

# (Über-) Leben mit Anforderungen



## Thom Scheiner

User Interface Design GmbH  
Claudius-Keller-Straße 3c,  
81669 München  
thom.scheiner@uid.com

## Abstract

Mit Anforderungen an Benutzungsschnittstellen hat jeder schon seine Erfahrungen gemacht. Das Zusammentragen von Anforderungen kann Spaß machen; sie dann zu dokumentieren, aktuell zu halten und immer wieder Überblick über den aktuellen Stand im Projekt zu geben erinnert aber schon den Verzehr von trockenem Brot. In kleineren Projekten und bei Kunden, für welche die Arbeit von User Experience Consultants Neuland ist, fällt es auch schwer, Begeisterung für diesen Teil der Arbeit zu wecken und entsprechend das Budget dafür bewilligt zu bekommen. Der Beitrag beschreibt Erfahrungen aus mehreren Projekten unterschiedlicher Komplexität: - Argumente für die Diskussion mit Kunden, warum Anforderungen überhaupt erhoben werden müssen - Vorschläge zur Dokumentationsweise von Anforderungen, abhängig von der Projektkomplexität - und mit "Wow"-Effekt - Pro und Con bezüglich der Dokumentation mittels Tabellenkalkulation (Excel) - und wie man's nicht machen sollte - Best practice zum Angleichen unterschiedlicher Abdeckungsniveaus einzelner Themenkomplexe - und was man von seinem Product Owner erwarten können sollte.

## Keywords:

/// Requirements  
/// Anforderungen  
/// Dokumentation  
/// Erfahrungsbericht  
/// Praxis

## 1. Überblick

Anforderungen von Nutzern in die Umsetzung zu führen ist Bestandteil jedes UX-Projekts. Für den UX-Designer bedeutet dies, gut verargumentieren können zu müssen – denn das strukturierte Arbeiten mit Anforderungen kostet viel Zeit. Hat er ein Budget dafür bewilligt bekommen, müssen Anforderungen zunächst erhoben und dann auch – gerne mit Wow-Effekt – dokumentiert werden – egal ob auf Papier, mit Excel oder einem professionellen Anforderungs-Tool. Wichtig ist, den Dokumentationsaufwand im Griff zu haben, über alle Themenfelder hinweg ein vergleichbares Detailniveau zu erreichen und vom Kunden eine Aussage dazu zu erhalten, welche Anforderungen wichtiger sind als andere.

Im Folgenden erläutere ich meine Erfahrungen als UX-Designer und Requirements Engineer aus zahlreichen kleinen und mittleren Projekten (vornehmlich

Industry-Bereich) sowie aus zwei mehrjährigen Programmen bei Vor-Ort-Einsätzen in der Kundenorganisation. UX-Designern und Projektleitern gebe ich Argumente für die Budgetierung sowie Tipps zur Erhebung und Dokumentation von Anforderungen an die Hand.

## 2. Anforderung als Bestandteil von UX – und Argumentation gegenüber dem Kunden

Ob Projektleiter, Product Owner, User Expert/Endnutzer, Software-Entwickler oder UX-Designer: Jeder Stakeholder hat einen anderen Blickwinkel auf das Produkt. Eine Aufgabe beim Erheben und Dokumentieren von Anforderungen ist es, diese sich stark unterscheidenden Blickwinkel einzunehmen und zwischen ihnen zu vermitteln. Übliche Quellen für die Anforderungserhebung sind:

– Interviews mit Projektbeteiligten sowie mit den Nutzern

- Dokumente aus dem Projekt selbst (Projektbeschreibung, Machbarkeitsstudien, etc.)
- Ergebnisse von Business Analysten
- Tools, die momentan zur Erledigung der Aufgaben herangezogen werden
- Infos über oder von Nutzer(n) – etwa von der Service-Hotline oder aus Nutzerforen

Das strukturierte Erfassen von Anforderungen hilft, die Informationen trotz ihrer Vielzahl im Blick zu behalten. Der Aufwand dafür ist nicht zwangsläufig hoch, muss aber in jedem Fall einkalkuliert und im Projektbudget verankert werden. Mögliche Argumente dafür sind:

- Anforderungen geben Klarheit darüber, welche Themen umfassend und welche weniger/nicht bearbeitet werden (siehe Abschnitt 5).
- Anforderungen, die diesmal noch nicht berücksichtigt werden, können für den nächsten Release herangezogen werden.

