



Design Matters: Design Thinking in der Entwicklung für bessere Usability

Katarina Poplawski, itelligence

Das Design ist ein wichtiger Teil jedes Produkts, dies gilt ebenso für Software-Lösungen. Das Design der Benutzeroberfläche und damit die User Experience im Bereich von Geschäftsanwendungen werden oft vernachlässigt. Dabei kann es als Alleinstellungsmerkmal oder als Entscheidungskriterium agieren.

Während das Interface-Design für Anwendungen im Privatkunden-Segment immer weiter und rasant verbessert wird, stockt dessen Weiterentwicklung im Bereich von Geschäftsanwendungen enorm und der Unterschied wird zunehmend deutlicher. Um die beschrie-

bene negative Entwicklung zu stoppen oder eine bestehende Anwendung benutzerfreundlicher zu gestalten, kann die Lösung der Einsatz des Design Thinking im Rahmen des Entwicklungsprozesses sein.

Die Realität

Viele Menschen müssen täglich die Herausforderung veralteter Geschäftsanwendungen bewältigen. Diese frustrieren den Endnutzer, denn die Entwicklerteams schenken der Bedienerfreundlichkeit ihrer Programmoberflächen immer noch zu wenig bis keine Beachtung. Sie entwickeln ihre Geschäftsanwendungen für einen IT-Spezialisten, der eine intuitive und einfache Usability nicht als Hauptkriterium ansieht. So muss der IT-Spezialist heute immer

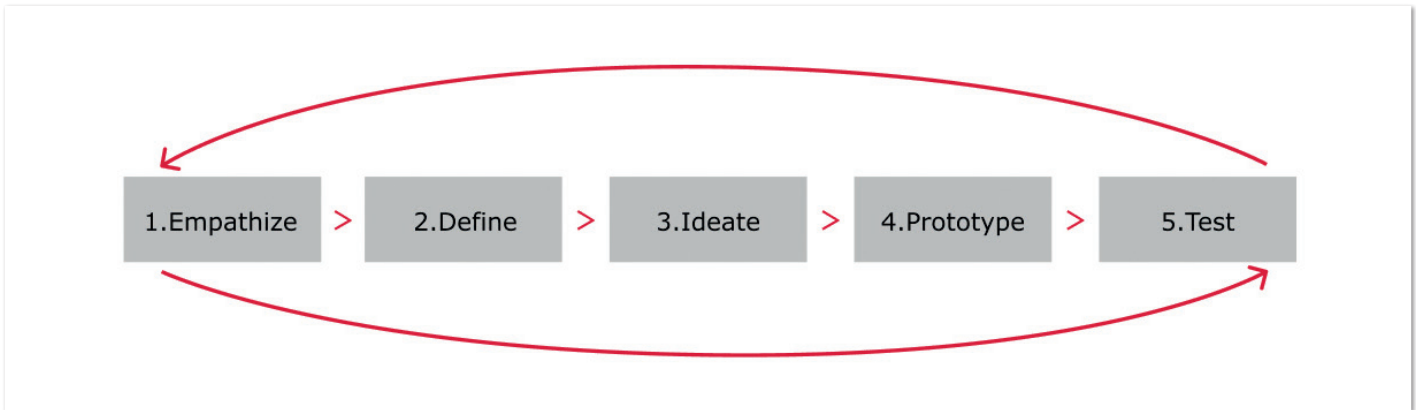


Abbildung 1: Design Thinking von d.school

noch mit Anwendungen arbeiten, deren UI-Oberfläche an die Relikte aus den 1990er-Jahren erinnert und deren Usability nach wie vor auf langen Listen, Form- und Tabellen-Views aufbaut. Egal ob ERP-, CRM- oder MRP-Anwendungen, diese erfordern zahlreiche, immer wiederkehrende manuelle Eingaben.

