der festgelegten Dokumentations- und Spezifikationsvorschriften dokumentiert?
- Abgestimmtheit: Stimmen alle Stakeholder mit den dokumentierten Anforderungen überein und sind alle bekannten Konflikte aufgelöst?

Bei der Qualitätssicherung des Anforderungsmanagements wird die Sorgfalt überprüft. Zu den Techniken zur Überprüfung der Anforderungen zählen beispielsweise Reviews, Stellungnahmen, Inspektion, Prototypen, Testfälle, perspektivenbasiertes Lesen und Checklisten.

Sinnvoll umsetzbare Anforderungen sollten dabei folgenden Kriterien genügen: Sie sind

- notwendig
- korrekt und mit den Stakeholdern abgestimmt
- realisierbar
- verständlich
- unzweideutig
- vollständig und messbar
- konsistent
- bewertet nach Wichtigkeit und/oder Stabilität
- prüfbar
- modifizierbar
- verfolgbar (siehe auch Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 26–28; Pohl und Rupp 2015, S. 98).

## 9 Change-Management

Change-Management bezeichnet den Prozess, der für die Steuerung des Lebenszyklus aller Änderungen verantwortlich ist. Anforderungen ändern sich im Verlauf eines Projekts, sie werden gestrichen, verändert oder es kommen neue hinzu. Änderungen sind positiv zu sehen, denn sie hängen unter anderem auch mit dem Wissenszuwachs der Stakeholder zusammen. Wichtig ist, dass die unterschiedlichen Versionen der Anforderungen über den gesamten Zeitraum lückenlos unter Versionsnummern doku-

mentiert werden und mit Titel, Datum, den Namen der AntragstellerInnen, einer Beschreibung, einer Begründung und einer Priorisierung versehen sind (siehe Pohl und Rupp 2015, S. 143). Change Management muss den gesamten Prozess begleiten. Ein Änderungsantrag (Change Request) kann den Source Code, die Anforderungen, die Architektur oder das Design betreffen. Wichtig ist immer, die Auswirkung auf das Gesamtprojekt zu prüfen. Wenn Änderungen akzeptiert werden, müssen erneut die Stakeholder miteinbezogen werden und die Änderung muss freigegeben werden.

## 10 Dokumentation

Die Dokumentation ist zeitaufwändig und oft mit keiner hohen Priorisierung versehen, da die Projektbeteiligten glauben, ohnehin alles im Kopf zu haben. In Wirklichkeit gehen jedoch viele Inhalte verloren, wenn sie nicht festgehalten sind. Vor allem Wissen über bereits erledigte Aufgaben wird rasch vergessen (siehe Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 20). Die Dokumentation hilft den Gesamtzusammenhang zu überblicken und unterschiedliche Interpretationen und Missverständnisse zu vermeiden. Alle Beteiligten müssen die gleiche Version der Dokumentation erhalten. Dokumentiert werden kann in natürlicher

Sprache, mittels Szenarien und unterschiedlicher Diagramme.

## 11 Release der Anforderungen

Für das Release eines Systems müssen sämtliche Informationen zu einem bestimmten Zeitpunkt »eingefroren« werden, man erhält eine Konfiguration. Erst danach kann man mit der Testphase beginnen. Wenn eine Konfiguration alle Anforderungen für ein Release enthält, spricht man von einer »Basislinie« oder »Baseline« (Pohl und Rupp 2015, S. 455). Mit dieser Basislinie und einem Änderungsprotokoll beginnen die nächsten Prozessschritte. Mit dem Release sollte man auch sogenannte »Release Notes«, eine Auflistung aller in einem Release vorgenommenen Änderungen, mitliefern.

## 12 Re-Use der Anforderungen

Aus ökonomischen und Qualitätsgründen ist es sinnvoll, Anforderungen nach einer rechtlichen Überprüfung auch für andere Fälle wiederzuverwenden. Gut dokumentierte und generalisierte Anforderungen werden gesammelt, beschrieben und nach Themen sortiert in einer Datenbank abgelegt. Vor allem nicht-funktionale Anforderungen, Modelle oder Glossareinträge bieten eine gute Basis.

# 5 Methoden und Tools des Requirement-Engineerings

RE ist, wie oben beschrieben, eng verbunden mit der Idee der agilen Softwareentwicklung. Die agile Methode (siehe Beck et al. 2001) gibt folgende Rollen vor:

- Product-Owner, der das Produktmanagement und das RE übernimmt, sowie die Kosten/Nutzen-Relation kontrolliert
- Scrum-Master, der für das prozessorientierte Projektmanagement und für die Organisationsentwicklung zuständig ist
- Das Entwicklungsteam, das für alle operativen Aufgabenbereiche zuständig ist (siehe auch Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, 505 f.).

Anforderungen werden als User-Stories dargestellt. Darunter versteht man kurzgehaltene und in der Alltagssprache formulierte Software-Anforderungen. Sie beschreiben relevante Funktionalitäten und be-

stehen aus einer schriftlichen Beschreibung, Gespräche über die Funktionalität und Akzeptanzkriterien, die festlegen, wann eine User-Story umgesetzt ist. User-Stories können mittels Aktivitäts- oder Zustandsdiagrammen dokumentiert werden. Mehrere User-Stories ergeben einen so genannten Use-Case. Um die weitere Planung, eine Sprint-Planung, zu ermöglichen, müssen die User-Stories weiter verfeinert werden. Innerhalb dieser User-Stories dürfen sich keine weiteren User Stories und keine Abhängigkeiten befinden. Die User-Stories werden präzisiert und strukturiert und zu Sprint-Backlogs zusammengefasst (siehe Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, 259 ff.).

Ein wichtiges Tool im RE ist ein mit allen Stakeholdern abgestimmtes und für alle frei zugängliches Glossar, das eine für alle am Projekt Beteiligten verbindliche Definitionen liefert, sowie auch Abkürzungen auflöst. Dabei kann man auch auf bestehende

Glossare zurückgreifen, wie beispielsweise auf das »IEEE Standard Glossary of Software Engineering Terminology« (IEEE 1990).

