

User Research

Identify Users: Designers, **Developers** - Front end, Product managers, Quality Assurance testers, Stakeholders

Needs:

- Leverage efficient productivity workflow across seat types - Viewer, Collaborator, Content, Developer, and Full, plan tiers and new products (Fig Jam, Dev Mode)

Quotes

“Two kinds of files in Figma 1. design files 2. design system”~*Designer Interview*

“Create a prototype for whole workflow ..a lot of iterations used to happen...the designer itself missed pointers....”~*Developer Interview*

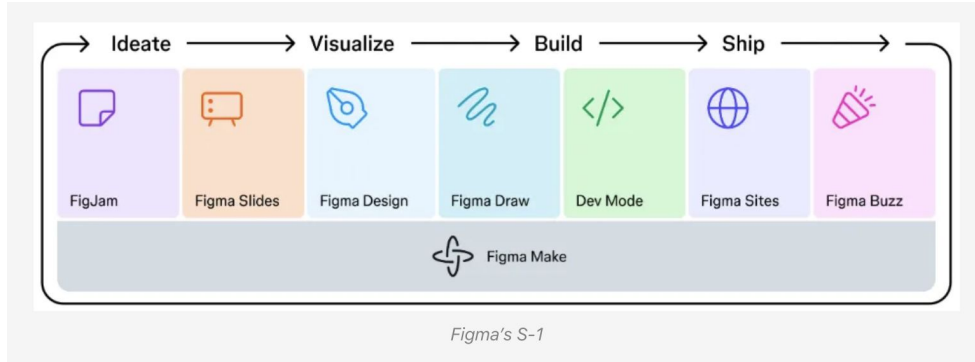
“It would have been helpful if there were notes on what color contrast they used while designing, which will help devs to use right colors than back an forth.”~*Developer interview*

User Research

Identify Users: Designers, **Developers** - Front end, Product managers, Quality Assurance testers, Stakeholders

Needs:

- Leverage efficient productivity workflow across seat types - Viewer, Collaborator, Content, Developer, and Full, plan tiers and new products (Fig Jam, Dev Mode)



Productivity Workflow

- **Ideate:** *FigJam* (for whiteboarding and brainstorming), and *Figma Slides* (for real-time presentations with embedded design elements).
- **Design:** *Figma Design* (the flagship UI tool), and *Figma Draw* (a dedicated vector editing space for detailed illustrations).
- **Build:** *Dev Mode*, a developer-focused workspace to inspect designs, view documentation, and streamline handoff.
- **Ship:** *Figma Sites*, which lets users turn designs into working websites with responsive layouts, motion effects, and code layers.
- **Promote:** *Figma Buzz*, a tool to create on-brand marketing assets like social media graphics and digital ads.
- **Prototype with AI:** *Figma Make*, an early AI-driven tool that turns prompts into working prototypes—potentially skipping traditional wireframing.

User Research

Pain Points:

- Design handoff involves multiple design files, one for design and one for dev handoff
- Documenting changes using Comments is not enough to handle version control issues
- Without commenting ability (limited in viewer mode), you can't ask questions directly on the design, leading to extra mentors to communicate
- Figma files change, but not notified. Hard to track what changed between your implementation and the latest design.
- You have to manually measure everything, clicking through each element to understand spacing and sizes.
- Different states of components might be scattered across frames rather than clearly documented as variants.
- Make assumptions or extra communication steps needed for hover effects, focus states, active states, disabled states, loading states, and error states are frequently not designed.
- Auto-layout doesn't perfectly map to CSS flexbox or grid.
- Interpreting absolute positioning by designers
- Dev mode requires seat growth and
- Limited inspection capabilities
 - Need to integrate design system, when exists, in final design easier
 - Need to add net new content in design and be able to identify easier and be able to adapt to design system when handing off final design.

User Research

Apprehension:

- Design handoff requires communication overhead
- Productivity workflow version control issues
- Access & Handoff Issues with Limited inspection capabilities, asset export restrictions
- Specification gaps on typography inconsistencies, component variants
- Design system and component management
- Adoption: Lack of best practices within the productivity workflow roles

SWOT Analysis - Strengths

Strengths:

- Collaborative workflow. Figma was first to embrace browser-native, collaborative user workflows charging per “editor” seat; only those actively creating or modifying files need a paid license. Using FigJam and Dev Mode. Average revenue per paying customer (1,000+ customers spending over \$100,000 ARR Q1 2025) has climbed steadily with seat growth, cross-team expansion,
- Community driven from free tier expands organically to enterprise tier leading to 11,000+ customers spending over \$10,000 ARR (annual recurring revenue) Q1 2025
- Multiplayer by default: multiple users can design, comment, and edit simultaneously.
- Web-first: No install required. Works across Windows, Mac, and Linux.

SWOT Analysis - Weaknesses

Weaknesses:

- Design-Development Misalignment: All designers interviewed had to create multiple figma files to accommodate developer needs.
- Inconsistent Handoffs: All developers had extra communication for design to code translation with or without a design system within figma files
- Viewer-only roles are free, which helps drive viral adoption, lacks ability to bridge design and development.
- Figma also offers a range of seat types—Viewer, Collaborator, Content, Developer, and Full—tailored to specific workflows but on Enterprise plans its difficult to get access for non-viewer across organizations.
- Adaptability with current access rights: Based on figma access have limited capabilities and lack an automatic way to extract design tokens (colors, typography, spacing) into CSS variables or other formats as identified from designers and developers interviewed and consistently described from interviews.