Der Endnutzer lernt mit der Zeit, mit der Anwendung zu arbeiten. Da viele Geschäftsanwendungen Prozess- und nicht Benutzer-orientiert gestaltet sind, ist deren Handhabung so kompliziert, dass eine oft mehrtägige Schulung unumgänglich ist. Dabei nutzt der gleiche „IT-Spezialist“-Endanwender privat sehr gerne eine Vielzahl unterschiedlicher Anwendungen, die sowohl nutzerfreundlich als auch einfach verständlich sind. Sie sollen nicht nur Spaß machen, sondern tatsächlich das Leben ihres Endanwenders bei der Lösung einer gestellten Aufgabe erleichtern.

In der Realität fragt sich der IT-Spezialist, warum seine Software-Tools so umständlich sind und ihm die Lösung seiner alltäglichen Arbeitsaufgaben erschweren. Aufgrund der privaten Erfahrungen steigen seine Anforderungen an Business-Anwendungen stetig, während die Endnutzer-Zufriedenheit zunehmend sinkt. In Wirklichkeit sind die Anforderungsunterschiede an die Usability und somit auch an das Interface-Design zwischen Geschäfts- und Privatanwendungen enorm.

Zwar nennen die meisten Softwarehersteller die Kunden- und/oder Endnutzer-Zufriedenheit bezüglich Usability als wichtige Zielstellung, in Wirklichkeit geht es in der Business-Anwendungswelt jedoch vor allem um schnelle Lösungen; getreu dem Motto „Hauptsache es funktioniert“ werden altbewährte Lösungen weiterentwickelt. Die Innovation findet zumeist im Backend der Anwendung statt und kommt somit nicht beim Endnutzer an. So scheitern die guten Vorsätze schlussendlich an den Entwicklern, die oft selbstständig nach bestem Wissen und Gewissen die Anwendungen weiterentwickeln. Es gibt eine mögliche Lösung des Problems: Die Applikationen mithilfe des Design Thinking zu realisieren.

Die Theorie

Es gibt keine einheitliche Definition für das Design Thinking, da es vielseitig eingesetzt und somit je nach Anwendungsgebiet auch unterschiedlich beschrieben wird. Für einige ist es ein Prozess, für andere wiederum eine Methode. Folgende zwei Definitionen beschreiben gemeinsam das Design Thinking als einen Entwicklungsprozess am besten:

- Design Thinking ist eine kreative und kollaborative Problemlösungsmethode aus der Nutzerperspektive.
- Design Thinking ist ein Innovationsprozess, der die Kunden und deren Bedürfnisse in den Mittelpunkt setzt

Folgende Grundeigenschaften von Design Thinking lassen sich anhand der genannten Definitionen ableiten, die im Rahmen der Entwicklung einer Softwareanwendung zu beachten sind. Der Design-Thinking-Prozess ist:

- **Nutzerzentriert**
Die Wünsche des Endanwenders stehen immer im Mittelpunkt des Prozesses
- **Ein Teamprozess**
Basierend auf dem Wissen eines Teams von Fachexperten aus unterschiedlichen Bereichen
- **Ein Lernprozess**
Bei jedem Schritt des Prozesses werden Erkenntnisse gewonnen und Neues erlernt, im nächsten Schritt wird das neuerworbene Wissen vom Team weiterverarbeitet
- **Iterativ und agil**
Die einzelnen Schritte sind keine Meilensteine wie bei anderen agilen Prozessen (etwa Scrum). Somit kann man die Reihenfolge der Schritte ändern und wiederholen, solange das gewünschte Ergebnis nicht erzielt ist

Es gibt auch kein einheitliches Prozess-Schema. Die zwei grundlegenden Prozess-Schemata wurden von zwei Bildungseinrichtungen entwickelt, die sich komplett dem Design Thinking widmen: der d.school an der Stanford University und dem Hasso-Plattner-Institut in Potsdam. Beide Bildungseinrichtungen wurden vom deutschen Unternehmer Hasso Plattner, einem der SAP-Gründer, gestiftet. Die d.school untergliedert den Prozess in fünf Schritte (siehe Abbildung 1), das Hasso-Plattner-Institut sieht sechs Schritte (siehe Abbildung 2).

Im Grunde splittet das Hasso-Plattner-Institut den ersten Schritt „Empathize“ von d.school in zwei Schritte auf, und zwar in „Verstehen“ und „Beobachten“. Wie bereits erwähnt, ist Design Thinking ein dynamischer und iterativer Prozess, dabei kann sich die Reihenfolge der einzelnen Schritte ändern oder einige Schritte können mehrmals wiederholt werden. So haben manche Unternehmen, wie beispielsweise IBM oder Google, den Prozess an die individuellen Gegebenheiten angepasst.