- Für eine Reihe von Produkt-Klassen verlangt der Gesetzgeber den Nachweis, dass Anforderungen strukturiert erhoben und im Projektverlauf erfüllt wurden (etwa im Medical-Bereich nach DIN EN 60601-1-6 und DIN EN 62366).
- Die Arbeit mit Anforderungen hilft, einen Themenkomplex rasch zu durchdringen – und auf neue Ideen zu kommen (siehe Abschnitt 4.6).
- Anforderungen stellen eine Vereinbarung dar zwischen PO und UX-Designer, wodurch die Erwartungen besser geklärt werden können.
- Anforderungen zeigen allen Projektbeteiligten, was überhaupt schon als Aufgabenfeld erkannt ist: Was als Anforderung beschrieben ist, kann auch bearbeitet werden; was nicht gelistet ist, hat niemand auf dem Schirm.
- Der Status von Anforderungen macht den Fortschritt der UX-Arbeit für den Kunden sichtbar.
- Anforderungen geben Sicherheit für Kunden und UX-Designer – „Selbstverständliches“ versteht sich nicht von selbst und führt schnell zu Missverständnissen oder Ärger – etwa dass in einer Software für Ingenieure neben SI-Einheiten (metrisch) auch das imperiale Einheitensystem unterstützt wird.
- Wenn die Zeit knapp wird, gibt die Priorisierung von Anforderungen vor, was dem Kunden eventuell weniger wichtig ist
- Werden Anforderungen frühzeitig erhoben, ist man sicherer vor späteren grundsätzlichen Änderungen – je später sie kommen, desto härter schlagen Anforderungen zu.

### **3. Erheben von Anforderungen**

Je nach Produkt und Kunde stehen die o.g. Quellen in unterschiedlichem Maß zur Verfügung. Es ist aber auch stark vom Budget abhängig, wie lange man wo auf die Jagd geht.

In den kürzeren meiner Projekte geschah die meiste Anforderungsarbeit in Workshops mit Kunden. Meine Erfahrungen daraus:

- Zumindest im Industry-Bereich gibt es viele Produzenten, die viel über die Bedürfnisse ihrer Kunden zu wissen glauben und deshalb sich selbst als Quelle für Nutzeranforderungen empfehlen. Zudem ist der Zugang zu Kunden in dieser Branche tatsächlich schwieriger, da sie in Marktforschungspanels kaum auftauchen und für Feldstudien neben der Arbeitskraft oft auch gleich eine Maschine oder gar Produktionsteile blockiert wären.
- Je mehr Personen an Workshops teilnehmen und über Anforderungen diskutieren, desto intensiver wird eine jede Anforderung beleuchtet – wodurch meist mehr relevante Blickwinkel eingebracht und die Anforderung am Ende stabiler werden; und desto länger dauern die Diskussionen – das kann auch in die Stunden gehen.
- Für mich UX-Designer sind solche Workshops eher frustrierend: man kommt nur langsam voran, die Diskussionen driften teils stark ab, es ist viel Arbeit die Leute beim Thema zu halten. Das Ergebnis scheint dürrig – eine Liste mit einigen Einträgen. Und auf beiden Seiten bleibt das Gefühl, mit dem eigentlich Erwarteten (irgendwas das eher nach UI aussieht) nicht vorwärts gekommen zu sein.

Die Kunden eigenständig Anforderungen dokumentieren zu lassen funktioniert nur, wenn sie genau instruiert werden und der UX-Designer regelmäßig reinschaut. In kleineren Projekten versuchte ich das einige Male – mit sehr gemischten Ergebnissen. Hier fehlte der Erfolgsfaktor der regelmäßigen (!) Teilnahme des UX-Designers, denn dafür war kein Budget da.

Mein erstes mehrjähriges Vor-Ort Programm, für das ich zunächst 90% als Requirements Engineer arbeitete, bot bessere Möglichkeiten für die Arbeit mit Anforderungen. Es gab strukturierte Informationen von Business Analysten – etwa über die unterschiedlichen Abläufe in den einzelnen Unternehmen des Kunden sowie über die Herstellungsweise des jeweiligen Produkts. Daneben hatte ich einfachen Zugang zu User Experts sowie zu regelmäßigen

Besprechungen mit Software-Entwicklern/-Architekten. Dem Product Owner (diese Rolle gab es in meinen kleineren Projekten nicht) legte ich regelmäßig Anforderungen zu nun komplett abgebildeten Themenbereichen vor, um diese abnehmen zu lassen. Insgesamt war das viel, aber auch vereinbarte Arbeit, und sie führte zu einer hohen Reife der Anforderungen.

