

hosteurope.de

Design Systeme organisieren – UX/UI Tools im Überblick

19-23 minutes

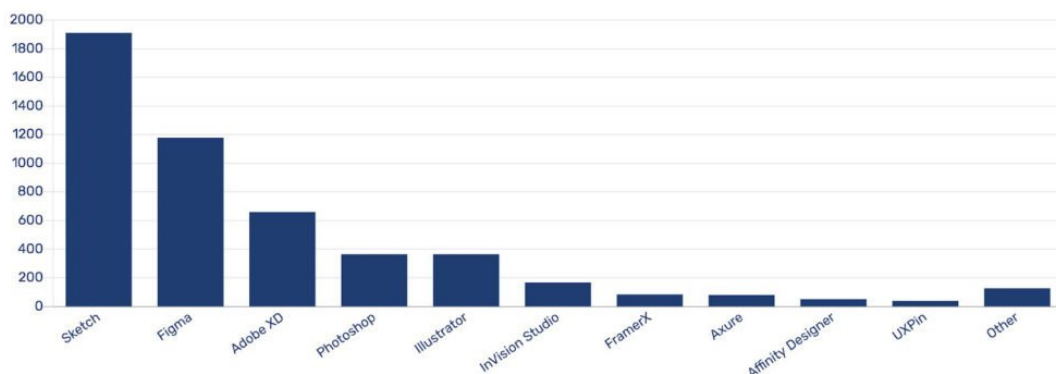
Design Systeme sind aktuell das zentrale Thema für viele Teams, die mit der Erstellung bzw. Weiterentwicklung digitaler Produkte beschäftigt sind. Alle relevanten UI-Tools integrieren Funktionen im Zusammenhang mit Design Systemen oder entwerfen Dienste zur besseren Zusammenarbeit zwischen Designern, Stakeholdern und Entwicklung. Verschaffen wir uns einen Überblick, was die Tools leisten und wo es noch Lücken gibt.

Welche UX/UI Tools wir uns anschauen

Dieser Artikel beschäftigt sich mit UI-Tools, die im visuellen Design-Prozess keine Coding-Kenntnisse voraussetzen, sondern grafisch arbeiten und versuchen, die Übergabe an die Entwicklung zu optimieren. In diesem Markt haben die Programme [Sketch](#), [Figma](#) und [Adobe XD](#) momentan die höchste Relevanz.

UI Design Tools

"Which tools do you use for interface design?"



Design Tools-Umfrage (Quelle: <https://uxtools.co/survey-2019/#ui-design>)

Die Motivation hinter der Entwicklung von Design Systemen

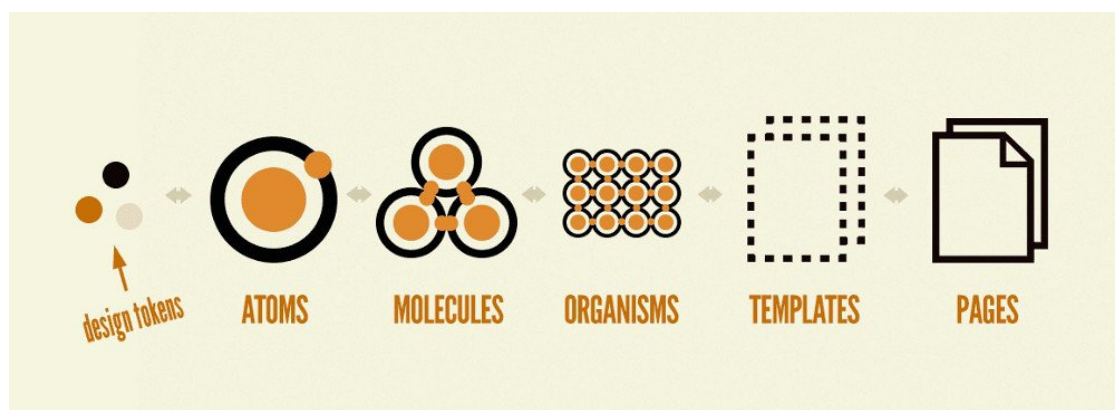
Die meisten Teams etablieren Design Systeme, um die Konsistenz von Benutzeroberflächen sicherzustellen – häufig über verschiedene digitale Produkte desselben Unternehmens hinweg. Dazu werden die verschiedenen Bestandteile eines Designs zentral für alle Personen verfügbar gemacht. Teams können so effizienter arbeiten, da auf bereits vorhandene Elemente zurückgegriffen werden kann. Auch eine Skalierung des Design Systems ist problemlos möglich, so dass Erweiterungen und Überarbeitungen leicht möglich sind.

Eine weitere Motivation kann das Aufbrechen von Silo-Strukturen in Unternehmen sein. Verschiedene Projektteams arbeiten gemeinsam mit dem Design System und entwickeln nicht versehentlich parallel ähnliche Lösungen.

Component Library, UI Kit oder Design System?