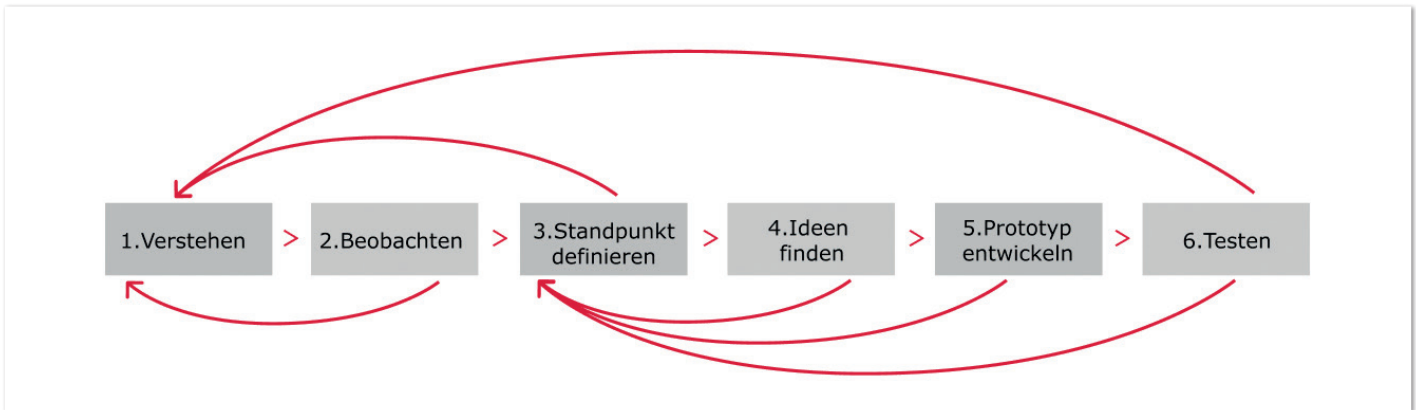


Abbildung 2: Design Thinking vom Hasso-Plattner-Institut

Wichtig ist also zu verstehen, dass, wenn der Prozess zur Entwicklung einer Softwareapplikation angewandt wird, dieser nur in Teamarbeit funktioniert und nur wenn von Anfang an der Endnutzer und seine Bedürfnisse in den Mittelpunkt des Prozesses gestellt werden. Im Grunde lehrt das Design Thinking, wie ein (Produkt-)Designer zu denken, so nach einer Lösung zu suchen und diese zu realisieren.

Die Praxis

Wie lässt sich das Design Thinking auf die Entwicklung einer Geschäftssoftware anwenden? Im ersten Schritt „Verstehen“ versucht das Team, das Problem zu erfassen und einzugrenzen. Was ist das Problem und wie wird es aktuell vom Endnutzer gelöst? Diese Frage ist im ersten Schritt zu beantworten. Dafür müssen die Bedürfnisse des Nutzers erkannt und verstanden werden. Dabei kommen unterschiedliche Methoden zum Einsatz, wie beispielsweise Personas erstellen und davon die möglichen Nutzerszenarien ableiten.

Mithilfe von Personas werden möglichst detaillierte Profile des Endanwenders ausgearbeitet, die mit den Profilen in den sozialen Netzwerken vergleichbar sind und eine Charakterisierung des Endanwenders zum Ziel haben. Relevante Fragen und Informationen, die bei der Ausarbeitung von Personas Anwendung finden, sind:

- Wie alt ist mein Endnutzer?
- Wie nutzt er normalerweise die Anwendung und in welchem Kontext?
- Wie kann die Aufgabe für ihn einfacher gestaltet werden?

