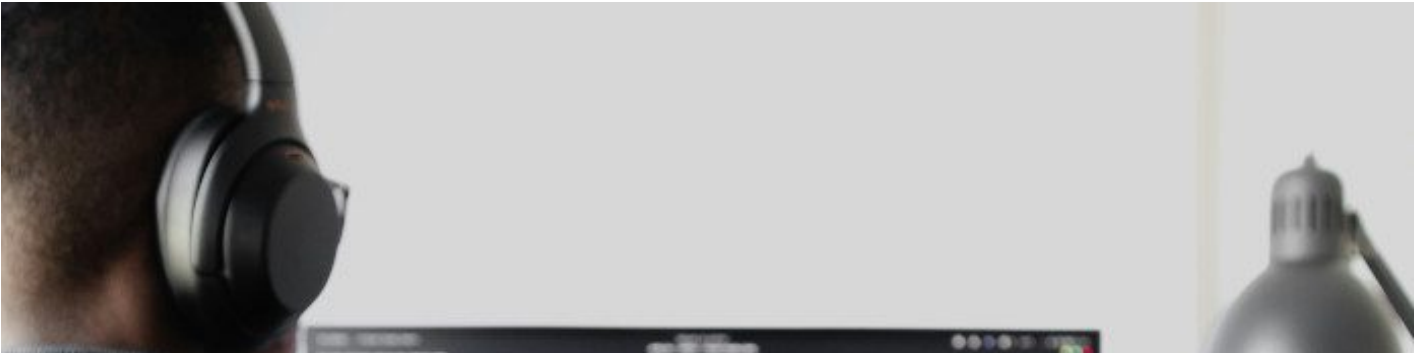




Figma für Entwickler: Dev Mode, Workflows und Best Practices

Jonas Hellwig am 2. Juni 2026
Lesezeit 10 Minuten

Mit Figma und dem integrierten Dev Mode lässt sich die Zusammenarbeit zwischen Design- und Development-Teams deutlich verbessern. Damit das auch funktioniert, müssen beide Seiten allerdings ihre Hausaufgaben machen. Die Qualität im Design muss stimmen, und die Entwicklerteams müssen die Features des Dev Modes kennen und nutzen.

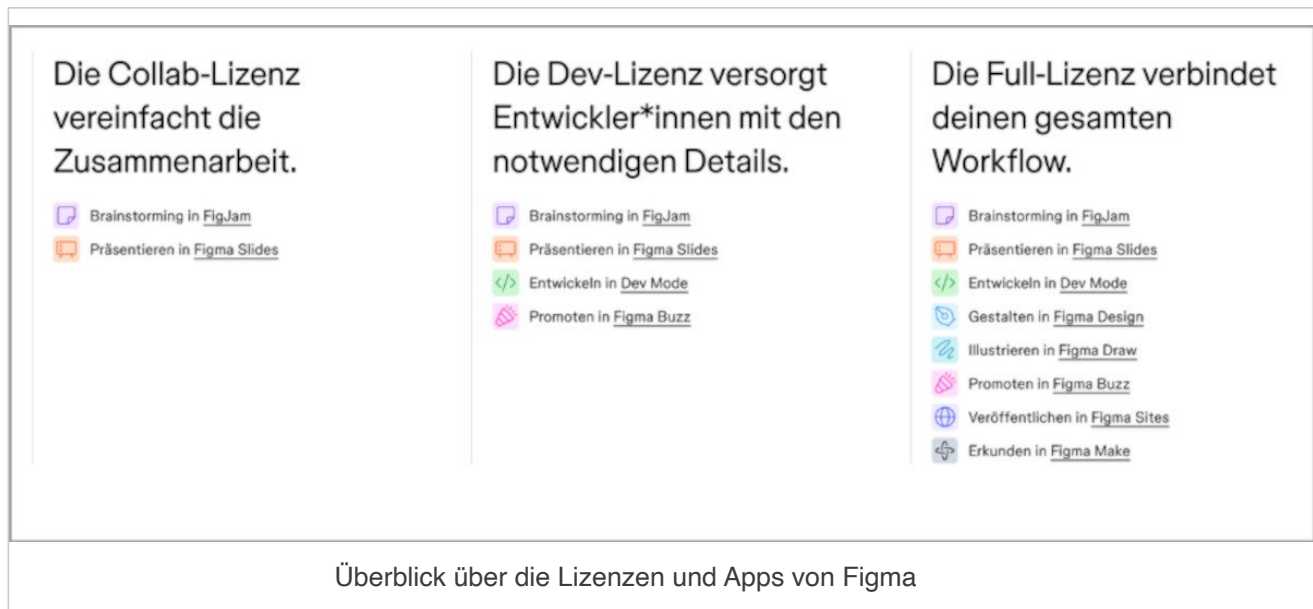
In diesem Artikel gebe ich Ihnen einen Überblick darüber, was Fignas Dev Mode leistet und wie er im Entwicklungsalltag eingesetzt werden kann.

Was ist Figma und der Figma Dev Mode?

Figma ist Anfang 2026 de facto zum Industry-Standard für digitales Design geworden. Besonders bei komplexen Projekten kann Figma seine Stärken voll ausspielen. Bei sehr kleinen Website-Projekten mit wenigen Seiten mag das Tool manchmal überdimensioniert wirken, aber sobald mehrere Personen involviert sind oder die Komplexität steigt, zeigt sich der Wert. Ein zentrales Merkmal von Figma ist, dass es vollständig in der Cloud läuft. Alle am Projekt beteiligten Personen nutzen daher gemeinsam dieselbe Datei in der neuesten Version.

Design und Entwicklung arbeiten im Idealfall also nicht nacheinander und ohne Kommunikation, sondern zusammen am gleichen Dokument und im direkten Austausch. Auch Projektmanager oder Stakeholder können direkt in die Datei schauen und den Fortschritt verfolgen. Diese Art der Zusammenarbeit reduziert Missverständnisse und beschleunigt den gesamten Prozess erheblich.

Figma ist heute eine ganze Suite von Design-Tools. In diesem Artikel konzentrieren wir uns auf Figma Design, das Tool für UX und UI Design mit starkem Fokus auf Frontend-Entwicklung. Hier gibt es integriert den sogenannten Dev Mode, mit dem Designs aus der Developer-Perspektive analysiert und umgesetzt werden können. Der Dev Mode bietet Funktionen, die speziell auf die Bedürfnisse von Entwicklerinnen und Entwicklern zugeschnitten sind.



Es gibt verschiedene Lizenzmodelle bei Figma, die auch Auswirkungen auf die im Dev Mode verfügbaren Features haben. Für Entwicklungsteams gibt es sogar reine Dev Mode Lizenzen, mit denen Sie auf die Designs zugreifen können, ohne eine vollwertige Designer-Lizenz zu benötigen. Das macht es für größere Teams erschwinglicher, alle Beteiligten einzubinden.