Um die Übersicht nicht zu verlieren, empfiehlt sich ein so genannter RE-Leitfaden, den man auch als RE-Handbuch, RE-Konzept oder RE-Richtlinie bezeichnen kann. »Der RE-Leitfaden ist das zentrale Artefakt, in dem alle Prozesse, Methoden, Artefakte, Tools, Rollen und Vorgehensweisen dokumentiert sind, die für die Durchführung der Anforderungsanalyse gebraucht werden.« (Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 370).

In den verschiedenen Phasen des RE werden unterschiedliche Diagramme verwendet, für die Dokumentation des Systemkontexts oft Use-Case-Diagramme oder Datenflussdiagramme. Sie zeigen den Datenfluss durch ein System und können allgemeine Bearbeitungsschritte, wie das Verändern von Daten, darstellen. Um den Systemkontext darzustellen, werden oft UML-Klassendiagramme herangezogen. Um geschäftliche Abläufe abzubilden, bieten sich Business-Use-Case-Diagramme an. Man erhält einen leicht verständlichen Überblick über die unterschiedlichen Systeme und deren Prozesse. Kausale Zusammenhänge zwischen den einzelnen Schritten eines Verarbeitungsprozesses können durch Ablaufdiagramme, auch Flussdiagramme genannt, visualisiert werden.

Um die Usability eines Systems zu testen kreierte man auch fiktive Personen, die Personas genannt werden (siehe Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 525). Man erstellt einen Persona-Steckbrief für eine fiktive Person mit einem erfundenen Namen, der die Ziele, Einstellungen und Motivationen vermerkt. Alter, Familienstand, Verdienst, Wohnort und Herkunft werden ebenfalls angegeben, zusätzlich fiktive Aktivitäten und persönliche Einstellungen. Aus dem Persona-Steckbrief werden Szenarien entwickelt, um die funktionalen und nichtfunktionalen Anforderungen verfeinern zu können.

Für abstrakte Anforderungen bieten sich Kreativitätstechniken an, die gemeinsam mit den Stakeholdern durchgeführt werden und bei denen wichtige Eigenschaften des Systems gesammelt werden können. In der Literatur werden mehrere Kreativitätstechniken (siehe z.B. Rupp und SOPHIST-Gesellschaft für Innovatives Software-Engineering 2014, S. 101) unterschieden:

- Brainstorming
- Brainstorming paradox
- Perspektivenwechsel
- Analogietechniken

Sind die Anforderungen weniger detailliert, könnte man mit Befragungs- und Beobachtungstechniken arbeiten. Bei sehr detaillierten Anforderungen bietet es sich eher an, Dokumentationen zu nutzen bzw. sich bereits existierende Systeme anzusehen. Beobachtungstechniken bieten sich an, wenn die Stakeholder aus Zeitmangel oder sonstigen Gründen ihr Wissen nicht an die RE weitergeben können.

Bei den Beobachtungstechniken unterscheidet man die Feldbeobachtung und die Teilnahme an der Arbeit der Stakeholder. Bei der Feldbeobachtung wird das System und die Umgebung beobachtet, wobei darauf zu achten ist, dass die Beobachtung das Verhalten des Systems nicht verändert. Dabei müssen im Vorfeld Fragen formuliert werden, es muss genau beschrieben werden, was beobachtet werden soll, der Zeitraum und der Ort müssen definiert sein bzw. die beteiligten Personen ausgewählt werden. Die Anforderungen müssen auch stets kritisch hinterfragt werden, da sonst eventuell überholte Anforderungen und suboptimale Prozesse festgehalten werden.

Bei dokumentenzentrierten Techniken verwendet man Dokumente wieder, die beispielsweise Informationen über abzulösende Systeme bieten können. So kann sichergestellt werden, dass keine Funktionalitäten vergessen werden. Diese Art der Technik sollte jedoch immer mit anderen Techniken kombiniert werden, um keine neueren Anforderungen zu übersehen.

Wenn Anforderungen priorisiert werden sollen, stehen wiederum zahlreiche Techniken zur Auswahl (siehe Pohl und Rupp 2015, 127 ff.):

- Ranking: dabei wird von den Stakeholdern im Hinblick auf ein bestimmtes Kriterium eine Rangfolge der zu priorisierenden Anforderungen festgelegt
- Top-Ten-Technik: Für eine vorher festgelegte Anzahl werden im Hinblick auf das betrachtete Kriterium die wichtigsten Anforderungen ausgewählt. Für diese wird danach eine Rangfolge festgelegt
- Ein-Kriteriums-Klassifikation: Dabei steht die Wichtigkeit der Realisierung dieser Anforderungen für den Erfolg des Systems im Vordergrund. Jede Anforderung wird einer der drei Prioritätsklassen zugeordnet: Mandatory (unbedingt zu realisieren),

optional (müssen nicht zwingend umgesetzt werden) und nice to have (diese gefährden den Erfolg eines Projekts nicht)

- Kano-Klassifikation: Hier werden die Anforderungen nach deren jeweiligen Marktwirkung klassifiziert und priorisiert. Die Anforderungen werden in drei Merkmalsklassen klassifiziert:
  - Basismerkmale (das System muss die Anforderung aufweisen, um in den Markt eintreten zu können),
  - Leistungsmerkmal (die Kunden fordern dieses Merkmal bewusst, Leistungsmerkmale steigern die Kundenzufriedenheit)
  - Begeisterungsmerkmal (den Stakeholdern war dieses Merkmal gar nicht bewusst und haben die Umsetzung gar nicht erwartet. Damit lässt sich die Kundenzufriedenheit überproportional steigern)
- Wiegers'sche Priorisierungsmatrix: Diese ist zwar sehr objektiv, erfordert jedoch einen sehr hohen Aufwand, da verschiedene Berechnungen stattfinden müssen um Nutzen, Nachteile, Kosten und Risiko einschätzen und miteinander in Verbindung bringen zu können.

