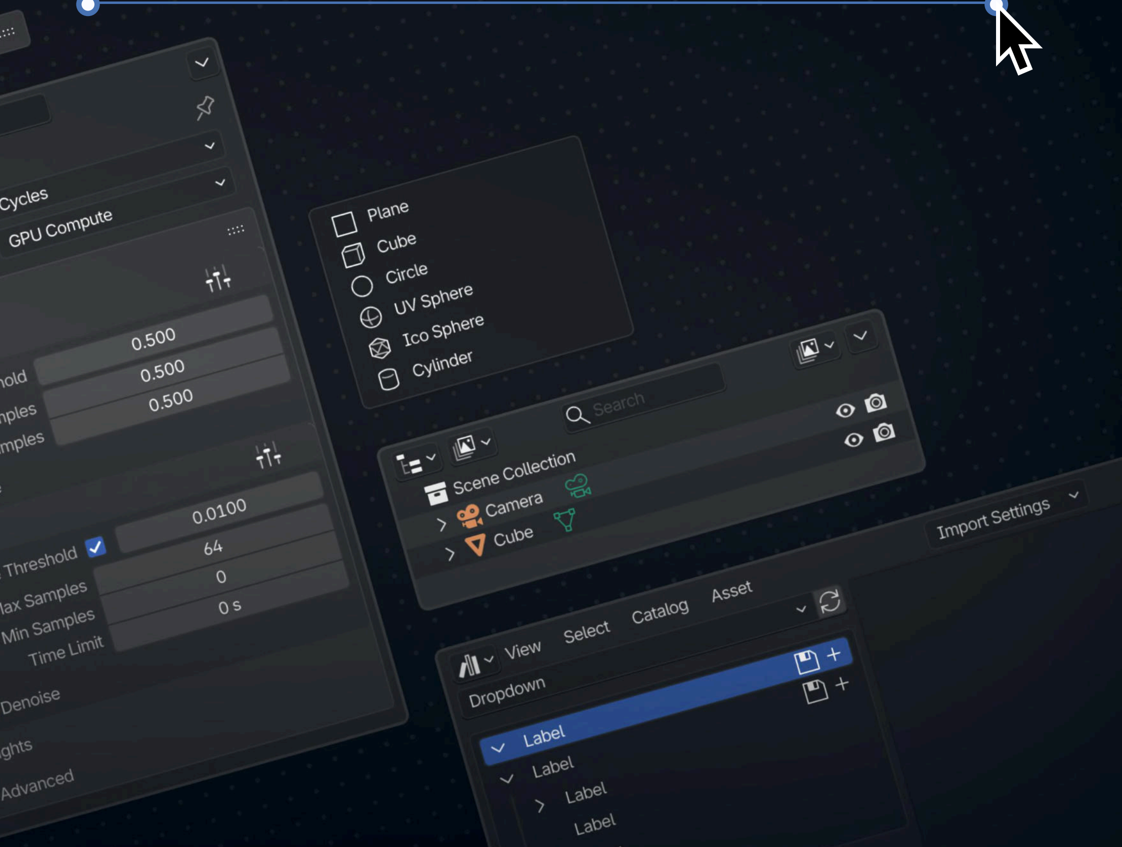
Leveraging Design Expertise in Open-Source 3D Software Organizations

Building Design Capacity through a Workflow-Integrated Design System - A Case Study of Blender



Sem Tip

Master Thesis, February 2026
TU Delft, MSc. Integrated Product Design

TU Delft blender

Master Thesis

February 17th, 2026
Delft, The Netherlands

Written By

Sem Tip
Student MSc. Integrated Product Design
Faculty of Industrial Design Engineering
Delft University of Technology

The Supervisory Team

PhD Jan Willem Hoftijzer
Coordinating teacher and expert in Design and Design visualization education
Human Information Communication Design (HICD), Section of Human-Centered Design (HCD)
Faculty of Industrial Design Engineering
Delft University of Technology

Marina Bos-De Vos
Associate Professor Design for Joint Value Creation in Societal Transitions
Designing Value in Ecosystems (DIVE), Section of Department of Design, Organisation and Strategy (DOS)
Faculty of Industrial Design Engineering
Delft University of Technology

In Collaboration With

Blender Institute, the working company of the Blender Foundation, Amsterdam
Represented by **Dalai Felinto**, Head of Product

Acknowledgements

Everyone

Even though this thesis is an individual project, it would not have been possible without the great help and contributions of many.