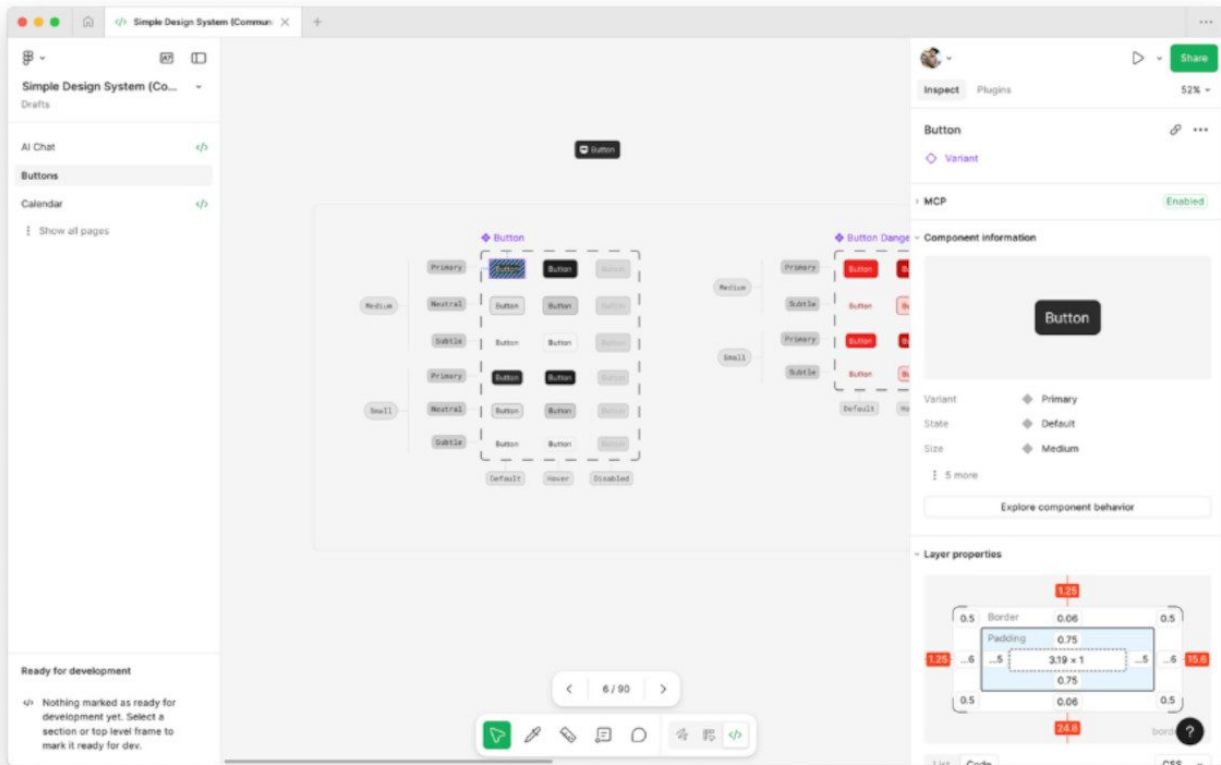
Was muss im Design vorbereitet sein?

Die Qualität dessen, was Ihnen im Dev Mode angezeigt wird, hängt maßgeblich von der Qualität der Design-Datei ab. Figma setzt voraus, dass bestimmte Konzepte und Techniken bekannt sind und auch konsequent angewendet werden.

Systematischer Aufbau

Nur wenn das Design systematisch aufgebaut ist, kann der Dev Mode seine volle Leistung entfalten.

Ein wichtiges Konzept ist die Design-Systematik nach dem Prinzip des Atomic Design. Das bedeutet, dass mit Komponenten, Komponenteneigenschaften (Varianten) und Bibliotheken in Figma gearbeitet wird. Einzelne kleine Elemente wie Buttons oder Input-Felder werden zu größeren Komponenten wie Formularen zusammengesetzt, die wiederum Teil noch komplexerer Strukturen werden können. Diese Hierarchie sorgt dafür, dass Änderungen zentral vorgenommen werden können und sich automatisch überall dort auswirken, wo die Komponente verwendet wird.



Eine Komponente im »Simple Design System« von Figma – einer kostenlosen Test-Bibliothek

