

User Experience – eine Einführung

Alexej Schneider, MT AG



Die heutige Welt ist ohne moderne Technologien nicht mehr vorstellbar. Tagtäglich verwenden wir Computer, Tablets und Smartphones für die unterschiedlichsten Aufgaben. Sei es, um das Wetter zu checken, uns miteinander zu vernetzen oder um schnell mal ein neues Paar Schuhe zu kaufen. Anwendungen im Konsumentenmarkt zeichnen sich oft durch eine gute Bedienbarkeit aus, Business-Anwendungen dagegen eher weniger. Diese sind in der Regel zwar komplexer, doch man kann den Anwender mit entsprechenden Werkzeugen oft bei seiner Arbeit unterstützen.

Große Firmen wie Apple, Google, Facebook oder Amazon machen es uns schon seit vielen Jahren vor. Ihre Produkte sind häufig intuitiv bedienbar, hübsch anzusehen und machen in der Regel sogar Spaß. Für die Unternehmen ist die Hauptmotivation natürlich der Verkauf ihrer Produkte. Der Markt ist hart umkämpft und wenn die Produkte zu schlechten Erlebnissen führen, kann das zum Verlust des Vertrauens und zu einer schlechten Reputation führen. Positive Erlebnisse sorgen hingegen für loyale Kunden und gute Bewertungen. Demzufolge wird viel investiert, entwickelt, geforscht und getestet, um die Anwender zufriedenzustellen. Das Ganze nennt man „User Experience“.

Lets talk about UX

User Experience (UX, zu Deutsch wörtlich „Nutzererfahrung“, besser „Nutzererlebnis“) bezieht sich auf die persönliche Emotion eines Anwenders einer Anwendung, eines Systems oder Service. Die Königsdisziplin einer guten Experience ist, dass man den Nutzer für das Produkt begeistern kann. Dies muss sich in erster Linie nicht unbedingt auf die digitale Welt beziehen. Wir alle kennen gutes und schlechtes UX aus unserem Alltag. Hat man sich nicht auch mal geärgert, als man vor einer Tür stand und nicht auf Anhieb wusste, wie die

se zu öffnen ist? Soll man sie nun drücken, ziehen, drehen oder geht sie vielleicht automatisch auf? Wenn es nicht direkt klappt, so ärgert man sich und sucht anschließend nach einer Anleitung, fragt nach Hilfe oder man lässt es eventuell komplett bleiben.

Dieses Beispiel ist natürlich überspitzt, doch möchte man meinen, dass ein so altes Interaktionsobjekt mittlerweile keinen vor Herausforderungen stellen sollte. Bei unserer Business-Software verhält es sich ähnlich. Die Software soll in erster Linie eine bestimmte Aufgabe effektiv, effizient und zufriedenstellend erfüllen. Eine schlechte UX bringt dabei viele negative Aspekte mit sich:

- **Produktivität**
Wenn eine Anwendung schlecht zu benutzen ist, dauert die Arbeit länger; eventuell muss man sogar mehr Mitarbeiter für eine Aufgabe einsetzen. Insgesamt führt das zu höheren Kosten.
- **Schulungskosten/Einarbeitungszeit**
Mitarbeiter müssen geschult werden, wenn die Anwendung nicht intuitiv zu benutzen ist. Bei schlechter Software dauert es eine längere Zeit, bis die Mitarbeiter produktiv sind.
- **Supportkosten**
Ist die Nutzbarkeit schlecht, muss der Support oft helfen.

- **Entwicklungskosten**
Wenn man die Endanwender spät in die Entwicklung einbezieht (im schlimmsten Fall erst nach der Implementierung), können neue, unerwartete Anforderungen entstehen. Das kann ein komplettes Redesign notwendig machen.
- **Zufriedenheit**
Ein wichtiger Faktor ist auch die Moral der Anwender. Es ist frustrierend, wenn man seine Arbeit nicht sachgemäß durchführen kann. Dadurch kann auch die weitere Produktivität leiden.