First, I would like to thank my supervisory team. My chair, Jan Willem, for his very insightful, critical feedback on my work and for pushing me further in research. My mentor, Marina, for her great support throughout the project and her clear suggestions during a project that sometimes felt very vague.

A big thank you to everyone at Blender. In particular Dalai, for allowing me to experience design at Blender in such a participatory way. I really enjoyed working together and being part of the design work! Pablo, for teaching me the design principles that make Blender. Francesco, for sharing the vision and strategy of the Blender mission. Ton, for being open, and positive in response to my initial request to work together. Lastly, all the wonderful members of the Blender team that were welcoming from day one and open to my (many!) questions.

Thank you Sarika and my family for your support and celebrating the big and little moments with me!

Furthermore, it is important to acknowledge that the Blender team was, and is actively working on improving the design standards, workflow and the challenges described in this thesis. They were a great support in realizing the developed concept, the Design System. Among user-testing, by taking care of technical tasks such as deploying and updating it on design.blender.org, and by building a dedicated mockup theme in the Blender software to support the design system development. The final result represents a team effort.

I'm very grateful to all. It was amazing to do this together :)

Open, Fundamentally

This report is created and published using the open-source publisher Scribus using the open font Inter, the same font used for the Blender software.

Other Tools Used

This report was constructed using input from AI. In particular Gemini 3. In sparring conversations it provided support in streamlining the story, argument chain, and rough text outlines. The sketches of the challenges in Section 3.4 were also created using AI. All other 2D and 3D visuals were made by hand, all 3D visuals with Blender (of course).

Executive Summary

This thesis investigates the challenge of leveraging design expertise in the development processes of a large-scale open-source 3D software organization, using Blender as a case study. While open-source projects like Blender thrive on community-driven and very technical contributions, they often lack approaches to integrate or leverage design expertise, leading to a "chronic scarcity of design expertise" at Blender. This research identifies the root of this challenge in a historically technology-first culture, which creates friction in designer-developer collaboration, results in inaccessible design knowledge, and hinders the leveraging of external design contributions.

To provide a more fitting analytical lens for this unique open-source environment, the thesis introduces the concept of Design Capacity: a measure of an ecosystem's systemic infrastructure and ability to effectively integrate and leverage design potential. Through stakeholder interviews, literature reviews, and participation, the thesis identifies a deficit in Blender's organizational design capacity, creating challenges. The central research question explores how practical interventions can improve the design workflow of the core Blender team to increase the ability to leverage their design expertise.

The findings suggest implementing a design system. This system is presented as a tactical intervention designed to bridge the gap between designers and developers. By creating a shared language and a single source of truth for design, the proposed design system aims to resolve the identified challenges, improve collaboration, and ultimately build Blender's long-term Design Capacity to create more value for its entire ecosystem.

