

Kapitel 9

Vom Papier zum interaktiven Prototyp: Wireframe, Mock-up & Co.

Jetzt fängt der bunte Teil der Gestaltung an. Sie haben Ihre Idee evaluiert, Zielgruppe und Konzept dingfest gemacht und sich ausführlich mit der gewünschten Plattform und deren Möglichkeiten und Prinzipien beschäftigt. Sie kennen alle Rahmenbedingungen und wissen, dass die Idee, die Sie als App umsetzen wollen, ein Problem löst und es dafür Nutzer gibt.

9

Jetzt geht's ans Reißbrett und an die Gestaltung der App. Hierbei können verschiedene Phasen des Prototypings durchlaufen werden. Bei jedem Schritt werden Sie sich näher mit den Inhalten der App beschäftigen und somit am Ende einen visuellen, eventuell sogar funktionstüchtigen Prototyp entwickelt haben.

Die Gestaltung einer App beginnt nicht in Adobe XD, Figma, Sketch oder in Xcode. Sie beginnt mit der Evaluierung einer Idee und der stetigen Konkretisierung und Formulierung der App selbst. Prototypen sind Ihr Wegbegleiter auf der Reise von der Idee zur ersten Version Ihrer App.

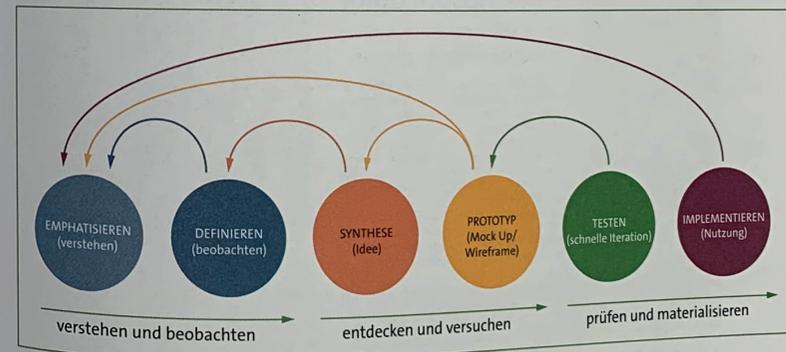


Abbildung 9.1 Produktiteration nach Design Thinking

Im Groben können folgende Prototypphasen unterschieden werden:

1. **Entwurfsskizzen:** Zunächst sollten Sie Ihre App skizzieren, denn eine interaktive Anwendung ist immer im Wandel. Layouts müssen daher in verschiedenen Abläufen getestet werden. Gerade Skizzen lassen sich schnell anpassen und erweitern. Egal, ob Sie eine App alleine entwickeln oder in einer Gruppe, die erste Konkretisierung findet idealerweise auf Papier statt. Sie können aber natürlich auch digital in einem Grafikprogramm loslegen. Hauptsache, Sie arbeiten mit einem Tool, das sich vertraut anfühlt und das bei Ihnen die Kreativität anregt. So bietet beispielsweise Affinity den Affinity Designer für das iPad an. Mit dieser App können Sie recht einfach Layouts mit Standard-Komponenten für Apps erstellen. Es gibt aber viele weitere Apps, die Sie hier unterstützen können. Einige stelle ich noch später in diesem Kapitel vor.

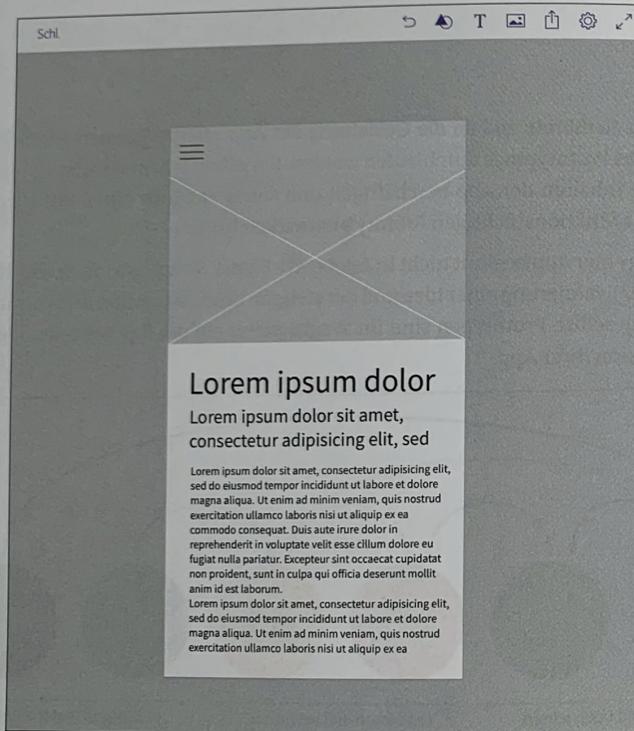


Abbildung 9.2 »Adobe Comp«-Wireframing-Layout

2. **Wireframes:** Aus den Skizzen wird anschließend ein Wireframe entwickelt, in dem die Inhalte und die Funktion genauer beschrieben und miteinander verknüpft werden. Im Vergleich zur Skizze sind Wireframes der erste schematische Design-

entwurf der App. Hier wird jedoch noch nicht konkret grafisch gearbeitet. Der Sinn hierbei ist es, genau zu erkennen, wie die User Journeys sich abbilden. Dort können die ersten Optimierungen vorgenommen werden. Hier lassen sich die ersten Fehler erkennen, verbessern und optimieren.

3. **Design-Mock-ups:** Der nächste Schritt zur finalen Darstellung ist das Mock-up, in dem das User Interface der App visuell und grafisch dargestellt und die visuelle Gestaltung getestet wird. Auch in diesem Stadium ist es möglich, Iterationen durchzuführen und erste funktionale und emotionale Ergebnisse zu erhalten. Diese werden dann meist in Grafikprogrammen wie Adobe XD, Adobe Illustrator, Bohemian Sketch, Figma oder Invision Studio erstellt und sollen das finale Design zeigen.

Iteration

Eine Iteration beschreibt einen sich wiederholenden Prozess, bei dem gleiche oder ähnliche Handlungen durchgeführt werden. Im Falle eines Designs oder eines Wireframes wird hierbei versucht, das Layout oder den User Flow zu optimieren.

9.1 Der iterative Design-Prozess

Wie Sie vielleicht erkannt haben, lassen sich unterschiedliche Arten von Prototypen einsetzen. Idealerweise nutzen Sie jedes der Modelle nach und nach. Bei kleinen, sehr überschaubaren Projekten kann es sinnvoll sein, den Wireframe-Prozess zu überspringen. Bei langwierigen Kick-off-Projekten mit umfangreichen Prozessen innerhalb eines Teams ist es aber sinnvoll, den kompletten Umfang des Prototyping zu durchlaufen.

Der ideale Design-Prozess einer App ist iterativ. Die App sollte möglichst früh prototypisiert werden. Dabei sollten die Design-Entwürfe wiederholt anhand von prototypischen Realisierungen eines Interfaces evaluiert werden. Iterationen können zum einen im Wireframe-Stadium und zum anderen im Design-Stadium durchgeführt werden und auch nacheinander erfolgen. Hierbei gibt es keinen wirklichen Königsweg, aber es lassen sich Gründe finden, unter welchen Bedingungen und in welchen Situationen der eine oder der andere Weg besser ist.

Die Wireframe-Iteration

Eine Wireframe-Iteration ist bei technischen Produkten im B2B-Bereich sinnvoll, bei denen es weniger um Branding und eine emotionale Ansprache geht. Hier stehen rein funktionale Aspekte der App im Mittelpunkt. Sollten Sie also etwa eine App für einen Wartungsdienstleister von Rauchwarnmeldern erstellen, wäre dieses Iterationsmodell wahrscheinlich das geeignetere. In meinen Projekten ist es meist so,

dass wir uns zunächst über die grundsätzliche Struktur Gedanken machen, bevor wir in die visuelle Ausarbeitung gehen. Erfahrungsgemäß ist es so, dass, wenn man ein fertiges Design vor Augen hat, auch über visuelle Inhalte diskutiert wird. So wird über Farben, Formen oder die Schrift diskutiert, aber nicht über die Funktionen und Struktur der App. Somit verlieren Sie Zeit und haben eine unnötige Diskussion geführt, ohne am wirklichen Problem gearbeitet zu haben.

B2B-Apps

Eine B2B-App (Business-to-Business) ist eine App, die vorwiegend in Unternehmen genutzt wird. Zum einen können das vertriebsunterstützende Apps sein oder Apps, die im eigenen Unternehmen genutzt werden. Ein Beispiel wäre eine App, mit denen man Prozesse oder Maschinenanlagen in der Produktion steuern kann.

Die Design-Iteration

Auch der Design-Prozess sollte iterativ sein: Nach einer initialen Design-Phase wird die App potenziellen Nutzern präsentiert, und deren Verhalten und Reaktionen werden beobachtet. Hierbei sollen im Wesentlichen die emotionalen Aspekte der App getestet werden. Im Falle einer B2C-App – also einer App, die ein Produkt bewerben soll – sollte dieser Iterationsprozess öfter durchlaufen werden, um die App besser an die Vorstellungen der Zielgruppe anzupassen, denn neben der Funktion ist die Gestaltung hierbei eines der wichtigsten Merkmale für das Verstehen des User Interfaces.

B2C-Apps

Eine B2C-App (Business-to-Customer) ist eine App, die ausschließlich Konsumenten ansprechen soll. Ein Beispiel wäre eine App von McDonalds, in der es Rabattgutscheine für Hamburger gibt.

9.1.1 Warum überhaupt Iterationen?

Digitale Medien werden zunehmend iterativ entwickelt. Es liegt in Ihren Händen zu entscheiden, welche Vorgehensweise Sie bevorzugen. Manchmal müssen Sie mehrere Iterationsvarianten ausprobieren, bevor Sie entscheiden können, welches für Sie das effektivste Verfahren darstellt. Der Standard ist der Weg von der Idee bzw. Skizze über den Wireframe und das Mock-up bis zur Entwicklung. In beiden oben genannten Varianten sollte der integrierte Iterationsprozess in jedem Fall durchlaufen werden. Es stellt sich nur eine Frage: Wie viele Iterationen sind sinnvoll?

Diese Frage hängt zum einem von Ihrem Budget und den erzielten Ergebnissen ab. Eins steht aber fest: Der iterative Gestaltungsprozess Ihrer App durch User Tests,

Feedback von anderen Projektbeteiligten oder vom Kunden hilft Ihnen, eine App zu realisieren, die in Design und Usability überzeugt. Dabei ist es gut, so früh wie möglich iterative Prozesse in einem Projekt zu etablieren, denn hier gewonnene Erkenntnisse legen frühzeitig Probleme und Fehler im Design offen und ermöglichen Ihnen, schnell darauf zu reagieren. Im Grunde sollten ständig Iterationen Ihrer App gemacht werden, selbst wenn diese längst im App Store verfügbar ist und bereits Nutzer hat. Denn es ist doch so, dass es stetig neue Funktionen oder Anpassungen gibt, die nicht zwangsläufig von Ihnen als Designer kommen. Entwickler optimieren die App und machen diese performanter und einfach wartbar. Hierbei müssen oder können bestehende Funktionen verbessert und optimiert werden oder sogar komplett neu gedacht werden. Denken Sie daran, eine App oder Software ist nie fertig, sie lebt und passt sich an.

Iterationen sollten immer im Kontext stattfinden, d. h., wenn die Funktionalität der App iteriert werden soll, sollten gestalterische Themen in diesem Vorgang ausgeklammert werden, und umgekehrt. Es ist ärgerlich, wenn schon gestaltet wird, obwohl parallel noch keine Funktionsweise beschrieben wurde. Ebenso ist es sehr ungünstig, während der ästhetischen Gestaltungsphase noch funktionale Änderungen vorzunehmen. Versuchen Sie, die Wireframe-Phase abzuschließen und erst dann mit der Gestaltung der App zu beginnen. Das spart Zeit, Nerven und Kosten.

9.2 Stift und Papier – eine erste Skizze der App

Nun geht's los. Die erste richtige Visualisierung Ihrer App-Idee steht bevor. Bevor Sie loslegen, noch eine kleine Checkliste:

1. Bietet die App-Idee eine Lösung für ein Problem?
2. Haben Sie Ihre Zielgruppe definiert? Haben Sie Personas identifiziert?
3. Was sind Ihre Erkenntnisse aus der User Research?
4. Haben Sie die Marktsituation gecheckt und mögliche Konkurrenzprodukte angeschaut?
5. Für welche Plattform haben Sie sich entschieden?
6. Kennen Sie Ihre Plattform (iOS, Android)?
7. Haben Sie Stift und Papier bereitliegen?

Können Sie all diese Fragen positiv beantworten, kann es losgehen. Einen Radiergummi brauchen Sie übrigens tatsächlich nicht – wenn Sie in dieser Phase einen »Fehler« machen, ist das vollkommen in Ordnung. Zeichnen Sie einfach neu. Sobald Sie anfangen zu radieren, beschäftigen Sie sich eher mit der Zeichnung als mit Ihrer App. Zeichnen Sie ruhig wild, es ist kein Schulunterricht, wo es Noten gibt. Bringen

Sie Ihre Idee so auf das Blatt Papier, dass Sie es anderen zeigen können und diese Ihre Idee nachvollziehen können.

