

Storytelling, User-Stories und Co. – verschiedene  
Arten, Anforderungen zu vermitteln

---



In diesem Kapitel beschreiben wir einige Möglichkeiten der Vermittlung von Anforderungen. Dabei ist den Techniken gemein, dass sie ohne eine Anforderungsdokumentation im klassischen Sinne auskommen. Viele davon (z. B. User-Stories oder Storytelling) werden vor allem in der agilen Welt häufig eingesetzt.

Allerdings schließen sich Anforderungsdokumentation mit Modellen oder natürlichsprachliche Anforderungsdokumentationen und die hier vorgestellten Möglichkeiten nicht gegenseitig aus. Vielmehr können Sie diese durchaus kombinieren, um die Stärken beider Varianten zu vereinen.

Möchten Sie wissen, wann welche Techniken am sinnvollsten einzusetzen sind, lesen Sie [Kapitel 16 „Wegweiser: Anforderungen dokumentieren und vermitteln“](#). Hier finden Sie auch die Einflussfaktoren, anhand derer Sie eine Eignung der einzelnen Ansätze für die drei Szenarien aus [Kapitel 4 „RE ist nicht gleich RE“](#) beurteilen können.

Wir möchten Sie darauf hinweisen, dass nicht alle Techniken, die Sie zum Vermitteln verwenden können, in dem vorliegenden Kapitel zu finden sind. Eine Erläuterung, in welchen Kapiteln Vermittlungstechniken beschrieben sind, finden Sie in [Kapitel 16 „Wegweiser: Anforderungen dokumentieren und vermitteln“](#).

## 17.1 Storytelling – Grimms Märchen der Anforderungsvermittlung

---

Hören Sie sich gerne einen nüchternen Sachvortrag an, wenn Sie den gleichen Inhalt auch in einer guten Story verpackt genießen könnten? Geschichten sind etwas, was Menschen seit eh und je fasziniert. Mittels Geschichten wird seit Anbeginn der Menschheit Wissen vermittelt. Somit ist es nicht verwunderlich, dass das Erzählen von Geschichten (Storytelling) auch im Requirements-Engineering zur Wissensvermittlung eingesetzt wird.

### Definition Storytelling [Mattschek]



Storytelling beschreibt eine Kommunikationsart zur Vermittlung von Informationen, Wissen, Werten, Meinungen etc. Dies kann über Sprache, Text, Bild oder Videos erfolgen. Dabei werden nicht emotionale Inhalte in Geschichten verpackt, um über die Geschichte Emotionen und Interesse bei Zuhörern, Lesern oder Betrachtern zu wecken.

Storytelling eignet sich im Requirements-Engineering vor allem für das Ermitteln und das Vermitteln von Visionen, Zielen, Epics, Geschäftsprozessen, Personas, Anforderungen und User-Stories. Beim Ermitteln von Wissen wird meist der Stakeholder die Geschichten erzählen. Beim Vermitteln von Wissen können sowohl der Product-Owner, der Requirements-Engineer, oder aber auch ein Stakeholder Wissen in Form einer Story an das Entwicklungsteam oder den/die AuftragnehmerIn vermitteln. Der Product-

Owner und/oder Requirements-Engineer sollte lernen, wie man Wissen effektiv mittels Storys vermittelt. Wir gehen bei den Stakeholdern nicht davon aus, dass sie perfekte Storyteller sind. Wir leiten sie durch Rückfragen und Hinweise dazu an, die wichtigsten Regeln des Storytellings einzuhalten. Wenn z. B. ein Stakeholder nicht von sich spricht, sondern immer wieder in Formulierungen verfällt wie „man braucht doch ...“, „man macht doch ...“, dann hinterfragen Sie einfach: „Brauchen Sie diese Funktion?“, „Wie gehen Sie denn mit der Situation um?“.

Als Beispiel dient uns hier eine Geschichte von Lina, in der sie uns ihre Wünsche zum automatischen Öffnen und Schließen der Zugangstür erzählt:

*Lina Meyer:*

*„Sonntags liebt es meine Familie nachmittags bei einer Tasse Tee und einem leckeren Stück Kuchen – gerne auch mal mit Freunden zusammen – im Garten zu entspannen. Dazu fahre ich nach dem Mittagessen mit dem Auto zu unserer lokalen Bäckerei. Von dort komme ich mit mehreren Kuchenpäckchen auf dem Arm zurück. Voll beladen stehe ich dann vor der Eingangstür. Ich versuche dann den Hausschlüssel aus der Tasche zu kramen, ohne den Kuchen auf dem Boden ablegen zu müssen, was meist nicht funktioniert. Klingeln bringt auch nix, denn die Familie befindet sich ja im Garten. Das neue Türöffnungssystem sollte mich erkennen, mir in Zukunft die Tür automatisch öffnen und sie hinter mir wieder schließen.“*

### 17.1.1 Arten des Storytellings

Es gibt unterschiedliche Arten von Storys [Löhr], deren Einsatz im Requirement-Engineering für die Vermittlung von Wissen Sinn ergibt.

**Hintergrundstorys** erzählen etwas über die Umgebung und die Nutzung des Systems – sie übermitteln wichtige Hintergrundinformationen. Sie explizieren damit oftmals implizites Wissen, welches den Zuhörenden dabei hilft die Anforderungen besser zu verstehen. In unserem Beispiel würde z. B. ein Video über das Haus der Familie Meyer und die Umgebung des Hauses Wissen zur Geografie, der Anmutung und der Historie des Anwesens vermitteln und damit ein erstes Bild der Situation in den Köpfen aller Beteiligten entstehen lassen. Das [Kapitel 27 „Videos im RE“](#) zeigt zum Beispiel, wie ein Umgebungsvideo die Einsatzumgebung des Smart-Home-Systems (SHS) vermitteln kann.



**Personality-Storys** stellen die Personas oder Stakeholder des Systems vor und machen deren Eigenschaften und Persönlichkeit erlebbar. Neben den Fakten zur Person (wie Alter, Beruf, Ausbildung) zeigt eine Personality-Story konkrete Meinungen, Vorlieben, Einstellungen zu relevanten Themen (z. B. Umgang und Einstellung zu IT-Systemen). Im Vergleich zu einer schriftlichen Personbeschreibung oder den Einträgen

in einer Stakeholderliste ist z. B. ein Video zur Vorstellung einer konkreten Person (eines Stakeholders) oder einer fiktiven Persona, die eine Geschichte über sich erzählt, viel informationsreicher und einprägsamer. Neben dem gesprochenen Text wird sehr viel Information über das Temperament, den Charakter, die Art zu interagieren und zu sprechen der gefilmten Person vermittelt. In unserem Beispiel würden Personality-Videos der BewohnerInnen mit den jeweiligen Einstellungen und Charakteren einen guten Eindruck der zukünftig Nutzenden vermitteln.

Achten Sie darauf, dass Personality-Videos als Minimalinhalt die Informationen enthält, die auch in der Stakeholdercheckliste vermerkt würden oder in der Personbeschreibung enthalten wären, falls das Video diese Artefakte ersetzen soll.