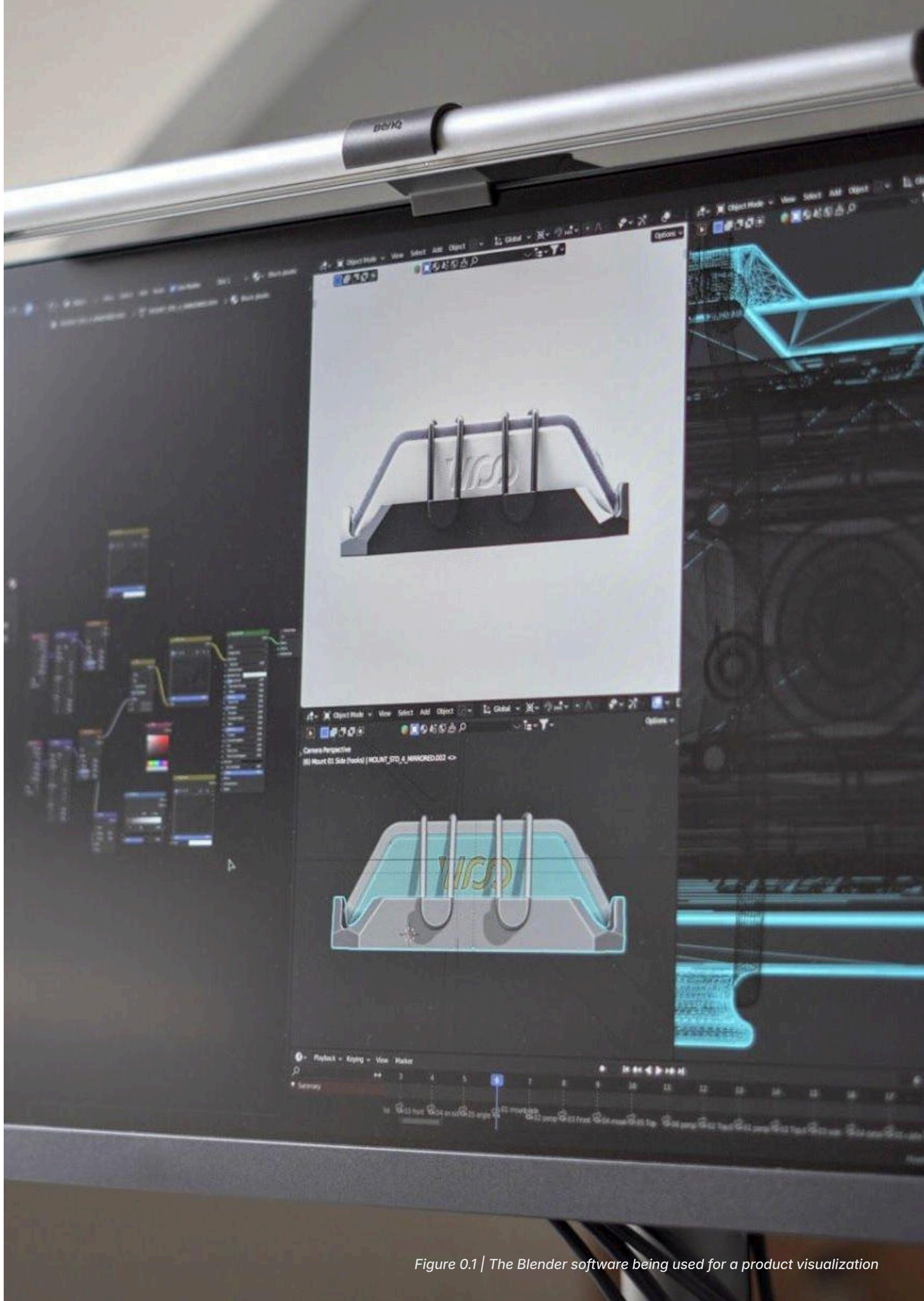


Figure 0.1 | The Blender software being used for a product visualization

Figure 0.2 | Visualisations created by me using Blender

Preface

If I could go back and speak to my younger self, the one just discovering the magic of 3D, I would tell him that the feeling he's chasing is real. It's the feeling of empowerment, the realization that you can create anything you can imagine with your own two hands. In a world full of constraints, here was something that promised limitless creation. That something for me was Blender.

For me, Blender quickly became more than just a tool, it was the path to a super exciting future I hadn't even considered. It was a way to realize my imagination, a common passion that connected me with friends, the foundation for my first company, and a skill I felt privileged (and very excited!) to teach others.

I vividly remember the first moment I saw something I created myself appear on the big screen or my work being part of big advertisements and getting seen by a lot of people. Seeing people so impressed by something I made was new for me. It illustrates Blender's core promise: that with accessible and powerful tools, anyone is free to create anything, for any purpose. The freedom to create as they call it.

Over time, my curiosity didn't just stop at the creation itself. As I became familiar with the tool, optimizing it for every client project, I found myself looking beyond creating, wondering about the architects behind it. Who were the people who gave this free gift to the world? I wanted to understand it, contribute to it, and to help make it even better.

My younger self would be super excited to see where this path would lead. The idea that I would one day collaborate with the very people who built the tool I admired for so long felt like a dream. This thesis is the culmination of that dream. This is my opportunity to give back to the organization and the community that have shaped my career and my creative identity.