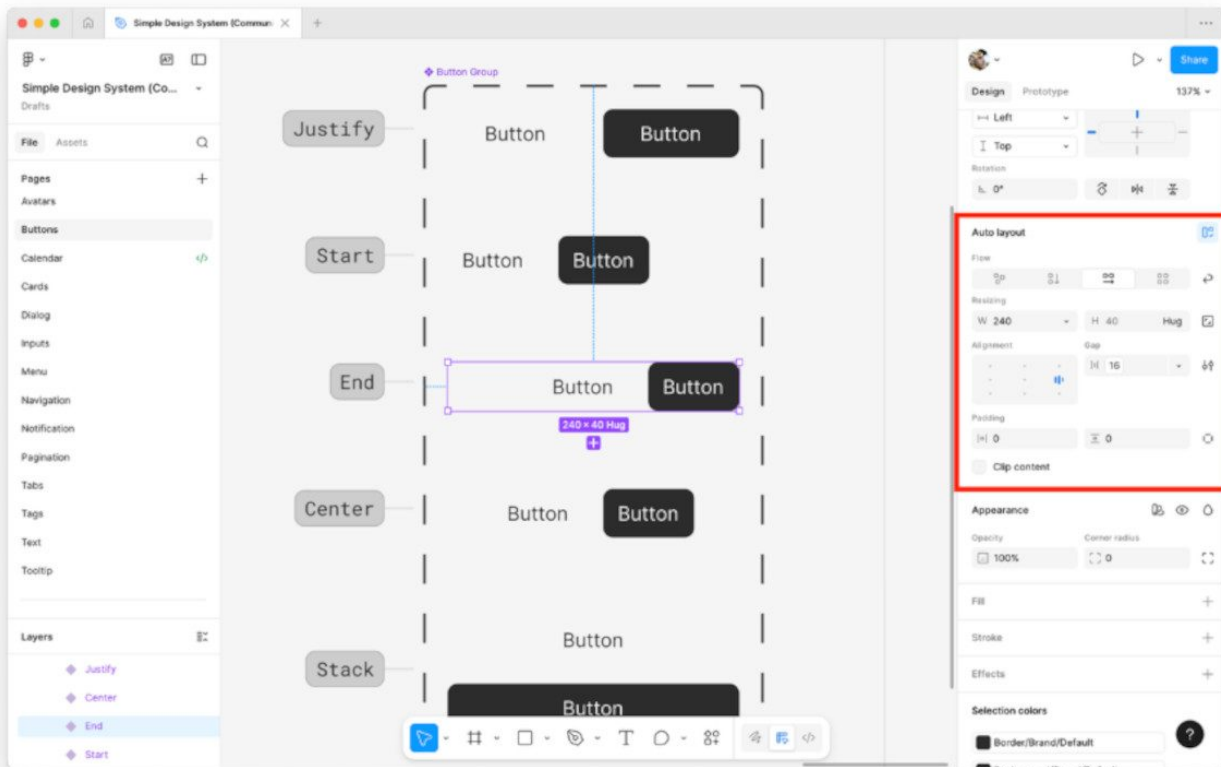
Variablen

Variablen spielen ebenfalls eine zentrale Rolle. Diese steuern einzelne Charakteristika innerhalb von Komponenten und sind auch als Design Tokens bekannt. Mit Variablen werden beispielsweise Farben, Abstände, Schriftgrößen oder andere wiederkehrende Werte zentral definiert. Wenn Sie später die Primärfarbe Ihres Projekts ändern wollen, ändern Sie nur die Variable, und alle Stellen, die diese Variable verwenden, werden automatisch aktualisiert.

Color			
Search			
Name	SDS Light	SDS Dark	
Default	Brand/800	Brand/100	
Hover	Brand/900	Brand/300	
Secondary	Brand/200	Brand/600	
Secondary Hover	Brand/300	Brand/500	
Tertiary	Brand/100	Brand/600	
Tertiary Hover	Brand/200	Brand/800	

Auto-Layout

Auto-Layout ist eine weitere wichtige Funktion, die unbedingt zum Einsatz kommen sollte. Mit Auto-Layout verhalten sich Komponenten in Figma ähnlich wie gecodete Web-Komponenten. Auto-Layout entspricht dabei Technologien wie Flexbox oder CSS Grid, allerdings in einer vereinfachten und abstrahierten Form. Wenn Auto-Layout richtig eingesetzt wird, kann Figma im Dev Mode sehr präzise CSS-Code vorschlagen, der dem tatsächlichen Verhalten im Browser nahekommmt.



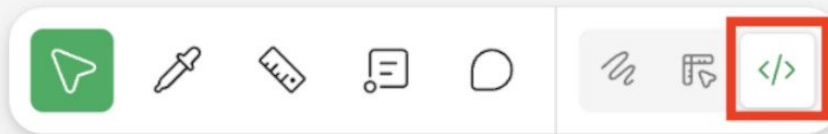
Eine Komponente mit rechtsbündiger Ausrichtung (in Flexbox: justify-content: flex-end)

Dokumentation

Schließlich ist auch die Dokumentation wichtig. Komponenten können in Figma dokumentiert werden, müssen sauber benannt sein und können mit externen Ressourcen verknüpft werden. Je besser die Dokumentation, desto leichter fällt Ihnen später die Umsetzung. Gute Benennung und klare Struktur sind nicht nur für die Übersichtlichkeit wichtig, sondern helfen auch dabei, Komponenten schnell zu finden und zu verstehen.

Designs analysieren – Dev Mode Features

In Figma wird der Dev Mode über den Switch in der unteren Werkzeugleiste aktiviert. Wenn er aktiv ist, können Sie das Dokument analysieren, aber nicht mehr verändern. Das ist wichtig, damit Sie als Entwickler das Design nicht versehentlich durcheinanderbringen. Im Dev Mode stehen Ihnen zahlreiche Funktionen zur Verfügung, die Ihnen bei der Analyse und Umsetzung helfen.

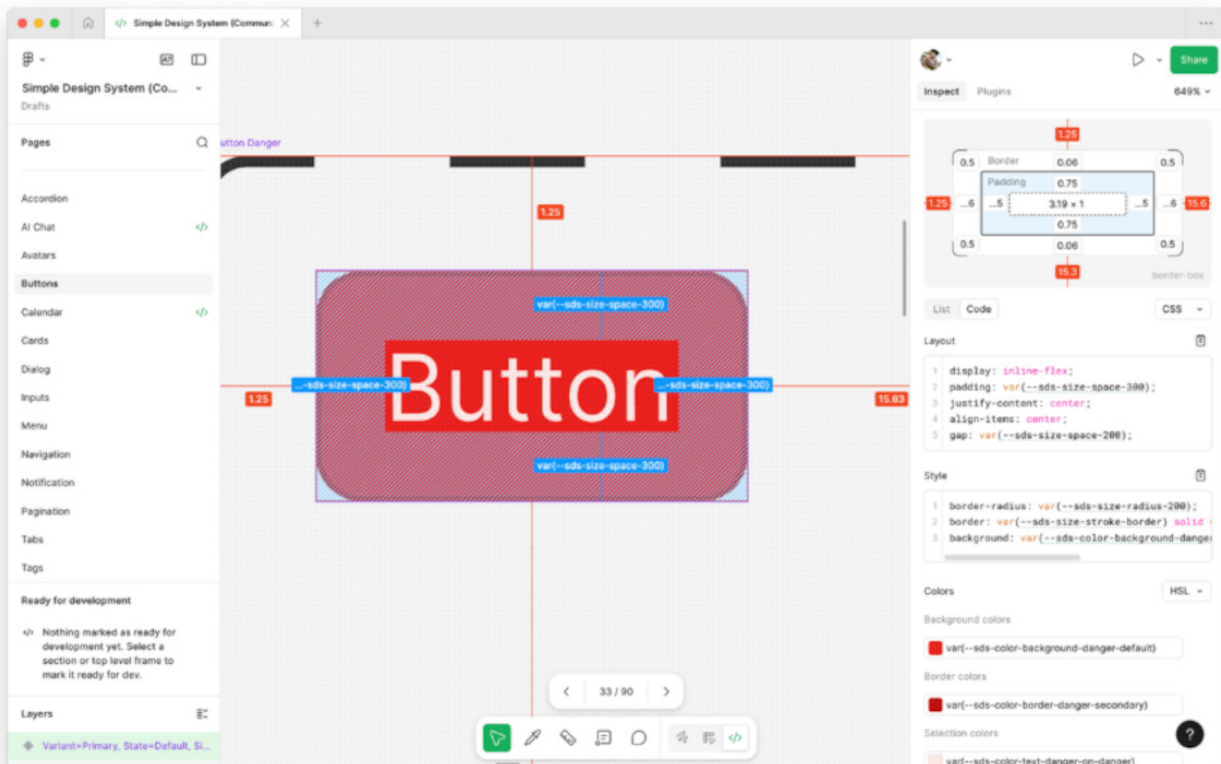


Werkzeugleiste mit aktivem Dev Mode

Analyse der Dokumentstruktur

Zunächst haben Sie Zugriff auf die komplette Ebenenstruktur. Der gesamte Aufbau des Dokuments wird sichtbar, inklusive aller Benennungen. Das ist extrem hilfreich, weil Sie so nachvollziehen können, wie die Designer das Layout strukturiert haben. Sie sehen, welche Elemente verschachtelt sind, wie die Hierarchie aufgebaut ist und wie die einzelnen Teile zusammenspielen.