Bei meinem zweiten mehrjährigen Vor-Ort Programm drehten wir von Projekt zu Projekt Lernschleifen, ehe wir einen guten Modus Operandi gefunden hatten. Grundsätzlich wurde hier Wert gelegt auf dokumentierte Anforderungen, wenngleich der kalkulierte Aufwand dafür eher als hoch empfunden wurde. Stets standen uns sämtliche Quellen zur Verfügung, und Anforderungen wurden auch hier dem jeweiligen Product Owner vorgelegt.

Im Sommer 2012 initiierten meine Kollegen der dortigen zentralen UX-Abteilung, dass sich eine zehnköpfige Gruppe von User Experts unter Leitung eines Product Owners für die nächsten neun Monate wöchentlich traf. Zunächst sollten die Nutzeraufgaben definiert werden (es wurden ca. 30), später dann in kleineren Gruppen zwischen den wöchentlichen Sitzungen alle einzelnen Aufgaben genau beschrieben werden. Bei den wöchentlichen Treffen waren stets ein Business

Analyst sowie ein UX-Designer anwesend und lenkten die Diskussion, so dass am Ende die einzelnen Aufgaben vergleichbar detailliert beschrieben vorlagen. Beim Start eines Projekts im Frühjahr 2013 lag damit idealer Input für uns UX-Designer vor.

### **4. Dokumentieren von Anforderungen (und wie besser nicht)**

Für eines der Angebote bei meinem zweiten Programm überschlug ich, wie viel Aufwand eine einzelne Anforderung verursacht zwischen dem initialen Dokumentieren mit allen Attributen in einem professionellen Tool einerseits und dem Verknüpfen mit einer konzeptuellen Lösung andererseits (die Erhebung der Anforderung sowie



| Prozessschritt   | Aufwand (digital) | Aufwand (Post-It) |
|--|-------------------|-------------------|
| Initiales Aufschreiben mit allen Attributen            | 5 min             | 1 min             |
| Besprechung mit PO                                     | 3 min             | 3 min             |
| Präzisieren und neue Fragen klären                     | 15 min            | 15 min            |
| Nivellieren  | 5 min             | 5 min             |
| Besprechung mit PO                                     | 4 min             | 4 min             |
| Beziehungen und Status im System aktualisieren         | 3 min             | 1 min             |
| Doubletten managen                                     | 2 min             | 1 min             |
| Anhand einer Lösung präzisieren und neue Fragen klären | 5 min             | 5 min             |
| Besprechung mit PO                                     | 2 min             | 2 min             |
| Lösung verknüpfen und Status im System aktualisieren   | 5 min             | 1 min             |
| Verbleibenden Datensatz digitalisieren                 |                   | 2 min             |
| <b>Gesamtaufwand je Anforderung</b>                    | <b>49 min</b>     | <b>40 min</b>     |

**Tab. 1.**  
Kalkulation für den Aufwand, den eine einzelne Anforderung während ihres Lebenszyklus verursacht (kalkuliert für digitale Erfassung in einem professionellen Tool sowie für Erfassung mittels Post-Its und nachträglicher Digitalisierung)

das Entwickeln einer konzeptuellen Lösung sind nicht enthalten); Ergebnis: 49 Minuten – je Anforderung, wohl gemerkt!

Wir fanden aber auch einen Weg, ein Fünftel dieses Aufwands einzusparen – und leider auch einen anderen Weg, um die Arbeit mit Anforderungen ohne große Not ziemlich unhandlich zu machen.

#### 4.1. Aufwand durch Prozessschritte

Nachdem eine Anforderung initial festgestellt wird, hat sie noch einige Bearbeitungsschritte vor sich, ehe sie ihren finalen Stand erreicht:

- Erste Dokumentation: Die Anforderung sollte strukturiert festgehalten und damit für den weiteren Projektverlauf zuverlässig erreichbar sein.
- Abgleich mit User Experts und Product Owner: Diese Quellen helfen, die Bedeutung der Anforderung zu erfassen und ggf. weitere noch zu erforschende Themen zu definieren.
- Präzisieren und Klären neuer Fragen: Die soeben definierten weiteren

Themen wirken auch zurück auf die vorliegende Anforderung; dadurch muss die Anforderung wieder überarbeitet werden – etwa indem sie verknüpft wird mit anderen (neuen) Anforderungen.