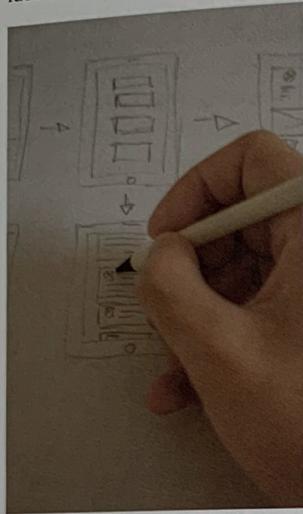


Abbildung 9.3 Die erste Visualisierung der App. Zeichnen Sie grob, und verlieren Sie sich nicht in Details.

Es gibt auch bereits vorgefertigte Schablonen von Smartphones, Tablets & Co. auf dem Markt, in denen Sie Ihre App skizzieren können. – Ich rate Ihnen allerdings davon ab, diese zu Projektbeginn zu nutzen, da sie bereits einen einengenden Rahmen vorgeben.

Schablonen

Unter <http://www.uistencils.com> finden Sie z. B. Schablonen, die Sie kaufen können, um Sie dann bei der Gestaltung auf Papier zu nutzen.

9.2.1 Die erste Skizze

Zunächst sollten Sie eine grobe Skizze Ihrer App-Idee anfertigen. Zeichnen Sie nicht zu kleinteilig oder sehr konkretisierte Elemente. Dem Detail Aufmerksamkeit zu schenken, ist natürlich wichtig – zu Beginn Ihres Projekts sollten Sie sich allerdings nicht in den Details verlieren. Es geht im ersten Schritt darum, eine grobe Vorstellung davon zu bekommen, wie Ihre App strukturiert ist.

Mittlerweile kennen Sie einige Apps sowie die Funktionsweisen der verschiedenen Plattformen. Nutzen Sie diese Kenntnisse bei der Visualisierung Ihrer App auf Papier.

Sie finden die Darstellung der Timeline von Twitter sehr gut und benötigen eine ähnliche Visualisierung? Kopieren und alterieren Sie. Sie müssen das Rad nicht neu erfinden, sondern nur soweit anpassen, dass es Ihren Anforderungen genügt. Es gilt nun, Ihre Problemlösung auf Papier zu bringen, und zwar mit den Rahmenbedingungen, die Sie bereits kennen.

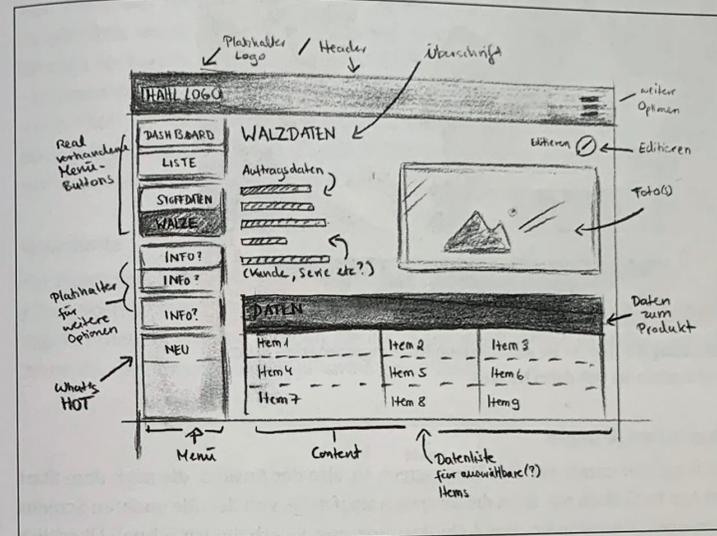


Abbildung 9.4 Zeichnung einer iPad-App

9.2.2 Storyboard

Anschließend sollten Sie mit *Storyboards* – also kurzen textlichen Beschreibungen und einfachen Skizzen Ihrer App-Bestandteile – arbeiten. Ein Storyboard unterscheidet sich von einer Skizze dahingehend, dass es eine Vielzahl von Skizzen aneinanderfügt und einen Ablauf zusammenhängend darstellt. Storyboards im App-Design stellen Use Cases von Anwendern nach. Es sind die Problemfälle, für die Ihre App die Lösung bietet. Bei Storyboards geht es darum, dass alle möglichen Problemfälle abgebildet werden. Ziehen Sie hierbei auch die entwickelten Personas heran. Sie helfen Ihnen dabei, die Problemfälle zu konstruieren und Fehlerstellen zu erkennen.

Storyboards funktionieren am besten, wenn sie tatsächlich eine Geschichte erzählen. Darin sollten Ihre Personas als Hauptdarsteller fungieren. Stellen Sie sich vor, wo der Nutzer Ihre App am häufigsten nutzen wird und wie die äußeren Umstände sind: Ist es Nacht? Befindet sich der Akteur in einer Werkshalle? Oder gerade im Schwimmbad? Spielen Sie mehrere solcher Fälle für jede Ihrer Personas durch.

Versuchen Sie nun, in Ihren Storyboards die zentrale Stelle zu finden, die alle Problemfälle miteinander vereint. Das wird der Startpunkt Ihrer App sein. Haben Sie keinen? Nun, dann wissen Sie, was noch fehlt.



Abbildung 9.5 Spielen Sie verschiedene Nutzungsszenarien durch, und machen Sie sich dabei Notizen.

Hauptscreen skizzieren

Ich fange hier immer mit dem Hauptscreen an, also der Ansicht, die nach dem Start der App erscheinen soll. Es ist die zentrale Anlaufstelle, von der alle anderen Screens abzweigen: Einstellungen, Profil, On-Boarding usw. So erhalte ich schnell Überblick über die Bereiche der App. An diesem Hauptscreen arbeite ich in der Regel recht oft bzw. fasse diesen wiederholt an, ist es doch meist so, dass, während ich andere Bereiche der App gestalte, der Hauptscreen um Inhalte erweitert wird, da sich neue Inhalte auf anderen Ansichten ergeben.

Wie bereits zuvor erwähnt, geht es darum, die Struktur der App zu verstehen. Hier werden Sie schnell erkennen, wie umfangreich und aufwendig die Umsetzung Ihrer App wird. Sie erkennen die ersten technischen Probleme, die von Ihrem Entwickler gelöst werden sollen. Sie sollten Ihre angefertigten Skizzen in jedem Fall allen, die an dem Projekt beteiligt sind, zeigen. Versuchen Sie, die Kenntnisse und Lösungen der Beteiligten in Ihre App einfließen zu lassen.

Nebenscreens skizzieren

Nachdem Sie den Hauptscreen skizziert haben, zeichnen Sie alle anderen notwendigen Screens. Sollen Prozesse abgebildet werden, zeichnen Sie auch diese ein. Bei Prozessen wie z. B. dem Hinzufügen eines Elements, z. B. eines Kalendereintrags, sollten Sie versuchen, alles auf einer View abzubilden. Sollte der Nutzer eine hohe Anzahl von Eingaben tätigen müssen, verteilen Sie diese kategorisiert über mehrere Seiten. Kein

Nutzer möchte durch eine lange Liste scrollen und in 20 Eingabefeldern Daten über seine Person eintragen. Zu viele Seiten sollten es natürlich auch nicht sein, für die Anzahl gibt es aber keine universalgültige Empfehlung. Je kompakter, desto besser.

Versuchen Sie nicht, die Anzahl der Screens als einen Messgrad für die UX zu nehmen. Die Anzahl der Screens ist nur ein kleiner Teil vom Ganzen. Bei der Abbildung von Prozessen sollten Sie versuchen, die Wege, die der Nutzer gehen muss, nach Möglichkeit abzukürzen, aber so, dass die Usability nicht darunter leidet. Vielleicht können Sie bereits durch Abfragen im On-Boarding-Prozess die wesentlichen Informationen des Nutzers registrieren, sodass dieser seine Daten nur einmal eingeben muss. Oder Sie nutzen für den Registrierungsprozess den *Social Connect* über Facebook (siehe dazu Abschnitt 6.3.2, »Die Registrierung«). Hier sparen Sie sich als Designer, dem Entwickler und dem Nutzer Zeit.

Flowcharts

Ein Flowchart, auch Flussdiagramm genannt, ist eine grafische Abbildung der logischen Schritte eines Problems oder eines Ablaufs innerhalb eines Programms. Mit sogenannten Flowcharts können Sie die »Reise des Nutzers« darstellen, also alle Wege, die der Nutzer in der Applikation gehen kann.

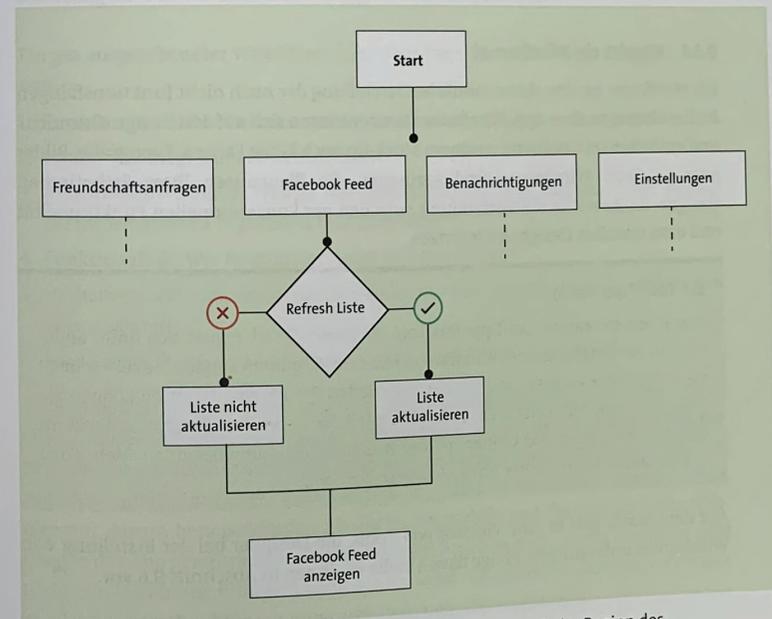


Abbildung 9.6 Flowcharts in der Facebook-App, hier exemplarisch der Beginn des Facebook-Feed-Prozesses

Klickdemos

Mit Klickdemos können Wireframes getestet werden. Hierbei wird die grundlegende Nutzerführung getestet und erlebt. So werden verschiedene Wireframes durch Klickflächen miteinander verbunden.

Dabei können auch höchst komplexe und verwinkelte Wege und daraus resultierende Funktionalitäten abgebildet werden. Das hilft Ihnen, schnell zu erkennen, wo sich Probleme auftun können oder wo noch Verbesserungsbedarf besteht. Es ist im Grunde die Blaupause Ihrer Applikation – mit all Ihren Facetten und Möglichkeiten.

9.3 Wireframe – es wird konkreter

Nun gilt es, Ihre ersten Überlegungen, die Sie bereits skizziert und gegebenenfalls in einem Storyboard festgehalten haben, weiter zu konkretisieren. Hierbei helfen sogenannte *Wireframes*. Ein Vorteil von Wireframes ist, dass ohne Programmieraufwand ein einfacher Prototypen erstellt werden kann. So können Sie sich ganz auf Ihre konzeptionellen Fähigkeiten konzentrieren.

9.3.1 Was ist ein Wireframe?

Ein Wireframe ist eine skizzenähnliche Darstellung der noch nicht funktionsfähigen Bedienelemente einer App. Wireframes konzentrieren sich auf den Design-Grundriss und enthalten im Gegensatz zu einem Mock-up noch keine Farben, Typografie, Bilder oder Grafiken. Wireframes sind sozusagen die Blaupausen Ihres ästhetischen Designs. Ihr Zweck ist die Verbindung zwischen der konzeptionellen Funktionalität und dem visuellen Design des Interface.

Ein Tool-Tipp vorab

Hier schon ein kurzer Tool-Tipp: Das Tool *Bohemian Sketch* eignet sich unter anderem gut zur Erstellung eines Wireframes. Mit dem Programm können Sie sehr schnell gute Ergebnisse erzielen, und mit der integrierten Prototyping-Funktion können Sie Ihren Prototypen umsetzen. Ein weiterer Vorteil: Sie haben Ihre Blaupause direkt im richtigen Programm und können mit der visuellen Gestaltung beginnen. Mehr dazu in Abschnitt 9.7.3 und unter <https://www.sketch.com>.

Auf dem Markt gibt es eine Vielzahl von Tools, die Designer bei der Erstellung von Wireframes unterstützen. Einige davon stelle ich Ihnen in Abschnitt 9.6 vor.



Abbildung 9.7 Vom Wireframe zum Design in der Foursquare-App

Ein gut ausgearbeiteter Wireframe sollte über folgende Inhalte Auskunft geben können:

1. Struktur: In welcher Relation stehen die einzelnen Screens zueinander?
2. Inhalt: Was zeigt jeder einzelne Screen?
3. Informationsstruktur: Welche Navigationsstruktur liegt zugrunde? Wie werden die Informationen organisiert und dargestellt?
4. Funktionalität: Wie funktioniert jeder einzelne Screen?
5. Verhalten: Wie interagiert der Anwender mit dem Screen? Und wie verhält sich dieser Screen?
6. Sind Empty States und Fehlermeldungen in den Wireframes inkludiert?