Je detaillierter die Beschreibung, desto einfacher lassen sich die Endnutzer-Bedürfnisse erkennen und desto schneller wird die Empathie zum Endnutzer vom Team aufgebaut. Normalerweise wird eine Software-Anwendung zumeist von mehreren unterschiedlichen Anwendergruppen genutzt. Somit müssen mehrere Personas erstellt werden, da sich beispielsweise die Bedürfnisse eines Administrators stark von den Bedürfnissen eines einfachen Endanwenders unterscheiden. Die Profile von Personas werden im weiteren Verlauf des Design Thinking verfeinert und ausgebaut.

Beim zweiten Schritt „Beobachten“ werden die Annahmen aus dem ersten Schritt „Verstehen“ praktisch untersucht. Auf die folgenden Schlüsselfragen werden Antworten gesucht:

- Sind die Annahmen korrekt?
- Existiert das angenommene Problem tatsächlich?

Auch hier stehen dem Team viele unterschiedliche Methoden zur Verfügung. So kann sich das Entwicklerteam selbst in die Situation des angenommenen Nutzers versetzen oder Interviews mit den Endanwendern führen, um eine Lösung für das angenommene Problem zu entwickeln. Allgemein gilt: Je früher der Endanwender einbezogen wird, desto besser. Die Mitglieder des Teams müssen bewusst die Erfahrungen des Nutzers machen, um dessen Probleme schneller definieren zu können.

Eine effektive Methode ist die Beobachtung des Endnutzers bei der Lösung des Problems: Idealerweise lässt das Design-Thinker-Team sich vom Nutzer zeigen, wie er bisher vorgegangen ist. Diese Vorgehensweise funktioniert besonders gut bei existierenden Geschäftsanwendungen. Dabei ist es sehr wichtig, die Erkenntnisse zu dokumentieren und zu visualisieren. Die Antworten werden aufgezeichnet, das Besondere fotografiert oder der Anwender gefilmt. So kann der Endnutzer besser kennengelernt und Empathie aufgebaut werden. Im Idealfall werden zwei extrem unterschiedliche Nutzer und ihre Bedürfnisse miteinander verglichen, etwa die Bedürfnisse von Gelegenheits- und Power-Usern.

Anschließend werden im nächsten Schritt „Standpunkt definieren“ die aus der Nutzersicht gesammelten Erkenntnisse zum Problem analysiert, interpretiert und gewichtet. Durch die folgende Synthese lassen sich außerdem unentdeckte Nutzer-Bedürfnisse herausfinden, darauf basierend wird der Standpunkt definiert. Bei dem Schritt sollte das Projektteam eine gemeinsame Wissensbasis aufbauen, die Anforderungen und Wünsche müssen also von allen Teammitgliedern verstanden werden. Zum Festhalten von Bedürfnissen und der Aufgabe können folgende Methoden angewandt werden: Customer Journey, User Stories, Definition von Point of View etc.

Mit den gewonnenen Ergebnissen und dem definierten Standpunkt werden nun im nächsten Schritt die Ideen generiert. Aber wie generiert man Ideen? Auch in dieser Phase können je nach Aufgabe unterschiedliche Verfahren angewandt werden, um die Kreativität des Teams zu steigern und so eine Vielzahl von Ideen zu definieren. Fakt ist: Gute Ideen kommen selten im Büro am Arbeitsplatz. Kreative Firmen schaffen daher kreative Räume für ihre Mitarbeiter, die für die Laufzeit des Projekts dem Team zur Verfügung stehen. Außerdem sollten gezielt Methoden zur strukturierten Ideenfindung in der Teamarbeit angewandt werden.

Die beliebteste und die bekannteste Kreativitätstechnik zur Generierung von Ideen ist das Brainstorming, individuell oder im Team.