- Nivellieren: Die Anforderung dringt in eine gewisse Tiefe eines Themas ein und hilft, dieses Thema besser zu beschreiben. Andere Themen mit ähnlicher Wichtigkeit sind möglicherweise viel präziser oder auch oberflächlicher durch Anforderungen abgedeckt. Es sollte also ein Prozess stattfinden, um die unterschiedlichen Themen auf ein vergleichbares Niveau zu bringen. Siehe Abschnitt 4.6.
- Abgleich mit Product Owner und Aktualisieren des Status: Eine weitere Besprechung des aktuellen Stands der Anforderung mit dem Product Owner; eventuell folgen daraus weitere Aufgaben, ansonsten ist nur der neue Status der Anforderung festzuhalten.
- Doubletten managen: Mit zunehmender Zahl von Anforderungen und v.a. bei mehreren Autoren können auch Doppelungen von Anforderungen auftreten.

Diese sollten von Zeit zu Zeit identifiziert, markiert und entsorgt werden.

- Anhand einer Lösung präzisieren und neue Fragen klären: Eine mögliche Lösung für die Anforderung liegt vor – und diese Lösung mag zu völlig neuen Fragen führen, welche ihrerseits wieder zu klären und in der Anforderung dokumentiert werden sollten.
- Besprechung mit PO: Der Product Owner akzeptiert die präsentierte Lösung oder weist sie zurück. Diese Information wird bei der Anforderung vermerkt, und die akzeptierte Lösung wird mit der Anforderung verknüpft.
- Datensatz digitalisieren: Sofern noch nicht geschehen, wird die Anforderung schließlich digital erfasst.

All diese Prozessschritte verursachen Aufwand. Da die Arbeit mit Anforderungen oft zwischen anderen Projektschritten erfolgt und sich dadurch zeitlich schwer fassen lässt, habe ich versucht einen „Stückpreis“ für die Anforderungen zu ermitteln – also den Aufwand, den eine einzelne Anforderung während ihres Lebenszyklus verursacht. [Tab. 1]

Die Kalkulation in Tabelle 1 enthält Posten, die nicht bei jeder Anforderung anfallen. So gibt es durchaus Anforderungen, die sofort klar und umfassend formuliert sind und ohne weitere Änderungen auskommen; ausgleichend gibt es aber auch Anforderungen, die wesentlich aufwändiger handzuhaben sind.

#### 4.2. Vergleich der Dokumentation per Post-Its und digitaler Erfassung

Auf Papier generell geht die Dokumentation schnell von der Hand und kann dazu beitragen, die Projektbeteiligten ins Boot zu holen und sie direkt in die Arbeit zu integrieren. Sie eignet sich aber nicht für jedes Szenario – etwa nur dann, wenn die Anforderungen mit anderen Standorten geteilt werden müssen.

- Stärken der Dokumentation mittels Post-It
  - **Bekannt:** Jeder kennt die Tools, und die Organisation auf der

| Kriterium   | Digital (Excel)                                | Digital (Profi-Tool)       | Post-It  |
|---|--|----------------------------|--|
| Sollen mehrere Personen zugleich an den Anforderungen arbeiten?                       | (nur eingeschränkt; Gefahr von Datei-Wirrwarr) | unterstützt                | unterstützt  |
| Soll die Abbildung der Anforderungen Diskussionen anregen?                            | (eher trist und technisch)                     | (eher trist und technisch) | diskussionfördernd                                       |
| Was ist wichtiger: Stabilität oder Flexibilität?                                      | Stabilität                                     | Stabilität                 | Flexibilität   |
| Müssen die Anforderungen archiviert werden?   | Archivierbar                                   | Archivierbar               | (nur per Foto oder spätere Digitalisierung archivierbar) |
| Wird für ein feststehendes Team dokumentiert, oder kommen später andere Nutzer hinzu? | Wechselnde Nutzer                              | Wechselnde Nutzer          | Feststehendes Team                                       |
| Ist es wichtig, Anforderungen vollständig definiert zu haben?                         | Vollständig definiert                          | Vollständig definiert      | Quick & dirty  |
| Müssen die Anforderungen an andere Nutzer verteilt werden können?                     | Verteilen möglich                              | Verteilen möglich          | (nur per Foto oder spätere Digitalisierung verteilbar)   |
| Ist Versionierung bedeutsam?  | (nur eingeschränkt)                            | Versionierbar              | (nur per Foto versionierbar)                             |
| Ist Zugriffskontrolle bedeutsam?  | (nur eingeschränkt)                            | Kontrollierbar             | nicht unterstützt  |