Wireframes sind keine Repräsentation des visuellen Designs. Lassen Sie sich in einem solchen Wireframing-Tool nicht dazu verleiten, mit der Gestaltung zu beginnen. Verzichten Sie hier auf Farbe, Icons und die konkrete Anordnung der einzelnen App-Bestandteile, außer es dient Ihnen dabei, Ihre Idee besser zu erklären. Es geht hier nur darum herauszufinden, ob die Funktionalität und die richtige Nutzerführung in der App gewährleistet sind und sich die App-Idee auch richtig »anfühlt« oder nicht. Ein Wireframe fühlt sich dann gut an, wenn Sie erkennen, dass der Nutzer nur wenige Schritte durchführen muss, um zu seinem Ziel zu gelangen.

Sollten Sie im Team arbeiten, vereinfacht ein solches Wireframing-Tool auch die Teamarbeit. Sie können Ihre sogenannten *Drafts* an Kollegen weitergeben, die Ihre Idee testen und Ihnen Verbesserungsvorschläge unterbreiten können. Das hilft Ihnen, kreativ zu arbeiten, denn so erhalten Sie eine neue, unverfälschte Perspektive auf Ihr mögliches Produkt.

Draft

Ein Draft ist ein Entwurf. Dieser dient als Diskussionsgrundlage für Teammitglieder oder den Kunden.

Wireframes zeigen die hierarchische Struktur der App, diese Struktur kann aus folgenden Eigenschaften bestehen:

- ▶ Größe: Je größer ein Objekt, desto wichtiger und zentraler ist es.
- ▶ Kontrast: Die Stärke des Kontrasts bestimmt die Unterscheidbarkeit der Elemente.
- ▶ Duktus: Die Stärke der Kontur einer Fläche oder auch die Einfärbung beschreibt die Funktionalität.
- ▶ Stil: Beschreibt das Verhalten von Elementen. Ein abgerundetes Rechteck wird z. B. eher als eine Schaltfläche wahrgenommen als ein normales Rechteck.
- ▶ Gruppe: Gruppierete Objekte stehen unter einem Motto.
- ▶ Konsistenz: Wiederkehrende Elemente werden vom Nutzer schnell erlernt.
- ▶ Symmetrie: Gefällt dem Auge und schafft Ordnung.

9.3.2 Wann werden Wireframes eingesetzt?

Der Zweck eines Wireframes hängt davon ab, wer einen solchen Wireframe entwickelt. Von einem Wireframe können viele Projektbeteiligte profitieren; betrachten wir diese Beteiligten und ihre Intention genauer.

Interaktions- und UX-Designer

Interaktions- und UX-Designer nutzen Wireframes, um User Journeys zwischen den Seiten darzustellen. Typischerweise sind diese Wireframes eine Kombination aus der Informationsarchitektur, aus Storyboards und Wireframes. Auf diese Weise sollen die Schwachstellen zwischen den einzelnen Screens und Abläufen innerhalb einer App erkennbar und die User Flows optimiert werden.

User Flow

Ein User Flow beschreibt den Weg, den der Nutzer innerhalb einer Applikation beschreitet, um eine Aktion durchzuführen. Hierbei wird jeder mögliche Schritt abgebildet.

Grafik- und GUI-Designer

Grafik- und GUI-Designer nutzen Wireframes als Blaupause für ihre visuellen Designs. Die Erstellung von Wireframes inspiriert Designer, was zu einem flüssigeren kreativen Prozess führen kann. Hierbei greift der Designer vorwiegend auf Papier oder Tools wie *Balsamiq Mockups* oder auf vorgefertigte Vorlagen für Sketch oder Photoshop zurück. Eine gute Anlaufstelle für solche Vorlagen ist z. B. bezüglich Sketch <http://www.sketchappsources.com>, <http://www.dribbble.com> oder <http://www.behance.com>. Für Balsamiq Mockups finden Sie sogenannte Stencils unter <https://mockupstogo.mybalsamiq.com/projects> oder auf der Balsamiq-Website <https://balsamiq.com>.

Back-End- und Front-End-Entwickler

Back-End- und Front-End-Entwickler nutzen Wireframes, um die Funktionalität einer App darzustellen und zu testen. Sie erkennen Zusammenhänge, Abläufe sowie Bedingungen, die von einem Screen in einen anderen resultieren. Sie erkennen Schwachstellen und liefern Lösungen.

Während Front-End-Entwickler sich an den Wireframes von Designern orientieren, da diese es ja sind, die technisch umgesetzt werden müssen, orientieren sich Back-End-Entwickler an der Funktionalität der App.

Produkt- und Projektmanager

Produkt- und Projektmanager begutachten Wireframes, um sicherzustellen, dass Projektziele eingehalten und umgesetzt werden. Während sich Designer und Entwickler auf ihre spezifischen Aufgaben konzentrieren, versuchen Produkt- und Projektmanager, einen Überblick vom großen Ganzen zu erhalten.

9.3.3 Was sollte ein Wireframe leisten?

Ein Wireframe sollte funktionieren und alle benötigten Screens enthalten. Jedes Nutzerszenario sollte abgebildet werden, auch Sonderfälle. Dies können Fehlermeldungen sein, die entstehen, wenn bestimmte Bedingungen für einen Funktionsablauf nicht gegeben sind oder Eingaben vom Nutzer fehlen. Sie werden mitunter nicht sofort diese Sonderfälle erkennen, besonders wenn diese technischer Natur sind, deshalb sollten Sie eng mit Ihrem Entwicklerteam zusammenarbeiten.

Iterieren Sie bereits hier mit den Projektbeteiligten Ihre Entwürfe. Versuchen Sie, komplexe Screens oder Prozesse zu vereinfachen und zu optimieren. Am besten führen Sie bereits jetzt erste Tests mit Ihrer Zielgruppe durch. Hier können Sie eine kleine Fokusgruppe aus Freunden zusammenstellen oder auch mit Kunden Ihre App testen. Versuchen Sie vorab, bevor Sie Ihren Wireframe zeigen, den Kunden Ihre befragen.

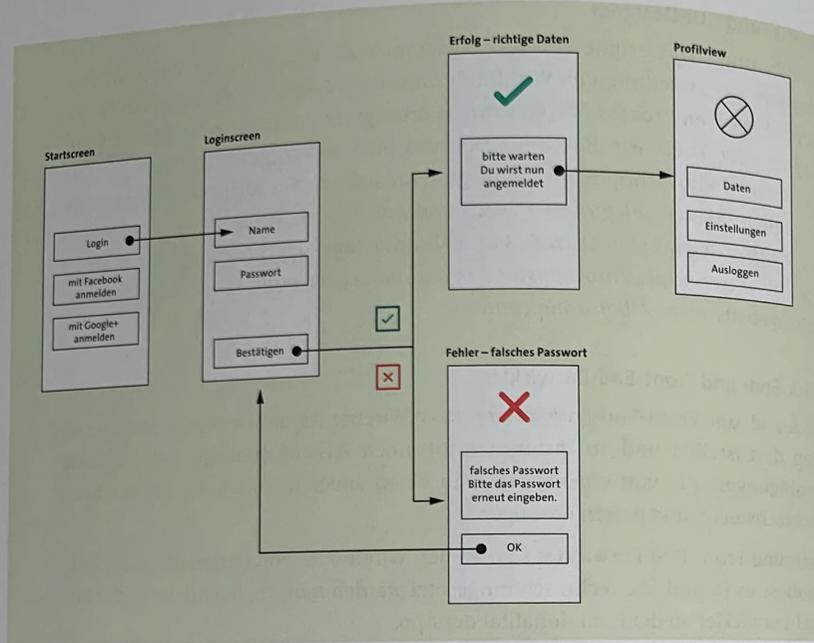


Abbildung 9.8 Sonderfälle wie Fehlermeldungen sollten in Wireframes skizziert werden.

Stellen Sie Fragen wie: Wie nutzen Sie die App *Wunderlist*, was müsste eine App, die Aufgaben sammeln soll, tun? Wann nutzen Sie Ihr Smartphone? Zeigen Sie danach Ihren Wireframe bzw. den aktuellen Stand der App. Versuchen Sie herauszufinden, ob Ihre Testgruppe Ihren Wireframe versteht. Jede Anmerkung bringt Sie und Ihre App weiter. Wireframes helfen Ihnen, die Gewichtung der Inhalte und die Nutzerführung herauszufiltern.

Optimieren Sie in dieser Phase Ihr Interface, sparen Sie sich im späteren Verlauf Zeit. Da Sie sich jede Entwicklungsstufe eines Wireframes vom Kunden oder Stakeholder absegnen lassen können, wird die Gefahr, die App in eine falsche Richtung zu entwickeln, reduziert. Eine tolle App, mit der Sie solche Tests machen können, ist *POP – Prototyping on Paper* (<https://marvelapp.com/pop>), auf die ich später genauer zu sprechen kommen werde. Mit ihr können Sie die gezeichneten Scribbles leicht zu einem kleinen Prototyp verwandeln.

9.4 Mock-ups – jetzt wird's bunt

In dieser Phase arbeiten Sie im Idealfall auf Basis eines fertig ausgearbeiteten Wireframes, in dem alle Funktionalitäten und Zustände der App abgebildet sind. Basie-

rend auf diesen Wireframes wird nun ein *Mock-up* erstellt. Die Basis von Mock-ups ist also eine konzeptionelle Grundlage, die Designer benötigen, um visuelle Konzepte zu liefern.

Ein Mock-up ist eine detailgetreue Nachbildung einer App. Mock-ups dienen dazu, dem Entwickler Vorgaben zu geben, wie die App zu programmieren ist. Hier konkretisieren Sie Farben, Formen, Typografie und sogar Ton. Im dritten Teil des Buchs werde ich verstärkt auf diese Aspekte eingehen und alle Facetten der Gestaltung näher beleuchten. An dieser Stelle möchte ich Ihnen nur grundlegend Mock-ups und deren Funktion erläutern.

Mithilfe von Mock-ups können Sie das Design- und User-Experience-Konzept Ihrer App diskutieren, bevor mit der aufwendigen technischen Implementierung der Funktionen begonnen wird.

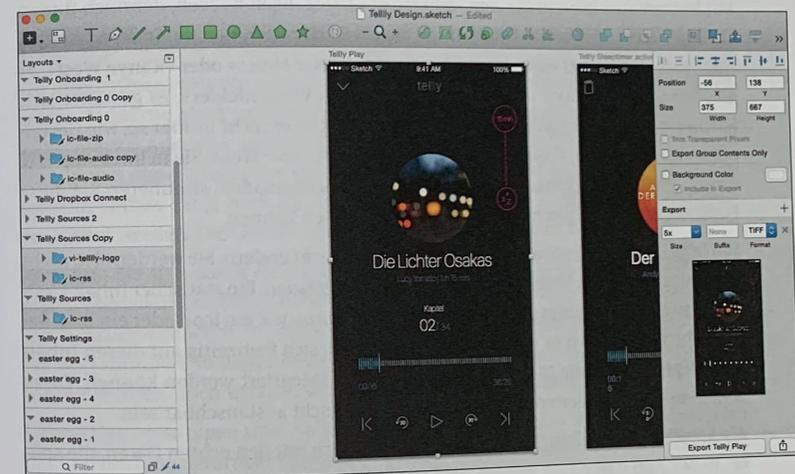


Abbildung 9.9 Das Mock-up einer App in Sketch

9.4.1 Mit richtigen Inhalten arbeiten

In dieser Phase neigen Designer dazu, das Mock-up mit einem pixelperfekten Interface ausstatten zu wollen. Es wird an der optimalen Textlänge und -größe geschraubt, dekorative Elemente werden in der App platziert und schöne Bilder eingebunden. In der Regel wird hier jedoch mit Platzhaltertext (»Lorem Ipsum«) und -grafiken gearbeitet, was nicht optimal ist. Versuchen Sie, so früh es geht, mit den richtigen Inhalten der App zu arbeiten. Es kann vorkommen, dass Texte doch deutlich länger werden als geplant oder Bilder nicht farbig, sondern nur schwarzweiß vorliegen.

Hier bietet Sketch beispielsweise eine sogenannte Daten-Funktion an. Hierbei können Sie Bilder und Text-Dateien laden, die dann in entsprechenden Feldern gezeigt werden. Hier könnte Ihnen beispielsweise der Entwickler bereits bestehende Daten zukommen lassen, und Sie können diese ohne große Mühe in Ihren Designs integrieren.

Statische und dynamische Inhalte

Die Inhalte einer App können in statische oder dynamische Inhalte eingeteilt werden. *Dynamische Inhalte* sind weniger berechenbare Inhalte. Sie können über verschiedene Schnittstellen – sei es von der App selber, durch externe APIs oder durch den Nutzer – generiert werden. Es sind also Inhalte, die auch über das Internet nachgeladen werden und schwer zu kategorisieren sind.