In elementarer Form deckt ein Design System mindestens folgende Themen ab, die allesamt in Sketch, XD und Figma in Form sogenannter Bibliotheken organisiert werden können:

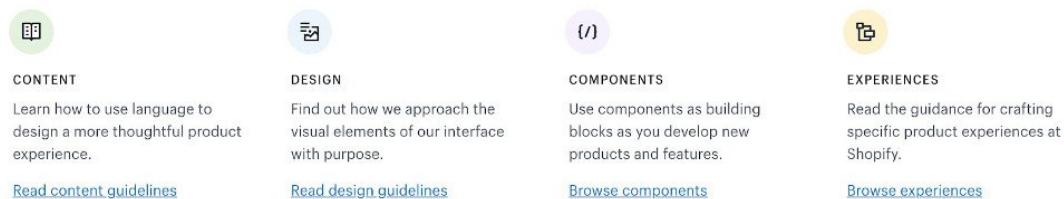
- **Design Tokens:** Übergreifende Charakteristika des Erscheinungsbilds wie Farben, Schriften, Abstände, Bildsprache und Icons
- **Komponenten:** Unterschiedlich komplexe UI-Elemente wie Buttons, Cards, Header etc. Häufig angelehnt an [Atomic Design](#)
- **Templates:** Verschiedene Seitentypen, die sich aus Komponenten zusammensetzen



Visualisierung des visuellen Design Systems im Atomic Design (Quelle: bradfrost.com/blog/post/extending-atomic-design)

Diese drei Punkte ergeben genau genommen noch kein Design System, sondern eher ein UI Kit. Für ein »echtes« Design System fehlen zahlreiche Inhalte, die das Design als ganzes auszeichnen, jedoch nicht zwingend visuellen Charakter haben müssen. Unter anderem werden auch folgende Aspekte typischerweise in einem Design System beschrieben:

- Regeln zu Interaktionen und Animationen
- Tonalität und Sprache als Gestaltungsmittel
- Regeln zur Barrierearmut
- In Frontend-Code definierte Komponenten!



Kapitel im Design System »Polaris« von Shopify

An dieser Stelle gehen die Meinungen häufig auseinander. Populäre Vertreter wie Brad Frost argumentieren, dass [eine Sketch-Bibliothek noch kein Design System ist](#), da das Design System nur dann »fertig« ist, wenn daraus das finale Produkt gebaut werden kann. Das trifft auf ein UI Kit nicht zu – im Design System müsste dafür zwingend »Production Ready Code« enthalten sein.

Allerdings hört man auch gegenteilige Stimmen, die argumentieren, dass [ein UI Kit sehr wohl ein Design System darstellt](#), da die zentralen Punkte erfüllt sind. Nämlich konsistent zu gestalten und Zeit sowie Geld zu sparen.

»Real Single Source Of Truth« oder »Hand-Off«?

Die meisten Design Systemen unterscheiden nach wie vor zwischen Design

und Development. Viele Unternehmen und Agenturen propagieren zwar einen interdisziplinären Design/Development-Workflow, entwickeln dann aber doch zunächst im UX/UI-Design statische Design-Komponenten, die im zweiten Schritt von Frontend-Devs im Code nachgebaut werden.

Erst wenn Komponenten mit Code gebaut wurden, sieht man, wie sie tatsächlich aussehen und wie sie sich in verschiedenen Browsern verhalten. Auch wichtige Aspekte wie Performance, SEO und Accessibility können in einem UI-Tool nicht verlässlich beurteilt werden.

Üblicherweise entstehen so in Design Systemen zwei verschiedene Bibliotheken: Eine für visuelle Designer – das sog. UI Kit – und eine für Frontend-Entwickler – die Component Library.

Buttons



Buttons can be used to show the user's choice of options for actions and assign these to a clear hierarchy. The target page to be accessed or the action to be triggered is shown on the buttons by means of a label in Audi Type Screen Normal or else by an icon, or by a combination of the two. All buttons provide visual feedback when clicked or tapped in the form of a *response effect*

Typische Organisation am Beispiel des Audi Design Systems (Quelle: <https://www.audi.com/ci/en/guides/user-interface/introduction.html>)

Oben rechts kann die Komponente als Code oder als UI-Kit heruntergeladen werden.

Da der Workflow von Design in Richtung Entwicklung gedacht wird, müssen visuelle Änderungen im Code »nachgezogen« werden. Es gibt daher faktisch keine zentrale Stelle (Single Source Of Truth), an der das Design System definiert wird.

Einige Design-Teams nutzen daher Workflows, um im Code angelegte Komponenten in UI-Tools hineinzurendern. Alles wird zentral im Code gepflegt, und visuelle Abweichungen zum Design-Tool sind weitestgehend

ausgeschlossen.

Video: Designing with Code

Wohin sich die UI-Tools entwickeln

Die Design-Tooling-Branche entwickelt sich rasant weiter. Sketch, Figma und XD integrieren in sehr kurzen Abständen neue Features und übernehmen Funktionen der Konkurrenz. Folgende Bereiche des Design-Workflows versuchen die Tools abzubilden. Alle sind im Zusammenhang mit der Entwicklung von Design Systemen relevant:

- UI Design
- UX Design
- Hand-Off
- Kollaboration

Die drei Schwergewichte Figma, Sketch und XD decken alle Bereiche ab, ohne auf externe Software oder Plugins angewiesen zu sein. Ein genauer Blick offenbart jedoch, wie unterschiedlich ausgebaut die Funktionen in den einzelnen Programmen sind.