Tab. 2.  
Kriterien für die Auswahl der Dokumentationsart

Arbeitsfläche (thematische Gruppierung, ggf. Farbkodierung) ist leicht erklärt. Man muss also kein Computerprogramm beherrschen dafür. Außerdem stehen bei Post-Its – anders als bei Software – sicher auch immer genug kostenlose Lizenzen zur Verfügung;-)

- **Alles im Blick:** Auf einer großen Wand lassen sich die Post-Its nach Themenbereichen gruppiert anordnen. Man kann mit Fäden, Linien oder separaten Papierbögen arbeiten – und damit schnell den Fokus auf Überblick oder Detail einstellen. Auch bei vielen Anforderungen lassen sich hier Zusammenhänge einfach abbilden und nachvollziehen.
- **Potenziell riesige Arbeitsfläche:** Je nach Platzverhältnissen kann man Wände, Fenster, die Decke oder auch mobile Raumteiler verwenden – es wird sicher mehr Platz sein als ein 24-Zoll-Monitor.

- **Gleichberechtigt und spontan:** Neue Informationen lassen sich ohne spürbaren Aufwand hinzufügen und verorten: einfach ein Post-It schreiben und aufkleben. Jeder hat dieselben Möglichkeiten und Rechte, man muss nicht um Tastatur und Maus bitten oder das Programmfenster wechseln.
- **Erweiterbar:** Eine Handskizze, ein ausgedrucktes Dokument, ein Foto, ein Workflow, irgendein anderes Artefakt – was hilft lässt sich i.d.R. mit dazu hängen und von allen nutzen.
- **Schnell:** Die Arbeit auf Papier geht sehr rasch vonstatten.
- Stärken der digitalen Dokumentation
  - **Typische Funktionen:** „Rückgängig“, „Senden an“, „Copy&Paste“; bei der Papier-Variante hingegen bleibt eine gezogene Linie sichtbar.
  - **Versionierung und Tracking:** Gelöschtes wiederherstellen oder auch nachvollziehen können, wer einen Status geändert hat. Gute

Tools bieten solche Funktionen – aber in meiner bisherigen Arbeit brauchte ich diese Funktionen noch nicht.

- **Luftzug-resistent:** Kein Windstoß bringt meine Anforderungen zum Wanken. Wenn Sticky Notes schon eine Weile hängen, muss man hingegen vielleicht nur schnell dran vorbeigegangen sein um sie hinabtrudeln zu lassen.
- **Lokal gebunden:** Verteilt arbeitende Teams brauchen einfachen Zugriff auf die Anforderungen. Als Datei oder Datenbank kann man sie einfach mit anderen Büros teilen, in der Papierform wird das schwierig.
- **Informationstiefe:** Digital kann man unheimlich viele Daten je Anforderung erfassen. Auf ein Post-It bekommt man hingegen nur wenige Informationen – vielleicht die Quelle noch auf die Rückseite, aber das erfordert schon viel Disziplin.



Der wesentliche Erfolgsfaktor für mein erstes solches Projekt war, dass wir genau jene Phase der Anforderungsarbeit mit ökonomischen Post-Its bewältigten, in welcher die Anforderungen am intensivsten bearbeitet wurden: Neue Informationen erforderten neue Formulierungen (Post-It neu schreiben), neue Informationen erforderten Ergänzungen (auf dem Post-It nachtragen), Anforderungen mussten neu gruppiert werden (Post-Its umhängen). Für die Arbeit an einem bestimmten GUI-Thema wollte man sich auf eine Gruppe von Anforderungen konzentrieren (Poster mit allen Post-Its eines Themas mitnehmen), Diskussionen mit dem Product Owner verwarfen eine Anforderung wieder, etc. Diese Aktionen kosteten mit Post-Its nur wenig Zeit, diese Phase der Anforderungsarbeit verursachte also kaum Aufwand, und sie signalisierte anderen Beteiligten Offenheit für neuen Input.