Mit Nutzerdaten kann in der App recht leicht umgegangen werden, dennoch finden sich auch hier unerwartete Hindernisse, die ein Design komplett verändern können. Nicht jeder Nutzer hat einen kurzen Namen wie »Peter Maier« oder »Kanye West«, es gibt mit Sicherheit auch einige »Karl-Wilhelm von Westenfelder«. Es gestaltet sich leichter mit kurzen Inhalten, aber in der realen Welt ist es nicht immer so, wie Gestalter es sich wünschen. Kreieren Sie auch den »Worst Case«: Wenn Sie in Ihrer App z. B. Anwendern die Möglichkeit geben wollen, Fotos hochzuladen, simulieren Sie diese Funktion mit den schlimmsten Bildern, die Sie finden können.

Statische Inhalte sind vorwiegend Inhalte, die sich nicht ändern. Sie werden also nicht dynamisch über eine Verbindung wie das Internet geladen. Ein statischer Inhalt kann ein Label eines Textfeldes sein oder auch Assets der App wie ein Icon oder ein Bild. Bei statischen Inhalten ist es wichtig, darauf zu achten, sich frühzeitig auf diesen Inhalt festzulegen, sodass die Inhalte früh in das Layout integriert werden können. Diese sollten in standardisierten Größen vorliegen und leicht austauschbar sein.

Als App-Designer sollten Sie sich also frühestmöglich mit den echten Daten, die später in der App dargestellt werden sollen, auseinandersetzen. Auch in der Mock-up-Phase sind Iterationen wichtig, da durch eingehendes Feedback zum einen die visuelle Gestaltung optimiert werden kann und zum anderen eventuell letzte Fehler erkannt und verbessert werden können, die in der konzeptionellen Phase nicht aufgefallen sind. Mehrsprachigkeit einer App zählt hierbei genauso. So kann es sein, dass statische Inhalte wesentlich länger sind als gedacht.

9.5 Interaktive Prototypen – der erste echte Eindruck

Wie Sie in den vorherigen Abschnitten gelesen haben, ist ein klassisches Mock-up bzw. ein Wireframe statisch. In manchen Fällen möchte man jedoch ein interaktives Model testen, bevor man es richtig entwickelt.

Mock-up-Tools

- ▶ Framer: framerjs.com
- ▶ InVision: invisionapp.com
- ▶ Principle (für Mac): principleformac.com
- ▶ Flinto for Mac: flinto.com
- ▶ ProtoPie: protopie.io
- ▶ Marvel App: marvelapp.com
- ▶ Adobe XD: adobe.com/de/products/xd.html
- ▶ Bohemian Sketch: sketch.com

Mehr zu diesen Tools erfahren Sie in Abschnitt 9.6, »Wireframe- und Mock-up-Tools«.

9.5.1 Was ist ein interaktiver Prototyp?

Ein interaktiver Prototyp ist z. B. dann sinnvoll, wenn ein eigenes Control entwickelt wurde. Moderne Mock-up-Tools wie Framer, InVision, Principle, Flinto for Mac, ProtoPie, Marvel App, aber auch Design-Programme wie Adobe XD und Sketch ermöglichen genau diese Integration interaktiver Funktionen wie etwa Verlinkungen zu den Screens, sodass sie je nach Komplexität zu den Low- bis High-Fidelity-Prototypen gerechnet werden. Die Grenze zu dynamischen Prototypen ist hier fließend.

Nachdem Sie sich durch erste grobe Skizzen Schritt für Schritt hin zu digitalen und realen Modellen gearbeitet haben, stehen in dieser Phase Tests mit einem Prototyp mit funktionierendem Interface an. Hierbei werden Verhalten und Funktionalität der App simuliert. In dieser Phase können Sie testen, ob die Lösungen, die zuvor entwickelt wurden, auch wirklich so funktionieren, wie es gewünscht war.

Interaktive Prototypen sind die letzte Bewährungsprobe Ihrer App. Der Aufwand bei der Entwicklung derartiger Prototypen ist unter Umständen sehr hoch und erfordert gegebenenfalls Programmierkenntnisse. Im Entwicklungsprozess gibt es verschiedene Stellen, an denen es sinnvoll ist, mit solchen Prototypen zu arbeiten. Komplexe, mehrteilige Mock-ups sind eine gute Basis, um interaktive Prototypen zu entwickeln, die je nach Ausarbeitung zu den sogenannten Low- bis High-Fidelity-Entwürfen gezählt werden. Diese Formen von Prototypen sind immer interaktiv und enthalten, je nach Ausführung und Entwurfsstadium, schon viele Darstellungen der geplanten App.

9.5.2 Low Fidelity – ein Wireframe-Prototyp

In der Low-Fidelity-Version wird ein rudimentärer Prototyp in der Wireframe-Phase entwickelt und geprüft. So können Konzepte sehr früh im Designprozess validiert werden. Durch erste Klickdemos werden die User Flows getestet und auf Schwachstellen hin geprüft. Es handelt sich hierbei um sehr grundsätzliche und grobe Tests.

Bei diesen Prototypen werden durch Interaktionen Seiten angezeigt, teilweise gibt es hier schon einfache Animationen für Seitenübergänge. Eigene neuartige Controls oder Elemente können hier nicht animiert werden. Bei den Low-Fidelity-Prototypen geht es vor allem darum, die Informationsarchitektur zu testen.



Abbildung 9.10 Low-Fidelity-Prototyp: Der Übergang von links nach rechts erfolgt statisch, die rechte Seite wird einfach nur angezeigt.

Low-Fidelity-Prototypen bieten App-Designern die Möglichkeit, schwerwiegende und große Probleme frühzeitig zu identifizieren und zu lösen. In dieser Phase können sehr schnell Iterationen von Konzepten durchgeführt werden.

Die Nachteile sind, dass solche Low-Fidelity-Prototypen nicht die kleinen, sondern eher die großen Probleme identifizieren und dass die emotionalen Komponenten nicht konkret getestet werden können. Lediglich die Frustration oder das Erfolgsgefühl des Nutzers können identifiziert werden. Das sind aber nicht die einzigen Emotionen, die ausschlaggebend sind für den emotionalen Wert einer App.

Klickdemos bzw. Low-Fidelity-Prototypen werden vorwiegend bei Wireframes eingesetzt und werden schon teilweise im iterativen Prozess des Wireframings genutzt. Dadurch hat man in der Gestaltung einer App bereits frühzeitig die Möglichkeit, Probleme zu erkennen, die sich auf die Weiterentwicklung der App auswirken können.

9.5.3 High Fidelity – Design-Prototyp

In der High-Fidelity-Version wird eine App mit ausgearbeitetem Design und mit schon recht ausgereiften Animationen getestet. Allerdings kann der Nutzer hier noch keine realen Daten abrufen oder eingeben. Wir dürfen aber gespannt sein, was die Zukunft in den Prototypen-Tools bringen wird.



Abbildung 9.11 High-Fidelity-Prototypen, bei denen auch Teile des Interfaces animiert (Bild Mitte) sind, sobald es genutzt wird.

High-Fidelity-Prototypen sind mitunter recht aufwendig zu entwickelnde Prototypen. Es können komplette Applikationen als Prototyp entwickelt werden oder nur Teilbereiche der App. High-Fidelity-Prototyping eignet sich gerade für die Ausarbeitung kleiner Details der App, die für die User Experience entscheidende Auswirkungen haben. So wurde mit Sicherheit das *Swipe to Choose* von *Tinder* frühzeitig genau so getestet.

Es ist immer empfehlenswert, selbst erstellte Custom Controls mit solchen Prototypen zu testen. Große und selbstverständliche Funktionalitäten können hier natürlich auch überprüft werden, aber vielleicht sind diese bereits so selbsterklärend, dass hierfür kein Prototyp notwendig ist. Ein Side-Menü bzw. Fly-out-Menü kennt z. B. der Nutzer, Sie müssen es nicht neu erfinden.

High-Fidelity-Prototypen werden dann eingesetzt, wenn man gerade die emotionalen Komponenten einer App testen möchte, auch weil hier bereits mit dem »echten« Design gearbeitet wird. Für den Test der User Experience einer App kann ein High-Fidelity-Prototyp ein wichtiger Gradmesser sein, der hilft, kritische Stellen bei Nutzerführung, Inhalten und Funktionen der App zu erkennen und darauf aufbauend konkrete Optimierungsansätze zu erarbeiten.

Es sei noch angemerkt, dass manche App-Designer High-Fidelity-Prototypen als Dead End ansehen. Gemeint ist dabei, dass ein Mock-up nicht mehr editierbar ist, sobald er End ansehen. Gemeint ist dabei, dass ein Mock-up nicht mehr editierbar ist, sobald er in der Prototyping-Phase befindet. Dies liegt im genutzten Tool begründet: So sind Änderungen im Design zunächst wieder in der Mock-up-Phase angesiedelt. Das heißt, größere Änderungen müssen wieder in dem Tool gemacht werden, in dem das Mock-up erstellt wurde. Danach müssen Sie wieder einen komplett neuen High-

Fidelity-Prototypen entwickeln. Sie verstehen nun vielleicht, warum alle wichtigen Punkte in der Wireframe-Phase abgeschlossen sein sollten.

Egal, ob Sie einen Low- oder einen High-Fidelity-Prototypen testen, verwenden Sie keinen Emulator, und führen Sie die Tests immer auf einem echten Endgerät aus, denn es macht einen gravierenden Unterschied, ob Sie ein Interface mit der Maus oder mit Touchbedienung testen bzw. testen lassen. In der Regel bieten alle Tools zur Entwicklung von interaktiven Prototypen eine App an, mit der Sie die Prototypen auf echten Endgeräten testen können.



Abbildung 9.12 Testen Sie Ihr Design direkt auf dem Endgerät.

Checkliste: Prototypen

- ▶ Erst eine Skizze, dann ein Wireframe und zum Schluss das Mock-up – nicht umgekehrt!
- ▶ Iterieren Sie in jeder Konzeptionsphase.
- ▶ Widerstehen Sie in den frühen Phasen der Versuchung, zu konkret zu werden und Farben, Icons und die konkrete Anordnung der einzelnen App-Bestandteile final zu planen.
- ▶ Teilen Sie im Team frühzeitig die Ergebnisse, und geben Sie jedem Teammitglied die Möglichkeit, am Projekt zu partizipieren.
- ▶ Arbeiten Sie bei den Mock-ups mit echten Inhalten.
- ▶ Interaktive Prototypen helfen dabei, das Look & Feel der App zu erleben und zu testen.
- ▶ Nutzen Sie Low-Fidelity-Prototypen für das große Ganze.
- ▶ Nutzen Sie High-Fidelity-Prototypen fürs Kleine, um z. B. Custom Controls zu prüfen.

9.6 Wireframe- und Mock-up-Tools

Für die Erstellung von Skizzen, Wireframes, Mock-ups und interaktiven Prototypen gibt es viele Tools, die Designern und Entwicklern helfen, und täglich kommen neue Werkzeuge hinzu. Im Folgenden möchte ich Ihnen einige der populärsten Tools vorstellen.

Den ersten Tool-Tipp haben Sie sicher bereits bei sich im Büro liegen: Zu Beginn ist ein leeres Blatt Papier vollkommen ausreichend, um erste Prototypen zu entwickeln. Nicht wenige Designer führen ein Sketchbook als treuen Begleiter ständig mit sich.

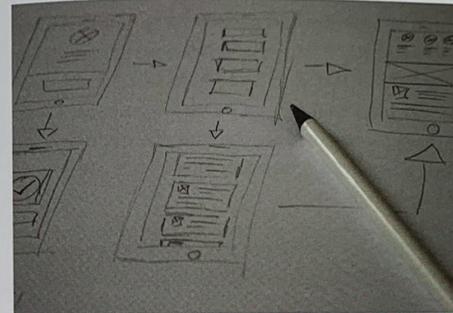


Abbildung 9.13 Nichts ist flexibler als ein Blatt Papier und ein Bleistift.

Natürlich besteht auch die Möglichkeit, Ihre Ideen mit einem Grafiktablett oder mit einem Tablet digital festzuhalten. Hauptsache, Sie arbeiten mit dem Werkzeug, das sich vertraut anfühlt und das Ihrer Kreativität entgegenkommt. Je freier Sie sich dabei fühlen, desto besser.

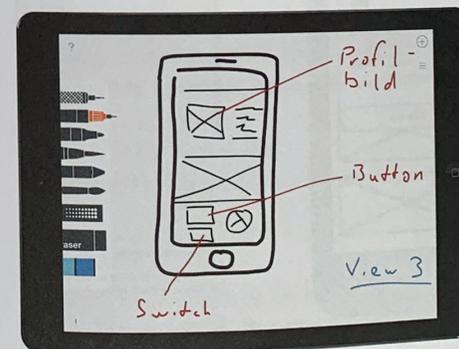


Abbildung 9.14 Sie können auch Zeichen-Apps auf Ihrem Tablet nutzen, um Ihre Ideen zu skizzieren. Hier wurde dafür die iOS-App »Tayashu Sketches Pro« genutzt.