Selbstverständlich sind Business-Anwendungen in der Regel komplexer. Gerade deswegen ist es jedoch essenziell wichtig, für eine gute UX zu sorgen. Wenn man anfängt, sich mit UX zu beschäftigen, kann es zunächst doch recht verwirrend sein. Es kommen viele neue Fachausdrücke hinzu, vieles ist auch eher subjektiver Natur und nicht alles ist klar definiert. UX besteht zudem aus zahlreichen Komponenten, die im Zusammenspiel miteinander agieren.

Oft wird UX fälschlicherweise nur mit einer hübschen Benutzeroberfläche (UI, User Interface) in Verbindung gebracht. Nur weil eine Anwendung ein schönes Design besitzt, bedeutet das allerdings nicht gleichzeitig, dass sie zu einem guten Erlebnis führt. Selbstverständlich dient ein gutes

Design der positiven Assoziation mit dem Produkt, es ist aber nur ein Teil von UX.

Eine weitere Komponente (wenn nicht sogar die wichtigste) ist die Nutzerfreundlichkeit („Usability“). Der Anwender des Produkts soll effektiv, effizient und möglichst zufrieden zum Ziel kommen. „Usability“ ist ein subjektiver Begriff. Was für einen Nutzer einfach ist, kann für einen anderen eine unüberwindbare Herausforderung sein. Daher ist es wichtig, während der Entwicklung seine Benutzer genau zu kennen und die Anwendung an deren Erfahrungen und Wünsche anzupassen. Oft spricht man dabei von „User-Centered Design“ (UCD).

In den 1990er-Jahren hat Jakob Nielsen „10 Usability Heuristics for User Interface Design“ [1] formuliert. Auch wenn sie vor mehr als zwanzig Jahren aufgestellt wurden, haben diese Prinzipien nicht an Bedeutung verloren (siehe Abbildung 1):

- **Feedback**
Das System soll den User stets darüber informieren, was gerade passiert.
- **Gemeinsame Sprache**
Das System sollte die Sprache des Benutzers sprechen.
- **Kontrolle und Freiheit**
User sollten Aktionen rückgängig machen oder neu durchführen können.
- **Konsistenz**
Benutzer erwarten gewisse Design-Standards (etwa Logout immer oben rechts).
- **Fehlermeldungen**
Fehlermeldungen sollten für den User verständlich sein.
- **Fehlervermeidung**
Hilfreiche Fehlermeldungen sind gut, Prävention ist besser (etwa durch Client-seitige Validierung).

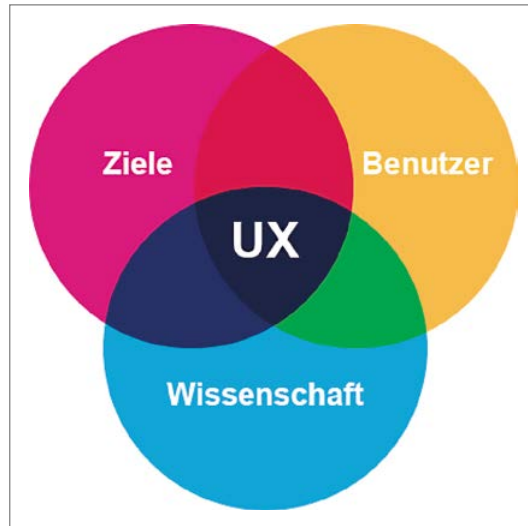


Abbildung 1: Zusammenspiel der Komponenten bei User Experience

- **Flexibilität und Effektivität**
Zusätzliche Funktionen sollen dem Benutzer die Arbeit erleichtern.
- **Erkennen statt Erinnern**
Alle notwendigen Informationen sollten sichtbar sein.
- **Minimalismus**
Design sollte auf das Notwendigste begrenzt werden.
- **Hilfe**
Ein gutes Design ist selbsterklärend, dennoch braucht der User oft Hilfe.