This work is personal. It is a deep dive into the inner workings of a globally recognized open-source organization. This thesis is my contribution, offered in the hope that it will provide meaningful value to an organization I will always admire.

- Sem

Sem Tip

Involved Stakeholders and Expectations

Involved Parties: TU Delft & Blender

This graduation project was conducted in collaboration with the Blender Institute in Amsterdam and carried out for my Msc. Integrated Product Design at the Faculty of Industrial Design Engineering at Delft University of Technology. TU Delft acted as the academic stakeholder, ensuring the project is based on research to provide insights into open-source software development.

The thesis is also in collaboration with the Blender Institute, the operational company of the Blender Foundation, oversees the development of Blender. I was fortunate enough to be part of the development environment at the Blender office for 3 days per week.

Expectations

Before and during the project, expectations were balanced from three perspectives: university, client, and my personal development.

Academic Expectations	Client Expectations	Personal Goals
Systemic Analysis: Move beyond surface-level observations to diagnose the root causes of the challenges using valid methods.	**Actionable Solutions:** Go beyond theoretical advice to deliver practical, usable tools that directly benefit the Blender team.	**Give Back:** Leverage years of experience as a Blender user to make a meaningful contribution to the software and community that have given me a lot.
Validation: Ensure findings are supported by data triangulation, combining literature, interviews, and participatory observation.	**Cultural Fit:** Develop interventions that respect and enhance Blender's unique culture of autonomy and open contribution, rather than imposing constraints.	**Bridge Strategy & Practice:** Demonstrate the ability to operate on both a strategic level (defining framework) and an operational level (building practical concept).
Academic Independence: Take an independent perspective from the situation and objectively collect data and analyze it to get to insights.	**Efficiency:** Reduce the friction and inefficiency in the feature development process	**Deep Immersion:** Fully integrate into the development team to understand the nuances of open-source collaboration and experience the role of the designer in this context.

Table 0.1 | Overview of expectations of all stakeholders involved

Figure 0.3 | TU Delft Industrial Design Engineering Faculty in Delft

Figure 0.4 | Blender Foundation and Blender Institute office, based in Amsterdam

Table of Contents

Glossary of Terms

Atomic Components
The smallest, indivisible building blocks of a user interface, such as buttons, icons, or text labels. In the proposed Design System, these are the fundamental elements used to build larger templates.

Boundary Object
An artifact (such as a mockup, prototype, or document) that serves as a communication interface between different communities of practice to build mutual understanding, in this case, designers and developers. It allows valid collaboration even when the groups have different knowledge, perspectives and backgrounds.

Contributor Levels (C1–C4)
A framework defined in this thesis to classify stakeholders based on their proximity to the core organization. Reference Section 3.2 for more details.

Design Capacity
A new framework introduced in this thesis (Section 3.7). It is the measure of an ecosystem's systemic infrastructure effectiveness to integrate (capture) and leverage (accelerate) design capability potential into consistent organizational output. It consists of two levers: Workflow Leverage and Expertise Quantity.

Design System
A centralized library of reusable components, guidelines, and assets for UI/UX and other design tasks that serves as the "Single Source of Truth" for digital product development. It bridges the gap between design and code by ensuring consistency and efficiency.

Design Token
A semantic variable (e.g., "active.blue.color") that stores design decisions such as colors, typography, or spacing. Tokens allow design changes to propagate automatically (procedurally) across Design Systems and can map 1:1 to variables in the software code.

Design vs. Development
For the purpose of this thesis (Section 1.5):
Design: The process of conceptualizing a user need into a concrete solution proposal (mockups, rationale).
Development: The technical implementation process of building that proposal into functional software (coding).

Digital Twin
In the context of this thesis, a design component in the Design System that mathematically and visually matches the live software's rendering pixel-perfectly. This ensures that mockups created in the system are technically feasible and visually identical.

Downstream
Terms used to describe the dependency and sequence of interventions. Refers to goals, features, or process steps (like the Community Portal) that are dependent on foundational work and are scheduled to be implemented at a later stage.

Feature Development Process
The specific workflow within Blender where new functionalities are conceived, designed, and implemented. This is the primary scope of the intervention.