Jetzt aber zu den Empfehlungen für Programme, mit denen Sie Wireframes und Mock-ups erstellen können. Mit einigen Tools lassen sich lediglich Wireframes erstellen, andere Wireframe-Tools ermöglichen darüber hinaus die Erzeugung interaktiver Low-Fidelity-Prototypen. Einige Programme sind komplett kostenlos, andere bieten die Möglichkeit, Trial-Versionen herunterzuladen, mit denen sich das Programm für einige Tage kostenlos testen lässt. Nutzen Sie diese Möglichkeit, um das richtige Tool für sich zu finden.

9.6.1 Balsamiq Mockups

Balsamiq Mockups ist ein Programm, das das schnelle Erstellen von Konzeptskizzen für User Interfaces erlaubt. Die Software enthält bereits vorgefertigte Elemente wie Buttons, Listen oder andere Komponenten, die sich per Drag & Drop frei positionieren lassen. In ihrer Anmutung erinnern diese Elemente an Handzeichnungen. Es lassen sich weitere sogenannte Stencils herunterladen, die das Aussehen und den Stil der Elemente verändern. Auch schön: Controls, die Sie selber entwickelt haben, können Sie innerhalb des Programms archivieren.

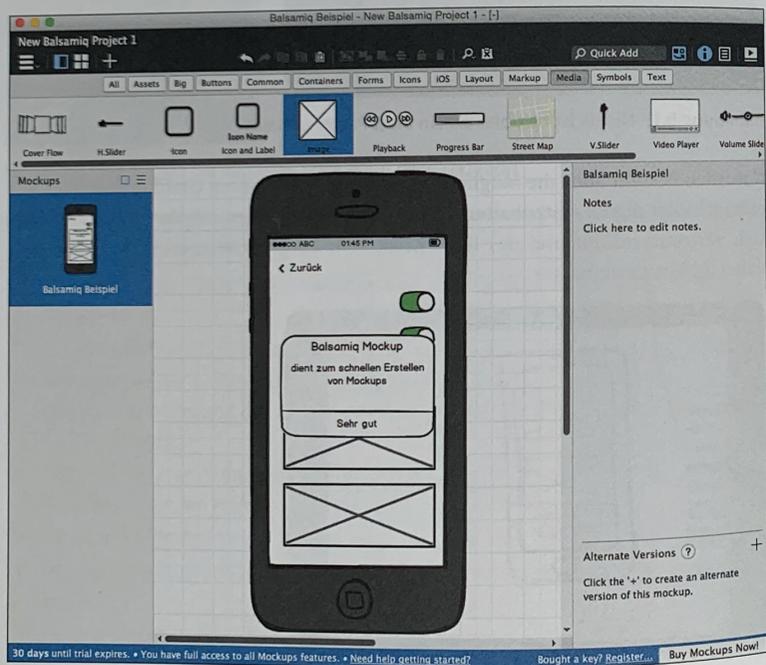


Abbildung 9.15 Für schnelle Konzeptskizzen: Balsamiq Mockups

- ▶ verfügbar für: macOS, Windows, Linux und als Browserversion
- ▶ offizielle Website: <https://balsamiq.com/wireframes/>

9.6.2 Omnigraffle

Omnigraffle ist ein Wireframe-Programm, das ähnlich funktioniert wie Balsamiq Mockups. Jedoch bietet Omnigraffle weitere Möglichkeiten: Es eignet sich für das Erstellen von Skizzen, Mindmaps, Klassendiagrammen und mehr. So können hier auch Flowcharts erstellt werden, was gerade beim Skizzieren von sehr detaillierten User Flows hilfreich sein kann. Auf diese Weise können recht komplexe Verhältnisse und Abläufe abgebildet werden.

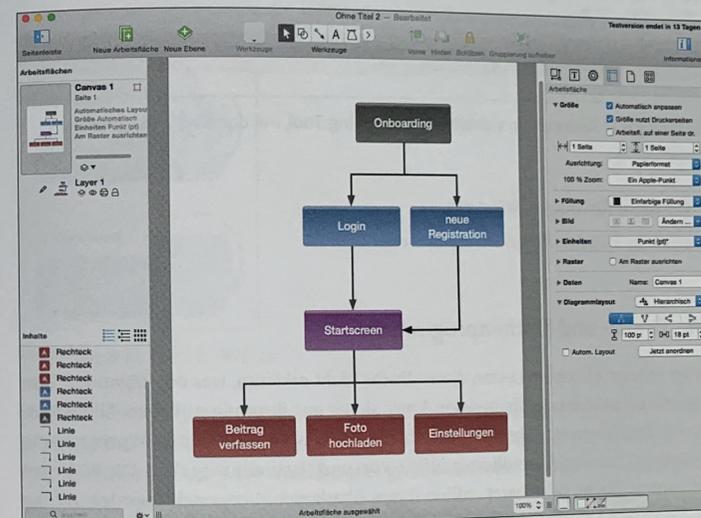


Abbildung 9.16 Mindmaps, Diagramme und mehr: Omnigraffle

- ▶ verfügbar für: macOS, iPad
- ▶ offizielle Website: <https://www.omnigroup.com/omnigraffle>

9.6.3 Axure

Mit dem Prototyping-Tool Axure kann fast alles getestet werden. Axure bietet die Möglichkeit, interaktive Prototypen zu entwickeln. So können Sitemaps und unterschiedliche Widgets erstellt werden. Es ist sogar möglich, den erstellten Wireframe auf dem eigenen mobilen Endgerät zu betrachten und zu testen.

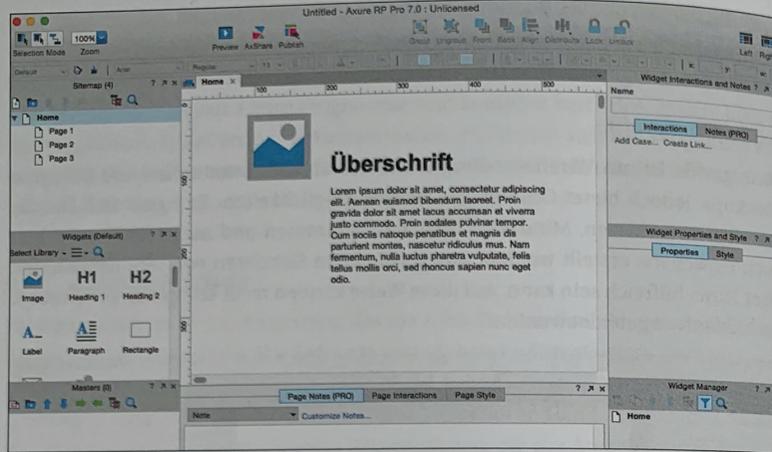


Abbildung 9.17 Axure ist ein vielseitiges Prototyping-Tool, mit dem fast alles getestet werden kann.

- ▶ verfügbar für: macOS, Windows
- ▶ offizielle Website: <http://www.axure.com>

9.7 Grafik- und Zeichenprogramme

Sicher müsste ich vielen Lesern dieses Buchs nicht erklären, was die folgenden Programme können, da sie die meisten Tools sicher aus ihrem beruflichen Alltag kennen. In den folgenden Abschnitten möchte ich dennoch einige Grafikprogramme näher beleuchten und detaillierter auf die Vor- und Nachteile eingehen. Die Wahl des Programms bleibt selbstverständlich Ihnen überlassen. Vorgestellt werden Adobe XD, Figma und Sketch. Es gibt noch weitere Programme wie Invision Studio oder Framer, jedoch sind diese Tools leider noch nicht so weit, um wirklich produktiv genutzt zu werden.

Empfehlungen

Auf YouTube finden Sie viele wirklich sehr gute Tutorials, die Ihnen alle möglichen Kniffe in Photoshop, GIMP & Co. zeigen. Für Photoshop ist auch PSD-Tutorials eine gute Anlaufstelle im Netz, ebenso das Tutplus-Netzwerk. Bevorzugen Sie Bücher, sind die Bücher aus dem Rheinwerk Verlag zu empfehlen.

- ▶ www.youtube.de
- ▶ www.psd-tutorials.de
- ▶ www.rheinwerk-verlag.de/grafik-design
- ▶ <http://tutplus.com>

Gerade für Entwickler, die ohne zusätzlichen Designer an einem App-Projekt arbeiten müssen erlernt und Kenntnisse in Grafikprogrammen erworben werden. Es gibt viele gute Sachbücher oder Websites, die Sie bei dem Erlernen dieser Programme unterstützen.

9.7.1 Pixel vs. Vektor

Photoshop ist ein *pixelbasiertes* Gestaltungsprogramm, das zunächst für die Bildbearbeitung vorgesehen ist. Dort können pixelgenau Fotomontagen oder Retuschen von Bildern erstellt werden, die später in verschiedenen Medien verwendet werden, egal ob Print oder digital.

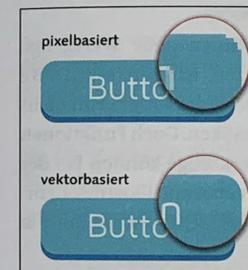


Abbildung 9.18 Pixel vs. Vektor

Die Darstellung eines Bildes basiert auf einzelnen aneinandergereihten Pixeln. In der Anfangsphase des Internets gab es keine Programme, mit denen man grafisch Websites gestalten konnte. Viele Webdesigner griffen daher auf Photoshop zurück. Noch heute wird Photoshop für die Gestaltung von Websites und User Interfaces verwendet. Der Grund: *What you see is what you get*. Ein pixelbasiertes Webdesign, das man in Photoshop gestaltet hat, lässt sich eins zu eins auf anderen Monitoren darstellen, vorausgesetzt es wurde die richtige Auflösung und Größe eingestellt.

In Illustrator und Sketch arbeiten wir vorwiegend mit Vektoren und nicht mit einzelnen Pixeln. Der Vorteil ist dabei, dass vektorbasierte Grafiken frei und vor allem verlustfrei skaliert werden können. Um jedoch zu sehen, wie diese Grafiken tatsächlich später am Monitor aussehen, muss man unbedingt eine Pixelvorschau verwenden. Denn die vektorbasierten Grafiken werden später als Pixel-Bilddateien exportiert, da die Bildinhalte in einzelnen Pixeln darstellen. Grafiken, die im Vektorprogramm immer schön und scharf aussahen, können nun Kacheleffekte aufweisen und unschön aussehen, und Schriften können dadurch nicht mehr lesbar sein.

Gerade die gute Lesbarkeit einer App ist wichtig. Es sind also Werkzeuge nötig, die Sie bei der Gestaltung unterstützen. Im Folgenden möchte ich Ihnen entsprechende Programme vorstellen.

9.7.2 Adobe Illustrator für Interface Design?

Adobe Illustrator ist ein vektorbasiertes Illustrationsprogramm. Mit ihm können Sie Illustrationen, Logos oder Infografiken auf *Vektorbasis* gestalten. Der Vorteil von vektorbasierten Grafiken ist die verlustfreie Skalierbarkeit.

Das Programm eignet sich im Grunde recht gut für die Gestaltung von Apps, jedoch ist der Export von Assets recht kompliziert und zeitintensiv. Exportierte Assets sind mitunter nicht scharf genug, und aufgrund von Anti-Aliasing werden Grafiken an den Rändern weichgezeichnet dargestellt. Designer, die sich mit der Gestaltung von Icons beschäftigen, nutzen vorwiegend Illustrator.

In neueren Versionen von Illustrator und Photoshop wurden neue Werkzeuge ergänzt, die den Designer bei der Entwicklung unterstützen. Ich persönlich rate Ihnen jedoch bei großen Projekten davon generell ab. Leider ist das Programm nicht sehr flexibel. Probleme gibt es vor allem beim Export von Grafiken. Doch Funktionen wie die Artboards und die sehr guten Kontur- und Formwerkzeuge können bei der Gestaltung einer App gute Dienste erweisen. Ich persönlich verwende Illustrator vorwiegend für die Gestaltung von aufwendigen Illustrationen, die ich dann per Copy & Paste in Sketch einfüge.

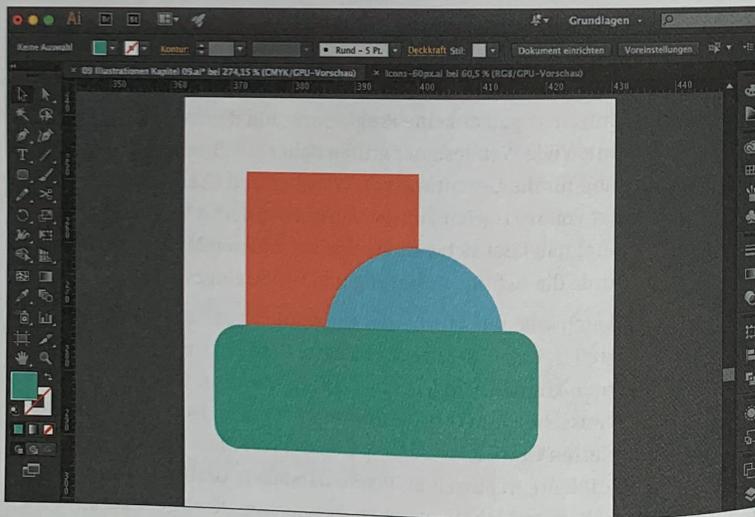


Abbildung 9.19 Adobe Illustrator ist nach wie vor ein beliebtes vektorbasiertes Grafik- und Zeichenprogramm.