Wenn Sie auf ein Element im Design klicken, zeigt Ihnen die Sidebar rechts automatisch alle relevanten Informationen an, die Figma zu diesem Element bereitstellen kann. Was genau das ist, hängt wieder von der Qualität der Design-Datei ab. Figma zeigt Ihnen dazu Code-Snippets mit den wichtigsten Informationen. Im Kontext der Webentwicklung ist das in der Regel CSS-Code. Sie können aber auch konfigurieren, dass nur die reinen Werte ohne Code-Syntax angezeigt werden, falls Ihnen das übersichtlicher erscheint. Für die Mobile-Entwicklung kann auch iOS- oder Android-Code angezeigt werden.

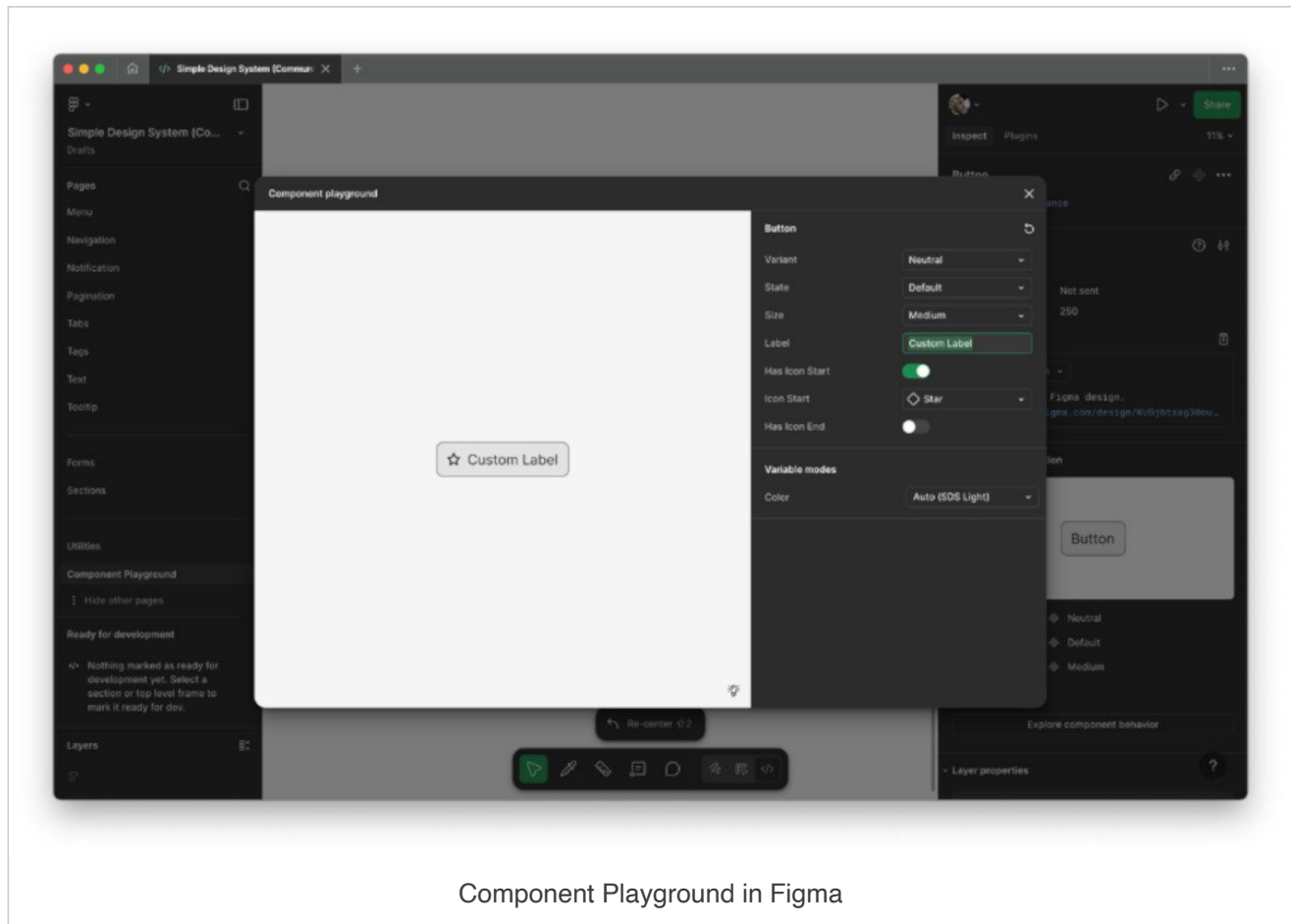


Im Dev Mode werden rechts alle vorhandenen Informationen angezeigt. Hier u.a. CSS-Variablen für Padding, Farbe, Border usw.

Infos zur Komponente

Ein besonders nützliches Feature ist die Erkennung von Komponenten und der Playground. Der Dev Mode zeigt Ihnen, wenn ein Element Teil einer wiederverwendbaren Komponente ist. Sie sehen, wie die Komponente heißt, ob es Abweichungen zum Original gibt und wo das Original liegt. Wenn die Komponente in einer externen Library organisiert ist, können Sie vom Dev Mode aus direkt auf dieses Original zugreifen und sehen, wie es definiert wurde. Das hilft Ihnen zu verstehen, welche Elemente immer gleich umgesetzt werden sollten und wo es vielleicht spezielle Varianten gibt.

Zudem bietet Figma den sog. Component Playground an, den Sie über den Button »Explore Component Behavior« erreichen. Figma öffnet die Komponente isoliert in einem Dialog. Dort können nun alle Einstellungen getestet werden, die im Design der Komponente mitgegeben wurden.