**Überzeugungsstorys** transportieren ganz am Anfang des Requirements-Engineerings die Ausgangslage und erklären motivierend, warum das System gebraucht wird. Im Beispiel unseres SHS könnte z. B. die Architektin in einem ambitionierten Plädoyer die Vision des neuen smarten Hauses der Familie Meyer vermitteln. Ein Beispiel für eine Überzeugungsstory für ein Smart Home finden Sie am Anfang des Kapitels in **Abschnitt 17.1 „Storytelling – Grimms Märchen der Anforderungsvermittlung“**. Achten Sie bei Überzeugungsstorys darauf, dass die Ziele/Visionen des Systems kommuniziert werden, ohne die Realisierung vorzugeben (sofern diese noch nicht feststeht). Ein Beispiel für eine Überzeugungsstory für das Requirements-Engineering sehen Sie in folgendem Video:

**Erklärungsstorys** verdeutlichen einzelne Abläufe und Verhaltensweisen des Systems. Wenn komplexe Sachverhalte mit Bildern und Geschichten untermalt werden, können Sie das Ganze besser nachvollziehen und sich merken. Auch Ihre Aufmerksamkeit wird durch Visualisierung (sei es auch nur im Kopf) stärker angesprochen. Wie wäre es für unser konkretes Beispiel mit mehreren Videos oder einer Erzählung, die verdeutlicht, wie sich die Familie Meyer später das automatisierte Öffnen ihrer Haustür vorstellt? Eine Erklärungsstory muss alle relevanten Vor- und Nachbedingungen des Ablaufs enthalten, ebenso wie die Prozessschritte, sofern diese schon bekannt sind. Eine Erklärungsstory ist somit ein natürlichsprachliches Szenario, das sich hervorragend eignet, um die gewünschte Interaktion des Erzählers mit dem System zu verstehen. Daraus lassen sich im nächsten Schritt Anforderungen, z. B. in Form von User-Storys, ableiten. Szenarien helfen auch dabei Abläufe systematisch zu durchdenken – insbesondere wenn der Stakeholder lieber in konkreten Beispielen statt in abstrakten Anforderungen denkt und spricht. Weitere Infos zum Thema Szenarien finden Sie unter [www.sophist.de/re7/kapitel17](http://www.sophist.de/re7/kapitel17).

### 17.1.2 Was macht gutes Storytelling aus?

---

Damit eine Story für die Vermittlung geeignet ist, muss sie gut sein. Aber was heißt das konkret für den Einsatz im Requirements-Engineering? Die Botschaft der Geschichte muss eindeutig sein (denn uns geht es ja um das vermittelte Wissen). Zudem müssen sowohl die Geschichte als auch die Person, die die Geschichte erzählt, authentisch wirken (damit der Vermittlungsprozess optimal funktioniert). Das lässt sich durch die folgenden Punkte erreichen [Fordon17]:

- Die Story muss nicht unbedingt wahr sein, aber reale Storys lassen sich authentischer erzählen als erfundene. Eine erfundene Geschichte sollte die erzählende Person gut vorbereiten, sodass es dem Publikum gar nicht auffällt, dass sie erfunden ist.
- Die Person, die erzählt, verstellt sich nicht, sondern verhält sich genauso, wie sie es auch privat tun würde. Falls Sie als Product-Owner oder Requirements-Engineer eine Geschichte wiedergeben, die Ihnen von Ihrem Stakeholder erzählt wurde, so stellen Sie einfach die Info voran, wer Ihnen aus welchem Anlass diese Geschichte erzählt hat, und geben diese möglichst genau wieder. Hier eignen sich z. B. Videoaufnahmen von Geschichten.
- Die Geschichte ist sprachlich und inhaltlich an die Zielgruppe angepasst.
- Drei bis fünf Minuten sind eine gute Erzähldauer [Oswald14].
- Bei den Zuhörenden müssen Emotionen geweckt werden durch [Fordon17]:
  - emphatische Charaktere,
  - eine schlüssige Hintergrundgeschichte,
  - adaptierbare Werte und Moralvorstellungen eines Charakters,
  - Ziele, die ein Charakter oder die Geschichte verfolgt,
  - Gefahren oder Konflikte, die überwunden werden (auch wenn das im RE meist nur kleine sind).

### 17.1.3 Die irrelevanten Teile einer Story

---

Eine Story enthält immer Informationen, die nicht direkt etwas mit dem System zu tun haben und auch nicht für dieses relevant sind. Allerdings werden diese Informationen benötigt, um den Inhalt der Story besser zu verstehen oder eine authentische Geschichte zu haben, die sich gut merken lässt. In der oben beschriebenen Story für das Smart Home von Lina zum automatischen Öffnen der Eingangstür ist beispielsweise die Information, dass Lina zur Bäckerei fährt und die Arme und Hände mit Kuchen beladen hat, wenn sie vor der Tür steht, irrelevant. Ausreichend wäre die Information, dass eine registrierte Person die Arme nicht zum Öffnen und Schließen der Tür verwenden kann. Die Story lässt sich allerdings besser merken, da sie Bilder im Kopf der Zuhörenden erzeugt.

Es kommt zu einem sogenannten „medialen Kollateralinhalt“, also einem Inhalt, der entsteht und nicht gebraucht wird, aber in Kauf genommen wird, um den relevanten Inhalt erfolgreich zu vermitteln. Denken Sie darüber nach, wie groß dieser „irrelevante Teil“ sein muss, um den Kern des relevanten Inhalts zu erzählen, und gestalten Sie ihn möglichst minimal.

### 17.1.4 Gute Geschichten für eine gute Vermittlung

Storytelling ist eine sehr effektive Möglichkeit, um Wissen zu vermitteln. Da die Entwicklung einer guten Story aufwendig ist, sollten Sie sich der Vor- und Nachteile des Storytellings bewusst sein.



- **Storys erzeugen mehr Aufmerksamkeit als sachliche Vorträge**
- **Storys entspannen die Situation, da wir uns unbewusst daran erinnern, wie uns als Kind vorgelesen wurde. Diese positive Erinnerung beruhigt uns.**
- **Durch Storys können Anforderungen besser vermittelt werden, denn sie helfen komplexe Sachverhalte auf einfachere Ebenen zu transferieren. Storys erzeugen Bilder in unseren Köpfen. Durch Bilder können wir Sachverhalte besser verstehen.**
- **Storys wecken Emotionen und sorgen dafür, dass Inhalte länger im Kopf bleiben und weitervermittelt werden.**
- **Die Zuhörenden benötigen keine speziellen Kenntnisse, um einer Story folgen zu können.**



- **Das Entwickeln geeigneter Storys ist aufwendig.**
- **Die erzählende Person muss authentisch wirken und gut kommunizieren können.**
- **Storys sind schwer wartbar und somit zur Konservierung von Wissen weniger geeignet.**
- **Storys beinhalten immer Informationen, die inkonsistent sind (zu anderen Storys) oder irrelevant sind (und nur der Erzählung dienen).**

Als Requirements-Engineer sollten Sie wissen, wie man eine gute Story aufbaut und wie man diese wirkungsvoll erzählt. Stakeholder haben dieses Wissen meist nicht. Hier sollten Sie den Stakeholder durch geschickte Fragestellung oder Hinweise in die richtige Richtung lenken.

Falls Sie Storys nicht nur erzählen, sondern anderweitig präsentieren wollen, müssen Sie sich Gedanken über die Dokumentation bzw. Konservierung von Storys machen. Hier eignen sich Videos deutlich besser als die Verschriftlichung der Geschichten, da in Videos Mimik und Gestik der Erzählenden zu sehen sind. Detaillierte Informationen, wie man Storys mittels Videos dokumentiert und was dabei zu beachten ist, erhalten Sie in Kapitel 27 „Videos im RE“.

Für die Wartung eignen sich Anforderungen oder Abläufe ohne Rahmenstory allerdings weit besser. Mehr dazu finden Sie in Kapitel 16 „Wegweiser: Anforderungen dokumentieren“.

und vermitteln“. Somit sollten Sie sich überlegen, ob es wichtiger ist, die Informationen erfolgreich zu vermitteln, oder diese auch zu verwalten.