Export- und Dithering-Probleme

Weitere Probleme bereitet der Export von Assets, der mithilfe eines veralteten Slice-Tools bewerkstelligt wird. Störendes Anti-Aliasing ist die Folge.

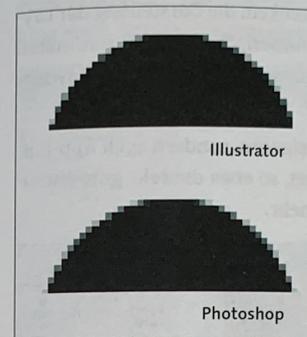


Abbildung 9.20 Anti-Aliasing bei exportierten Assets in Photoshop und in Illustrator. Achten Sie auf die Unterschiede der Kanten.

Bei Farbverläufen macht das fehlende Dithering Probleme, das Farbverläufe blockig darstellt.



Abbildung 9.21 Farbverlauf ohne Dithering



Abbildung 9.22 Farbverlauf mit Dithering

Dithering

Das Dithering ist eine Technik, die bei der Bildverarbeitung verwendet wird, um z. B. Farbverläufe, die aufgrund technischer Restriktionen blockig bzw. hart verlaufen, zu optimieren. Hierbei werden Farbanteile vermischt und so ein homogener Farbverlauf erzeugt.

Unterm Strich eignet sich Illustrator vor allem, um komplexe Illustrationen zu erstellen, die als Grafiken in einer App genutzt werden können.

- ▶ verfügbar für: macOS, Windows
- ▶ offizielle Website: www.adobe.com/de/products/illustrator.html

9.7.3 Bohemian Sketch

Sketch ist mittlerweile der Star am Sternenhimmel der UI- und UX-Designer. Es ist ein vektorbasiertes Programm, das explizit für die Gestaltung von Interfaces und Websites entwickelt wurde. Es besteht auch die Möglichkeit, die Darstellung der Layouts auf Pixelbasis zu betrachten. Elemente werden numerisch auf Pixelkoordinaten positioniert. Das Programm ist sehr performant, es lassen sich blitzschnell Layouts kreieren.

Sketch ist ein vielseitiges Tool, das nicht nur App-Designern, sondern auch App-Entwicklern viele nützliche Arbeitserleichterungen bietet, so etwa die sehr gute Unterstützung im Workflow mit SVG-Dateien und vieles mehr.



Abbildung 9.23 Sketch ist ein vielseitiges Tool.

Symbole

Symbole können verschiedene Kombinationen aus Elementen sein, die global bei jedem Layout eingesetzt werden können. Das können z. B. Icons, Schaltflächen oder weitere Symbole sein. Bei der Veränderung eines der Symbole ändern sich auch alle anderen Symbole des gleichen Typs. Die Idee dahinter ist es, ein Symbol, z. B. ein Icon, zu nutzen. Dieses Symbol samt Icon kann dann beim Gestalten genutzt werden. Nun setzen Sie dieses Icon mehrfach in Ihren Layouts ein, was ist jedoch, wenn Sie das Icon ändern möchten? Theoretisch müssten Sie nun jedes Layout begutachten und das Icon mit dem Neuen austauschen. Da Sie jedoch ein Symbol nutzen, müssen Sie lediglich im Symbol das Icon austauschen, und überall, wo Sie dieses Symbol-Icon

eingesetzt haben, wird mit dem neuen Icon aktualisiert. Die Symbole befinden sich auf einer eigenen Seite und können dort editiert werden. Der sehr große Vorteil ist: Es lässt sich damit ein vollwertiges Design-System entwickeln, das Ihnen eine Weiterentwicklung der App sehr erleichtert. Ich bin in Kapitel 8 detailliert auf die Entwicklung eines Design-Systems eingegangen.

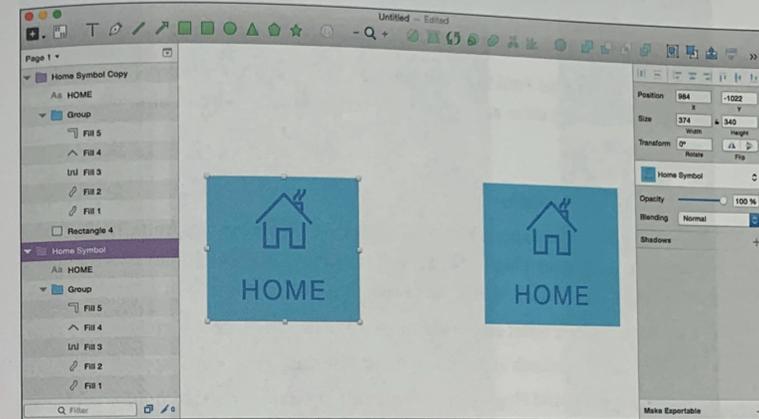


Abbildung 9.24 Ein Symbol in Sketch

Stile

Stile enthalten die Eigenschaften eines Elements. So können z. B. Texte oder Schriften eingefärbt und mit einem Schlagschatten versehen werden. Ein solcher globaler Stil kann auf andere Elemente appliziert werden. Sobald z. B. für den Stil eine andere Farbe gewählt wurde, verändern sich farblich auch alle Elemente mit demselben Stil. Diese Stile können auf Texte und Formobjekte und Konturen angewendet werden.

Symbol Overrides

Wenn Sie Stile und Textfelder in Ihren Symbolen nutzen, können Sie diese mittels sogenannter Overrides überschreiben. Als Beispiel nehmen wir ein Textfeld samt Stil, der die Box definiert, und einen hellgrauen Platzhaltertext. All das ist in einem Symbol zusammengefasst. Sie können nun weitere Stile definieren, z. B. ein Style für ein Fehlertextfeld, also ein Error State des Textfeldes durch Falscheingabe. So können Sie ein Symbol für mehrere Fälle nutzen durch Überschreibung des Urzustands. Auf die Overrides bin ich in Kapitel 8 eingegangen.

Artboards

Ebenso gibt es Artboards und vorgefertigte Templates für Smartphones und Tablets sowie für die Apple Watch. Ein großer Vorteil von Sketch ist zudem der sehr einfache Export von Assets. So reichen wenige Klicks, und alle markierten Icons, Visuals oder andere Elemente werden für die spezifischen Plattformen exportiert.

Prototyping

Sketch beinhaltet eine Funktion, mit der sich Low-Fidelity-Prototypen entwickeln lassen. Diese können dann auf dem Smartphone, Browser oder in einem Emulator getestet werden.

Bohemian Sketch erweitern

Neben den vielfältigen Exportfunktionen bietet Sketch eine große freundliche Community, die stetig neue Plugins für Sketch entwickelt. Zu Sketch gibt es auch viele gute Vorlagen und Plugins. Hier lohnt es sich durchaus, ein bisschen zu stöbern:

- ▶ Die Sketch Toolbox bietet eine große Auswahl an Plugins: <http://sketchtoolbox.com>.
- ▶ Ein Blick lohnt sich auch auf die Sketch App Sources: www.sketchappsources.com.
- ▶ Auch auf GitHub sind Plugins zu finden: <https://github.com/sketchplugins/plugin-directory>.

Einige Plugins generieren sogar Styleguides, in denen alle Textstile, Symbole und Farben gesammelt werden, was sehr beim strukturierten und ordentlichen Anlegen von Layouts hilft. Die Vielfalt der Sketch-Plugins ist recht hoch, und es finden sich stetig neue, mitunter sehr interessante Plugins.

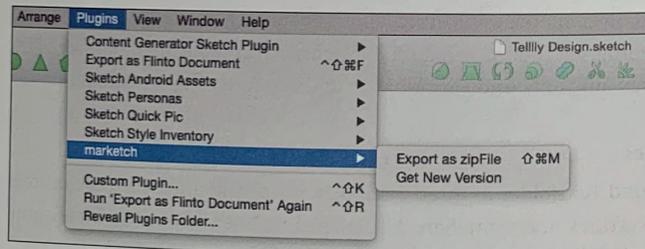


Abbildung 9.25 Das Plugin »Marketch« erstellt eine lokale Website, die quasi wie ein Styleguide fungiert.

Sketch hat jedoch auch einige Nachteile. So gibt es immer noch kleine Fehler bei der Nutzung und dem Umgang mit Bitmaps. Bitmaps bzw. Bilder können nur maskiert oder mit Überblendeffekten genutzt werden. Eine Bildbearbeitung wie in Photoshop ist nicht möglich. Die Entwickler sind jedoch sehr bemüht, alle Fehler auszubügeln

und neue Features zu implementieren. Auch ist das Programm leider nur für den Mac verfügbar. Sketch kündigte jedoch Anfang 2019 eine Webversion sowie ein Collaborations Feature für Sketch an, was bedeutet, dass es auch auf Windows lauffähig sein wird.

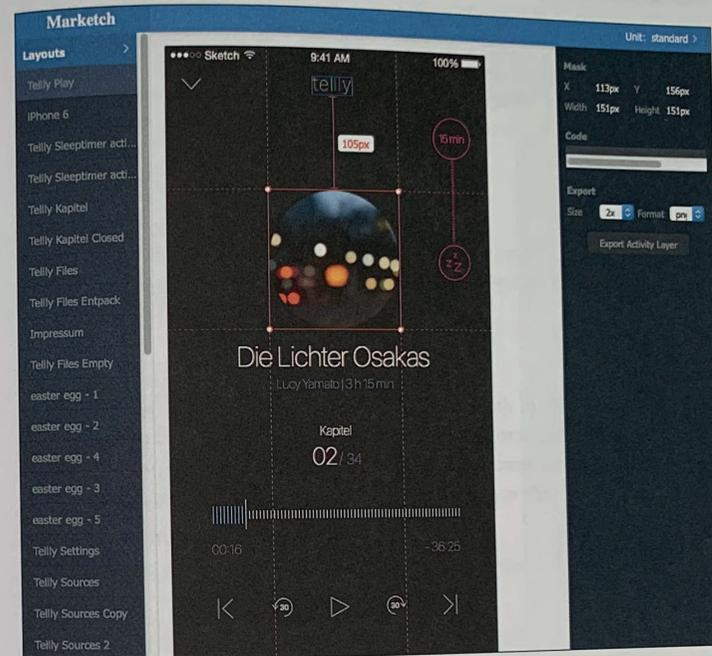


Abbildung 9.26 Ansicht der lokalen Website von »Marketch«. Hier können Elemente eines Layouts genauer betrachtet und Farben und Abstände abgelesen werden.

Ich rate dennoch jedem Interessierten, das Programm einmal zu testen. Anfangs war ich skeptisch, mittlerweile gestalte ich alle meine Apps in Sketch und bin schneller als zuvor und habe mehr Zeit für den kreativen Teil des Jobs.

- ▶ verfügbar für: macOS
- ▶ offizielle Website: www.sketch.com

9.7.4 Adobe XD

Adobe XD wurde von Designern für Designer entwickelt. Mit Adobe XD können Sie digitale Designs für Apps, Websites und Prototypen entwickeln. Die Oberfläche erin-

nerst sehr an Sketch. Man muss auch gestehen, dass die »Kopie« wohlmöglich so gewollt ist, ist doch Adobe XD die Antwort auf Sketch, seit viele Designer von dem langsamen und mühsamen Photoshop/Illustrator zu Sketch gewechselt sind. Das Tolle ist jedoch: Adobe XD ist kostenlos.

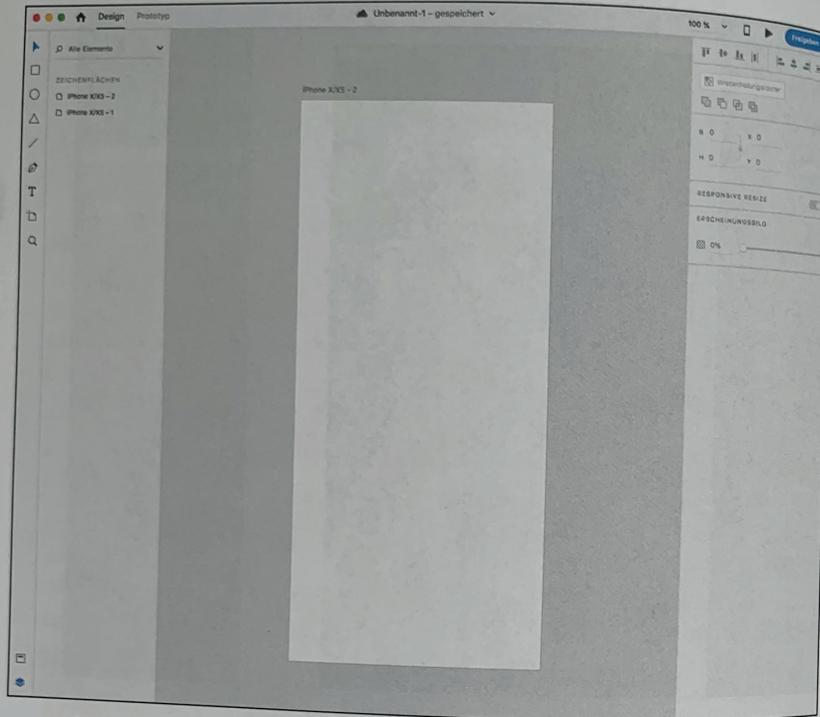


Abbildung 9.27 Adobe XD Nutzeroberfläche

Adobe XD unterscheidet zwischen zwei Ansichten: dem Design-Modus und dem Prototypen-Modus. Der Anwender kann zwischen diesen Ansichten im oberen Tab-Bereich wechseln.