UX-Design

Der Artikel bezog sich bisher darauf, was User Experience bedeutet. Doch wie können wir eine solche nun für unsere Anwendungen umsetzen? Eine gute UX entsteht nicht von einem Tag auf den anderen. Zum Glück ist das jedoch kein neues Thema und es gibt diesbezüglich viele Quellen (Bücher, Blogs, Zeitschriften oder

Tools). Den Vorgang, um eine möglichst gute Erfahrung zu erschaffen, nennt man „User Experience Design“ (UXD).

Im Mittelpunkt jeder Entscheidung stehen die Endanwender. Dementsprechend ist es wichtig, sich mit diesen so früh wie möglich auseinanderzusetzen – allerdings während des kompletten Produkt-Lebenszyklus und nicht nur während der Entwicklung. Im UX-Jargon heißen die Endanwender oft „Persona“. Bei diesen handelt es sich um fiktionale Personen, die einem typischen Vertreter einer Zielgruppe entsprechen. Man fasst die relevanten Merkmale der Personae zusammen und gibt ihnen ein Gesicht. Eine Persona besitzt üblicherweise einen fiktiven Vor- und Nachnamen sowie ein Foto. Außerdem kommen weitere Punkte hinzu, die für das jeweilige Produkt relevant sind:

- Alter/Geschlecht/Herkunft
- Beruf/Ausbildung
- Computerkenntnisse
- Ziele/Wünsche/Erwartungen an das Produkt

	Ähnlichkeit zum Endprodukt	Kosten	Einsatzzweck	Eigenschaften
Skizze	---	+++++	Brainstorming	Schnell, einfach
Wireframe	+	++	Feedback	Das Skelett der Anwendung (ohne Design)
Mockup	++	--	Feedback	Das Skelett der Anwendung (mit Design)
Click-Dummy	++++	-----	User testing	Interaktiv, aber keine Funktionalität

Tabelle 1: Prototypen

Komplexe Interaktionskonzepte lassen sich nur schwer beschreiben. Von einer Idee zur fertigen Lösung ist es ein langer Weg.

Die Erwartungshaltungen an ein Produkt sind meistens unterschiedlicher Natur zwischen Entwickler, Kunden und Endanwender. Beim UX-Design bekommt das Prototyping daher eine wichtige Rolle. Das agile Vorgehen während des Prototyping unterstützt die Kommunikation und hilft der gemeinsamen Vision vom Endprodukt. Prototypen können dabei unterschiedliche Formen annehmen. Jede hat wiederum ihre Vor- und Nachteile sowie Einsatzzwecke (siehe Tabelle 1).

Das Testen nimmt bei UX eine wichtige Rolle ein. Dabei gibt es auch hierbei zahlreiche Varianten davon, wie und was man testen kann. Die gewonnenen Erkenntnisse können anschließend in eine bessere UX umgesetzt werden:

- **Experten-Review**
Ein UX-Experte untersucht die Anwendung auf Grundlage von Usability-Kriterien und persönlichen Erfahrungen.
- **Usability-Test**
Personen der jeweiligen Zielgruppe werden bei der Benutzung der Anwendung beobachtet.
- **Crowd-Testing**
Es gibt Anbieter, bei denen man seine Anwendungen von vielen Usern testen lassen kann.
- **Eye-Tracking**
Der Blickverlauf einer Person wird sichtbar gemacht; daraus ist zum Beispiel die Empfehlung entstanden, dass man Label rechtsbündig anordnet [2].
- **A/B-Test**
Unterschiedliche Nutzer bekommen unterschiedliche Varianten eines Produkts; anhand von Metriken wird bestimmt, welche Lösung besser geeignet ist.