## 17.2 User-Stories und Story Mapping

User-Stories beschreiben gewünschte Funktionalitäten bzw. Eigenschaften (siehe Kapitel 19 „Schablonen für Anforderungen und User-Stories“) eines Systems aus der Sicht derjenigen Person, welche die Funktionalität bzw. Eigenschaft benötigt. Dabei muss diese Person nicht unbedingt diejenige sein, welche das System letztendlich benutzen wird. Nehmen wir als Beispiel folgende User-Story:

*Als GesetzgeberIn möchte ich, dass die Daten über Wohnungszutritte nach x Wochen gelöscht werden, um die Daten von Personen zu schützen.*

Die Funktionalität aus diesem Beispiel wird nicht von den nutzenden Personen des Systems gefordert, sondern von jemand anderem, hier der dem Gesetzgeber oder der Gesetzgeberin.

In der agilen Welt ist die Vermittlung von Anforderungen mittels User-Stories weitverbreitet.

### Definition User-Story nach Cohn [Cohn10]

Eine User-Story beschreibt eine Funktionalität, die für den Kunden oder Benutzer eines Produkts oder Systems von Wert ist. Sie besteht aus der schriftlichen Beschreibung der Funktionalität, Gesprächen über die Funktionalität und Akzeptanzkriterien, die Details vermitteln und festlegen, wann eine Story vollständig umgesetzt ist.



Angelehnt an diese Definition wurde eine Satzschablone für die Formulierung von User-Stories entwickelt. Diese ist in Abschnitt 19.6 „Schnell und einfach zur User-Story“ beschrieben.

Entsprechend könnte eine User-Story lauten:

*Als BewohnerIn möchte ich eine Kameraüberwachung rund ums Haus, um während meiner Abwesenheit nachprüfen zu können, ob sich Personen auf meinem Grundstück befinden.*

Sie werden aber sicherlich an der einen oder anderen Stelle auf das Problem stoßen, dass Sie Ihre User-Story nicht mit dieser Schablone formulieren können oder das formulierte Ergebnis komisch klingt. In solchen Fällen kann es sein, dass für diese User-Story nicht alle Punkte aus der obigen Definition zutreffen. Nehmen Sie einfach ein paar Anpassungen vor. Das könnte zum Beispiel sein, das Template für User-Stories abzuändern oder gar ganz ohne Satzschablone zu formulieren.

### 17.2.1 Verschiedene Detaillierungsebenen von User-Story – von Epics bis zu detaillierten User-Stories

Die in der User-Story beschriebene Funktionalität bzw. Eigenschaft kann auf unterschiedlichen Verfeinerungsgraden liegen (siehe [Kapitel 3 „Requirements-Engineering im Überblick“](#)). So könnte eine User-Story zum Beispiel „Schlüssellosen Zugang zum Haus“ als Thema haben, oder aber eher detailliert die Funktionalität „Erkennen einer autorisierten Person“ spezifizieren. Für sehr grobe User-Stories wird manchmal der Begriff Epic verwendet. Allerdings gibt es keine konkreten Richtlinien, wann eine User-Story ein Epic ist und wann nicht. Deshalb sprechen wir im Weiteren nur von User-Stories.

Grobe User-Stories haben durchaus ihre Daseinsberechtigung, zum Beispiel um detaillierte Stories einordnen zu können oder um einen Überblick zu bekommen. Allerdings gibt es Gründe eine grobe User-Story in detailliertere zu zerlegen. Man spricht in diesem Fall vom Schneiden der User-Stories. Warum grobe User-Stories in feine zerlegt werden und wie weit man dabei geht, lesen Sie in [Kapitel 12 „Anforderungen analysieren“](#). Dort ist das Konzept für Anforderungen und User-Stories beschrieben. Zusätzlich finden Sie im Internet unter [www.sophist.de/re7/kapitel17](http://www.sophist.de/re7/kapitel17) weitere Tipps und Tricks zum Schneiden von User-Stories.

*Als BewohnerIn möchte ich einen schlüssellosen Gebäudezugang, um mit vollgepackten Händen das Gebäude betreten zu können.*

Diese User-Story ist sehr wahrscheinlich noch zu grob, um sie in einer Iteration umsetzen zu können, und für das Entwicklungsteam nur schwer abschätzbar. Dementsprechend sind nicht alle Kriterien des INVEST-Prinzips aus [Kapitel 3 „Requirements-Engineering im Überblick“](#) erfüllt. Auch können detaillierte User-Stories hier und da gegen Teile dieses Prinzips verstoßen. Zum Beispiel haben Sie möglicherweise User-Stories, die nicht komplett unabhängig von anderen User-Stories sind. Sie sollten auf jeden Fall darauf achten, dass eine User-Story testbar und schätzbar ist.

### 17.2.2 Vermitteln mit User-Stories

Die Vermittlung mit User-Stories funktioniert grob nach folgendem Schema:

- Formulieren der Anforderungen in User-Stories, Diskussion mit den empfangenden Personen der Anforderungen anhand der formulierten User-Story, und dann ein gemeinsames Zustimmen über die Inhalte der User-Story,

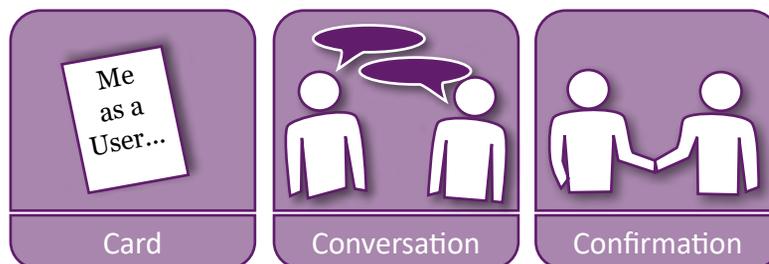


Abbildung 17.1: 3-C-Modell

Das Konzept wird üblicherweise als die 3 Cs – Card, Conversation und Confirmation – bezeichnet. Es beginnt mit der Karte, auf der die User-Story mit den Akzeptanzkriterien formuliert wird. Dann folgt das gemeinsame Gespräch über die User-Story. An dessen Ende sagen dann alle Parteien, dass sie die User-Story verstanden haben und diese so umgesetzt wird.

### 17.2.3 Formulieren einer User-Story

---

Prinzipiell sind Sie frei, was die Formulierung einer User-Story betrifft. Als Hilfsmittel für die Formulierung können Sie ein Template für User-Stories verwenden, welches im [Abschnitt 19.6 „Schnell und einfach zur User-Story“](#) beschrieben ist. Wie aber bereits geschildert, sollten Sie den Einsatz einer Schablone nicht als in Stein gemeißeltes Gesetz ansehen. Mit der Satzschablone formuliert, könnte die User-Story folgendermaßen lauten:

*Als BewohnerIn möchte ich, dass das Haus automatisch verriegelt wird, wenn sich niemand im Haus befindet, um sicherzugehen, dass das leere Haus immer verriegelt ist.*

Es hat sich in der Praxis bewährt, immer wieder über die Formulierung seiner User-Stories nachzudenken und entsprechend dem Feedback, das man beim Einsatz von User-Stories erhält, die Art der Formulierung anzupassen. Das Gleiche gilt auch für die Inhalte, die in einer User-Story beschrieben sind. Es kann sein, dass ein einzelner Satz für Ihre User-Story vollkommen ausreichend ist, sodass jeder weiß, was zu tun ist. Es kommt aber auch vor, dass in einer User-Story viel mehr beschrieben werden muss. Letzteres kann vor allem dann der Fall sein, wenn Sie als die Person mit dem Wissen nicht oft für Nachfragen erreichbar sind oder der Inhalt der User-Story etwas Neues, bisher Unbekanntes für die WissensempfängerInnen beschreibt.