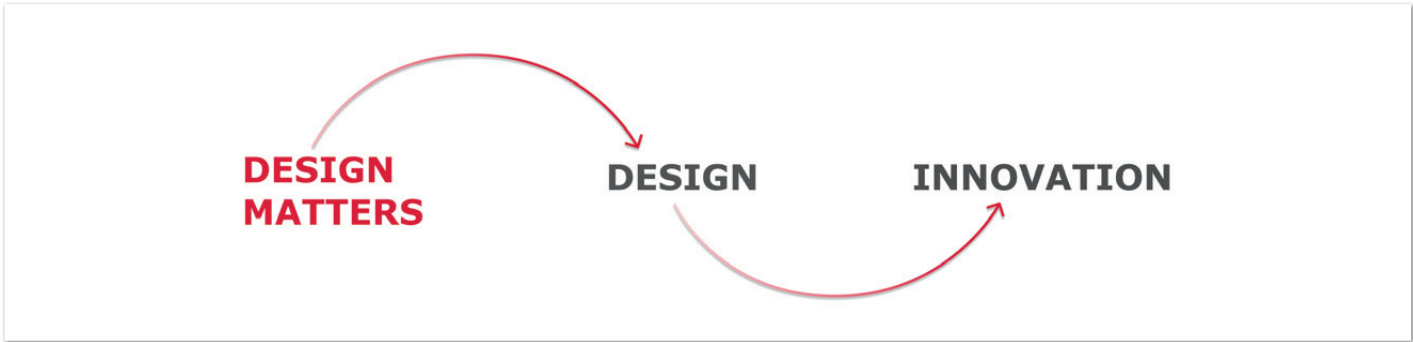


Abbildung 3: „Design Matters“ – erst durch das Design wird Innovation greifbar

Nachdem eine Vielzahl von Lösungsansätzen beim freien Brainstorming entstanden sind, kann beispielsweise ein gezieltes Brainstorming zur kritischen Funktionalität der Software-Anwendung durchgeführt werden. Anschließend erfolgen die Beurteilung und das Clustering der Ideen.

Oft werden Ideen mit Anforderungen vertauscht. Insbesondere bei technischen Produkten und auch Geschäftsanwendungen fällt es den Teams schwer, die Ideen von den Anforderungen zu unterscheiden. Dazu folgendes Beispiel: Die neue Geschäftsanwendung sollte als Web-Applikation entwickelt werden und daher responsiv sein. Bei dem aufgeführten Beispiel handelt es sich nicht um eine Idee, sondern um eine Anforderung. Die Idee wäre, ein Framework vorzuschlagen, das die Responsivität unterstützt, oder direkt einen Wireframe-Prototyp eines responsiven Layouts zu erstellen.

Die Ideen sollten nicht nur schriftlich festgehalten, sondern je früher, desto besser ebenfalls visualisiert werden, um die Kreativität zu steigern. Bei dieser Vorgehensweise folgen zumeist sofort weitere Ideen. Durch die Visualisierung werden auch für die Teamkollegen die Ideen greifbar gemacht.

Um die Kreativität des Teams zu fördern, bietet es sich an, für die Applikationen dreidimensionale Prototypen zu basteln. Aus diesem Grund basteln selbst die Design-Thinker-Teams der NASA, um ihre Ideen zu visualisieren. Im Allgemeinen ist die Visualisierung beim Design Thinking sehr wichtig, denn erst dadurch lassen sich ungewöhnliche Verbindungen, Muster und Lösungen erkennen. Eine gemeinsame Gedankenkette eines Teams kann nur durch das Mitteilen entstehen, durch die Visualisierung dokumentiert werden und so weiterwachsen.

diva^e

Java-Entwickler* für komplexe Digitalplattformen



Wer wir sind: Mit der geballten Erfahrung aus über 20 Jahren realisieren wir digitale Lösungen für namhafte Kunden. 600 Mitarbeiter in 10 Städten Deutschlands gestalten in unseren Büros, remote oder bei unseren Kunden die Zukunft des E-Business. Besonders stolz sind wir auf unsere Unternehmenskultur. Respekt, Vertrauen und Verantwortung werden in unseren Teams im täglichen Miteinander gelebt – und das über den Arbeitsplatz hinaus.

Was du bei uns machst:

- Du analysierst die technischen und fachlichen Anforderungen des Kunden hinsichtlich Machbarkeit und entwickelst mit deinen Kollegen komplexe Digitalplattformen.
- Du erstellst das Design der Java-Funktionalität und das Datenbank-Design und entwickelst deine Lösung iterativ von einem Minimal Valuable Product zu einem funktionsfähigen System. Dabei arbeitest du eng mit deinem Projektteam und unseren Kunden zusammen. Dein Know-how in Bezug auf Webtechnologien und dein Gespür für Website-Architektur und Usability helfen dir dabei. Klar, dass du dabei auch auf bestmögliche Systemkonfiguration und -optimierung achtest.
- In der Java-Entwicklung gehst du gerne neue Wege und stellst auch Altbewährtes infrage. Den dafür notwendigen Freiraum bieten wir dir gerne!