Gitea
The web-based project management and code hosting platform used by Blender for tracking tasks, issues, and contributions. This is where proposals are posted and discussed.

Hybrid Workflow
An emergent behavior observed during user testing where users mix raw screenshots of the existing software with vector components from the Design System to rapidly create high-fidelity mockups.

Module
A specific area of functionality within the Blender software (e.g., Sculpting, Rendering). Historically, Blender's development teams are siloed into these modules, with specific "module owners" that are responsible.

Open Source Software (OSS)
A software development model in which the source code is made publicly available, allowing anyone to inspect, modify, and distribute it. Beyond technical access, the thesis defines it as a culture characterized by transparency, community-driven collaboration, and distributed governance, where value is co-created by a global network of volunteers and professionals rather than a single closed entity.

Penpot
The open-source, web-based UI/UX design and prototyping tool selected to build the Blender Design System.

Single Source of Truth
The practice of structuring information models and associated data schema such that every data element is represented by (or edited in) only one place. In this thesis, the Design System becomes the single reference point for UI decisions.

Abbreviations

C1 / C2 / C3 / C4
The classification of contributor levels within the Blender ecosystem (see Glossary).

DDP
Design and Development Process

Des-Dev
The interaction between Designers and Developers

DS
Design System

GIMP
GNU Image Manipulation Program, open-source image editor tool (similar to Photoshop)

Hi-Fi
High Fidelity: high or final quality level, opposite to low-fidelity (referring to visual quality of mockup design proposals)

HQ
Headquarters

HR
Human Resources

IP
Intellectual Property

MECE
Mutually Exclusive, Collectively Exhaustive

OS
Operating System (not to be confused with Open Source)

OSS
Open Source Software

ROI
Return on Investment

RQ / SQ
Research Question, Sub-question (e.g., SQ1, SQ2)

UI / UX
User Interface / User Experience

VS Code
Visual Studio Code, Code-writing tool

01

Introduction

Figure 1.1 | Blender Conference, Amsterdam 2025. Photo by Andy Goralczyk

1.1 Context and Background

3D creation software is essential across a wide range of industries, including media & entertainment, advertisement, product design and architecture. The quality of a software product directly affects the efficiency, capabilities, and outcomes for its applications. Moreover, software quality strongly affects user satisfaction, which is a direct determinant of the success of the software development organization. In turn, since the development process is the main determinant of software quality, an optimized development process is crucial for achieving optimal outcomes (Akbar et al., 2018).

Open Source

The digital landscape is increasingly shaped by Open-Source Software (OSS), a software development approach in which the source code, the human-readable instructions that define a program's functionality, is made publicly available and licensed to allow for free distribution and modification (Gacek & Arief, 2004).

OSS projects typically begin as collaborations of independent contributors who volunteer their time and skills, providing code, ideas, or other resources to the project (Scacchi et al., 2006). The approach is motivated by principles of transparency, collaboration, and freedom. Contributors work in arbitrary locations and collaborate almost entirely online (Sack et al., 2006). Over time, these communities grow, sometimes evolving into large-scale projects or formal organizations with full-time developers.

Case Study: Blender

Combining the significance of 3D software and the principles of open source, Blender is an open-source 3D creation suite developed and maintained by the Blender Foundation, headquartered in Amsterdam. Originally a small, proprietary in-house tool, it became open source in 2002. Today, with an estimated 500.000 daily users while being fully funded by donations, Blender has evolved into a significant initiative supported by an ecosystem of volunteer contributors, professional software developers, and industry partners. For this thesis, Blender is examined as a case study to research improvements to the design workflow in OSS development.

As Blender has grown in scale and complexity, it faces increasingly sophisticated challenges that require both design and technical development expertise. Despite its success as an open-source 3D creation platform and its strong technical capabilities, integrating design consistently and efficiently into its historically established technical-focused development process remains difficult, even though Blender wants to.

This thesis investigates why fully leveraging design skills from both internal and external contributors is challenging and explores strategies to better optimize the design workflow in the development process and increase Blender's ability to address complex challenges. The central question at the heart of this thesis is:

How can improving the design workflow enable Blender to leverage design expertise and balance its historically tech-heavy approach?