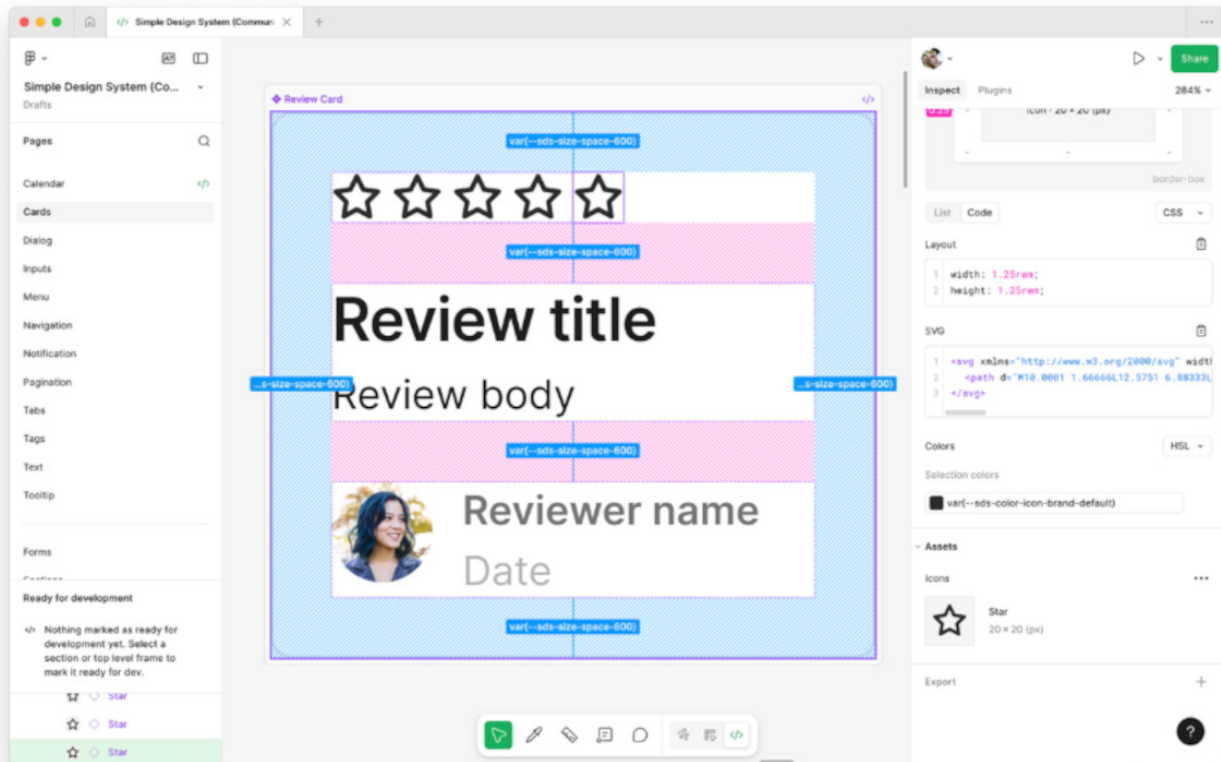
Code-Snippets für Layout & Styles

In den Abschnitten »Layout« und »Style« liefert Ihnen der Dev Mode besonders viele hilfreiche Informationen. Wenn im Design Auto-Layout zum Einsatz kam, sehen Sie Code-Vorschläge für Flexbox oder Grid. Auch Informationen zum Resizing-Verhalten, zu minimalen oder maximalen Abständen sowie zur Anordnung der Elemente werden angezeigt. Das macht es deutlich einfacher, das Layout nachzubauen. Auch Abstände zwischen Elementen, Padding-Werte, Schriftgrößen, Farben und vieles mehr werden übersichtlich aufbereitet.

Für die korrekten Informationen müssen Sie den Dev Mode unbedingt nach Ihren Bedürfnissen konfigurieren. Legen Sie beispielsweise fest, ob Abstände in Pixeln oder em angezeigt werden sollen und ob Farben als Hex, RGB oder HSL ausgegeben werden. Diese Einstellungen erreichen Sie, indem Sie *kein* Element auswählen. Dann stehen in der Sidebar globale Einstellungen zur Verfügung.

Grafik-Export

Wenn Sie Grafiken im Design anklicken, können Sie diese rechts unten in der Sidebar als Datei exportieren. Das funktioniert mit verschiedenen Formaten und in unterschiedlichen Auflösungen. Bei SVG-Grafiken wird Ihnen der komplette Code angezeigt, den Sie direkt kopieren und in Ihren Code einfügen können. Das spart Zeit und vermeidet unnötige Dateien, wenn Sie das SVG lieber inline verwenden möchten.