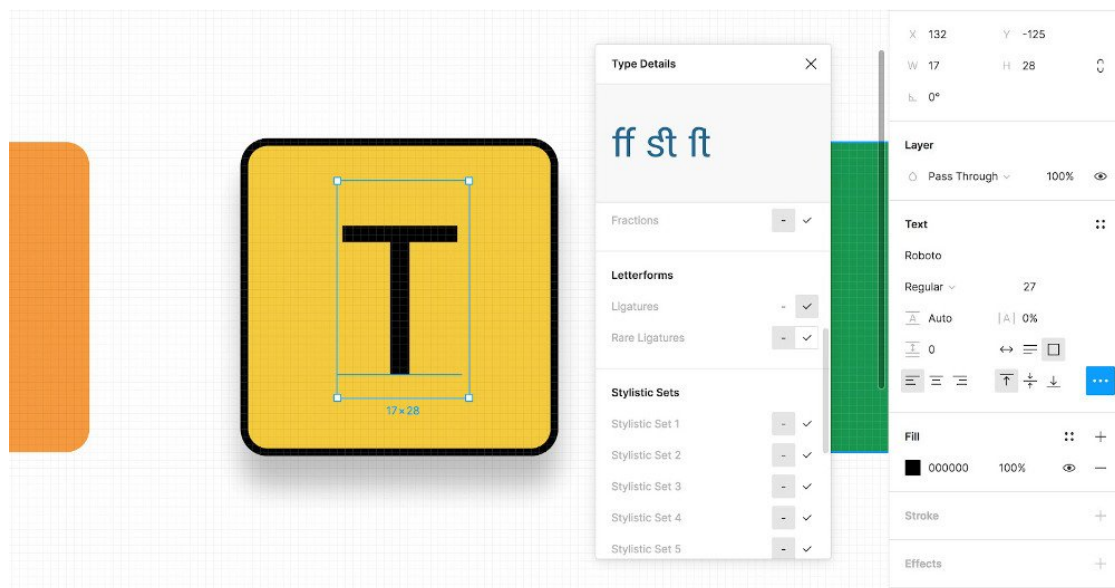
UI Design

UI Design-Funktionen sind für die Entwicklung von Design Systemen höchst relevant, da visuelle Perfektion und gute Organisationsmöglichkeiten von nahezu allen Teams gewünscht sind.

Erstellung von grafischen Elementen – Arbeit mit Fotos, Schrift, Icons usw.

In allen Tools können professionelle, vektorbasierte Benutzeroberflächen hergestellt werden. Die Integration von Schriften und Fotos stellt kein Problem dar, und in allen Tools können grafische Elemente wie Icons oder flächige Illustrationen hergestellt werden. Figma bietet grafisch, insbesondere

typografisch, einige Möglichkeiten, die man in den anderen Programmen vermisst. Generell ist Figma überraschend leistungsstark, was visuelle Details betrifft.



Typografie-Einstellungen in Figma

Adobe XD sticht aufgrund der Integration in die Creative Cloud von Adobe hervor. Wenn umfangreich mit in Photoshop oder Illustrator erstellten Ressourcen gearbeitet werden soll, kann der programmübergreifende Workflow von Adobe ein Vorteil sein.

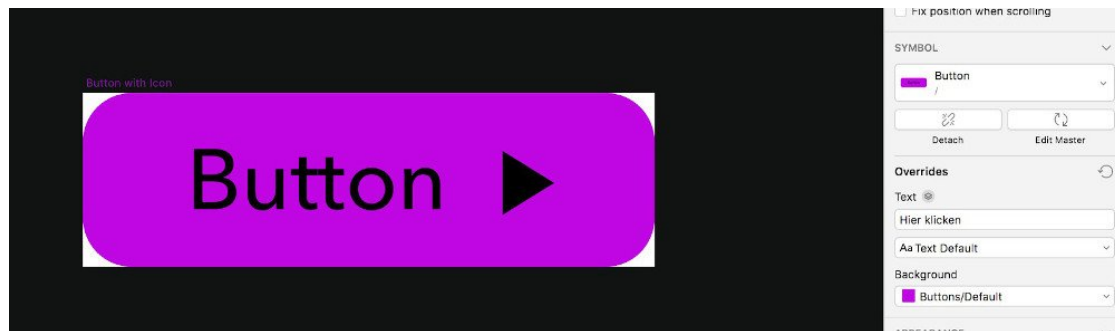
Organisation von wiederkehrenden Elementen und Stilen

Die zentrale Pflege von Design-Elementen wie Farben, Schriften und Komponenten ist grundsätzlich in allen Programmen möglich.

Adobe XD offenbart Schwächen bei verschachtelten Komponenten und einem sehr atomaren Ansatz. Einerseits ist die Benutzeroberfläche für große Mengen an Farben, Schriften und Komponenten nicht ausgelegt, andererseits erlaubt XD zu viele Freiheiten. Komponenten können intuitiv gestalterisch angepasst werden, was grundsätzlich zu begrüßen ist. Allerdings lassen sich in XD kaum Regeln für die Anpassung definieren. Es ist daher in XD leicht, vom Design System abzuweichen und Inkonsistenzen zu erzeugen.