Als dann die Anforderungen über mehrere Iterationen in die finale Form hinein diskutiert worden und damit ziemlich stabil waren, zogen wir sie in einem Kraftakt um in die digitale Welt (siehe letzter Prozessschritt in Tabelle 1). Für all jene Anforderungen, die der natürlichen Selektion zum Opfer gefallen waren, fiel dieser Aufwand schon nicht mehr an.

#### 4.3. Kriterien für die Wahl der Dokumentationsart

Ob man nun digital oder per Post-Its dokumentiert, hängt ab von mehreren Kriterien. Kriterien für die Auswahl der Dokumentationsart<sup>2</sup> zeigt, welche Kriterien man berücksichtigen muss und welche Dokumentationsart das Kriterium unterstützt. Es empfiehlt sich, diese Kriterien vorab mit dem Kunden zu klären, denn ein späterer ungeplanter Wechsel der Dokumentationsart sorgt für Frust und zusätzlichen Aufwand. [Tab. 2]

#### 4.4. Professionelle RE-Tools vs. herkömmliche Tabellenkalkulationen

Sofern in der Organisation für das Anforderungsmanagement bereits eine Software eingeführt wurde und man durch dieses



Abb. 1. Anforderungsschablone (schematisch, nach Rupp)

| ID | Rolle | Source        | Pre-Subject | Action       | Object  |
|----|-------|---------------|-------------|--------------|---|
| 1  | 0001  | WCAG 2.0 14.3 | The system  | MUST provide | minimum of requested contrast (4,5:1)                                 |
| 2  | ###   | WCAG 2.0 14.4 | The system  | MUST provide | resizing of text up to 200% without loss of content or functionality. |
| 3  | ###   | WCAG 2.0 14.5 | The system  | MUST use     | text instead of images of text in essential cases (i.e. legibility).  |
| 4  | ###   | WCAG 2.0      | The system  | allow        | the user to obtain the background and                                 |

Abb. 2. Excel-Template für vollständige Anforderungstexte – leider kaum benutzbar

nicht zu sehr eingeschränkt wird im kreativen Arbeiten, sollte man diese Software natürlich unbedingt verwenden und dem Kunden damit die Nutzung der eigenen Arbeitsergebnisse sehr erleichtern.

Hat ein Kunde noch keine Software hierfür im Einsatz, so kann man aus meiner Sicht mit den üblichen Tabellenkalkulationen (Excel und Co.) loslegen.

#### 4.5. Eine Excel-Formel für mehr Chaos

Die Qualität der Anforderung hängt ab vom Wissen des Autors; sind mehrere Autoren beteiligt, so schwankt mitunter die Qualität der Dokumentation.

Ich wollte unter Berücksichtigung der Anforderungsschablone von Rupp<sup>1</sup> [Abb. 1] sicherstellen, dass alle Autoren gleich von Beginn an zu jeder Anforderung alle benötigten Informationen beitragen.

Hierzu legte ich in Excel ein stark schematisiertes Arbeitsblatt an – und wie Dr. Jekyll war ich mir nicht bewusst was ich da erschuf. Je Informationsbaustein spendierte ich eine Spalte – also etwa je für „Subject“, für „Prio“, für „Action“ (Spalten I bis O in Abbildung 2). Der Autor konnte anhand leerer Zellen leicht erkennen, ob er vielleicht wichtige Informationsbausteine noch nicht bereitgestellt hatte.

In einer separaten Spalte (Spalte F in Abbildung 2) baute ich eine Formel ein mit dem zentralen Befehl „Concatenate“ / „Verketten“. Diese Formel zog die Informationen aus den gerade genannten Spalten an (z.B. „The system“, „MUST“, „provide“, ...) und verband sie zu einem flüssig lesbaren Text („The system MUST provide“ ...). In dieser Spalte stand dann die Anforderung schön lesbar drin. [Abb. 2]

Doch jedes Mal wenn man etwas am Anforderungstext ändern wollte, war man erst mal frustriert: Man wechselte in der Spalte mit der schön lesbaren Anforderung in den Bearbeitungsmodus – und das war Spalte F, die aber nur eine Formel enthielt. Man musste also erst nach der richtigen Spalte suchen (Spalte I bis O) und dort den Text ändern. Und startete bei der nächsten Anforderung sicher wieder in der falschen Spalte.