UI-Design

Das User Interface ist die zentrale Schnittstelle zwischen Mensch und Computer. Eine gute Schnittstelle ist es dann, wenn der Anwender bei seiner Aufgabe unter-

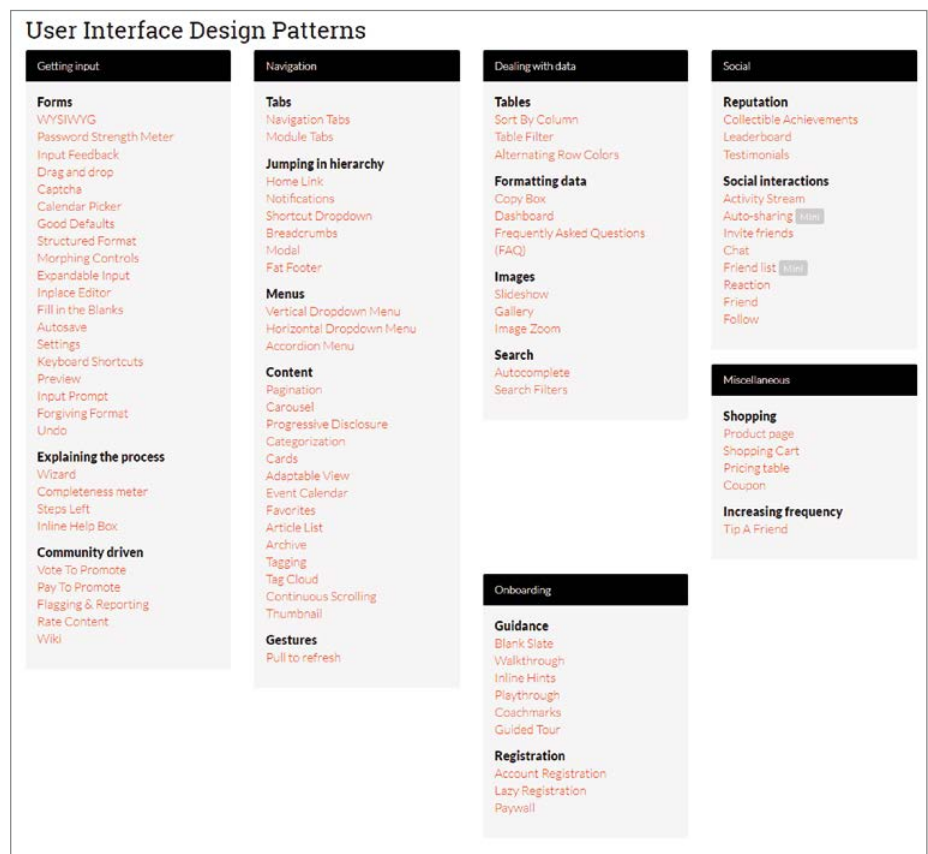


Abbildung 2: Eine Auswahl von UI-Design-Patterns [3]

stützt und ihm keine Steine in den Weg gelegt werden.

Moderne Oberflächen bestehen nicht mehr aus Kommandozeilen, sondern aus einer Vielzahl von Elementen (Fenstern, Formularen, Buttons, Reports, Charts etc.). In der Regel ist es die Aufgabe eines Designers, die zahlreichen Elemente zu einem Interface zu verbinden. Man muss dies allerdings nicht unbedingt studiert haben, um ein ansprechendes UI zu kreieren. Oft reicht es aus, sich von anderen Anwendungen inspirieren zu lassen, oder man verwendet Frameworks à la Bootstrap.

Design hat jedoch mehr mit Regeln und Prinzipien zu tun, als man vielleicht glauben mag. Viele Design-Trends sind auch durch UX-Studien entstanden. Menschen lesen Anwendungen beispielsweise nicht wie ein Buch von links nach rechts. Web-Anwendungen sind dabei ein klassisches Beispiel. Sie sind üblicherweise nach dem „F Pattern Layout“ aufgebaut. Demzufolge werden die wichtigsten Elemente oben links platziert (Logo, Navigation, Handlungsaufforderungen), anschließend die nächste Reihe und so weiter.