Sketch ist für umfangreiche und verschachtelte Design Systeme sehr gut

geeignet – kann aber schnell tiefe und unübersichtliche Strukturen erzeugen. Das System muss sorgfältig gepflegt sein. Anpassungen an Komponenten (Overrides) werden kleinteilig über Eingabefelder in der Seitenleiste ermöglicht. Das ist weniger intuitiv und frei als in XD und Figma, aber es lassen sich die meisten Konfigurationsmöglichkeiten detailliert steuern.

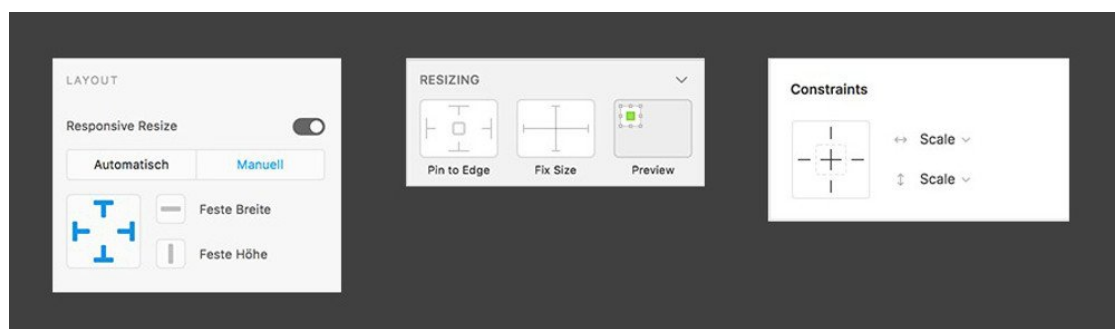


Overrides-Panel in Sketch für eine simple Button-Komponente

Figma bietet intuitivere Anpassungen als XD, aber weniger Steuerungsmöglichkeiten als Sketch. Eine Besonderheit von Figma ist die Möglichkeit, Komponenten direkt im System mit einer Beschreibung zu versehen. Auch die Organisation von Komponenten in Ordnerstrukturen ist vergleichbar mit Sketch.

Simulation des Browser-Verhaltens in Bezug auf Abstände, flexible Größen, variable Inhalte etc.

Alle Programme unterstützen das sog. »Resizing«. Gemeint ist damit, dass Objekte an übergeordneten Objekten wie Gruppen oder Komponenten ausgerichtet werden können und somit flexible Komponenten erzeugt werden können. Das Feature wird häufig im Zusammenhang mit Responsive Design erwähnt.



Resizing in Adobe XD, Sketch, und Figma (v.l.n.r.)

In Adobe XD ist Resizing automatisch aktiv, und das Tool erkennt, welches Verhalten gestalterisch gewünscht ist. Wenn das Programm sich irrt, kann manuell eingegriffen werden. In Sketch und Figma gibt es keinen Automatismus – alles muss manuell konfiguriert werden.

Bei der Reaktion auf textliche Anpassungen liegen Figma und Sketch momentan vorne. Komponenten können so konfiguriert werden, dass sie automatisch höher oder breiter werden, wenn der Inhalt sich ändert. Das funktioniert auch in XD. Allerdings reagieren in Sketch und Figma Objekte auf andere Objekte. Wenn die Überschrift also beispielsweise länger wird und umbricht, schiebt sie den Fließtext nach unten. Dieses Verhalten ist in Figma besonders gut umgesetzt.

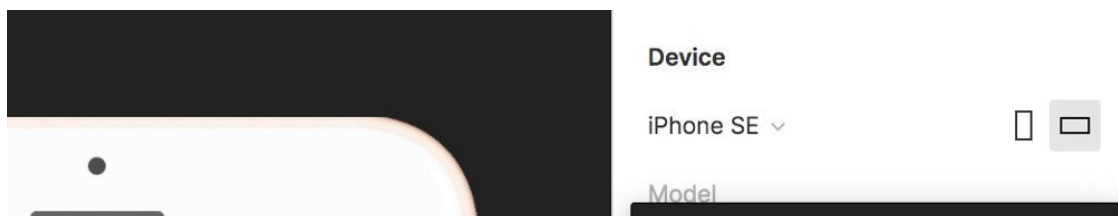
UX Design

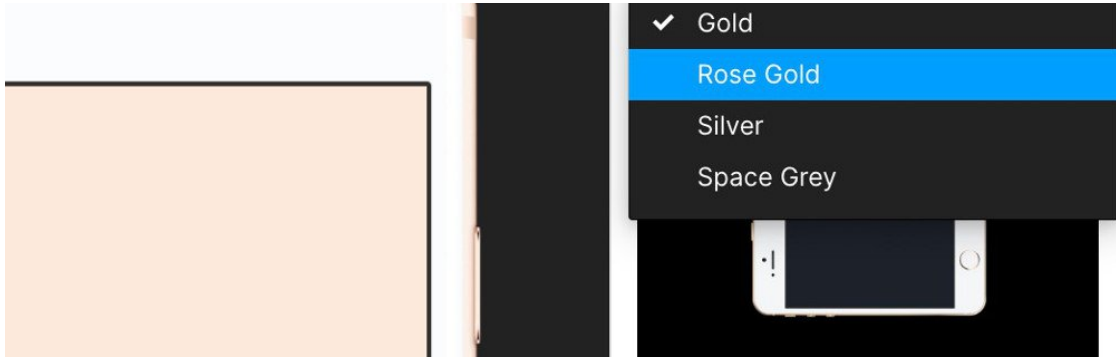
UX Design-Features ermöglichen u.a. das Erstellen von interaktiven Prototypen. Im Zusammenhang mit Design Systemen sind diese Features relevant, um Mikroanimationen und das Verhalten einzelner Layout-Elemente visualisieren zu können.