Das Problem war wirklich groß – alle drei Autoren stolperten andauernd über diese Formel-Spalte. Die beiden Vorteile (Vollständigkeit der Anforderung und gute Lesbarkeit für den Leser) traten rasch, sehr rasch in den Hintergrund. Wir behielten dieses Schema bei bis zum Ende des Projekts, aber seither ist klar dass wir es so nicht wieder machen wollen.

#### 4.6. Nivellieren

Bei den Nutzaufgaben eines bestimmten Produkts, an dem ich arbeitete, gab es drei

Aufgaben, die thematisch eng beisammen lagen: Anlegen von Reference Cases, von Remindern und von Subscriptions. Diese Aufgaben waren nun unterschiedlich tief mit Anforderungen abgedeckt, und wir wollten abklären, ob sich Teilaufgaben davon überschneiden.

Eine Kollegin trug deshalb alle Anforderungen zu diesen drei Aufgaben zusammen, analysierte sie und übertrug sie der Vergleichbarkeit halber in eine Matrix (siehe Matrix zum Niveau-Vergleich der Anforderungsabdeckung dreier Aufgaben<sup>3</sup>). Die Analyse bezog sich auf diese Aspekte:

- Gegenstand (Aufgabe vs. Akte)
- Darstellung (Überblick, visueller Hinweis, Push-Notification)
- Informationsbedarf (notwendige Informationen zur Beurteilung, ob man diese Akte öffnen will)
- Auslöser (selten oder besonders, notweniger Input von anderen, Neugier)
- Methode (Push vs. Pull)
- Zweck (Qualitätssicherung, Vollständigkeit, Neugier)
- Aktuell genutzte Tools und Prozesse

Dann prüften wir je Punkt, ob die drei Aufgaben ähnlich präzise durch Anforderungen abgedeckt waren. War bisher z.B. bereits klar dass die Liste der Reminder Änderungen hervorheben muss, so bemerkten wir dass der Bedarf für die Darstellung von Subscriptions noch unklar war und wir hier weitere Informationen benötigten.

Anhand dieser Matrix erkannten wir, bei welchen Aufgaben ein Aspekt zu ungenau oder unnötig detailliert dargestellt war. Wir konnten daraufhin die Anforderungen präzisieren bzw. entschlacken, so dass sie schlussendlich die drei Aufgaben unter jedem Aspekt in vergleichbarer Qualität abdeckten.

## 5. Aufgaben für den Product Owner

Anforderungen bestimmen maßgeblich das Endergebnis des Produkts und verpflichten die Projektbeteiligten. Deshalb sollte von Kundenseite geklärt werden welche Anforderungen tatsächlich relevant sind und wo für eine Entscheidung noch