Zu einer User-Story gehören Akzeptanzkriterien. Akzeptanzkriterien legen fest, wann eine User-Story vollständig umgesetzt ist. Sie beschreiben mit den Akzeptanzkriterien, was gegeben sein muss, damit Sie die User-Story als umgesetzt betrachten können. Auch für Akzeptanzkriterien gibt es keine feste Vorgabe, wie diese formuliert werden müssen, aber eine Schablone, die Sie nutzen können. Diese Schablone finden Sie in [Abschnitt 19.6.2 „Aufbau und Inhalt von Akzeptanzkriterien für User-Stories“](#). Sie sollten bei der Formulierung darauf achten, dass aus den Akzeptanzkriterien klar hervorgeht, wie und was Sie prüfen, um eine Aussage treffen zu können, ob die User-Story zu Ihrer Zufriedenheit umgesetzt wurde. Ein Akzeptanzkriterium für unsere User-Story könnte folgendermaßen lauten:

*Angenommen: Im Haus befindet sich eine Person*

*Wenn: Die Person verlässt das Haus*

*Dann: Die Tür ist verriegelt*

### 17.2.4 Das Gespräch zu einer User-Story

---

Eine wichtige Tätigkeit beim Einsatz von User-Stories ist das gemeinsame Gespräch zur Story. In diesem Gespräch werden die detaillierten Inhalte der User-Story vermittelt. Man sagt häufig, User-Stories sind ein Kommunikationsversprechen. Dementsprechend gehört das Sprechen über eine Story zu User-Stories dazu. Für dieses Gespräch ist es wichtig, dass Sie sich entsprechend vorbereiten.

Für die Vorbereitung sammeln Sie alles Material zusammen, welches Ihnen im Gespräch hilfreich sein kann. Das können Bilder, Diagramme, Prototypen und so weiter sein. Denken Sie daran, möglichst viele Sinne anzusprechen, also nicht nur den Hörsinn. Des Weiteren sollten Sie prüfen, ob Sie ausreichend Wissen zu einer User-Story haben, um alle möglichen Fragen beantworten zu können. Hierzu könnte das *REgelwerk* aus Kapitel 9 „Das *SOPHIST-REgelwerk*“ hilfreich sein. Anhand der Regeln können Sie für sich nachprüfen, ob Ihnen noch Informationen zu der User-Story fehlen, oder ob sie mögliche Fragen konkret genug beantworten können.

Für das Gespräch sollten Sie sich überlegen, ob die User-Story, über die gesprochen wird, eine vernünftige Größe hat. Damit ist der Umfang des Inhalts gemeint. Die User-Story „Die nutzende Person möchte schlüssellosen Zugang zum Haus haben“ bietet sehr viel Gesprächsstoff und ist daher für ein Gespräch, bei dem alle Beteiligten das Problem vollständig durchdringen sollen, eher ungeeignet. Da tun Sie sich mit einer User-Story der Größenordnung „Die Tür soll beim Entriegeln ein akustisches Signal abgeben“ viel leichter.

Für das Gespräch selbst ist es wichtig, gerade bei sehr detaillierten User-Stories, den Personen, die das Wissen empfangen, die Einordnung der User-Story in deren Kontext zu ermöglichen. Dazu haben Sie verschiedene Mittel. Sie könnten zum Beispiel mittels Storytelling oder mit einer Story Map die Einbettung der User-Story in deren Kontext darstellen. Achten Sie auch auf ausreichende und sinnvolle Visualisierungen während des Gesprächs und halten Sie dafür die entsprechenden Mittel wie Flipcharts, Whiteboards etc. parat.

Wie Sie sehen, ist beim Einsatz von User-Stories das gemeinsame Gespräch ein zentraler Punkt. Das kann allerdings zu einem Problem führen, da das Wissen, welches sich in einem Gespräch verbreitet, schnell auch wieder vergessen wird (vergleichen Sie hierzu die Vergessenskurve in Kapitel 1 „In medias RE“). Dementsprechend ist diese Technik nicht so gut geeignet, um Wissen über einen längeren Zeitraum zu bewahren. Üblicherweise werden nur die User-Stories gemeinsam besprochen, die relativ zeitnah realisiert werden sollen. User-Stories, die erst in ferner Zukunft realisiert werden, werden nicht im Detail besprochen, sondern höchstens mal kurz angerissen, um einen Überblick zu geben, was alles noch kommen wird. Dem Problem des Vergessens können Sie entgegenwirken, indem Sie einerseits mehr Informationen in der User-Story schriftlich festhalten oder andererseits eine zusätzliche Dokumentation parallel zu den User-Stories anlegen (siehe Kapitel 16 „Wegweiser: Anforderungen dokumentieren und vermitteln“). Gerade die erstgenannte Variante ist interessant, wenn Sie als ÜbermittlerIn der Anforderungen für Nachfragen bei der Umsetzung eher sporadisch zur Verfügung stehen. Die

zweite Variante – also eine zusätzliche Dokumentation – ist dann ratsam, wenn das vermittelte Wissen auch lange nach der Umsetzung zur Verfügung stehen muss.

### 17.2.5 Story Mapping – das Gesamtbild betrachten

Beim Arbeiten mit User-Stories kann man schnell den Überblick über das Gesamtbild verlieren. Mitunter können Ihre User-Stories nur einen ganz kleinen Teil des zu entwickelnden Systems betreffen und dadurch ist es sehr schwer, anhand der User-Story selbst den Zusammenhang mit anderen User-Stories zu erkennen.

Um den Überblick zu behalten, kann die Technik des Story Mappings [Patton14] verwendet werden. In einer Story Map werden die User-Stories nach speziellen Aspekten angeordnet. Zum Beispiel können die User-Stories entsprechend ihrer zugehörigen Epics sortiert werden. Oder die User-Stories werden nach einem darüberliegenden Prozess sortiert. Eine andere Variante ist, die User-Stories entsprechend einer Releaseplanung zu sortieren. Je nach Information, die man aus einer Story Map gewinnen möchte, wird die Anordnung der User-Stories in einer Story Map anders sein.

Wir wollen Ihnen hier ein Beispiel einer Story Map vorstellen, mit deren Hilfe die Einordnung einer User-Story in ihren Kontext gezeigt werden kann. Dafür werden anhand von Szenarien aus Benutzungssicht die essenziellen Grundfunktionalitäten des Systems festgelegt. Diese können in Form von Epics beschrieben sein. Diese Epics werden horizontal ganz oben in einer Story Map angeordnet und bilden das sogenannte Rückgrat der Story Map (engl. backbone). Unterhalb des Rückgrats werden die User-Stories angeordnet, die wichtig für eine Minimalversion des Systems sind. Diese sollten Sie, falls möglich, entsprechend ihrer Reihenfolge im Geschäftsprozess von links nach rechts anordnen. Unterhalb der User-Stories, die die Minimalversion darstellen, werden nun alle weiteren User-Stories entsprechend ihrer Zugehörigkeit zu der Minimalversion angeordnet. Dabei ist es üblich, User-Stories mit hoher Priorität weiter oben und Storys mit niedriger Priorität weiter unten anzuordnen.