Wann du zu uns passt:

- Du bist ein echter Teamplayer. Einer für alle, alle für einen – das ist dein und unser Motto.
- Du kennst dich in der Webentwicklung aus und hast (idealerweise) bereits einige Jahre Erfahrung in der Konzeption und Entwicklung anspruchsvoller IT-Softwarelösungen mittels Java.
- Du fühlst dich wohl bei der Entwicklung mit Java, bist aber auch mit JavaScript, HTML5 und Spring vertraut. Zudem verfügst du über Know-how im Bereich Java Build-Tools wie Maven oder Gradle.
- Zu Projektstarts und bei Sprintwechseln auch vor Ort für wenige Tage bei unseren Kunden zu arbeiten, ist für dich kein Problem.
- Du bist kommunikationsstark, mindestens auf Deutsch.

Was du von uns erwarten kannst:

Als Mitarbeiter stehst du bei uns im Fokus. Unsere HR-Strategie ist auf die Bedürfnisse unserer Mitarbeiter ausgerichtet und umfasst die unterschiedlichsten Zielgruppen. Die Förderung des Teamworks steht dabei besonders im Vordergrund, wodurch eine familiäre Arbeitsatmosphäre entstanden ist und Kollegen sich gegenseitig voranbringen.

Wir bieten dir u.a.:

- | | | |
|--|--|--|
| 
Weiterbildung
satt | 
Viel
Freiraum | 
Forschungs-
freizeit |
| 
Reichlich
Perspektiven | 
Flexible
Arbeitszeiten | 
Umfangreiches
Fachwissen |

Und wir haben noch viel mehr Vorteile für dich.

Anschließend werden im vierten Schritt aus den Ideen Prototypen entwickelt, aus der Theorie wird Praxis, aus den Plänen wird Realität. Die besten Ideen werden vom Team ausgewählt und zu Beginn anhand von einfachen Prototypen dargestellt. Hier spielt natürlich der Zeit- und Kostenfaktor eine sehr große Rolle; je mehr Zeit für die Gestaltung von Prototypen zur Verfügung gestellt wird, desto aussagekräftiger wird im nächsten Schritt das Nutzerfeedback in der Testphase. Deswegen empfiehlt es sich, zu Beginn einfach, schnell und skizzenhaft die ersten Lösungsideen zu veranschaulichen. Dadurch lassen sich bereits erste Lösungsideen aussortieren und andere wiederum favorisieren.

Je einfacher der Prototyp, desto unkomplizierter lässt sich eine schlechte Idee verwerfen. Zu den einfachen Lo-Fi-Prototypen gehören Skizzen, Wireframes und Moodboards. Nachdem ein Lo-Fi-Prototyp steht, wird darauf basierend ein Hi-Fi-Prototyp erstellt. Je mehr Zeit dafür bleibt, desto realistischer der Prototyp und schlussendlich desto qualitativer das Nutzer-Feedback. Um den Pfad nicht zu verlieren, sollten hier bereits die echten Endanwender stark miteinbezogen werden, um detailliertes Feedback, noch vor der Testphase des Prototyps, liefern zu können.

Die Hi-Fi-Prototypen werden mittels grafikbasierter Software geschaffen, etwa Adobe Illustrator, Sketch oder das Online-UX-Tool Figma. Schlussendlich sollte im besten Fall ein interaktiver Prototyp entstehen, der nicht nur das Look-and-Feel der Anwendung abbildet und die User-Szenarios visualisiert, sondern auch die möglichst realistische Usability für die Tester greifbar macht.