Neuerdings verfügt Adobe XD über die Möglichkeit, Plugins zu installieren. Leider ist der Umfang dieser Plugins noch sehr gering. Aber sicherlich werden wir in den kommenden Jahren noch mehr Plugins für Adobe XD erhalten.

Für Adobe gibt es auch ein Mirror-Programm, welches in den App Stores verfügbar ist, sowie die Möglichkeit, *Zeplin.io* zu nutzen. Mit Zeplin können Sie erstellte Layouts, Assets und Styleguides samt Komponenten an Ihre Entwickler übermitteln.

Komponenten

Adobe XD verfügt ebenso über Symbol-Funktionalitäten wie Sketch, jedoch werden diese Symbole Komponenten genannt. Diese verhalten sich aber anders als jene in Sketch. So gibt es eine Master-Komponente und Instanzen. Instanzen sind Kopien der Komponenten und enthalten nur die rudimentären Informationen, wie Position oder Schriftart. Overrides können Sie direkt in den Instanzen selber machen. Hier finden Sie ein gutes Video, das die Funktionalität beschreibt: <https://www.youtube.com/watch?v=PCICHoZDxhM>. Mir persönlich gefällt die Lösung von Sketch besser, da jedes Symbol gleich ist und nicht zwischen Master und Instanz unterschieden wird.

Stile

Wie in Sketch auch können Farb-, Form- und Textstile erstellt werden. Ebenso wird hier mit Artboards gearbeitet, und diese können auch wie in Sketch zu einem einfachen Prototypen verknüpft werden.

Overrides

Overrides funktionieren in Adobe XD anders als in Sketch. In Adobe XD können Sie in den Instanzen die Inhalte ändern, ohne dies im Inspektor zu tun. Im Grunde ist es so, als würden Sie ganz normal arbeiten und keine Dropdowns bedienen. Textänderungen werden direkt im Textfeld gemacht, Farbänderungen ebenso.

- ▶ verfügbar: macOS, Windows
- ▶ offizielle Website: www.adobe.com/de/products/xd.html

9.7.5 Figma

Figma ist das erste All-in-one-Design-Tool, welches im Browser genutzt wird. Hierbei können Teams zusammen in einem Dokument arbeiten. Sie sehen sogar den Cursor Ihres Teammitglieds. Ebenso bietet Figma auch eine App ihres Tools an, welche jedoch in einem Browser-Container abläuft.

Sicherlich liegt die Zukunft von Apps in Browser-Technologien, jedoch bin ich im Falle von Figma noch nicht ganz überzeugt, sind es doch Performance-Probleme, die das Arbeiten mit Figma ein wenig trüben. Jedoch hält Figma viel tolle Features bereit, die sicherlich auch demnächst in anderen Programm-Tools enthalten sein werden.

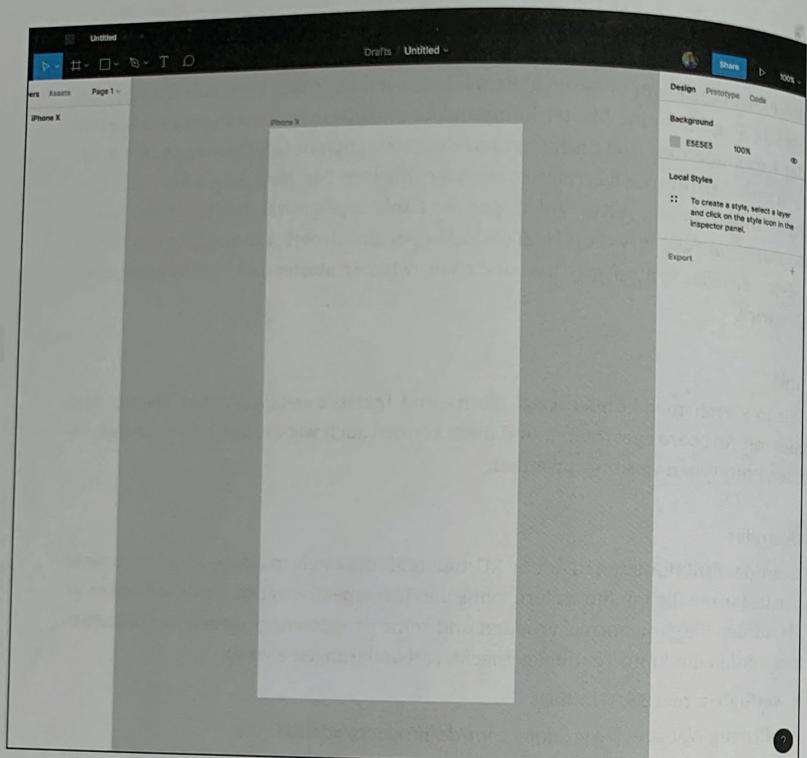


Abbildung 9.28 Figma Nutzeroberfläche

Komponenten

In Figma gibt es wie in den Programmen zuvor Komponenten bzw. Symbole. Auch hier arbeitet Figma ähnlich zu Adobe XD und nutzt eine Master-Komponente und die jeweiligen Instanzen. Änderungen an der Master-Komponente sorgen für Änderungen aller Instanzen. Änderungen an Instanzen bleiben auch nur bei Instanzen.

Overrides

Bei folgenden Objekten sind Overrides möglich: Text-Eigenschaften, Farbfüllungen, Konturen, Effekte und – das ist besonders interessant – Bounding Box. Ausgeschlossen von Overrides sind Größe, Positionen, Rotation, Constraints, Layer Hierarchy, Kurvenpunkte und -position. Overrides werden wie auch in Adobe XD auf der Bühne direkt gemacht. Hier finden Sie Näheres zu Komponenten unter Figma: <https://www.youtube.com/watch?v=k74IrUNaJVk>.

Stile

Ähnlich Sketch oder Adobe XD verfügt Figma über Farb-, Form- und Textstile. Hier erfahren Sie mehr über Stile in Figma: www.youtube.com/watch?v=gtQ_A3imzsg.

Weitere Features

Sie haben die Möglichkeit, mit anderen Team-Mitgliedern an einem Projekt zu arbeiten. Sie sehen in der Tat, woran die anderen Mitglieder arbeiten, und befinden sich im selben Dokument. Sollten Sie Google Docs kennen, dann ist es genauso.

Das Bearbeiten von Vektorgrafiken erreicht bei Figma eine neue Dimension und wurde so bisher nicht von den großen Grafiktools umgesetzt. Ich hoffe, andere Anbieter schneiden sich davon eine Scheibe ab. Der Clou dabei: Ein Vektorpunkt kann mehrere Verknüpfungen haben. Wer sich dieses sogenannte Vector Networks mal anschauen möchte: www.youtube.com/watch?v=b-xDRjf5B-8.

Figma verfügt ebenso über eine Komponenten-Library. Hierbei können Team-Mitglieder neue Komponenten hinzufügen oder alte verbessern bzw. überarbeiten samt Versionskontrolle. Ebenso ein inkludiertes Hand-Off für die Entwicklung.

- ▶ verfügbar: macOS, Windows, Linux
- ▶ offizielle Website: www.figma.com

9.7.6 Mirror-Programme

Sketch verfügt über einen sogenannten *Mirrormodus*, und auch Adobe bietet mit Adobe XD for iOS/Android ein Mirror-Programm an. Mit der Mirror-Funktion können Sie sich Ihre aktuellen Interface-Designs direkt auf Ihr mobiles Endgerät spiegeln und live Anpassungen im Design anzeigen lassen.



Abbildung 9.29 Mirror-Applikationen wie Sketch Mirror, Adobe XD, Skala Preview oder Crystal erlauben das Begutachten des Designs direkt auf dem mobilen Gerät.

Im Grunde handelt es sich um einen erweiterten Bildschirmmodus. Alle Änderungen, die Sie im Layout auf dem Desktop-Bildschirm vornehmen, werden aktualisiert auf dem Endgerät dargestellt. So können Sie quasi auf dem Smartphone oder Tablet designen. Es gibt auch andere Entwicklerstudios, die solche Lösungen bieten, z. B. Skala Preview von Bjango.

- ▶ Skala Preview: <https://bjango.com/mac/skalapreview>
- ▶ Adobe XD for iOS: <https://itunes.apple.com/de/app/adobe-experience-design-preview-your-prototypes/id1146597773?mt=8>

- ▶ Adobe XD for Android: <https://play.google.com/store/apps/details?id=com.adobe.sparklerandroid&hl=de>
- ▶ Sketch Mirror für iOS: <https://itunes.apple.com/de/app/sketch-mirror/id677296955?mt=8>
- ▶ Crystal für Sketch unter Android: <https://play.google.com/store/apps/details?id=com.smithyproductions.crystal&hl=de>

Der Vorteil solcher Mirror-Programme liegt klar auf der Hand: Farben, Formen, Schriften und Schriftgrößen werden direkt auf dem Endgerät dargestellt und können also im »echten Kontext« begutachtet werden. So wird der klassische Fehler, nur auf dem Desktop-Monitor zu gestalten und sich am Ende zu wundern, dass Schriften zu klein, Farben falsch dargestellt oder Formen zu klein oder zu groß sind, vermieden.

Die Größe der Bedienelemente ist eines der entscheidenden Kriterien bei der Touch-Bedienung. Mit solchen Mirror-Programmen kann die Größe exakt ermittelt werden, und insgesamt bekommen Sie als Designer so ein besseres Gefühl für die Gestaltung und Bedienung der App.

Ob Adobe XD, Figma oder Sketch – egal, welche Programme Sie für Ihre Gestaltung bevorzugen, entscheiden Sie sich für die Software, die Ihren Bedürfnissen am besten entgegenkommt. Ich persönlich bin auf den Bohemian-Sketch-Zug aufgesprungen. Für mich ist in dieser Software der UI-Design-Workflow am besten gelöst.

9.8 Tools für das Rapid Prototyping

Neben dem High- oder Low-Fidelity-Prototypen gibt es das sogenannte *Rapid Prototyping*. Im Rapid Prototyping werden zum einen Micro-Interaktionen für Custom Controls und zum anderen komplette Screens entwickelt. An sich handelt es sich dabei um High-Fidelity-Prototypen, die sich jedoch nur auf einen spezifischen Teil der App konzentrieren. Es gibt unzählige Applikationen oder Tools, die Sie beim Erstellen von Low- oder High-Fidelity-Prototypen unterstützen. Von einfachen Programmen, die selbst solche Prototypen generieren, bis zu komplexen Tools, für deren Verwendung Programmierkenntnisse benötigt werden, ist alles zu finden.

9.8.1 POP – Prototyping on Paper

POP (Prototyping on Paper) von Marvel App ist ein Tool für das iPhone, mit dem man Aufnahmen von einzelnen Seiten eines Papier-Prototyps logisch miteinander verlinken kann. Alternativ können auch Designs geladen und ebenso miteinander verlinkt werden. Ebenso können Sie direkt auf dem Smartphone Ihre App zeichnen. Zudem können hier neben Taps auch Swipe-Gesten simuliert werden. Andere Screens kön-

nen also einfach eingeblendet oder hereingefahren werden. Zwar ist der verfügbare Animationsumfang recht gering, dennoch reicht er, um ein Gefühl dafür zu bekommen, ob der geplante User Flow gut oder schlecht funktioniert.



Abbildung 9.30 Mit POP lassen sich Aufnahmen von einzelnen Seiten eines Papier-Prototyps logisch miteinander verlinken oder direkt auf dem Gerät erstellen.

POP ermöglicht Low-Fidelity-Prototypen mit eingeschränktem Umfang. Es eignet sich nicht für individuelle Lösungen und Konzepte. So können keine scrollbaren Listen oder interaktive Elemente dargestellt werden, lediglich einseitige Layouts. Das Tool erfordert vom Anwender keinerlei Programmierkenntnisse. Es ist einfach zu bedienen, und man kann recht schnell einen einfachen Prototypen entwickeln.

- ▶ verfügbar für: Webapplikation, iOS, Android
- ▶ offizielle Website: <https://marvelapp.com>

9.8.2 Pixate

Pixate ist ein Prototyp-Entwicklungsprogramm, das über einen größeren Funktionsumfang verfügt als POP. Mit Pixate können Sie schnell interaktive Prototypen für mobile Endgeräte erstellen. Dabei können Sie individuelle Listen und zusammenhängende Animationen erstellen. Entwickeln Sie z. B. ein Swipe-to-Choose-Elemente. Auch bietet das Pro-

gramm einen Mirror-Modus an, mit dem die erzeugten Prototypen direkt auf das Endgerät übertragen und getestet werden können.