1.2 Think Open Source!

Improving Closed vs. Open Organizations

While proprietary software development typically relies on centralized control and hierarchical decision-making,

Open Source Software (OSS) operates in a fundamentally different way: voluntary and distributed collaboration.
Besides being a licensing model, open source is a culture and production method where all results are public, and development is driven by a decentralized community of contributors.

For this thesis, this distinction is important. In a traditional closed environment, workflows can be guided or forced by management. In an OSS environment, improvement must be achieved through consensus and distributed contributors. This creates a unique challenge: how to create a lasting change when the workforce is voluntary, transient, and geographically separated?

Open Source Benefits and Drawbacks

OSS promotes openness and collective value creation, while organizations using traditional, closed innovation models depend on intellectual property (IP) protection and access restriction for their survival. This limits accessibility, collaboration and ultimately overall value creation and the benefit for society (Hippel & Krogh, 2003; Balka et al., 2010). Even though corporations may have valid reasons for keeping certain elements proprietary, this restricts them from the many benefits of open-source collaboration (Dinkelacker et al., 2002).

For core developers and organizations, OSS offers substantial benefits in maintainability, code reusability, and higher quality results. Its community-driven development naturally introduces product alignment with diverse user needs (Dinkelacker et al., 2002). Defects tend to be found faster since more people are looking for problems, and code quality and creativity improves as developers contribute to projects they are genuinely passionate about (Sack et al., 2006).

For users and interested contributors, OSS offers unmatched accessibility and cost savings, rapid product innovation, educational opportunities through open code, freedom to customize and contribute, and transparency for understanding functionality and verifying security.

So why isn't everything open source? Because **(viable) open-source business models are hard.** The openness introduces income and motivation challenges, quality inconsistencies from varying contributor expertise, less standardization and efficiency, and coordination difficulties within large, diverse communities where everyone has ideas and access (Balka et al., 2010).

The Reality: Open Source Isn't All or Nothing

The tension between corporate control and open-source ideals reflects a broader societal challenge. Society benefits from innovation, which can be viewed through two distinct approaches (Hippel & Krogh, 2003). **Private Investment** incentivizes individuals by providing private returns and IP rights, where disclosing knowledge can reduce these profits. In contrast, **Collective Action** relies on contributors making knowledge publicly available to reduce the societal downsides associated with the private model.

However, and most importantly, a middle ground exists where private incentives and collective action coexist, creating a **Private-Collective** approach that balances individual and societal benefits (Hippel & Krogh, 2003). To effectively do so, organizations should balance community and commercial needs by maintaining a perception of fairness towards the community, something they value greatly (Balka et al., 2010).

The notion of open source is similarly nuanced as the traditional black-and-white view of open source is overly simplistic (Balka et al., 2010). Early definitions framed it as requiring full IP surrender, yet in practice, selective and strategic adoption is far more common. Projects do not need to be entirely open or fully closed. **Variable openness** enables a balance of collaboration with protection of core assets (Dinkelacker et al., 2002). Organizations can leverage these varying degrees of openness as a strategic opportunity to capture and create unique value (Balka et al., 2010).

Even though Blender is completely open source, it similarly does not fit the binary definition of a "small hobbyist project" or a "corporate product". Over the years Blender established a centralized professional core (The Blender HQ/ Blender Institute) that guides a vast decentralized community. The HQ has the strategic intent to maintain quality, but it relies on the collective community for execution and scale.

Key Takeaway

We are trying to solve a design leverage problem, but we have to do it in an OSS environment where typical top-down, centralized approaches are incompatible. To align with the OSS culture, we must create solutions that guide behavior instead of force it, and that can be used anywhere, anytime

Figure 1.2 | Blender classroom demo scene running on laptops. Photo by Andy Goralczyk

Motivations in Open Source

Typically, open-source contributors contribute their time and expertise voluntarily, without direct compensation. So what drives these individuals?

OSS developers tend to really enjoy their work. Their intrinsic motivation predominates, driven by enjoyment, creativity, flow, intellectual challenge, and a sense of responsibility toward the community and its values. They contribute voluntarily and, crucially, often choose what to work on. Even though intrinsic drivers are the most important, extrinsic motives also contribute. They include career growth and skill development, as OSS contributors can achieve better career opportunities than peers without OSS experience (Scacchi et al., 2006). Overall, the intrinsic feeling of creativity is the strongest predictor of effort, followed by perceived compensation, team satisfaction, and reputation (Lakhani & Wolf, 2005).