Verknüpfung von Screens zu interaktiven Prototypen

Alle drei Programme unterstützen die Verknüpfung von statischen Screens zu klassischen Prototypen, inkl. vertikalem Scrollen und der Fixierung von Elementen, die bei Scrollen im Vordergrund stehen bleiben sollen.

Sketch ist mit Abstand am schwächsten aufgestellt, was Prototyping betrifft. Die Übergänge zwischen einzelnen Screens sind sehr rudimentär, und selbst Scrolling ist schwach ausgebaut. Beim Scrollen sind fixierte Objekte nur im Vordergrund möglich.





Auswahl des Präsentationsgeräts in Figma

XD und Figma hingegen bieten sehr umfangreiche Prototyping-Funktionen und zahlreiche Animationsmöglichkeiten zwischen Screens. Es können kleinteilige, komplexere Animationseffekte erstellt werden. Figma erlaubt als einziges der drei Tools echtes horizontales Scrolling.

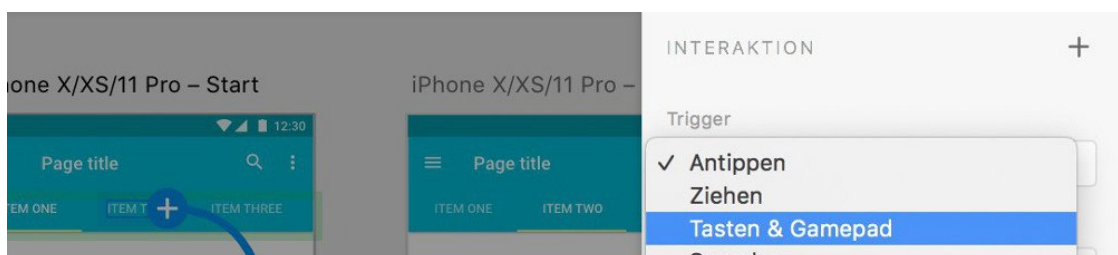
Interaktionen, States und Mikroanimationen

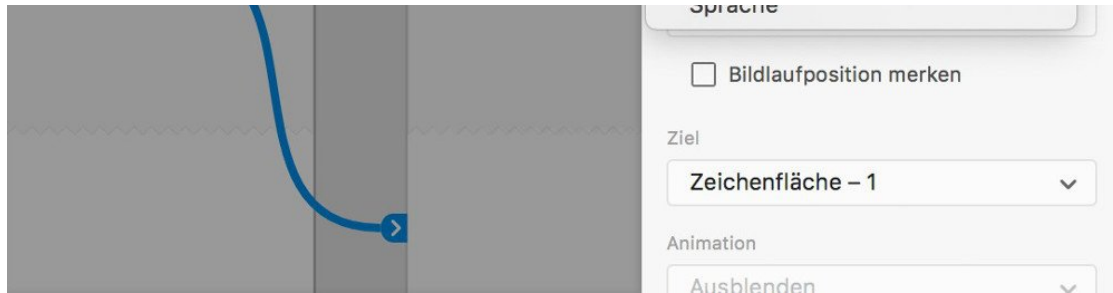
UI-Komponenten wie Buttons verfügen über Stadien – z. B. Mouse Over. Adobe XD unterstützt States für Komponenten und erlaubt es, im Prototypen auf dieselbe Screen zwischen diesen States zu wechseln. Mikroanimationen und Varianten von Elementen können problemlos abgebildet werden. Ein sehr praktisches Feature für die Erstellung von Design Systemen.

Sketch erlaubt detaillierte Konfigurationen für verschachtelte Komponenten. Somit können States selbstständig angelegt werden. Es ist nicht möglich, im Prototypen zwischen States zu wechseln, ohne dafür verschiedene Screens anzulegen. Figma verhält sich ähnlich wie Sketch, bietet aber durch bessere Prototyping-Möglichkeiten mehr Spielraum.

Trigger

Die sog. »Trigger« lösen Interaktionen im Prototypen aus. Die Programme unterscheiden sich hier spürbar in der Unterstützung.





Auswahl der Trigger in XD

Sketch kann nur per Klick zwischen Screens wechseln. Adobe XD unterstützt neben Klick und Mouse Over auch automatische Übergänge, Gesten, Tastatureingaben sowie Spracheingabe. Figma bietet *zusätzlich* noch Verlinkungen auf echte URLs sowie detaillierte Trigger wie MouseIn, MouseOut etc.