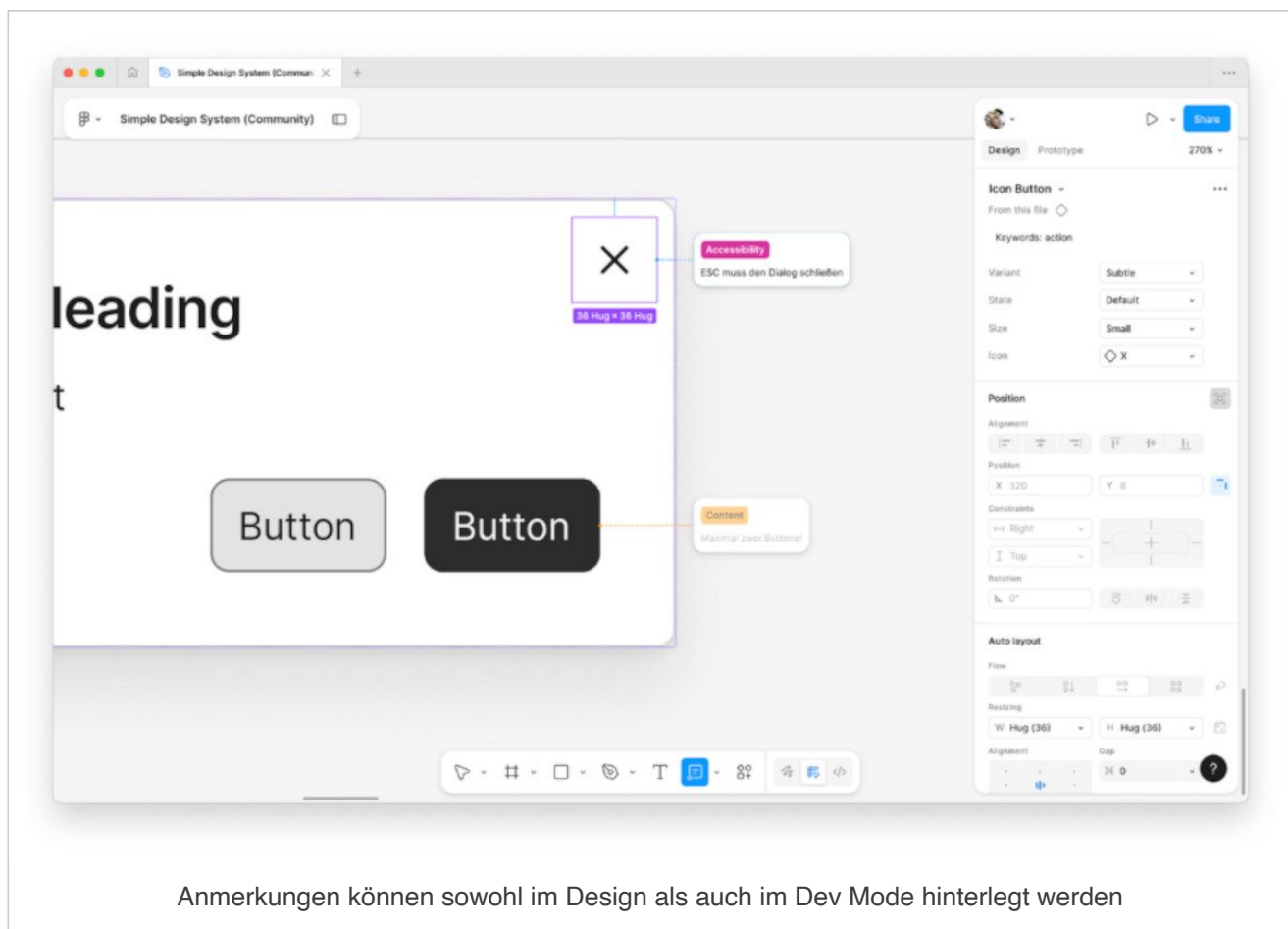


Informationen bei grafischen Assets wie SVG-Icons

Kommentare und Anmerkungen

Für die Kommunikation gibt es Kommentare und Anmerkungen. Mit Kommentaren können Sie direkt bestimmte Bereiche im Dokument diskutieren. Sie können Fragen stellen, Feedback geben oder auf Probleme hinweisen. Die Kommentare sind eher für kurzfristige Kommunikation gedacht.

Anmerkungen hingegen dienen dazu, langfristig relevante Informationen zu hinterlegen. Diese Anmerkungen können kategorisiert und üblicherweise für Informationen zu Content, Interaktivität oder Barrierefreiheit verwendet werden. Wenn beispielsweise ein Element eine bestimmte Animation haben soll oder besondere Accessibility-Anforderungen erfüllen muss, wird das in einer Anmerkung festgehalten.



Workflow

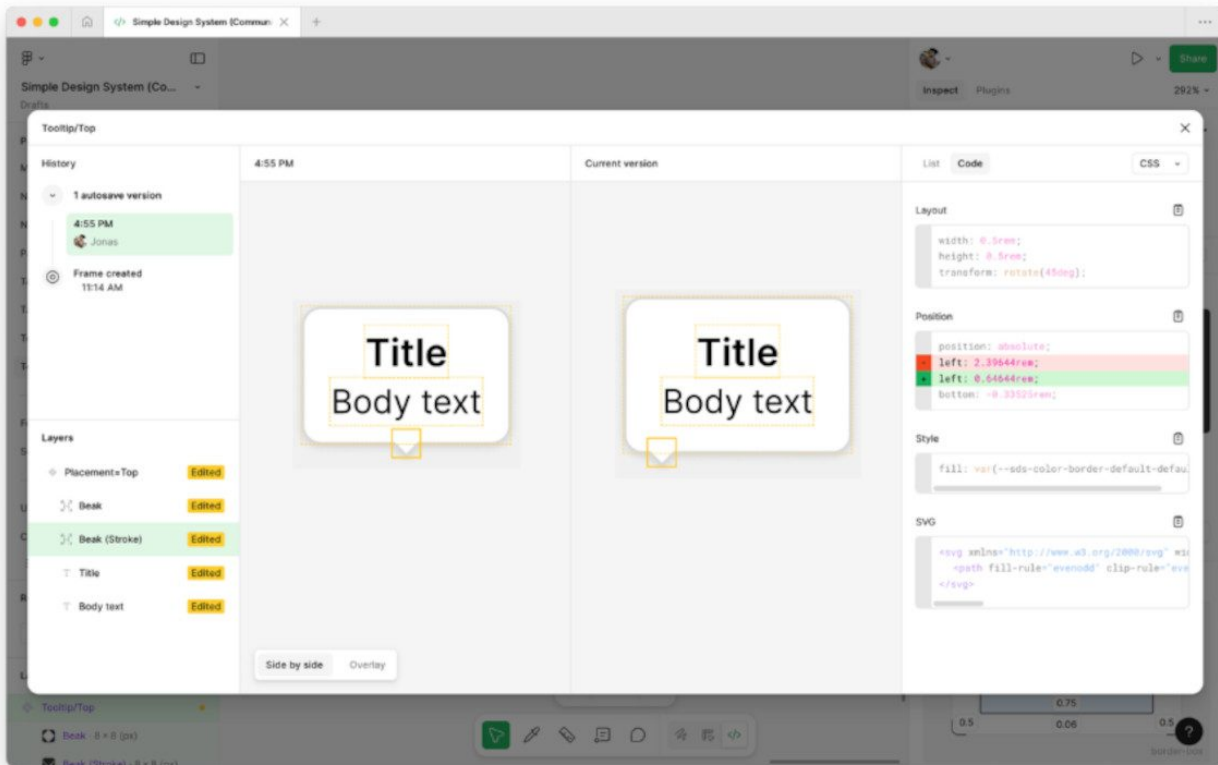
Die Figma-Datei lebt in der Cloud und erzeugt automatisch Versionen, sobald Figma relevante Änderungen im Design erkennt. Diese Autosave-Versionen helfen dabei, den Verlauf nachzuvollziehen. Zusätzlich können manuell Versionen mit aussagekräftigen Namen erzeugt werden.

Es ist wichtig, dass alle Beteiligten auf diesem Zeitstrahl arbeiten und nicht mit Kopien der Datei hantieren. Nur so bleiben alle synchron, und es entstehen keine parallelen Versionen, die später mühsam abgeglichen werden müssen.

Versionierung und »Changes«

Die Versionierung und die Änderungshistorie werden genutzt, um zu älteren Zuständen zurückzukehren oder Änderungen zu vergleichen. Sie können zwei verschiedene Versionen eines Frames sowohl visuell als auch im Code-Snippets-Bereich miteinander vergleichen. So sehen Sie genau, was sich geändert hat.

Auch bei Komponenten lässt sich sehen, ob diese eins zu eins aus der Bibliothek verwendet wurden oder ob jemand Anpassungen vorgenommen hat. Im schlimmsten Fall wurde eine Komponente sogar »detached«, also von der Bibliothek gelöst. Das sollte eigentlich vermieden werden, kommt in der Praxis allerdings dennoch vor und ist in den Informationen zur Komponente sichtbar. Figma zeigt detailliert, warum sich eine Komponente anders verhält als möglicherweise erwartet.



Diff einer angepassten Komponenten im Vergleich zum Original

»Ready for Dev« und »Focus View«

Mit dem Feature »Ready for Dev« hat Figma einen praktischen Mini-Workflow integriert. Das Vorgehen funktioniert so: Wenn eine Komponente oder ein bestimmter Bereich des Designs fertig ist, kann dieser auf »Ready for Dev« gesetzt werden. Im Dev Mode sehen Sie dann auf einen Blick, welche Bereiche fertig sind und umgesetzt werden können. Sie können diese systematisch abarbeiten und müssen nicht raten, welche Teile schon final sind.