Für die Dokumentation von Anforderungen stehen mehrere Standards zur Verfügung:

- RUP (Rational Unified Process), ein kommerzielles Produkt von IBM, das erstmals 1998 vorgestellt wurde, wird meist für Softwaresysteme verwendet. Die AuftraggeberInnen leiten die Anforderungen aus dem zuvor erstellten Business-Modell ab (vgl. Kuhrmann 2008–2018a).
- Der IEEE Standard 830-1998 (IEEE 1998) ist ein vom Institute of Electrical and Electronic Engineers entwickelter Standard zur Spezifikation von Software, der mehrmals überarbeitet wurde und sowohl das Pflichtenheft als auch das Lastenheft umfasst und Qualitätskriterien erarbeitet hat.
- Der ISO/IEC/IEEE-Standard 29148:2011 (IEEE 2011) gibt eine in fünf Blöcken unterteilte Gliederung für Softwareanforderungen vor und unterscheidet (siehe auch Pohl und Rupp 2015, 40 f.):
  - Ein Kapitel mit einführenden Informationen, z.B. die Systemabgrenzung oder den Systemzweck betreffend und eine allgemeine Beschreibung der Software
  - Ein Kapitel mit einer Auflistung aller referenzierenden Dokumente

- Ein Kapitel für spezifische Anforderungen, betreffend die Schnittstellen, die Performanz usw.
- Ein Kapitel mit sämtlichen geplanten Verifikationsmaßnahmen sowie
- Anhänge, beispielsweise bereits identifizierte Abhängigkeiten.

- Das V-Modell des Bundesministeriums des Inneren (BMI) ist ein Vorgehensmodell für die Durchführung von IT-Projekten, das erstmals 1992 von der Bundeswehr veröffentlicht wurde und sich an alle Projektbeteiligten richtet, gibt ebenfalls Strukturen vor. Es setzt sich aus modular aufeinander aufbauenden Vorgehensbausteinen zusammen, die projektspezifisch angepasst werden können (vgl. Kuhrmann 2008–2018b).

Anforderungen können in natürlicher Sprache dokumentiert werden, aber auch mittels Szenarien, die mittels natürlicher Sprache oder als Diagramme notiert werden. Meist erfolgt die Beschreibung in natürlicher Sprache, entweder als Freitext oder als Tabelle bzw. mittels Formular. Formulare haben den Vorteil, dass damit auch jene Informationen abgefragt werden können, auf die sonst leicht vergessen werden könnte. Die Dokumentation von Anforderungen als Aktivitätsdiagramme eignet sich vor allem zur Darstellung von komplexen Abläufen. Für jeden zuvor erstellten Use-Case sollte ein Aktivitätsdiagramm erstellt werden, eventuell ist es sinnvoll, die Aktivitäten in Unteraktivitäten zu zerlegen und sämtliche Ausnahmen, Ablaufvarianten und Beschreibungen hinzuzufügen. Aktivitätsdiagramme lassen sich zu mehreren Zeitpunkten in Projekten einbinden. Sequenzdiagramme, als eine Art grafische Tabellen konzipiert, lassen sich überall dort gut einsetzen, wo zeitliche Reihenfolgen, Ablaufbedingungen, Schleifen, Alternativen und Nebenläufigkeiten festgehalten werden sollen (siehe auch Neumann 1998, 200 ff.). So kann die Kommunikation zwischen den einzelnen PartnerInnen, zum Beispiel NutzerInnen und Systemen, gut präsentiert werden, aber auch die zeitliche Abfolge und die unterschiedlichen Verantwortlichkeiten lassen sich festlegen. Mittels Zustandsdiagramm kann das Verhalten eines Systems bzw. einzelner Komponenten dargestellt werden, wie etwa der Lauf des Buches durch die Bibliothek. Hat man dies einmal sichtbar gemacht, lassen sich feste Regeln aufstellen und Inkonsistenzen erkennen. Das Zustandsdiagramm kann sehr detailliert sein, die einzelnen Use-Cases können im Zusam-

menspiel miteinander betrachtet werden. Das Klassendiagramm der UML (Unified Modeling Language) ermöglicht es, Begriffe und die jeweiligen Beziehungen zueinander zu beschreiben. Die einzelnen Begriffe werden dabei als Klassen dargestellt, so können alle möglichen Aktivitäten zwischen NutzerInnen und System dargestellt werden.

Das Prüfen der Anforderungen kann ebenfalls wieder auf unterschiedliche Arten erfolgen, zum Beispiel durch Reviews. Hier lassen sich drei Ausprägungsformen unterscheiden: Stellungnahme, Inspektion und Walkthrough. Bei der Stellungnahme wird der Prüfgegenstand von einer fachlich versierten Person geprüft. Das erfordert relativ wenig Aufwand, das Ergebnis ist jedoch sehr stark vom Qualitätsbewusstsein der einzelnen Personen abhängig. Bei der Inspektion von Software, auch technische Reviews genannt, werden Entwicklungsartefakte anhand eines Prozessschemas systematisch nach Fehlern durchsucht. Die Inspektion kann in die Phasen Planung, Übersicht, Fehlersuche, Fehlersammlung unterteilt werden. In einem weiteren Schritt sind dann noch Korrektur, Nachkontrolle und Reflexion nötig. Mit einem Walkthrough wird geprüft, inwieweit die Anforderungen akzeptiert werden. Es ist weniger strikt angelegt als eine Inspektion, auch die Rollen sind nicht so vielfältig und nicht so strikt voneinander getrennt. Wichtig sind ReviewerInnen, AutorInnen und ProtokollantInnen. Ziel dieser Methode ist es, gemeinsam Qualitätsmängel in den Anforderungen zu entdecken und für alle ein akzeptierbares Verständnis für die perfekte Anforderung zu erzielen. Gemeinsam mit der Stellungnahme und dem Walkthrough wird auch oft das perspektivenbasierte Lesen verwendet. Hier nimmt man unter-

schiedliche Rollen ein um eine ganz spezielle Sicht auf eine Anforderung zu erhalten und sich auf bestimmte Aspekte gut konzentrieren zu können.