SWOT Analysis - Opportunities

Opportunities:

- Enhance developer specific needs at the designer-to-developer handoff such as inspection details and different states of components within an enterprise level.
- Combine prototypes with final design with production-ready components without coding
- Making Dev mode features available for Viewers in Design stage

SWOT Analysis - Threats

Threats:

- Design-Development Misalignment: All designers interviewed had to create multiple figma files to accommodate developer needs.
- Tools that are explicitly designed to overcome the core weaknesses of traditional Figma-to-code converters (like Builder.io, Framer, and Anima)
- Inconsistent Handoffs: All developers had extra communication for design to code translation with or without a design system within figma files.
- Licensing constraints for its specialized developer features (Dev Mode).Adaptability with current access rights: Based on figma access have limited capabilities and lack an automatic way to extract design tokens (colors, typography, spacing) into CSS variables or other formats as identified from designers and developers interviewed and consistently described from interviews.
- Organizations pay for unnecessary editing functionality and introduces the risk that developers could accidentally delete or change design files

Goal Setting

- Goal: Increase cross-functional engagement primarily by increasing usage based tools for viewers for designer-to-developer handoffs on enterprise-grade level collaboration for large organizations and secondarily opportunities for revenue generation. (revenue, engagement)
- Improve cross-functional engagement by 20% to 30% over three years for enterprise design-to-developer teams by increasing adoption of viewer-focused, usage-based collaboration tools.
- Users: **Developers (primary)**, Designers, QA
- Prioritized Problems
 - Developers spend significant time on repetitive manual work, such as extracting assets, copying styles, converting layouts, and aligning typography,
 - Designs often lack complete specifications for crucial development details such as spacing, responsiveness rules, interaction states, and breakpoint.
 - Manually Converting Static Visual Design into Responsive Frontend Code
 -

User Personas

Front End Developer

Developer Dave

Front End Developer

Dave is a developer that frequently works with designers, product managers, QA and backend developers. Dave has received many invites from designers with figma files from same organization of variable size.

“ It would have been helpful if there were notes on what color contrast they used while designing, which will help devs to use right colors than back and forth.”

Jobs to be done: When I receive a final design I want to be able to identify components and translate them to code with just the final design file only so that when I don't have access to Dev Mode I can reduce my time interpreting and meeting designers reducing frustration.

Core Needs:

- Developers need to convert the visual design assets and styling into maintainable, working frontend code
 - Need spacing, responsiveness rules, interaction states, and breakpoints
- Manually Converting Static Visual Design into Responsive Frontend Code
- Need to extract assets, copy styles, convert layouts and aligning typography

Frustrations:

- Inconsistent Component Implementation with inaccurate code translation, missing information,
- Manual and Repetitive Tasks when extracting assets, etc
- Incomplete specifications for variants, responsive guidelines forcing to context switch across design systems, PRDs, IDEs.

User Personas

“ I just check what developers build based on figma. ”

Product Manager

Product John

Product Manager

John is a product manager that interacts with designers trying to ground the design to a realistic applicable level and is always required to create PRD from any final design and use that PRD to help front end developers estimate level of development work to implement final figma design.signers

Jobs to be done: When I receive a final design I want to be able to identify components and translate them to component level and understand the user experience along with any possible integrations required So that I can create user stories with acceptance criteria for all possible component variants and integration with as little interpretation needed.

Core Needs:

- Efficient estimation of development effort and scope identification based on design and all related interaction and integrations
- Maintain a streamlined development workflow based on clearly defined features
- Identify and maintain components across ecosystem with efficiency and high resusability

Frustrations:

- Final designs not enough for PRD specifications.
- Too many files in figma for final design. Handoff contains design system, prototype, static UI.

User Journey Map

Developer Dave

Developer

1. Handoff & Access

2. Interpretation & Specification Extraction

3. Component Development & Code Construction

4. Validation, Quality Assurance (QA), and Rework

5. Integration and Deployment Preparation

Phases

Receive access to the Figma file (often with "can view" permission) and attend handoff meeting

Open the Figma file to inspect design elements, colors, text styles, and measurements

Manually write HTML/CSS/JS components, exporting necessary assets (like SVG icons)

Test the component implementation across devices and check for design fidelity.

Integrate the component into the larger application architecture and connect UI elements to data sources.

Activity

Understand the **full scope** of the feature, including requirements for mobile, tablet, and desktop viewports

Understand the **full scope** of the feature, including requirements for mobile, tablet, and desktop viewports

Implement the design using web standards. Ensure the component adheres to design system and is reusable

Minimize delays caused by errors or rework

Code aligns with backend logic and is ready for the Software Development Life Cycle (SDLC),

Pain Points

Developers must rely on **hour-long meetings** for clarification or manually annotated redlining

Designs often **lack complete specifications** for responsiveness, spacing rules, or interaction states, forcing developers to **guess** the design intent

This stage involves **repetitive manual work**, such as copying individual styles, converting layouts, and aligning typography

Components are manually detached, key accessibility attributes (like alt text or color contrast compliance) may be lost

Developers or backend teams must reverse engineer intent to link the UI to backend services

Opportunity

Automate and standardize handoff processes

Automate Specification & Eliminate Guesswork

Generate Production-Ready Code & Enforce Consistency

Component testing framework

Build internal tooling that generates API contracts from Figma components

Thinking



Problem Prioritization - Choosing Your P0

"Developers are struggling with manually repeating tasks such as extracting assets, copying styles, converting layouts, and aligning typography because of inconsistent component implementation or lack of efficient extraction.

This is leading to less engagement in design phase and creating gaps in the productivity workflow for Figma with eventual revenue generation occurring in the wrong productivity stages and undermining the principal of product led growth by non-designers "