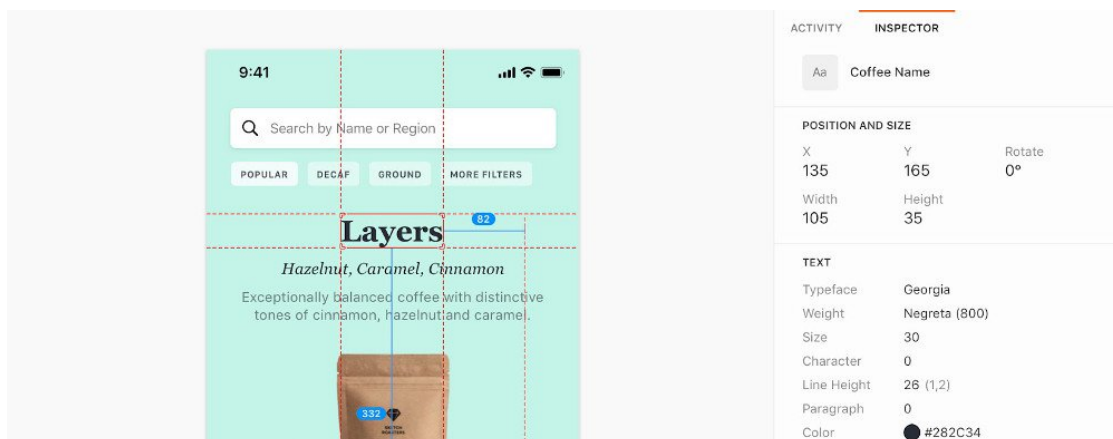
Hand-Off

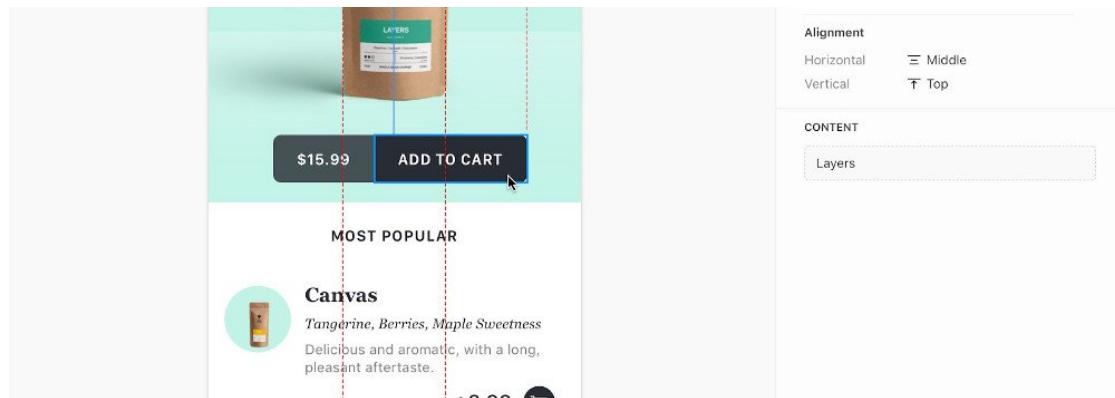
Ein Ziel von Design Systemen ist das Aufbrechen von Silo-Strukturen im Unternehmen. Eine optimierte Übergabe von Design an Entwicklung kann die Zusammenarbeit verbessern.

Übergabe von Code und Assets an die Entwicklung

Die Programme bieten hier einen sehr ähnlichen Funktionsumfang: Einzelne Grafiken, Screens und Elemente können in den gängigen Dateiformaten exportiert werden.

Für die Entwicklung kann das Dokument ebenfalls bereitgestellt werden, so dass Abstände, Texte, Farben, Schriften und weitere für die Umsetzung hilfreiche Informationen komfortabel ausgelesen werden können.





Hand-Off in der Sketch-Cloud (Quelle: blog.sketchapp.com/introducing-cloud-inspector-free-developer-handoff-in-the-browser-59917220334a)

Der Unterschied zwischen den Programmen besteht darin, wo Developer sich die Informationen beschaffen können. Sketch und XD bieten den sog. Inspector in der Cloud an. Figma läuft selbst bereits in der Cloud – hier können die Infos direkt im Programm in Erfahrung gebracht werden.

Präsentation und User Testing

Alle Programme bieten eine interne Vorschau-Funktion und eine Vorschau in der Cloud. Adobe XD ermöglichen den Export von Prototypen als Video. Figma unterstützt zudem eine Vielzahl an Mockup-Vorlagen, so dass Prototypen z.B. im Kontext eines bestimmten Smartphones präsentiert werden können. Die Präsentation wirkt dadurch echter.