Beim Prototyp werden die Anforderungen bereits zum Teil umgesetzt um die Realisierbarkeit zu überprüfen. Fehlende bzw. ungeeignete Funktionalitäten können so rasch aufgezeigt werden. Prototypen können in allen Phasen der Analyse verwendet werden. Für die Stakeholder sind die Ergebnisse direkt erlebbar, auch bei recht komplexen Systemen.

Die Qualität von Anforderungen und auch des Requirements-Engineering-Prozesses kann durch die Verwendung von Metriken gemessen werden. Daran lässt sich ein Indikator für die Produkt- bzw. Prozessqualität ablesen. Beispielsweise lässt sich mit Prozessmetriken erkennen, wie viele schon abgestimmte Anforderungen in einem bestimmten Zeitrahmen geändert wurden.

Für das Anforderungsmanagement können unterschiedliche Werkzeuge eingesetzt werden.

Neben Standard-Bürowerkzeugen wie Word für die Textverarbeitung werden MS-Excel sowie Access oder Oracle-Datenbanken genutzt. Spezielle Anforderungsmanagementwerkzeuge sind meist proprietär. Als Modellierungswerkzeug wird unter anderem Microsoft Visio eingesetzt, als Fehlermanagementwerkzeug unter anderem Bugzilla23, ein freies, webbasiertes Werkzeug zur Verwaltung von Fehlermeldungen. Als Kollaborationswerkzeuge eignen sich zum Beispiel Wikis.

Bei der Wahl der Werkzeuge empfiehlt es sich für wissenschaftliche Bibliotheken, eng mit den KollegInnen aus dem IT-Bereich zu kooperieren, damit die spätere Zusammenarbeit reibungslos funktioniert.

## 6 Warum sollte RE in Bibliotheken eingesetzt werden?

Bei den meisten Projekten treffen BibliotheksexpertInnen auf IT-ExpertInnen, um für WissenschaftlerInnen technische Lösungen für Recherche, Archivierung von Forschungsdaten usw. anbieten zu können. Diese Personengruppen besitzen in der Regel unterschiedliches Wissen, haben verschiedene Sichten auf die diversen Aufgaben und verwenden andere Tools. Aus diesem Grund kann es zu Missverständnissen in der Kommunikation kommen, zu Umwegen

und zu Verzögerungen von Projekten, die nicht nur Geld kosten, sondern auch das Arbeitsklima auf beiden Seiten belasten. Für die Behebung eines Fehlers, der erst während des Programmierens offensichtlich wird, muss ein um ca. den Faktor 20 höherer Aufwand berechnet werden, als wäre dieser Fehler bereits in der Phase des RE aufgefallen (siehe Pohl und Rupp 2015, 1 f.).

## 7 Anforderungen an einen Requirements-Engineer in Bibliotheken

Aus der Beschäftigung mit RE ergibt sich, dass eine speziell dafür eingesetzte Person bestimmte Eigenschaften mitbringen muss. Sie muss bibliothekarisches Wissen besitzen und sich im IT-Bereich so weit auskennen, dass sie eventuelle Probleme nachvollziehen kann und in der Lage ist, mit den unterschiedlichen Stakeholdern entsprechend zu kommunizieren. Auch komplizierte Zusammenhänge müssen schnell erkannt und analysiert werden können. Kommunikative Fähigkeiten sind Vorausset-

zung, jede Aussage muss auf Vollständigkeit und Eindeutigkeit überprüft werden. Gleichzeitig ist darauf zu achten, dass alle Anforderungen auch ausgesprochen und dokumentiert werden. Sie sollte fähig sein, das Arbeitsklima positiv zu gestalten, Konflikte müssen rasch erkannt und – eventuell mit Hilfe von ExpertInnen - aufgelöst werden. Kritik darf nicht persönlich genommen werden. Die Arbeit im Team ist unerlässlich.

## 8 Zur Umfrage

Die explorative Umfrage, die sich an MitarbeiterInnen in Schlüsselpositionen bzw. bei kleineren Einrichtungen auch an die LeiterInnen direkt richtete, war zum Teil als quantitativer, als auch als qualitativer Fragebogen konzipiert. Soweit möglich wurde die Befragung persönlich durchgeführt, um gegebenenfalls nachfragen zu können. Dadurch wurden zusätzliche wertvolle Hinweise gegeben und es konnte das gewonnene Stimmungsbild weiter verdeutlicht werden. Einige Personen wurden aufgrund der räumlichen Distanz telefonisch befragt, in wenigen Ausnahmefällen wurde der Fragebogen von den Befragten selbständig ausgefüllt, aber auch in diesen Fällen wurde telefonisch Kontakt aufgenommen. Alle Befragten erhielten die Fragen vorab per Mail zugesandt um sich mit dem Thema vertraut machen zu können. Der verwendete Fragebogen befindet sich im Anhang.

Die ersten 11 Fragen beschäftigten sich mit der Art und der subjektiv empfundenen Qualität der Zusammenarbeit zwischen IT und Bibliothek und mit dem Umgang mit Projekten im Allgemeinen, die Fragen 12 bis 27 zielten speziell auf den Umgang mit Anforderungen in der eigenen Institution und der subjektiven Zufriedenheit damit ab, die Fragen 28 bis 35 waren auf den derzeitigen Umgang mit Projektmanagement bzw. RE abgestimmt.

Sämtliche angefragte Personen aus 13 Institutionen waren erfreulicherweise bereit, die Fragen zu beantworten, wenn auch nicht immer alle detailliert Auskunft geben konnten. Da die Umfrage sich vor allem auf den Bibliotheksbereich bezog, schien es angemessen, dass sechs Personen, also knapp

ein Drittel der insgesamt 19 Befragten aus dem IT-Bereich – hier wurden nur IT-Abteilungen, keine Firmen berücksichtigt – stammen.