Das menschliche Auge sucht standardmäßig nach Mustern. Durch ein entspre-

chendes Design können wir den Benutzer durch die Anwendung führen. Dazu hat man unterschiedliche Design-Werkzeuge wie Positionierung, Größe, Farbe, Kontrast, Abstände oder die Wahl von Schriftarten zur Auswahl. Je dominanter ein Element ist, desto wahrscheinlicher ist es, dass das Auge dahin wandert.

UI-Design-Patterns

Beim Designen von neuen Interaktionselementen muss man nicht jedes Mal das Rad neu erfinden. Es gibt etablierte Best Practices, die man für seine Anwendungen nutzen kann; oft spricht man hier von „UI-Design-Patterns“. In der Programmierung gibt es ein ähnliches Konzept, die sogenannten „Entwurfsmuster“. Dies sind Lösungsschablonen zu wiederkehrenden Entwurfsproblemen.

Bei UI-Design-Patterns handelt es sich um Konzepte, die der User aus seinem alltäglichen Leben kennt. Durch ihren Einsatz muss der Anwender nicht mehr großartig nachdenken und kann im besten Falle die Anwendung intuitiv nutzen (siehe Abbildung 2).

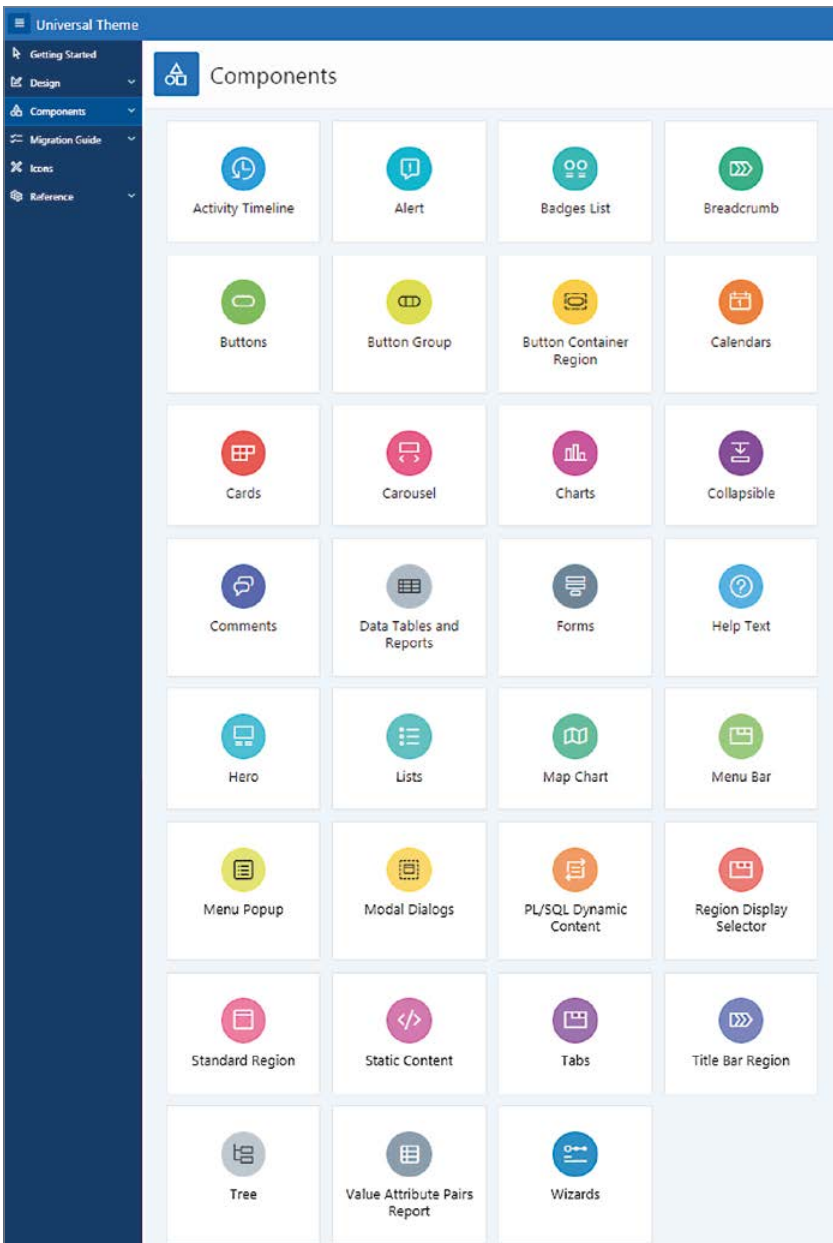


Abbildung 3: Apex-Komponenten [4]