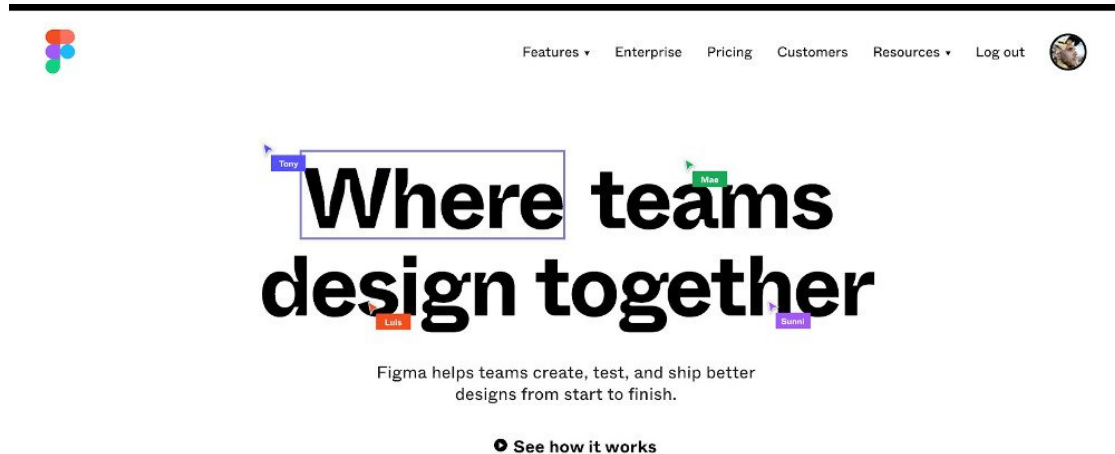
Die Präsentation und Benutzung von Prototypen auf echten Endgeräten ist ebenfalls in allen Tools möglich. Dazu muss das Gerät per Kabel/WiFi verbunden oder über die Cloud angesteuert werden. Sketch ermöglicht ohne Drittanbieter-Software leider nur eine Präsentation auf iOS-Geräten.

Kollaboration

Funktionen zur Zusammenarbeit sind essentiell um Design, Entwicklung, Marketing und Stakeholder zusammenzuführen. Das Ziel aller Programme ist Echtzeit-Kollaboration im selben Dokument, nach dem Vorbild von Figma. Im geöffneten Dokument können somit zeitgleich verschiedene Personen arbeiten.

Co-Editing

Für eine Zusammenarbeit in Echtzeit muss die Software entweder im Browser laufen oder für alle relevanten Plattformen verfügbar sein. Hier hat Figma klar die Nase vorne, da die Anwendung im Browser läuft und Co-Editing seit längerer Zeit zur Kernkompetenz zählt.



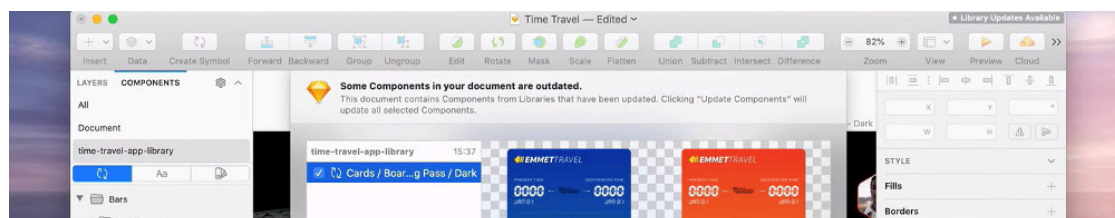
Adobe XD – Creative Cloud

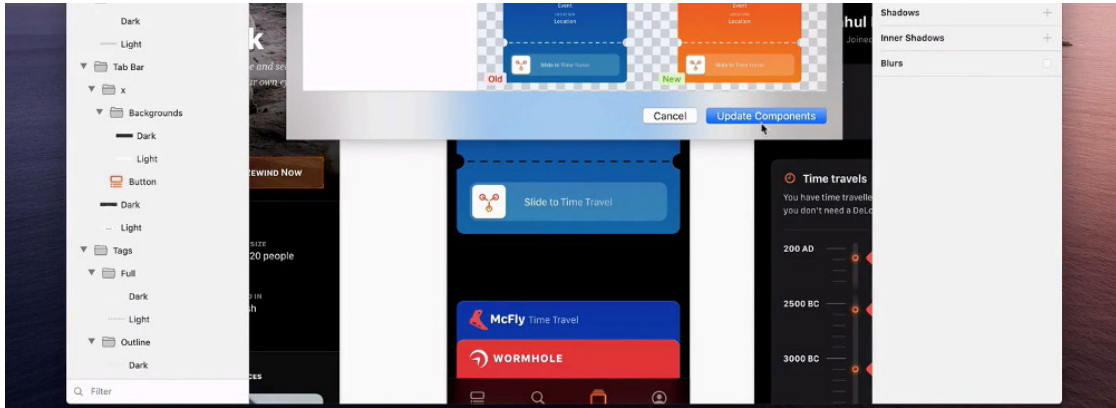
Adobe XD ermöglicht mittlerweile ebenfalls eine Zusammenarbeit in Echtzeit über die Creative Cloud.

Sketch hat das Feature angekündigt, bietet mit »Sketch for Teams« aktuell aber nur eine Synchronisation über die Sketch-Cloud.

Organisation von Bibliotheken

Bibliotheken – also Zusammenstellungen von Komponenten, Farb- und Schriftstilen – sind in allen drei Programmen möglich. Somit kann ein Design System zentral in einer Bibliotheksdatei organisiert und dann in anderen, abhängigen, Dateien genutzt werden. Über Berechtigungen kann festgelegt werden, wer die Bibliothek ändern darf. Sobald dort eine Änderung stattfindet, erscheint in allen abhängigen Dokumenten eine Information, und die Änderungen können übernommen oder verworfen werden.