### 8.1 Ergebnisse

Nur einem Drittel aller Befragten waren die Begriffe RE und RM völlig unbekannt, alle anderen hatten zumindest schon davon gehört.

In allen Fällen werden Projekte sowohl mit der IT-Abteilung als auch mit Firmen durchgeführt.

In den meisten Fällen reicht diese Zusammenarbeit schon 16 bis 20 Jahre zurück, bei vielen sogar 26 bis 30 Jahre und bei manchen sogar mehr als 30 Jahre. In den meisten Fällen wird täglich bzw. fast täglich oder zumindest wöchentlich zusammengearbeitet. Die Bandbreite der Antworten auf die Frage »Welche Abteilungen der Bibliothek arbeiten eng mit der IT-Abteilung/mit der Firma zusammen?« zeigt, wie viele unterschiedliche Aufgaben, bei denen die IT eine wesentliche Rolle spielt, wissenschaftliche Bibliotheken heute wahrnehmen. Genannt wurden Digitalisierung, e-Ressourcen, Forschungsdokumentation, Repositorien, Open Access-Abteilungen, Teaching Library und weitere Handlungsfelder.

Der nächste Fragenblock befasste sich konkreter mit dem Umgang mit Anforderungen im technischen Bereich. In fast allen Fällen werden regelmäßige Treffen mit der IT-Abteilung, wo über Anforderungen gesprochen wird, organisiert. Die Zusammensetzung des Projektmanagements ändert sich – abhängig von der Größe der jeweiligen Institu-

tion – in den meisten Fällen je nach Projekt. Bei der Hälfte der befragten Institutionen gibt es eine klare Rollenverteilung, wobei der Umgang mit Rollen von Abteilung zu Abteilung stark unterschiedlich ist.

Wie im theoretischen Teil der Arbeit gezeigt wurde, nimmt professionelles RE viele Zeitressourcen in Anspruch. Die Frage, ob für das Management von Anforderungen im Vorfeld Zeit eingeplant wird, verneinte die Hälfte aller Befragten. Nur für mehr als knapp ein Drittel der InterviewpartnerInnen wird genügend Zeit eingeplant, das ist nicht ausreichend für ein professionelles RE. Günstiger fielen die Antworten bei der Frage aus, ob die Zuständigkeiten im Projekt schon zu Beginn festgelegt werden würden. Hier antworteten mehr als drei Viertel mit Ja. Auch eine vertiefende Frage nach der Transparenz der Zuständigkeiten wurde ähnlich beantwortet. Bei der Frage, ob der Prozess rund um die Anforderungen dokumentiert wird, antwortete die Mehrheit der Befragten mit Ja, ca. ein Drittel dokumentierte wenig oder gar nicht, was auf einen deutlichen Mangel in der Projektplanung hinweist. Wenn dokumentiert wird, wird diese Aufgabe je nach Institution von unterschiedlichen Personen wahrgenommen. In mehr als einem Drittel der Fälle dokumentieren Bibliothek und IT-Abteilung/IT-Firma gemeinsam. Die Frage, wie dokumentiert wird, ergab, dass in den meisten Fällen eine Word-Datei erstellt wird, an zweiter Stelle wurde ein Wiki genannt bzw. eine Excel-Datei. Nur sehr wenige der Befragten setzten eine spezielle Software ein. Etwas mehr als die Hälfte der befragten Personen gab an, dass diese Dokumentation auch inhaltlich überprüft wird, ein Drittel verneinte dies. Wenn Dokumentationen geprüft werden, wird dies in den meisten Fällen gemeinsam von allen Betroffenen durchgeführt. Ob die Dokumentationen regelmäßig aktualisiert werden, war rund der Hälfte unbekannt. Weniger als ein Drittel der InterviewpartnerInnen bejahten die Frage, alle anderen gaben an, dass in Projekten unterschiedlich mit der Aktualisierung von Dokumentation umgegangen wird.

Auf die offen gestellte Frage, wer die Anforderungen erstellt, antworteten die meisten, dass diese gemeinsam von der Bibliothek und der IT-Abteilung bzw. der IT-Firma erstellt werden und zwar oft in einem ersten Schritt von der Bibliothek alleine und erst in einem zweiten Schritt gemeinsam mit den TechnikerInnen. Für das Eruiere von Anforderungen gab es in etwas mehr als der Hälfte der Fälle noch keine Workflows. Einige wiesen im Gespräch

darauf hin, dass Anforderungen eher unkoordiniert verfasst werden. Nur knapp ein Drittel der Befragten gab an, dass für das Eruiere der Anforderungen bestimmte Methoden verwendet werden würden. Mehr als drei Viertel der Interviewten bestätigten, dass die Anforderungen untereinander abgestimmt werden, woraus man schließen kann, dass ansonsten eventuell widersprüchliche Anforderungen weitergegeben werden. Wenn es eine Abstimmung gibt, erfolgt diese meistens gemeinsam mit der IT-Abteilung bzw. der IT-Firma. Die Dokumentation der Anforderungen selbst erfolgte meist in einem Textdokument.

Zusammenfassend zu diesem Frageblock lässt sich sagen, dass sich der Umgang mit Anforderungen im technischen Bereich recht unterschiedlich gestaltet, in einigen Fällen schon weitgehend professionell abläuft, aber bei den meisten Institutionen noch stark verbessert werden könnte.

Nachdem bereits zu Beginn der Befragung geprüft wurde, ob der Begriff RE überhaupt bekannt ist und nur ein Drittel angab, nichts darüber zu wissen, wurde im nächsten Abschnitt dieses Thema vertieft. Die Frage, ob Standards des RE bekannt seien, wurde nur von weniger als einem Viertel aller Befragten bejaht, den meisten sind Standards unbekannt. Von jenen, die den Begriff kannten, gaben fast alle an, dass diese in ihrer jeweiligen Institution nicht eingesetzt werden würden. Das bestätigt die eingangs formulierte Vermutung, dass RE aktuell in wissenschaftlichen Bibliotheken noch nicht in ausreichender Weise berücksichtigt wird.