In höheren Lizenz-Stufen steht sogar ein »Focus Mode« zur Verfügung. Dieser ermöglicht es, eine Komponente komplett isoliert zu betrachten, ohne Ablenkung durch den Rest des Designs. Nach der erfolgreichen technischen Umsetzung können Sie die Komponente dann als »Completed« markieren, was den Status für alle sichtbar macht.

Branches

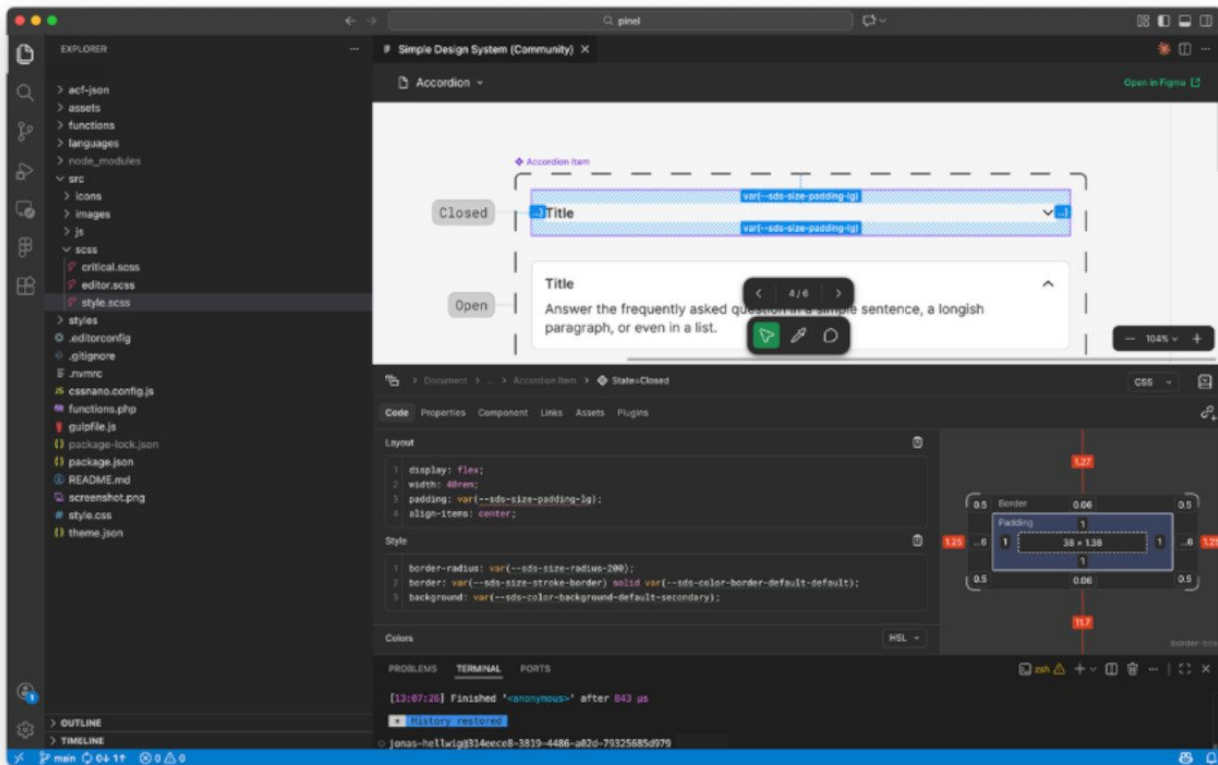
Auch Branches, also abgezwigte Kopien von Design-Dateien, sind in höheren Lizenzen verfügbar. Diese werden verwendet, um Ideen sicher auszuprobieren, ohne das Hauptdokument zu beeinflussen. Branches sind auch nützlich, wenn nicht allen Designern direkter Zugriff auf die zentrale Library gegeben werden soll. Änderungen können in einem Branch entwickelt und getestet werden, bevor sie in die Hauptdatei übernommen werden.

Plugins und Drittanbieter-Integrationen

Es gibt zahlreiche Plugins für Figma, und viele davon sind speziell für den Dev Mode gedacht. Diese helfen bei Aufgaben wie der Verbesserung der Barrierefreiheit, dem besseren Export von SVG-Dateien oder der Generierung von CSS-Variablen. Andere Plugins integrieren externe Tools direkt in Figma, was den Workflow erheblich vereinfachen kann.

Projektmanagement-Tools wie Jira oder Asana können mit Figma verbunden werden. Das erlaubt es, Frames oder Komponenten direkt mit Tickets zu verknüpfen. In Figma sehen Sie dann, welche Tickets in welchem Status zu einer bestimmten Datei oder einem bestimmten Bereich existieren. Im Projektmanagement-Tool sehen Sie wiederum eine Live-Ansicht der relevanten Designs oder Bereiche. Sie müssen nicht mehr zwischen verschiedenen Tools hin- und herspringen, um den Zusammenhang herzustellen.

Code-Repositories wie GitHub oder Tools wie Storybook können ebenfalls integriert werden. Das ermöglicht es, Komponenten in Figma direkt mit Dateien in Ihren Repositories zu verknüpfen. Sie sehen dann auf einen Blick, ob es zu einer Design-Komponente bereits Code gibt und wo dieser liegt. Das reduziert doppelte Arbeit und hilft dabei, Design und Code synchron zu halten.



Integration des Dev Modes in VS Code

Figma in VS Code

Besonders interessant ist die direkte Integration von Figma in den Code-Editor Visual Studio Code. Damit haben Entwicklerinnen und Entwickler keinen Kontext-Switch mehr zwischen Tools. Sie installieren die Figma-Erweiterung in VS Code und haben anschließend alles aus Figma direkt im Editor zur Verfügung.

Das umfasst sogar Auto-Vervollständigung beim Programmieren, basierend auf den Komponenten und Werten aus dem Design. Sie müssen nicht mehr zwischen Browser und Editor wechseln, sondern haben alle Informationen direkt da, wo Sie sie brauchen.