Visualisierung einer aktualisierten Komponente in Sketch (Quelle: www.sketch.com/docs/libraries)

In allen Programmen kann grundsätzlich jedes Dokument als Bibliothek für andere Dokumente verwendet werden. In XD gehen jedoch wichtige Funktionen verloren, wenn die Bibliotheksdatei lokal abgelegt wird und nicht in der Cloud liegt. Das ist bei Sketch nicht der Fall. Bei Figma hingegen liegt ohnehin immer alles in der Cloud.

In Sketch for Teams und insbesondere in Figma lassen sich Berechtigungen und Projekt-Zuordnungen detailliert steuern.

Das Design System mit externen Tools weiterführen

Mit den internen Funktionen der Tools kann eine solide Grundlage für die Erstellung von Design Systemen geschaffen werden. Wenn man das Design System zunächst als UI-Kit aufbauen möchte, sind die Tools hervorragend geeignet und reichen ggf. alleine schon aus.

Dennoch möchte ich zum Abschluss einige populäre Tools nennen, die häufig in Kombination genutzt werden.

Zeplin

[Zeplin](#) ist ein Tool, das u.a. dafür geschaffen wurde, um Sketch-Dateien für Entwickler aufzubereiten. Mittlerweile integrieren alle Tools ähnliche Hand-Off-Funktionen selbstständig. Zeplin bietet eine automatische Organisation von Styleguides anhand der Struktur einer Bibliotheksdatei. Zudem stehen Kommunikations-Funktionen zur Verfügung. Mit der Funktion »Connected

Spaces« können in Zeplin verfügbare Design-Komponenten mit ihrem Code-Gegenstück verknüpft werden. Dieses Gegenstück kann z.B. in GitHub oder Storybook abgelegt sein. Auch hier besteht das Ziel darin, Design und Entwicklung zumindest visuell in einer Oberfläche zusammenzuführen.

Abstract

[Abstract](#) bietet u.a. Versionierung für Design-Komponenten, Kommunikations-Funktionen und eine Integration in Development- und Projektmanagement-Tools wie GitHub oder JIRA. Sketch und XD-Dateien können zu Abstract hochgeladen und dort zentral dokumentiert, diskutiert und freigegeben werden.

InVision DSM

InVision bietet mit [DSM](#) die Möglichkeit, die Inhalte von Sketch-Bibliotheken in einer Weboberfläche zentral abzulegen, zu dokumentieren und zu versionieren. Zudem bietet InVision DSM eine Hand-Off-Funktion, um Entwickler bei der Überführung von UI-Kits in Code-Komponenten zu unterstützen. Die fertigen Frontend-Komponenten können im externen Tool Storybook organisiert werden, was wiederum in DSM integriert ist. Somit können Designer und Entwickler in der gleichen DSM-Umgebung an UI-Kit und Code-Komponenten arbeiten.

Storybook

[Storybook](#) ist ein Tool für Frontend-Entwickler, um fertig gecodete Komponenten zu organisieren. Es bietet sehr umfangreiche Möglichkeiten der Anpassungen und Dokumentation, sowie des Testings von Komponenten.

UX/UI Tools – Fazit

Sofern man der Vorgehensweise mit einem UI-Kit und einer daraus abgeleiteten Component Library grundsätzlich etwas abgewinnen kann, sind alle drei Tools ähnlich gut geeignet und mit Plugins oder flankierenden

Diensten entsprechend der eigenen Anforderungen erweiterbar.

Adobe XD ist momentan meine Wahl, wenn das Projekt visuell stark und ggf. animiert sein soll, aber keine sehr kleinteilige und komplexe Designsystematik nach einem atomic Ansatz gewünscht ist. Insbesondere typografisch macht XD bei Design Systemen noch Probleme. Von den drei behandelten Tools ist XD wahrscheinlich am leichtesten zu erlernen.

Sketch spielt seine Vorteile in eben diesen kleinteiligen Systemen aus, wobei die Prototyping-Funktionen momentan wirklich sehr schwach sind. Wenn der Schwerpunkt des Projekts auf Prototyping liegt, würde ich Sketch nicht einsetzen. Wenn das Ziel jedoch darin besteht, umfangreiche Bibliotheken aufzubauen, kann Sketch sehr sinnvoll sein. Zum aktuellen Zeitpunkt sollte Sketch sich beeilen, Funktionen der Konkurrenz nachzurüsten.

Figma ist ein exzellenter Allrounder mit überraschend starken visuellen Funktionen und sehr guten Prototyping-Features. Wie eingangs beschrieben, ist das Programm durchaus populär – außerhalb der UX/UI-Branche aber in meiner Erfahrung weniger bekannt als Sketch und XD. Da Figma in allen Bereichen ziemlich stark aufgestellt ist, lohnt es sich allemal, das Tool genauer in den Blick zu nehmen. Die Einarbeitungsphase könnte sich hier jedoch etwas schwerfälliger gestalten.

Es bleibt spannend.