Mehr als die Hälfte der InterviewpartnerInnen verneinte die Frage, ob die Anforderungen qualitätsgeprüft werden. Wenn eine Qualitätsprüfung stattfindet, wird sie meist im Team vorgenommen, wobei am häufigsten auf Ressourcenplanung und Transparenz geachtet wird. Eine Dokumentation der Qualitätsprüfung erfolgte nur in der Hälfte der Fälle.

Ein Viertel der Befragten gab an, dass es keinen Feedbackprozess hinsichtlich der Anforderungen gibt. Wenn Anforderungen als unzureichend angesehen werden, startet in den meisten Institutionen der Prozess erneut, wenige nannten explizit ein bereits existierendes Changemanagement. In einigen Fällen war den ProjektpartnerInnen der Umgang mit unzureichenden Anforderungen nicht bekannt, was auf mangelnde Transparenz schließen lässt bzw. war das Problem noch nie aufgetreten. Mehrheitlich haben alle am Projekt beteiligten Personen Zugriff

auf die erstellten Anforderungen. Von einigen wurde auch angegeben, dass selbst Personen, die nichts mit dem Projekt zu tun haben, die Anforderungen einsehen können, was eine sinnvolle Offenheit beweist.

Auf die Frage, wie reagiert wird, wenn im Nachhinein neue Anforderungen gestellt bzw. die Anforderungen geändert werden, antworteten fast alle, dass dies ein alltäglicher Fall sei. An dieser Stelle verwiesen einige auf die Vorteile des agilen Arbeitens. Diese offene Frage führte auch dazu, dass auf derzeitige Mängel in der eigenen Institution hingewiesen wurde, auf unkoordiniertes Vorgehen bzw. auf fehlendes Changemanagement.

Sämtliche Befragten gaben an, dass die Umsetzung der Anforderungen überprüft werden würde, in den meisten Fällen von zuständigen BibliotheksmitarbeiterInnen, manchmal auch von der Bibliothek gemeinsam mit der IT-Abteilung bzw. IT-Firma. In mehr als der Hälfte der Fälle wurde dieser Umsetzungsprozess dokumentiert, in knapp einem Viertel nur zum Teil.

Auch die Antworten auf diesen Fragenblock zeigen einerseits die Unterschiede in den einzelnen Institutionen, so wird zum Teil sehr professionell mit Anforderungen umgegangen, während es an anderen Einrichtungen noch an Wissen und an der Umsetzung mangelt. Andererseits sieht man auch, dass den meisten Befragten diese Mängel durchaus bewusst sind. Die Einführung von RE würde hier sicherlich auf fruchtbaren Boden fallen.

Bei den letzten Fragen war vor allem der subjektive Eindruck der Befragten zum Thema »Umgang mit Anforderungen« gefragt. Die Hälfte gab an, mit der derzeitigen Situation, wie an ihrer Institution mit Anforderungen umgegangen wird, zufrieden zu sein. Jene, die nicht oder nur teilweise zufrieden waren, gaben meist auch den Grund dafür an. So wurde beispielsweise eine mangelnde Kultur in Sachen Projektmanagement kritisiert, eine engere Zusammenarbeit, eine klare Rollenzuweisung und ein strukturierteres Arbeiten wurden gefordert. Von den zufriedenen TeilnehmerInnen waren einige der Mei-

nung, dass sich der Umgang mit Projekten in den letzten Jahren wesentlich verbessert hat. Kein eindeutiges Ergebnis ergab die Frage, ob für die Umsetzung von Projekten genügend Personal und Zeit zur Verfügung steht. Von denen die nicht oder nicht ganz zufrieden waren, meinte die Hälfte, dass zu wenig Personal vorhanden wäre, die andere Hälfte merkte an, dass es an Zeit mangle. Beides stört den Prozess des Anforderungsmanagements. Auf die Frage, wo Verbesserungsmöglichkeiten gesehen werden würden, antworteten die meisten, dass das Personal eine bessere Ausbildung im Bereich des Aufbaus von e-Infrastrukturen benötigen würde, dass mehr Personal nötig wäre, eine raschere Umsetzung von Projekten, klarere Workflows, eine besser funktionierende Kommunikation und mehr Transparenz. Ein Management auch bei kleineren Projekten wurde nicht ausschließlich positiv bewertet. Es sprach sich zwar knapp die Hälfte dafür aus, viele waren jedoch der Meinung, dass es nicht unbedingt notwendig sei bzw. sahen keinen Sinn darin, auch bei kleinen Projekten ein Management einzuführen. Eine offensichtliche Sorge war, dass sich damit alles verkomplizieren würde und durch den Mehraufwand Projekte ineffizient werden könnten.

Bei einem Drittel aller befragten Institutionen ist RE bereits ein Thema, wenn es auch nicht konsequent umgesetzt werden würde. Entsprechend ausgebildete ExpertInnen für RE gibt es derzeit jedoch kaum an den Institutionen. Unter den ausgebildeten Personen besaßen nur zwei ein entsprechendes Zertifikat. Keine der befragten Institutionen bietet derzeit Weiterbildungsmöglichkeiten im Bereich RE an. Knapp drei Viertel der InterviewpartnerInnen sehen in der Zukunft Bedarf an RE. Als Gründe wurden die vermehrte Zusammenarbeit von Bibliotheken mit dem IT-Bereich und die erstrebte Professionalisierung von Abläufen genannt. Der Rest, der sich nicht sicher war, ob Bedarf an RE besteht, äußerte Bedenken hinsichtlich der Kosten-Nutzen-Relation, da die Ausbildung Zeit und Geld kostet und es eventuell zu einem erhöhten Verwaltungsaufwand kommen könnte.