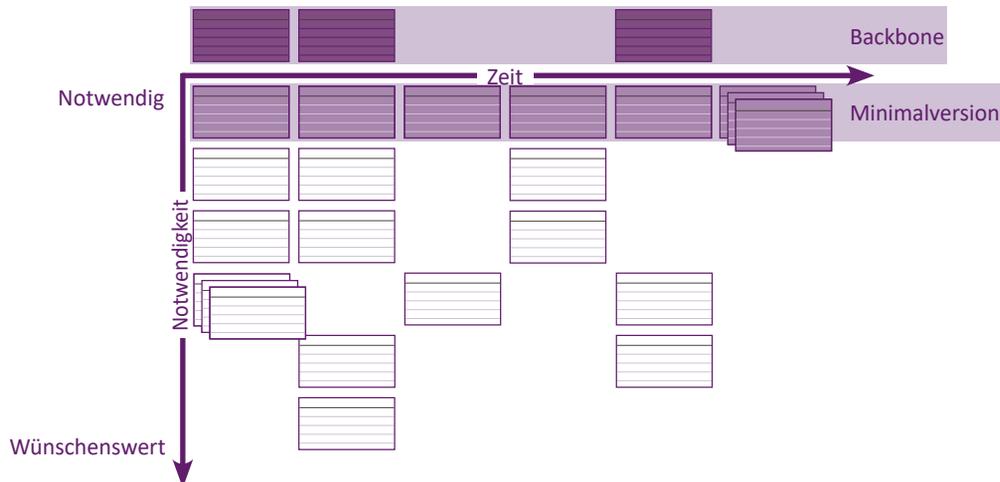


Abbildung 17.2: Grafik einer Story Map

Solch eine Story Map können Sie nutzen, um in einer Diskussion über eine User-Story den Beteiligten den Kontext dieser einzelnen User-Story zu vermitteln. Story Maps sind auch gut geeignet, um fehlende Anforderungen (hier in Form von User-Stories) aufzudecken bzw. um Anforderungen initial zu finden, und können daher auch für das Ermitteln von Anforderungen eingesetzt werden. Wie eine Story Map erstellt wird, können Sie auch in folgendem Video sehen:

### 17.2.6 Gute User-Stories für eine gute Vermittlung

---

User-Stories sind ein sehr effektives Mittel, um Wissen zu vermitteln. Da die Entwicklung aufwendig ist, sollten Sie sich der Vor- und Nachteile von User-Stories bewusst sein.



- **User-Stories entwickeln bei allen Beteiligten ein gemeinsames Verständnis zu den Anforderungen.**
- **Durch die nutzungsorientierte Beschreibung von User-Stories wird man sich der Notwendigkeit der Anforderungen bewusst.**
- **Durch das Schneiden der User-Stories nach dem INVEST-Prinzip erhalten diese eine für die Kommunikation angemessene Größe.**



- **User-Stories eignen sich nicht für die Aufbewahrung detaillierter Anforderungen.**

## 17.3 Prototypen – everybody's darling

---

Prototypen sind in vielen Bereichen der Systementwicklung einsetzbar, sie eignen sich zum Ermitteln von Anforderungen, zu deren Prüfung, aber vor allem auch zum Vermitteln. So vielfältig wie ihre Einsatzmöglichkeiten sind, so unterschiedlich sind auch ihre Ausprägungen. Wir stellen Ihnen hier die Arten vor, die wir für den Bereich der Wissensvermittlung für geeignet halten. Dabei fokussieren wir uns auf die Oberflächen-Prototypen.

### 17.3.1 Wireframe – das Drahtmodell für den Bildschirm

---

Ein Wireframe war in der Schneiderbranche eine sehr einfache Schneiderpuppe, die die Außenlinien des Körpers mittels Draht abbildete. In der Informatik spricht man von einem Wireframe, wenn es sich um einen sehr frühen Entwurf einer Benutzeroberfläche handelt, bei dem weder Gestaltung noch Details bezüglich der Funktion eine Rolle spielen. Der Fokus liegt auf der Anordnung von Elementen und der Benutzerführung. Alle

wichtigen Elemente des Endprodukts, zum Beispiel Eingabefelder oder Schaltflächen, sollten enthalten sein. Das Wireframe legt damit die Basis für die weitere Entwicklung und eignet sich für die Vermittlung der wichtigsten Elemente, die auf einer Oberfläche dargestellt werden. Wireframes werden nach ihrer Nutzung entweder Teil der Spezifikation oder sie werden verworfen.

Die folgenden beiden Arten von Wireframes werden eigenständig oder in Kombination verwendet:

- Statische Wireframes visualisieren jeweils eine einzelne Seite der Software-Benutzungsoberfläche schematisch. Sie zeigen die grundlegenden Elemente und den konzeptuellen Aufbau einer Seite.
- Dynamische Wireframes umfassen mehrere interaktiv verknüpfte Seiten, die eine Navigation von einer zur anderen Seite ermöglichen. Sie bilden somit eine einfache Form eines funktionalen Prototyps.

Da es bei einem Wireframe um die Struktur und Logik und nicht um das Design geht, genügen für die Darstellung rudimentäre grafische Elemente, was die Aufwände bei der Erstellung reduziert. Auch auf Farben wird in dieser Phase grundsätzlich verzichtet, aber durch die Verwendung unterschiedlicher Graustufen können Elemente bewusst hervorgehoben werden. Dadurch lässt sich eine gewünschte Wahrnehmungsreihenfolge aufbauen. Die einfachste Form eines Wireframes ist ein Papierprototyp oder ein Low-fidelity-prototyp, der mit einem Tool erstellt wird. Wireframes werden nach ihrer Nutzung für die Ermittlung oder Vermittlung nicht zum Endprodukt weiterentwickelt.

Durch die sehr vereinfachte Darstellung lassen sich Wireframes mit den entsprechenden Tools von fast jedem erstellen. So könnte in unserem Beispiel Ilona Neuhaus als Architektin, oder auch Lina ihre Vorstellung vom System mittels eines Wireframes visualisieren und vermitteln.

Neuen Nutzenden anlegen	
Name: <b>Seppi Schreiner</b>	
Status: <b>Gast</b>	PIN-Nummer: _ _ _ _ _ _
Zugangszeiten: <b>7 bis 18 Uhr</b>	Geben Sie eine 6-stellige
Gültig ab: <b>25.09.2020</b>	PIN-Nummer für den neuen
Gültig bis: <b>21.11.2020</b>	Gast ein
<input type="button" value="Bestätigen"/> <input type="button" value="Abbrechen"/>	

Abbildung 17.3: Wireframe

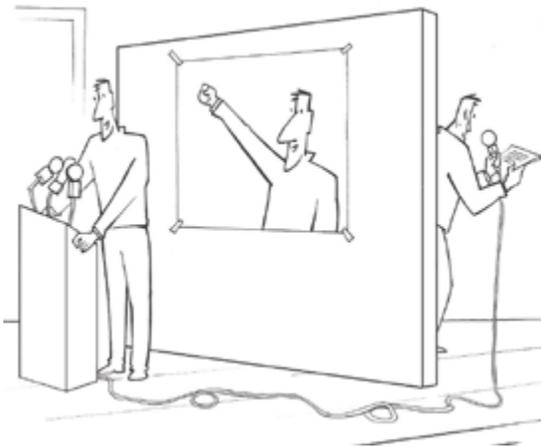
### 17.3.2 Funktionaler Prototyp – erlebte Funktion

---

Ein funktionaler Prototyp ist ein funktionsfähiges Modell, das im Entwicklungsprozess als Demonstrator eingesetzt wird. Funktionale Prototypen simulieren die Funktionalität des Systems und zeigen deren Ergebnisse an der Benutzungsschnittstelle. Sie sind eine Middle- bis High-fidelity-Abbildung des zukünftigen Systems, wobei das Aussehen dem Endprodukt nicht völlig gleichen muss, da hier der Schwerpunkt nicht auf dem Oberflächendesign liegt. Funktionale Prototypen ermöglichen es den Stakeholdern die Interaktion mit dem späteren System zu erleben, zu verstehen und zu kommentieren. Um den Entwicklungsaufwand zu minimieren, sind sie meist nicht mit dem Backend verbunden und liefern damit keine realen Daten, sondern arbeiten mit hinterlegten Testdatensätzen. Dennoch lässt sich das Systemverhalten bezüglich der Funktionalität erleben und beurteilen. Um Wissen über ein System zu vermitteln, eignen sich funktionale Prototypen hervorragend, da der Prototyp sehr viel Wissen in einer leicht aufnehmbaren Form darstellt. Funktionale Prototypen können iterativ zu realen Systemen weiterentwickelt werden.