UX am Beispiel von Apex

Oracle Application Express (Apex) ist eine webbasierte Softwareentwicklungs- und Laufzeit-Umgebung. Sie ist standardmäßig Bestandteil jeder Oracle-Datenbank-Edition, wodurch keine zusätzlichen Lizenzkosten entstehen. Bei Apex handelt es sich um ein Rapid-Application-Development-Tool (RAD), mit dessen Hilfe man schnell datenzentrierte Web-Anwendungen erstellen kann. Der Entwickler hat Zugriff auf zahlreiche Komponenten (siehe Abbildung 3) und kann eine Anwendung nach dem Baukasten-Prinzip zusammensetzen.

Viele Funktionen und Eigenschaften lassen sich deklarativ anpassen. Das Layout kann man durch Drag & Drop nach

Belieben verändern. Durch die enge Verbindung zur Oracle-Datenbank sind Prototypen sehr schnell realisiert, die weitaus mehr Funktionalität bieten als einfache Mockups oder Click-Dummys. Dadurch kann man Ergebnisse zügig präsentieren und früh auf Usability testen.

Besonders seit der Apex-Version 5.0 hat sich beim Thema „UX“ sehr viel getan. So wurde das Design komplett überarbeitet und trägt seitdem den Namen „Universal Theme“. Es sieht nun modern aus und ist responsive. Mit dem „Theme Roller“ und den „Template Options“ lassen sich viele Design-Elemente ohne Programmierkenntnisse leicht anpassen. Im Internet findet man weiterhin zahlreiche Plug-ins, die den Umfang von Apex stetig erweitern.

Reicht das nicht aus, muss man bei besonderen Anforderungen mit CSS und JavaScript arbeiten. Auch sind fundamentale Kenntnisse über HTML-DOM und Web-Entwicklung im Allgemeinen zwingend notwendig. Nichtsdestotrotz eignet sich Apex hervorragend, um benutzerfreundliche Anwendungen zu erstellen.

Fazit

„UX“ ist ein wichtiges und großes Thema. Eine gute UX-Strategie ist im Konsumentenmarkt überlebenswichtig. In der Business-Welt gewinnt es langsam mehr an Bedeutung. Die Anwender sind durch ihre privaten Applikationen mittlerweile einen hohen Standard gewohnt. Diesen Standard erwarten sie auch immer mehr von ihren Anwendungen im Berufsleben. Selbstverständlich erfordert eine gute UX-Strategie mehr Ressourcen. Entwickler, Architekten und Projektmanager müssen sich in diesem Bereich weiterbilden. Außerdem brauchen sie oft Kenntnisse über neue Technologien und Tools. Durch das Feedback der Nutzer kann die Entwicklungsphase auch länger dauern. Auf lange Sicht gesehen, bringt es allerdings überwiegend nur Vorteile, sich mit diesem Thema auseinanderzusetzen. Wie bereits erwähnt, führt eine gute User Experience zu mehr Effektivität, Effizienz und Zufriedenheit der Mitarbeiter. Das ist letztendlich genau das, was wir alle wollen: Gute Software schreiben, die gerne genutzt wird.

Weiterführende Links

- [1] <https://usersnap.com/de/blog/usability-nielsen>
- [2] <https://www.uxmatters.com/mt/archives/2014/09/eye-tracking-in-user-experience-design.php>
- [3] <http://ui-patterns.com/patterns>
- [4] <https://apex.oracle.com/pls/apex/f?p=42>



Alexej Schneider
alexej.schneider@mt-ag.com