## 9 Zusammenfassung der Ergebnisse

Die Literaturrecherche hat ergeben, dass bereits einige Werke am Markt sind, die sich auch für den Bibliotheksbereich als Einstieg in das Thema RE eig-

nen. Sich das Wissen über RE anzueignen kann sowohl über Literatur, aber auch über diverse Kurse, die in Weiterbildungsinstitutionen angeboten wer-

den, erfolgen. Auch wenn in der Wirtschaft RE zum Teil schon fixer Bestandteil von Unternehmen ist und Jobprofile entwickelt wurden, wird es im bibliothekarischen Umfeld, zumindest in Österreich, noch nicht bzw. nur marginal in der bibliothekarischen Ausbildung angeboten. Das ist umso bedauerlicher, als in der Zukunft nahezu alle BibliothekarInnen mit IT-Projekten beschäftigt sein werden, wie auch die Befragungsergebnisse nahelegen.

Dass die Zusammenarbeit zwischen IT und Bibliothek von vielen als oft nicht sehr kompetent und zielführend bewertet wird, liegt auch daran, dass zu wenig Zeit, Geld und Personal eingeplant wird und dass deshalb das gesamte Projektmanagement nicht optimal erfolgen kann. Es fehlt oft das Wissen über RE, es gibt derzeit zu wenige Personen, die bibliothekarisches Wissen und IT-Wissen besitzen und gleichzeitig auch noch über die erforderliche kommunikative Kompetenz verfügen um als RE agieren zu können.

Insgesamt zeigen die Antworten, dass RE in wissenschaftlichen Bibliotheken noch nicht angewendet wird, obwohl bereits sehr intensiv mit IT-Abteilungen zusammengearbeitet wird.

Für die Zukunft scheint es ratsam zu sein, den BibliotheksmitarbeiterInnen Zugang zu diesem Wissen zu ermöglichen um das nötige Bewusstsein bezüglich des Umgangs mit Anforderungen optimieren

zu können. Viele der in den Handbüchern beschriebenen Schritte werden auch jetzt schon im Arbeitsalltag intuitiv ausgeführt, mit ein wenig Übung und den richtigen Tools lassen sich diese Ansätze verbessern und vor allem bewusstmachen.

Abschließend soll eine Auflistung von Tipps dabei helfen, eine erste Orientierung im Bereich RE zu erhalten.

- Alle Stakeholder in den gesamten Prozess einbeziehen
- Keine Angst vor Änderungen
- Alles lückenlos dokumentieren
- Keine raschen Ergebnisse erzielen
- Bei der Analyse von Anforderungen nicht an Kosten sparen
- Auch scheinbar Selbstverständliches muss explizit formuliert werden
- Keine Widersprüche bei Anforderungen
- Keine schleichenden Änderungen
- Alle Beteiligten müssen während der gesamten Laufzeit gut informiert werden
- Der/die Requirements-Engineer sollte bei Bedarf Coaching erhalten
- Die Rollen und Abläufe müssen allen vertraut sein
- Ein ständig gepflegtes und allen transparentes Glossar verwenden
- Gut funktionierende Workflows beibehalten
- Die Grundlagen von RE sollten allen bekannt sein

## Literatur

- Beck, K.; Beedle, M.; Bennekum, A. v. et al. (2001). *Manifesto for agile software development*. <http://agilemanifesto.org/iso/en/manifesto.html> zuletzt abgerufen am 17. Dezember 2018.
- Heche, D.; Ollech, O. (2006). Projektbegleitendes Anforderungsmanagement: unnötige Kosten und Ärger vermeiden. In *Projekt Magazin* (21/2006). [https://www.projektmagazin.de/artikel/projektbegleitendes-anforderungsmanagement-unnoetige-kosten-und-aerger-vermeiden\\_6949](https://www.projektmagazin.de/artikel/projektbegleitendes-anforderungsmanagement-unnoetige-kosten-und-aerger-vermeiden_6949) zuletzt abgerufen am 17. Dezember 2018.
- IEEE (1990). *610.12-1990 - IEEE standard glossary of software engineering terminology*. New York, NY: IEEE. doi: 10.1109/IEEESTD.1990.101064. <http://ieeexplore.ieee.org/document/159342/>.
- IEEE (1998). *830-1998 - IEEE recommended practice for software requirements specifications*. New York, NY: IEEE. doi: 10.1109/IEEESTD.1998.88286. <http://ieeexplore.ieee.org/document/720574/>.
- IEEE (2011). *29148-2011 - ISO/IEC/IEEE international standard – systems and software engineering – life cycle processes – requirements engineering*. New York, NY: IEEE. doi: 10.1109/IEEESTD.2011.6146379 zuletzt abgerufen am 5. Januar 2019. <http://ieeexplore.ieee.org/document/6146379/> zuletzt abgerufen am 5. Januar 2019.
- Kuhrmann, M. (2008–2018a). Rational Unified Process (RUP). In *Enzyklopädie der Wirtschaftsinformatik: Online-Lexikon*. Hrsg. von Gronau, N.; Weber, E. Potsdam: Lehrstuhl für Wirtschaftsinformatik der Universität Potsdam. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/>