### 17.3.3 Mock-up der Oberfläche – das Designmodell

---



Mock-ups der Oberflächen dienen zur Vermittlung der visuellen Seite Ihres Systems und produzieren dadurch Feedback. Ein Mock-up repräsentiert das Design einer Oberfläche. Es ist ein Middle- oder High-fidelity-Oberflächenprototyp. Mock-ups werden häufig auf der Grundlage von Wireframes erstellt und detaillieren diese. Stellen Sie sich ein Mock-up wie einen Screenshot der späteren Oberfläche des Produkts vor. In ihm werden unter anderem bereits die verwendete Typografie und das Farbschema abgebildet. Das Ziel ist es, so genau wie möglich das spätere Erscheinungsbild der zu erstellenden Oberfläche abzubilden. Mock-ups sind somit Bilder zur Wissensvermittlung und damit treffen die Aussagen aus Abschnitt 17.4 „Bilder zur Vermittlung von Wissen“ zu.

Ein Mock-up wird entweder mit einem Grafikprogramm oder einem spezialisierten Tool erstellt und der Erstellungsprozess ist viel aufwendiger als der eines Wireframes. Nach ihrer Verwendung zur Ermittlung oder Vermittlung von Wissen landen Mock-ups in der Spezifikation oder werden iterativ zur Oberfläche des Produkts erweitert.

### 17.3.4 Gute Prototypen für eine gute Vermittlung

Prototyping ist eine sehr effektive Möglichkeit, um Wissen zu vermitteln. Da die Entwicklung eines guten Prototyps aufwendig ist, sollten Sie sich der Vor- und Nachteile des Prototypings bewusst sein.

<ul style="list-style-type: none"> <li>■ <b>Keine Notationskenntnisse erforderlich</b></li> <li>■ <b>Gerade funktionale Prototypen regen zum Spielen an und damit zu einer intensiven Auseinandersetzung mit der Funktionalität.</b></li> <li>■ <b>Grafische Informationen (z. B. Aussehen einer Oberfläche) können besser als Bild vermittelt werden.</b></li> <li>■ <b>Üblicherweise universell in allen Sprachen einsetzbar</b></li> <li>■ <b>Gut mit anderen Vermittlungstechniken kombinierbar</b></li> </ul>	
<ul style="list-style-type: none"> <li>■ <b>Interpretierbar</b></li> <li>■ <b>Schlecht wartbar, da Einzelelemente eines Bildes schlecht attribuierbar/referenzierbar sind und damit Änderungen im Detail schlecht verfolgt werden können.</b></li> <li>■ <b>Schlecht erkennbar, welche Bestandteile des Prototyps Anforderungen sind und welche nicht</b></li> </ul>	

## 17.4 Bilder zur Vermittlung von Wissen

Ein Bild sagt mehr als tausend Worte .... aber wollen Sie das wirklich alles sagen? Genau vor diesem Dilemma stehen Sie, wenn Sie Bilder für die Vermittlung von Wissen verwenden.

Vielen Menschen fällt es leichter, Ideen und Wünsche in einem Bild festzuhalten als sie nur sprachlich zu explizieren – somit ist ein Bild für sie eine optimale Erzählhilfe. Zudem zeigen Forschungsergebnisse, dass Bilder wesentlich besser gemerkt werden können [Wolfe10], was gerade bei der Wissensvermittlung der entscheidende Faktor ist. Natürlich ist die Verwendung von Bildern eingeschränkt, da sich nicht jeder Aspekt gut in einem Bild ausdrücken lässt. Aber gerade, wenn es um räumliche Anordnungen, Anmutungen, also die Teile eines Systems, die man sehen und bildlich darstellen kann, geht, helfen Bilder enorm bei der Wissensvermittlung.

Wenn wir hier von Bild sprechen, dann meinen wir eine informelle visuelle Darstellung, also z. B. eine Zeichnung des Hauses auf Papier, um über die Positionierung der Überwachungskameras zu diskutieren. Es ist keine formale oder semiformale Darstellung wie ein Architekturmodell oder ein Ablauf, der in einem UML-Diagramm notiert wird.

Abbildung 17.4 zeigt die Frontseite des Hauses der Familie Meyer mit der groben Position der Haustür und des geplanten Vordachs. In der Abbildung stimmen weder die exakten Maße des Hauses noch sind die restlichen Elemente wie Fenster, Tür an der exakt richtigen Position. Dieser erste bildliche Entwurf eignet sich dennoch hervorragend als Grundlage für ein Gespräch über die Positionierung von Überwachungskameras (durch rote Punkte gekennzeichnet) und damit zur Vermittlung von Anforderungen und Ideen, was alles auf welche Art überwacht werden soll.

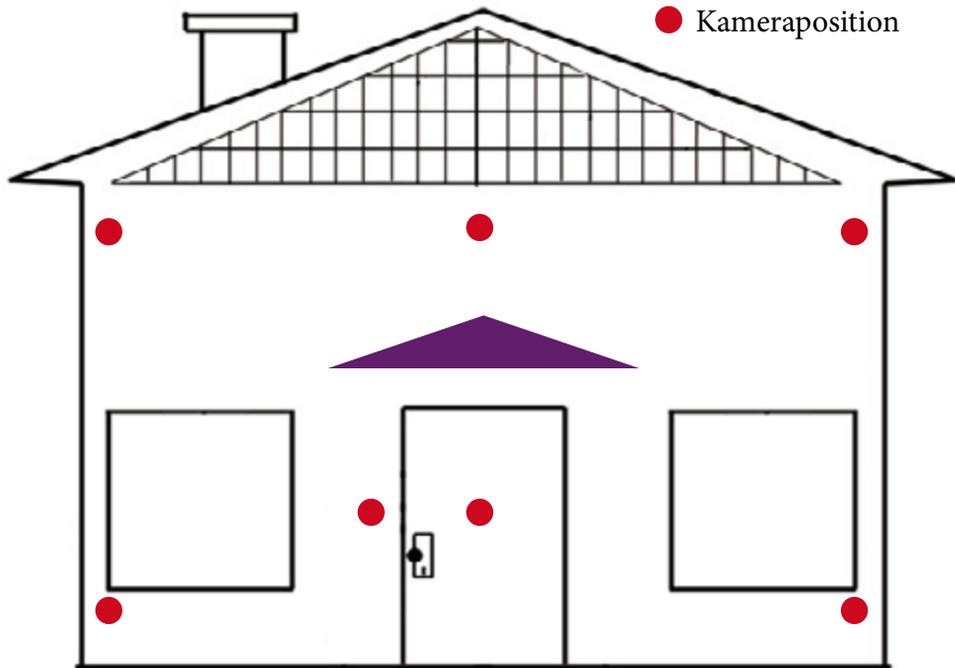


Abbildung 17.4: Die Frontseite des Hauses der Familie Meyer mit der Position der Haustür und des geplanten Vordachs sowie potenziellen Kamerapositionen

Ein Bild hat die Stärke, räumliche Zusammenhänge (also z. B. wo befindet sich die Kamera relativ zum Vordach, das die Aufnahme verhindern könnte?) darzustellen.