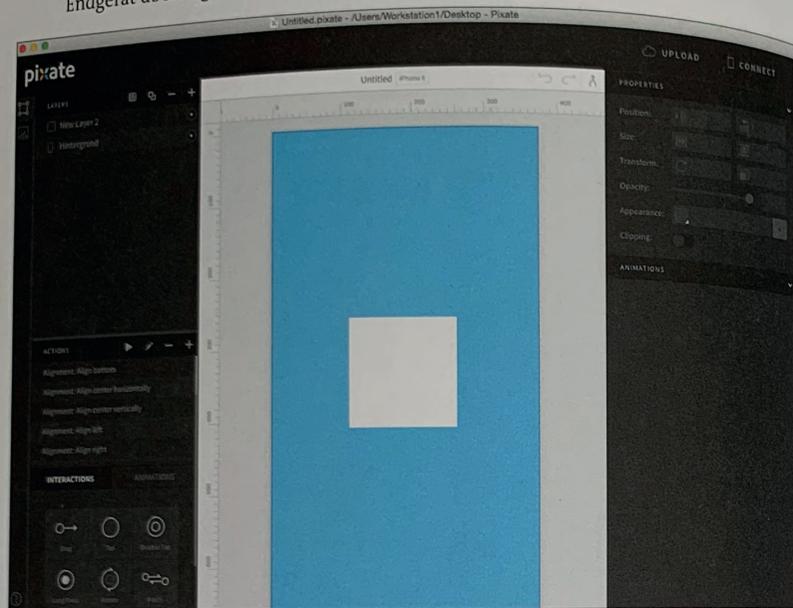


Abbildung 9.31 Mit Pixate können interaktive Prototypen für mobile Endgeräte erstellt werden.

Pixate ermöglicht es, High-Fidelity-Prototypen zu entwickeln, die mehr können, als nur Screens anzuzeigen.

In Pixate lässt sich zwar vieles über Drag & Drop erledigen, um einen umfangreichen Prototypen zu erstellen, benötigen Sie jedoch ein wenig Programmierkenntnisse. Pixate macht es allerdings sehr einfach, das Nötige zu erlernen. Im Netz finden sich ausführliche Tutorial-Videos und Beispielprojekte zum Herunterladen.

- ▶ verfügbar für: macOS, Windows
- ▶ offizielle Website: www.pixate.com

9.8.3 Framer.js

Framer.js ist ein Open-Source-JavaScript-Framework für Rapid Prototyping. Es handelt sich also um ein Prototyp-Tool, das Programmierkenntnisse erfordert. *Framer.js* bietet neben dem Open-Source-Framework und einer Bedienoberfläche, die Adobe XD sehr ähnlich ist, eine kleine, aber feine Entwicklungsumgebung mit dazugehörigem Viewer. Ebenso gibt es Mirror-Programme für die jeweiligen Endgeräte.

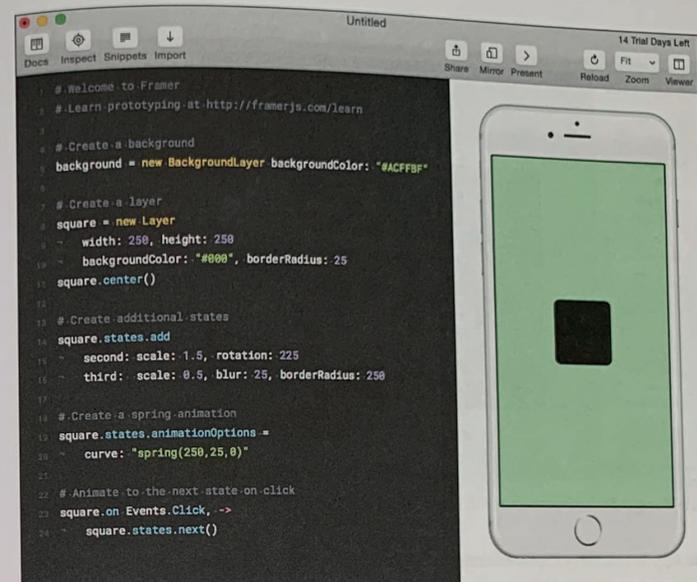


Abbildung 9.32 Screenshot Framer.js

Der Einstieg in das interaktive Prototyping mit Framer ist gar nicht so schwer, wie man vermuten würde. Sie gestalten Ihre Applikation und können diese in einfache Prototypen umwandeln. Dabei können Sie Animationen oder Transitions programmieren oder fertige Module nutzen, um Ihr Ziel zu erreichen. Komplexere Animationen und Funktionen lassen sich dann mit JavaScript realisieren. Wenn ausreichend Programmierkenntnisse vorhanden sind, können sehr komplexe Animationen und Controls umgesetzt werden.

Einen Nachteil hat Framer: Die entwickelten Prototypen sind nur auf WebKit-Browsern wie Safari oder Chrome lauffähig. Mit Firefox sind diese Prototypen aktuell nicht kompatibel.

- ▶ verfügbar für: macOS, Windows (demnächst)
- ▶ offizielle Website: www.framer.com

9.8.4 Flinto for Mac

Flinto for Mac ist eine macOS-Applikation, mit der sich High-Fidelity-Prototypen entwickeln lassen. Hier können komplette User Flows abgebildet werden. Auch können Artboards von Sketch importiert und animiert werden.



Abbildung 9.33 Flinto-for-Mac-Oberfläche

Die App verbindet einzelne Screens miteinander, sodass Sie eine einfache Klickdemo erhalten. Anschließend können Sie den Übergang zwischen den zwei Screens mit dem sogenannten Transition Designer definieren, wobei Sie die Elemente auf dem Screen in vielfältiger Weise verändern können. So sind Rotationen, Vergrößerungen und Verkleinerungen möglich.

Ebenso sind 3D-Manipulationen der Elemente möglich, und es können sogenannte Behavior-Objekte erstellt werden. Sie stellen quasi eine weitere Interaktionsebene zur Verfügung, die vergleichbar mit den Symbolen unter Sketch ist.

Diesen Prototypen können Sie mit der entsprechenden App nun auf dem Smartphone testen. Durch den Import und die sehr einfache Nutzung ist Flinto eines der besten Tools für das Erstellen von High-Fidelity-Prototypen. Ein weiterer sehr großer Vorteil ist, dass Sketch-Dateien mit entsprechendem Flinto-Plugin problemlos in Flinto importiert werden können. Ein Klick reicht aus, und in wenigen Sekunden stehen die Designs in Flinto zur Verfügung.

- ▶ verfügbar für: macOS
- ▶ offizielle Website: www.flinto.com/mac

9.8.5 Principle

Principle ist eine Mac-Applikation zum Erstellen von High-Fidelity-Prototypen. Während bei Flinto nur rudimentäre Animationen zwischen den Screens erstellt werden

können, können Sie in Principle komplexe Animationen gestalten. Diese werden mittels einer Timeline dargestellt, auf der Sie die Animationen genau definieren und gestalten können.

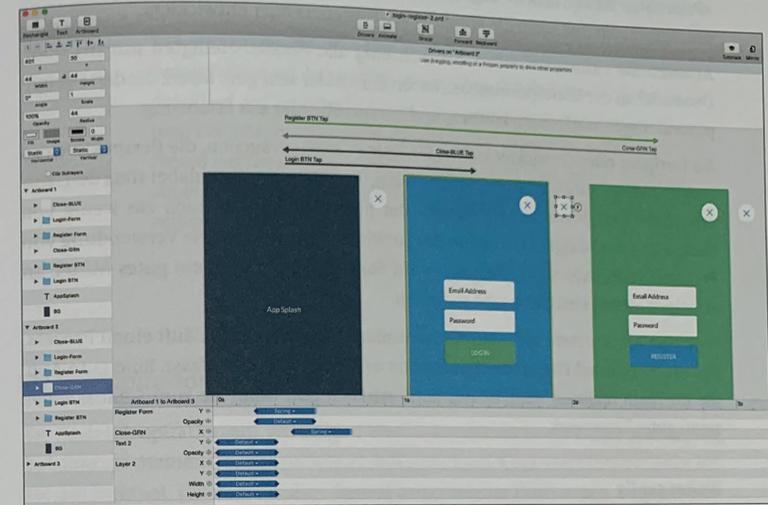


Abbildung 9.34 Principle-App für Mac

Neben der Animations-Timeline gibt es eine weitere Timeline über dem Hauptbereich, die sogenannte Drivers enthält. Mit Drivers lassen sich Animationen auf Basis von Gesten o. Ä. erstellen. Sogar das Erstellen von Micro-Interaktionen ist möglich, ebenso wie die Gestaltung einer Swipe-to-Choose-Geste wie in *Tinder*. Ebenso sind keine 3D-Animationen möglich.

- ▶ verfügbar für: macOS
- ▶ offizielle Website: <http://principleformac.com>

9.8.6 Design- und Entwicklungsperspektive

Als App-Designer müssen Sie sich zwangsläufig mit der Perspektive des Designers und der des Entwicklers auseinandersetzen.

Wunsch der Designerseite ist es, eine ästhetisch ansprechende App umzusetzen, die gefällig animiert ist und dem Auge schmeichelt. Die Entwicklerseite wünscht sich einen sauber strukturierten Quellcode, der gut kommentiert und flexibel genug ist, um neue Funktionen einfach zu implementieren. In der Regel kommt es zwischen diesen beiden Perspektiven zu Konflikten und zu manchmal schlechten Kompromis-

sen, die oft nicht zugunsten des Nutzers ausfallen und sich negativ auf die User Experience auswirken.

»Prioritäten lenken Leben«, sagt ein Sprichwort. Die Priorität einer App sollte immer der Nutzer sein und nicht die Wünsche des Designers oder Entwicklers.

Arbeiten Sie zusammen, holen Sie frühzeitig alle Projektbeteiligten mit ins Boot. Denn auch in der Konzeptionsphase haben Entwickler sehr gute Ideen, die den Designprozess ungemein fördern können, und ich spreche hier aus Erfahrung.

Als Designer oder Entwickler sollten Sie immer auch versuchen, die Perspektive des jeweils anderen einzunehmen, und im Fokus Ihrer Bemühungen dabei stets den Nutzer haben. Im optimalen Fall versteht der Entwickler immer auch ein wenig von Design und der Designer etwas von der Entwicklung von Apps. Die Verwendung von Personas (siehe Abschnitt »Personas« auf Seite 216) ist hierbei ein gutes Mittel, an strittigen Stellen einen Konsens zu finden.

Eine gute App entsteht nicht aus Kompromissen, sondern durchläuft einen Prozess, der wiederkehrend Phasen der Konzeption und des Testens umfasst. Eine gute App, die im Team realisiert wird, ist nur dann wirklich gut, wenn sie zum Schluss eines zeigt: gutes Teamwork.

Weiter geht's

In den folgenden vier Kapiteln nehmen wir das grafische App-Design unter die Lupe. Wir beschäftigen uns neben den Trends und Stilrichtungen mit Grafiken, Icons, Typografie, Formen und Farben.

Kapitel 10

Inspiration und Trends

Sich immer wieder Inspiration zu holen, ist auch während der Gestaltungsphase einer App wichtig. Bevor ich in den Kapiteln 11, 12 und 13 detailliert auf Typografie, Farbe, Icons und Bilder in der App-Gestaltung eingehe, möchte ich in diesem Kapitel einen Blick auf aktuelle Trends werfen und Ihnen einige Hinweise und Tipps geben, die Ihnen dabei helfen, kreativ zu bleiben und nicht den Spaß an der Sache zu verlieren.

10.1 Inspiration im Netz

Sie finden Inspiration auf vielfältigen Wegen. Bei mir reicht ein kurzer Spaziergang, bei dem ich meinen Gedanken freien Lauf lassen kann. Konkretere Inspirationen bietet das Internet. Der regelmäßige Besuch verschiedener Plattformen, Blogs oder bestimmter Newsfeeds bringt stets neuen Input. In diesem Abschnitt werde ich Ihnen einige Websites vorstellen, auf denen ich mir regelmäßig Inspiration hole.

Wenn es um neue Features und Technologien geht, sind gerade die jährlich stattfindenden Developer-Konferenzen interessant. Bei uns im Büro ist das sogar ein Gemeinschaftsereignis: Wir schauen uns den Livestream des Events zusammen in unseren Büroräumen an und essen dabei Pizza.

Für Apple-Produkte ist die Worldwide Developer Conference (WWDC) maßgeblich und für Google-Produkte die Google I/O. Die Präsentationen sind später auch im Internet verfügbar.

- ▶ WWDC 2019: <https://developer.apple.com/wwdc19/>
- ▶ Google I/O 2019: <https://events.google.com/io/>

Es ist empfehlenswert, diese Videos und die Hauptpräsentation anzuschauen. Das inspiriert und hilft Ihnen, auf dem Laufenden zu bleiben.

10.1.1 Inspiration für Designs

Farbe, Form, Muster, Anordnung, Raster, Kontrast – auf zahlreichen Websites können sich App-Designer Inspiration für die Gestaltung ihrer App holen. Ein paar dieser Sites möchte ich hier nun vorstellen.