| 4. ANALYSIS |               |  |  |  |
|-------------|---------------|--|--|--|
|             |               | Aufgabe  |  |  |
|             |               | Reference  | Reminder   |  |
|             |               |  | Subscription   |  |
| Aspekte     | item          | <ul style="list-style-type: none"> <li>o example of refusal.</li> <li>o example of suspension,</li> <li>o etc...</li> </ul>  |  |  |
|             | presentation  | <ul style="list-style-type: none"> <li>- overview</li> </ul>   | <ul style="list-style-type: none"> <li>- overview</li> <li>- visual indicator in work manager</li> </ul>   | <ul style="list-style-type: none"> <li>- overview</li> <li>- (notification?)</li> </ul>  |
|             | require       | <ul style="list-style-type: none"> <li>- overview is kept up-to-date by user</li> <li>- overview is editable                             <ul style="list-style-type: none"> <li>o add item</li> <li>o delete item</li> <li>o possibility of creating folders / subfolders or categories or tagging (?)</li> </ul> </li> <li>- user can add comments when adding an item</li> <li>- user can edit comment in an already added item</li> <li>- only visible to user</li> </ul> | <ul style="list-style-type: none"> <li>- an automatic reminder is made by system to the assignee of the task (approaching due date)</li> <li>- a manually set reminder is made to user A when he has re-assigned the task to user B</li> <li>- reminder is also visible to team</li> <li>- several reminders can be set on a task by several users (probably not needed in new system when tasks supersede messages?)</li> <li>- user can add a comment                             <ul style="list-style-type: none"> <li>o the user to himself</li> <li>o the system to the user</li> <li>o user (manager, trainer...) to another? to be discussed!</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>- only concerns the user who subscribes</li> <li>- overview is kept up-to-date by user                             <ul style="list-style-type: none"> <li>o add item</li> <li>o delete item</li> </ul> </li> <li>- user can add comments when adding an item</li> <li>- user can edit comment in an already added item</li> <li>- only visible to user</li> </ul> |
|             | inform needed | <ul style="list-style-type: none"> <li>- * task ID (technical requirement)</li> <li>- task name</li> <li>- task type (doccode)</li> <li>- date of completion (?)</li> <li>- date of addition to overview (?)</li> <li>- user's comment</li> </ul>  | <ul style="list-style-type: none"> <li>- * task ID (technical requirement)</li> <li>- task name</li> <li>- task type (doccode)</li> <li>- due date</li> <li>- trigger event</li> <li>- user's comment</li> </ul>   | <ul style="list-style-type: none"> <li>- * dossier nr</li> <li>- user's comment</li> <li>- (trigger event?)</li> </ul>   |
|             | trigger       | <ul style="list-style-type: none"> <li>- from user perspective, occurrence of a task that is:                             <ul style="list-style-type: none"> <li>o rare</li> <li>o unusual (anomaly)</li> <li>o curious</li> </ul> </li> </ul>   | <ul style="list-style-type: none"> <li>- from user perspective: a task cannot be completed without input from other user</li> <li>- system trigger: event or date                             <ul style="list-style-type: none"> <li>o when re-assigned task is completed</li> <li>o if re-assigned task is not completed after X days</li> <li>o if re-assigned task is not completed after X days before the due date</li> </ul> </li> </ul>   | <ul style="list-style-type: none"> <li>- from user perspective, occurrence of a dossier that interests him</li> <li>- (system trigger: event?)</li> </ul>  |
|             | method        | <ul style="list-style-type: none"> <li>- the user pushes the task to the reference overview</li> <li>- when the user needs information about a task type he pulls it from the reference</li> </ul>   | <ul style="list-style-type: none"> <li>- PULL</li> </ul>   | <ul style="list-style-type: none"> <li>- PUSH</li> </ul>   |

Abb. 3. Matrix zum Niveau-Vergleich der Anforderungsabdeckung dreier Aufgaben

Informationen fehlen. Der Product Owner sollte diese Aufgabe wahrnehmen; doch manche verstehen gar nicht wie wichtig diese Aufgabe ist – so dass wir sie in diese Richtung leiten müssen.

Priorisierung: Der PO muss klären welche Anforderungen wichtig und welche weniger wichtig sind. Andernfalls muss der UX Designer entweder glatt alle Anforderungen lösen, was häufig Zeit und Budget sprengen würde; oder er muss selbst priorisieren, wodurch aber das Produkt in eine andere Richtung gehen könnte als vom Kunden intendiert.

In einem meiner Projekte hatte der PO auch nach neun Monaten noch keine einzige Anforderung priorisiert; in einem anderen Projekt ging ich mit dem PO gemeinsam alle Anforderungen durch – und hatte am Ende nahezu ausschließlich Anforderungen mit „MUST“-Einstufung (was nicht praktikabel ist, da man dann doch wieder selbst priorisieren muss).

Lücken und Fragen: Der PO sollte die Domäne kennen und Lücken in den Anforderungen identifizieren können (siehe auch Abschnitt 4.6). Sofern er offene Fragen

nicht selbst beantworten kann, sollte er passende Ansprechpartner nennen können.

Diskussionen zu bedeutenden Themen: Durch die Anforderungsarbeit stößt man immer wieder auf grundsätzliche Fragen, die sich dann nicht aus dem Projekt heraus schnell klären lassen. Oder man entwickelt weitreichende Vorschläge, die z.B. einen Umbau der Prozesse innerhalb des Unternehmens mit sich bringen könnten. Die hierzu nötigen Diskussionen kann man selbst führen, sofern der Kunde einen gewähren lässt; oder man gibt sie an den PO ab, der dann im Sinne des besseren Produkts diese Vorschläge durch die Instanzen diskutieren soll.

<sup>1</sup> Rupp, Chris; SOPHIST Group: „Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis“, 5. Auflage; Carl Hanser Verlag GmbH & Co. KG