Zudem stellt ein derartiges Bild immer eine konkrete Situation dar, wohingegen ein semiformales oder formales Modell abstrakt ist. Das Bild zur Diskussion der Kameraposition zeigt ein ganz spezielles, konkretes Haus, über das diskutiert wird, denn es lässt sich kein generisches Haus zeichnen. Anhand dieses konkreten Beispiels lassen sich manche Fehler finden, die wir im Modell nicht finden (und natürlich auch umgekehrt).

### 17.4.1 Definition der eigenen Bildsprache

Bei informellen visuellen Darstellungen braucht keiner der beteiligten Personen Notationskenntnisse. Vielmehr erschafft man sich als Team beim Erstellen und in der Diskussion seine eigene Notation, die im Team verstanden wird – oder eben erklärt werden muss (z. B. erläutert die Legende, dass die roten Punkte die möglichen Positionen für die Anbringung einer Überwachungskamera darstellen).

Überlegen Sie sich bei einer informellen Darstellung immer, was selbstverständlich ist – also auch ohne Festlegung oder Erläuterung verstanden wird (hier z. B. dass das größere blaue Rechteck die Haustür darstellt). Erläutern oder definieren Sie, was nicht selbstverständlich ist (z. B. dass die roten Punkte die potenziellen Anbringungspunkte für eine Überwachungskamera darstellen). Dabei müssen Sie Ihr anvisiertes Zielpublikum einschätzen können, mit Rückfragen rechnen oder beim Erläutern Ihres Bildes an der Nasenspitze ablesen, welche Informationen Sie erfolgreich transportiert haben.

### 17.4.2 Verbindliches von nicht Verbindlichem trennen

Legen Sie bei einer informellen Darstellung fest, was in einem Bild als Anforderung verbindlich ist.

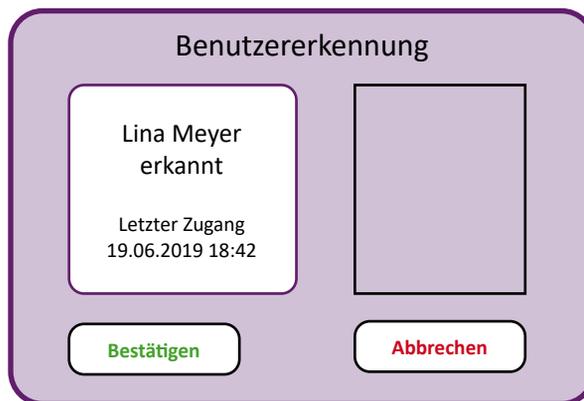


Abbildung 17.5: Beispielhafter Inhalt, der auf dem Zugangsterminal angezeigt wird

Wollen Sie mit der **Abbildung 17.5** festlegen, dass es zwei Schaltflächen gibt, die den Prozess bestätigen oder abbrechen? Oder sind damit die Begriffe Bestätigen und Abbrechen festgelegt? Ist damit geklärt, dass Bestätigen grün und Abbrechen rot sein muss? In genau diesem Farbton? Müssen die Schaltflächen genau diese Form und Größe haben? Müssen sie sich an genau dieser Position befinden? Sind die abgerundeten Ecken Pflicht? Muss Bestätigen immer links von Abbrechen stehen? Was steht bei „Letzter Zugang“, wenn die Person noch nie Zugang erlangte (nichts oder eine Info, dass die Person noch nie Zugang hatte)?

Ein Bild sagt mehr als tausend Worte – aber welche davon meinen Sie als Anforderung ernst?

Definieren Sie, ob das Bild

- farbverbindlich ist (Fordern Sie genau diesen Farbton?),
- positionsverbindlich ist (Muss das Element pixelgenau an dieser Stelle sein?),
- größenverbindlich ist (Stimmt die Größe des Objekts?),
- textverbindlich ist (Muss genau dieser Text dort stehen oder ist das ein Platzhalter? Bei der Überschrift Zugangsberechtigung könnten Sie den Text verbindlich meinen, beim Datum des letzten Zugangs hätten Sie vermutlich gerne immer das Datum des letzten Zugangs und nicht den 19.06.).

Diese Festlegungen sollten Sie mit Ihrem Bild kommunizieren, damit Ihre RezipientInnen von den gleichen Annahmen ausgehen. Dieses Vorgehen empfehlen wir Ihnen auch beim Einsatz von Prototypen, obgleich diese nicht ganz so informell wie Bilder sind. Nähere Informationen zu den Definitionen finden Sie unter [www.sophist.de/re7/kapitel17](http://www.sophist.de/re7/kapitel17). Ein Beispiel dazu finden Sie in [Abschnitt 13.5 „Steckbrief Anforderungen an die Benutzeroberfläche“](#).

### 17.4.3 Kombination Bild mit anderen Techniken der Wissensvermittlung

---

In der Praxis werden Bilder oftmals mit anderen Techniken der Wissensvermittlung kombiniert. Hier folgen ein paar Spielarten, die wir häufig antreffen.

Ein Bild dient als Erzählgrundlage. Die wissende Person malt, während sie ihr Wissen ausspricht, ein Bild und nutzt dieses, um ihre Erzählung zu strukturieren oder die Aufmerksamkeit der KollegInnen durch Zeigen auf bestimmte Bildbereiche zu lenken.

- Die wissende Person malt ein Bild und lässt sich von den RezipientInnen erklären, was sie dort an Informationen sehen. Falls die Informationen, die sie vermitteln will, noch nicht eindeutig vermittelt werden, bessert sie im Bild nach, bis die Information ankommt.
- Die wissende Person malt ein Bild, das einen ersten Eindruck vermittelt. Alle Informationen, die sie für wichtig hält, expliziert sie schriftlich und fügt diese dem Bild hinzu. Das Bild dient hier zur groben Orientierung und liefert den Kontext für die zugehörigen Anforderungen – hat aber den Charakter eines Kommentars, der juristisch nicht verbindlich ist.
- Die wissende Person nutzt sowohl ein Bild als auch Text, um ihre Anforderungen zu vermitteln. Achten Sie in diesem Fall darauf Redundanzen und Widersprüche

zu vermeiden. Aspekte, die besser im Bild zu vermitteln sind (z. B. Positionierung relativ zueinander) sollten dort dargestellt werden. Aspekte wie geforderte Schriftgröße/Schriftart sollten lieber textuell gefordert werden, auch wenn Schriften im Bild sichtbar sind.

#### 17.4.4 Bilder für eine gute Vermittlung

Da sehr viele Menschen visuell veranlagt sind und Bilder einen guten Wiedererkennungswert besitzen und gut memoriert werden können, sind sie ein Mittel der Wissensvermittlung, das häufig eingesetzt wird. Bedenken Sie vor der Verwendung allerdings die Vor- und Nachteile dieser Technik.

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>■ Keine Notationskenntnisse erforderlich</li> <li>■ Bestimmte Informationen können besser als Bild vermittelt werden.</li> <li>■ Üblicherweise universell in allen Sprachen einsetzbar</li> <li>■ Gut mit anderen Vermittlungstechniken kombinierbar</li> </ul>   |  |
| <ul style="list-style-type: none"> <li>■ Interpretierbar</li> <li>■ Schlecht wartbar, da Einzelelemente eines Bildes schlecht attribuierbar/referenzierbar sind und damit Änderungen im Detail schlecht verfolgt werden können</li> <li>■ Schlecht erkennbar, welche Bestandteile des Bildes verbindliche Anforderungen darstellen und welche nicht</li> </ul> |   |

## 17.5 Gemeinsam Artefakte erstellen

Eine weitere Möglichkeit Anforderungen zu vermitteln ist das gemeinsame Erstellen zusätzlicher Artefakte. Besonders gut geeignet ist dabei das gemeinsame Erstellen von Testfällen zu den Anforderungen. Es ist aber auch denkbar, dass andere Artefakte, zum Beispiel Architektur- oder Designdokumente oder Bedienungsanleitungen, erstellt werden. Gerade wenn Sie im Systems-Engineering tätig sind, könnte die Erstellung von Architekturdokumenten sinnvoll sein (siehe Kapitel 23 „Systems-Engineering“). Suchen Sie sich Artefakte aus, die sowieso erstellt werden müssen, um nicht am Ende das erstellte Artefakt wieder wegzuerwerfen.