- [Systementwicklung/Vorgehensmodell/Rational-Unified-Process-%28RUP%29/index.html](http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Rational-Unified-Process-%28RUP%29/index.html) zuletzt abgerufen am 17. Dezember 2018.
- Kuhrmann, M. (2008–2018b). V-Modell XT. In *Enzyklopädie der Wirtschaftsinformatik: Online-Lexikon*. Hrsg. von Gronau, N.; Weber, E. Potsdam: Lehrstuhl für Wirtschaftsinformatik der Universität Potsdam. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Rational-Unified-Process-%28RUP%29/index.html> zuletzt abgerufen am 17. Dezember 2018.
- Kulke, M. (2014). Wie Sie die natürliche Sprache bändigen. Die SOPHIST-Satzschablone: Funktionale Anforderungen präzise formulieren. In *Projekt Magazin* (15/2014). [https://www.projektmagazin.de/artikel/die-sophist-satzschablone-funktionale-anforderungen-precise-formulieren\\_1091786](https://www.projektmagazin.de/artikel/die-sophist-satzschablone-funktionale-anforderungen-precise-formulieren_1091786) zuletzt abgerufen am 17. Dezember 2018.
- Moser, J. (2007). *Integration von Anforderungsverhandlung in den Rescue Requirements Engineering Prozess*. Masterarbeit. Universität Wien.
- Neumann, H. A. (1998). *Objektorientierte Softwareentwicklung mit der Unified Modeling Language (UML)*. München, Wien: Hanser. 452 S.
- Niebisch, T. (2013). *Anforderungsmanagement in sieben Tagen: der Weg vom Wunsch zur Konzeption*. Berlin, Heidelberg: Springer.
- Partsch, H. (1991). *Requirements engineering*. München Wien: Oldenbourg. 335 S.
- Patig, S.; Dibbern, J. (2008–2018). Requirements Engineering. In *Enzyklopädie der Wirtschaftsinformatik: Online-Lexikon*. Hrsg. von Gronau, N.; Weber, E. Potsdam: Lehrstuhl für Wirtschaftsinformatik der Universität Potsdam. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/Requirements-Engineering/index.html?searchterm=requirements> zuletzt abgerufen am 17. Dezember 2018.
- Pohl, K.; Rupp, C. (2015). *Basiswissen Requirements Engineering: Aus- und Weiterbildung zum »Certified Professional for Requirements Engineering«*. 4. Aufl. Heidelberg: dpunkt-Verlag. 171 S.
- Rupp, C.; Geißel, C. (2007). Nichtfunktionale Anforderungen strukturiert erfassen und wieder verwenden. In *Projekt Magazin* (8/2007). [https://www.projektmagazin.de/artikel/nichtfunktionale-anforderungen-strukturiert-erfassen-und-wieder-verwenden\\_6987](https://www.projektmagazin.de/artikel/nichtfunktionale-anforderungen-strukturiert-erfassen-und-wieder-verwenden_6987) zuletzt abgerufen am 17. Dezember 2018.
- Rupp, C.; SOPHIST-Gesellschaft für Innovatives Software-Engineering (2014). *Requirements-Engineering und -Management: aus der Praxis von klassisch bis agil*. 6. Aufl. München: Hanser. 556 S.

## Anhang: Fragebogen

1. Sagt Ihnen der Begriff Requirements-Management/Requirements-Engineering etwas?
2. Werden in Ihrer Einrichtung gemeinsame Projekte zwischen der Bibliothek und der IT-Abteilung bzw. einer IT-Firma durchgeführt?
3. Wenn ja, seit wann, gibt es diese Zusammenarbeit zwischen der Bibliothek und der IT-Abteilung bzw. IT-Firmen?
4. Wie häufig arbeiten Sie zusammen?
5. Welche Abteilungen der Bibliothek arbeiten eng mit der IT-Abteilung/mit der Firma zusammen?
6. Wie viele Mitarbeiterinnen und Mitarbeiter sind auf der Bibliotheksseite involviert?
7. Wie viele Mitarbeiterinnen und Mitarbeiter sind auf der IT-Seite involviert?
8. Um welche Projekte handelt es sich dabei?
9. Gibt es gemeinsame Treffen, bei denen über Anforderungen gesprochen wird?
10. Gibt es in Ihrer Einrichtung eine bestimmte Person oder ein fixes Team, die oder das für das Projektmanagement zuständig ist, oder ändert sich die personelle Zusammensetzung von Projekt zu Projekt?
11. Gibt es klar definierte Rollen?
12. Wird für das Managen von Anforderungen Zeit eingeplant?
13. Werden die Zuständigkeiten im Projekt schon zu Beginn festgelegt?
14. Wird der Prozess rund um die Anforderungen dokumentiert?
  - a) Wer dokumentiert?
  - b) Wie wird dokumentiert?
  - c) Wird diese Dokumentation inhaltlich überprüft?
  - d) Wer überprüft die Dokumentation?
  - e) Wird die Dokumentation regelmäßig aktualisiert?
15. Wer erstellt die Anforderungen, die umgesetzt werden sollen?
16. Gibt es für das Eruiere von Anforderungen bereits Workflows?
  - a) Werden für das Eruiere von Anforderungen bestimmte Methoden verwendet?
17. Werden die Anforderungen untereinander abgestimmt?
18. Wenn ja, wer nimmt diese Abstimmung vor?
19. Wie werden die Anforderungen selbst dokumentiert?
20. Kennen Sie Standards des Requirements-Engineerings?
  - a) Wenn ja, wird an Ihrer Institution darauf zurückgegriffen?
21. Werden die Anforderungen qualitätsgeprüft?
22. Gibt es einen Feedbackprozess?
23. Wie wird mit unzureichenden Anforderungen umgegangen?
24. Wer hat Zugriff auf die erstellten Anforderungen?
25. Was passiert, wenn im Nachhinein neue Anforderungen gestellt werden bzw. die Anforderungen geändert werden?
26. Wird die Umsetzung der Anforderungen überprüft?
27. Wird der Umsetzungsprozess dokumentiert?
28. Sind Sie im Allgemeinen zufrieden mit dem Umgang mit Anforderungen?
29. Steht Ihnen für die Umsetzung dieser Projekte genügend Personal und Zeit zur Verfügung?
30. Wenn Sie nicht zufrieden sind, wo sehen Sie Verbesserungsmöglichkeiten?
31. Ist es Ihrer Meinung nach wichtig, dass es auch bei kleineren Projekten ein Management gibt, das sich um gewisse Dinge kümmert?
32. Ist Requirements-Engineering in Ihrer Einrichtung ein Thema?
33. Gibt es ausgebildete Personen im Bereich Requirements-Engineering? Wenn ja, welche Ausbildung haben sie?
34. Werden in Ihrer Institution Weiterbildungen im Bereich Requirements-Engineering angeboten?
35. Sehen Sie einen Bedarf von Requirements-Engineering in der Zukunft?