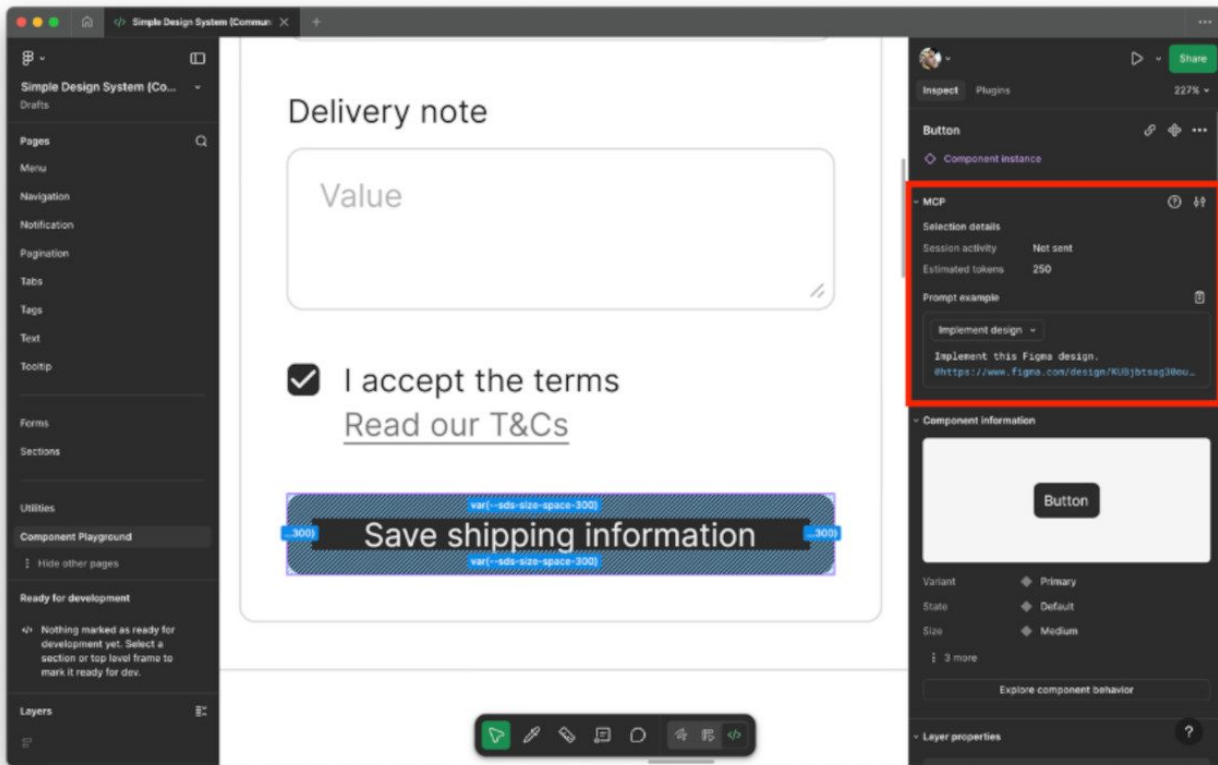
Figma und AI

Im Kontext der künstlichen Intelligenz gibt es einige sehr relevante Features im Dev Mode, die die Arbeit revolutionieren können. Der Einsatz von AI-Tools kann den Entwicklungsprozess erheblich beschleunigen, birgt jedoch auch einige Besonderheiten, die Sie kennen sollten.

MCP

Ein wichtiges Konzept ist das MCP-Protokoll in Figma, das für »Model Context Protocol« steht. Dieses Protokoll standardisiert, wie KI-Tools auf verschiedene Anwendungen zugreifen und Daten austauschen können. Vereinfacht gesagt ermöglicht es MCP, über verschiedene KI-Agenten mit Figma zu kommunizieren und direkt aus den Designs Code zu generieren.

Zu den wichtigsten AI-Tools, die mit Figma arbeiten können, gehören Claude oder Copilot, sowie spezialisierte Entwickler-Tools wie Cursor. Diese Tools können einen Link zu einem Figma-Dokument oder zu einem bestimmten Bereich im Dokument verarbeiten und daraus automatisch Code erzeugen.



MCP-Einstellungen im Dev Mode von Figma

Wie gut das Ergebnis ist, hängt maßgeblich vom verfügbaren Kontext ab. Damit ist die Qualität der Figma-Datei gemeint, die dort bereitgestellten Informationen wie Anmerkungen und Dokumentation und gegebenenfalls weitere Informationen aus anderen Anwendungen. Deshalb macht es Sinn, verschiedene MCPs zu integrieren.

Ein Beispiel wäre die Kombination von Figma und Jira. Wenn der KI-Agent Zugriff auf beide Systeme hat, kann er nicht nur das Design sehen, sondern auch verstehen, welche Anforderungen in den Tickets beschrieben sind.

Zusätzlich ist die Qualität des Prompts, also Ihrer Anweisung an die KI, sehr entscheidend. Je präziser Sie beschreiben, was Sie erreichen wollen und welche Technologien oder Frameworks verwendet werden sollen, desto besser wird das Ergebnis.

Wenn ausreichend Kontext vorhanden ist und der Prompt gut formuliert wurde, können AI-Agenten erstaunliche Ergebnisse erzielen. Komplette Komponenten können innerhalb von Sekunden generiert werden, inklusive HTML-Struktur, CSS-Styling und sogar JavaScript für Interaktionen. Das bedeutet nicht, dass der generierte Code immer perfekt ist, aber er bietet einen sehr guten Ausgangspunkt und spart enorm viel Zeit.

Code Connect und »production-ready« Code

Ein Nachteil ist allerdings, dass sowohl die Code-Snippets im Dev Mode als auch die AI-Agenten den Code jedes Mal neu erfinden.

Wenn Sie beispielsweise einen Button haben, der bereits als finale Code-Komponente in Ihrem Projekt existiert, möchten Sie eigentlich nicht, dass Figma oder ein AI-Agent diesen Button immer wieder von Grund auf neu erstellt. Stattdessen wäre es sinnvoller, wenn der existierende, getestete und produktionsreife Code wiederverwendet würde. Genau hier setzt das Feature »Code Connect« an.

Mit Code Connect können UI-Komponenten in Figma mit Code-Komponenten in einem Repository verknüpft werden. Das erzeugt zusätzlichen Kontext und führt in der Theorie bei der Umsetzung durch AI-Agenten zu »production-ready« Code, der Ihre bestehenden Komponenten nutzt, statt alles neu zu erfinden. In der Praxis ist diese Technologie noch relativ neu, zeigt aber bereits vielversprechende Ergebnisse.

Fazit: Design und Dev wachsen zusammen

Der Dev Mode macht die Zusammenarbeit zwischen Design und Development deutlich effizienter – vorausgesetzt, beide Seiten arbeiten auch wirklich zusammen und missverstehen den Dev Mode nicht als Lösung für fehlende Team-Kommunikation. Und beide Parteien müssen über den Tellerrand schauen: Designer gestalten technischer, Devs beschäftigen sich mit dem Design-Tool.

Besonders spannend wird es zukünftig bei der AI-Integration. Code Connect und MCP-basierte Workflows stecken zwar noch in den Kinderschuhen, zeigen aber schon jetzt, wohin die Reise geht. Designs werden nicht mehr mühsam abgetippt, sondern direkt in produktionsreifen Code überführt.

Je besser das Design System, desto besser werden solche Workflows funktionieren. Digital-Teams sollten daher spätestens jetzt in solide Systeme investieren und vor allem die Dokumentation nicht vernachlässigen. Denn Doku und Kontext werden immer wichtiger, damit Mensch und Maschine verstehen, was zu tun ist.