### 17.5.1 Vorbereitung

---

Bevor Sie mit dem gemeinsamen Erstellen der Artefakte beginnen, sollten Sie ein paar Dinge festlegen. Zunächst sollten Sie sich überlegen, welche Artefakte erstellt werden sollen. Dafür kann es auch sinnvoll sein sich die Meinungen anderer Personen darüber, welche Artefakte geeignet sind, einzuholen. Wir werden in diesem Kapitel speziell auf das Erstellen von Testfällen eingehen, daher sprechen wir ab jetzt nur noch von diesen.

Weiterhin sollten Sie den inhaltlichen Umfang festlegen. Das bedeutet: Zu welchem Thema wollen Sie die Testfälle erstellen? Das kann zum Beispiel die Auswahl ein oder mehrerer User-Stories sein. Es ist ratsam den Umfang nicht zu groß zu wählen, da Sie sonst beim Erstellen der Testfälle leicht den Überblick verlieren. Planen Sie lieber ein paar mehr aber kürzere Termine ein.

Sammeln Sie alle Informationen, die Sie zu dem festgelegten Scope haben, denn Sie sollten darauf vorbereitet sein, alle möglichen Fragen gestellt zu bekommen. Überlegen Sie sich dazu auch, welche Visualisierungsmöglichkeiten Sie nutzen können. Gerade wenn der Scope etwas größer ist und die Arbeitssession dementsprechend länger dauern kann, ist es hilfreich Informationen zu visualisieren, um sie für die Zeit des gemeinsamen Arbeitens präsent zu halten.

Letztendlich sollten Sie sich noch bewusst machen, wer die Beteiligten der Arbeitssession sein sollen. Das heißt: Wer soll eigentlich die Informationen zu den Anforderungen erhalten? Bedenken Sie hierbei, dass vor allem die Personen, die aktiv mitarbeiten, am meisten von der Vermittlung der Anforderungen profitieren. Dementsprechend sind größere Gruppen weniger geeignet, wenn alle Beteiligten viel detailliertes Wissen erhalten sollen.

### 17.5.2 Überblick geben

---



Sobald die Vorbereitungen abgeschlossen sind, setzen sich alle Beteiligten zusammen, um gemeinsam die Testfälle zu erstellen. Es empfiehlt sich hierbei, an einem Ort zusammenzukommen. Sicherlich ist dies nicht immer möglich und Sie werden auf andere Mittel wie Webmeetings ausweichen müssen. Die Erfahrung hat aber gezeigt, dass das Nebeneinandersitzen am besten geeignet ist.

Zunächst gibt die vermittelnde Person einen Überblick über die Anforderungen, welche vermittelt werden sollen. Dabei sollen noch keine Details besprochen werden, sondern alle Beteiligten sollen erst mal den Gegenstand der

Vermittlung kennenlernen. Falls nötig, können Sie hierfür eine moderierende Person hinzuziehen, falls die Beteiligten immer wieder in Detaildiskussionen abzuschweifen drohen.

Hilfreich für das Verständnis und den Einstieg ist es zu diesem Zeitpunkt, den Beteiligten auch Hintergrundinformationen zu den Anforderungen zu liefern, wie zum Beispiel eine Begründung für die Anforderungen. Auch sollte zu diesem Zeitpunkt der Kontext, in dem man sich mit den Anforderungen bewegt (nicht zu verwechseln mit dem Systemkontext), klar herausgestellt werden. Als Einstieg empfehlen wir Ihnen Storytelling (siehe [Abschnitt 17.1 „Storytelling“](#)) zu verwenden. Unterstützend sollten Sie von diversen Visualisierungsmöglichkeiten Gebrauch machen. Das kann von einfachen Stichpunktlisten über kleine Zeichnungen bis hin zu skizzierten Aktivitätsdiagrammen reichen.

### 17.5.3 Erstellen der Testfälle

---

Nachdem die Beteiligten einen Überblick erhalten haben, wird gemeinsam der erste Testfall erstellt. Dazu sollten Sie zunächst den Normalfall betrachten, zum Beispiel: Eine autorisierte Person kommt zur Wohnungstür und diese entriegelt sich automatisch. Der erste Testfall behandelt den Fall, dass die Person autorisiert ist, dass die Tür verriegelt ist und entriegelt werden kann etc. Es werden also keine Ausnahmen oder Sonderfälle betrachtet.

Nachdem dieser Testfall beschrieben ist, werden Schritt für Schritt die Ausnahmen und Sonderfälle betrachtet. Hierzu kann das *REgelwerk* in [Kapitel 9 „Das SOPHIST-REgelwerk“](#) eine gute Hilfe sein, um herauszufinden, ob sämtliche Fälle berücksichtigt worden sind. Hierbei sollten Sie konkrete Beispiele nutzen. Anstatt den Fall zu betrachten, dass Teile vom Gesicht der Person bedeckt sind, sollten Sie beispielsweise besser darüber sprechen, was in den Fällen passieren soll, wenn die Person einen Schal oder eine Mütze trägt. Diese konkreten Beispiele erleichtern das Auffinden sämtlicher Sonderfälle gegenüber einer abstrakten Betrachtung des Sachverhalts.

Wer im Einzelnen die Testfälle schreibt, ist egal. Das kann die testende Person sein, da es sich um ihre Domäne handelt, es ist aber auch denkbar, dass eine der anderen beteiligten Personen das macht. Überlegen Sie sich, ob Sie nicht einen Wechsel der schreibenden Person einführen können, sodass jeder in den Genuss kommt Testfälle zu schreiben.

Für die Formulierung der Testfälle bieten sich verschiedene Techniken an. Hier eignen sich das Given-When-Then-Template (siehe [Abschnitt 19.6.2 „Aufbau und Inhalt von Akzeptanzkriterien für User-Stories“](#)) oder aber auch andere Formulierungstemplates für Testfälle. Sie sollten auf jeden Fall darauf achten, dass eine Formulierung gewählt wird, die für alle Beteiligten verständlich ist. Denn alle Beteiligten sollen sich am Ende einig sein, was die gewünschten Anforderungen konkret sind. Weitere Informationen zum Schreiben von Testfällen finden Sie in [Abschnitt 14.5 „Testfälle“](#).

### 17.5.4 Gemeinsam erstellte Artefakte für eine gute Vermittlung

---

Die gemeinsame Erstellung von Artefakten ist eine sehr effektive Möglichkeit, um Wissen zu vermitteln. Da die Entwicklung der Artefakte aufwendig ist, sollten Sie sich der Vor- und Nachteile bewusst sein.



- **Gemeinsames Bearbeiten der Anforderungen stärkt das gemeinsame Verständnis.**
- **Es werden Artefakte erstellt, die für weitere Arbeiten benötigt werden.**
- **Durch das gemeinsame Arbeiten erhöhen sich die Chancen, dass alle Sonderfälle berücksichtigt wurden.**



- **Es wird vorausgesetzt, dass alle Beteiligten gemeinsam arbeiten können.**
- **In manchen Szenarien zu aufwendig und schießt über das Ziel hinaus (zum Beispiel Szenario 1 „Kundenanfrage bearbeiten“); möglicherweise ist dazu ein anderes Zielartefakt hilfreich.**