Der finale Schritt beim Design Thinking ist das Testen. Ähnlich wie das Prototypen ist auch das Testen stark vom Zeit- und Kostenfaktor abhängig. Der finale Prototyp wird dem Endanwender präsentiert, die Ergebnisse validiert sowie reflektiert. Folglich werden darauf basierend entsprechende Anpassungen am Prototyp vorgenommen (Schritt 4 wird mit neuen Erkenntnissen erneut durchgeführt) und anschließend wieder getestet. Dabei besteht das Ziel des Testens darin, die Schwachstellen einer Idee anhand des Prototyps herauszufinden und zu korrigieren.

Neue Ideen können sowohl beim Testen als auch beim Prototypen entstehen, dann kehrt man zum vierten Schritt zurück und teilt die Idee dem Team mit. Nachfolgend wiederum kann die Idee in den Prototyp eingearbeitet und anschließend wieder getestet werden. Allgemein wird in der Testphase wertvolles qualitatives Feedback gesammelt, insbesondere, wenn dieses von den echten Endanwendern der zukünftigen Applikation kommt. Auch hierfür steht eine Vielzahl von Methoden und Tools zur Verfügung, wie A/B-Test, Feedback-Erfassungsraster, Online-Test- und Feedback-Web-Applikationen etc.

So eignet sich die A/B-Testmethode vor allem zum Testen des Interface-Designs oder der Usability. Dabei werden von gleichen Ansichten der Anwendung (UI-Views) zwei Varianten erzeugt und dem Tester vorgeschlagen. Anschließend wird die Bearbeitungszeit einer gestellten Aufgabe gemessen. Die Ansicht, in der der Tester schneller die Lösung findet, wird dann final implementiert, da sie höchstwahrscheinlich benutzerfreundlicher ist.

Für das Testen einer Geschäftsanwendung sollten webgestützte Tools angewandt werden; damit lassen sich etwa ganze User-Sto-

ries testen, dabei die Reaktionszeit messen sowie direkt das Tester-Feedback zu den einzelnen Anwendungsansichten sammeln. Da Design Thinking ein iterativer Prozess ist, wird im Idealfall der Prototyp immer weiter verfeinert, entsprechend dem Tester-Feedback. Der direkte Kontakt zum Nutzer ist unumgänglich, denn weder der Auftraggeber noch der Entwickler kennen die wahren Wünsche des Endanwenders:

- Wann haben Sie das letzte Mal gesehen, wie jemand Ihr Software-Produkt im Alltag benutzt?
- Haben Sie überhaupt schon den Endanwender bei der Nutzung Ihrer Anwendung beobachtet?

Als Ergebnis des Design Thinking entsteht eine Endnutzer-orientierte und validierte Idee in Form eines Anwendungs-Prototyps. Nachdem dieser fertiggestellt und getestet wurde, startet im Anschluss die Anwendungs-Implementierung.

Fazit

Design Thinking ist kein Allheilmittel und kein Garant für den Erfolg einer Geschäftsanwendung, jedoch ist die Wahrscheinlichkeit, im Rahmen eines Benutzer- und Design-orientierten Prozesses unter realer Einbeziehung des Endnutzers eine gute Lösung zu entwickeln, deutlich höher. Schon heute sollte das Ziel jedes Entwicklungsteams einer Geschäftsanwendung darin bestehen, die Unterschiede zwischen den Geschäftsanwendungen und den Applikationen des privaten Gebrauchs zu verringern.

„Design Matters“ – denn erst durch die benutzerfreundliche User-Experience und das moderne User-Interface-Design im Frontend der Anwendung wird die Innovation und Einzigartigkeit des Backends der Anwendung für Endnutzer greifbar und erlebbar.

„Design Matters“ – denn mithilfe des Design Thinking, also durch das gezielte Gestalten des Denkens im Team, können innovative und außergewöhnliche Ideen entstehen (siehe *Abbildung 3*). Deshalb immer an die Bedürfnisse der Endanwender denken und deren Leben einfacher machen.



Katarina Poplawski

katarina.poplawski@itelligence.de

Katarina Poplawski hat ihr Bachelor-Studium Medieninformatik an der Hochschule Hof absolviert, gefolgt von einem Master-Studium der Technischen Redaktion und Wissenskommunikation an der Hochschule Merseburg. Seit Januar 2017 ist sie als UX/UI-Designerin im Entwicklungsteam der itelligence in Dresden tätig.