At a deeper level, open-source contributors uphold values of transparency, accessibility, and replicability (Balka et al., 2010). The defining characteristics of open source (transparency, community-driven collaboration, and distributed governance) embody principles of

Freedom, accessibility and transparency.

Transparency manifests in full access to source code and design documentation, while collaboration creates decentralized, collective development. Through peer production, individuals and groups worldwide can contribute, and the principle of freedom of use ensures that anyone may utilize the software for any purpose.

Key Takeaway

Unlike corporate motivations of salary and obligation, open source contributors are driven primarily by intrinsic motivations of enjoyment.

Thus, for the adoption of our solution we must rely on these workflow experiences. The solution should feel intuitive, empowering and very enjoyable. The goal is to maximize the time spent in 'creative flow'.

Figure 1.3 | Contributors during the 2025 Blender Conference. Photo by Andy Goralczyk

The Social Gradient of People Involved

In traditional software, there is a clear difference between the creator (Company) and the user. In Blender's ecosystem, this boundary is porous and dynamic. Developers are typically end-users and vice versa (Scacchi et al., 2006). Projects typically start with a few developers addressing their own needs. New contributors initially join as passive users before gradually evolving into active participants. Through contributions like bug reports and feature suggestions, they gain recognition and progress to co-developer, core developer, and eventually influence the project's direction (Gacek & Arief, 2004).

Communities form as developers connect, share expertise, and gain recognition. Projects develop through interactions among developers, software, and communities. Many contributors participate in multiple projects, forming interconnected ecosystems where small core teams drive most development and contributors self-select roles to build reputation and trust (Scacchi et al., 2006).

Newcomers in the ecosystem need to build trust and align interests, by understanding the network's structure before contributing. Progression from peripheral participation to core contribution functions as a rite of passage. Thus, a key skill for OSS developers is building a stable contributor network to ensure the project's longevity (Sack et al., 2006). Free access to information and active participation are essential for community growth, whereas restricting access hinders the attraction of new members.

Understanding this map requires recognizing the unique dynamics of how influence operates within an OSS environment. It manifests materially (through control of code), discursively (through control of discussions), and technically (through expertise) (Sack et al., 2006). In short, influence doesn't just come from technical skill but is mostly a social construct.

Key Takeaway

Open source ecosystems like Blender are not flat but a gradient of trust and influence. Social dynamics have a great influence. Over time, contributors can progress in commitment and involvement.

Figure 1.4 | OSS Contributor Evolution Map

1.3 The Open Source Paradox

From Top-Down to Collaborative Development

Software development was traditionally characterized by a top-down structure, managed by small, tightly coordinated teams with limited user involvement. The emergence of open-source software introduced a fundamentally different, collaborative paradigm. In its approach, development is community-driven, with decision-making distributed across contributors, and users actively participating in the design process, thereby further democratizing the process (Raghunathan, 2024).

The Core Nature of OSS: A Double-Edged Sword

The core nature of open-source development presents a double-edged sword:

On one hand, its defining characteristics are the source of its greatest strengths. On the other, they also cause its most significant challenges.

Open-source projects typically lack the traditional structures, system-level design considerations, or detailed schedules that guide conventional software development. They often operate without formal hierarchies, relying instead on geographically dispersed volunteers with both prescribed and unprescribed roles who use diverse information and communication sources such as email and code (Sack et al., 2006). While realizing transparency with activities and results are publicly visible, formal management, budgets, or schedules are rare. Design decisions emerge organically from the ongoing interactions within this distributed community.

The key advantages of the OSS approach are said to emerge precisely because of this loose and implicit structure (Sack et al., 2006). Free from rigid project plans or lists of deliverables, developers can focus on things for which they have a real passion. This autonomy is credited with leading to faster discovery of issues and code that is written with more care and creativity.

However, this same lack of formal process is what creates complex and unpredictable development. The absence of explicit conventions, standard processes, or clear agreements on critical details like graphic styles and user experience principles leads to contributions that are difficult to coordinate (Sack et al., 2006). This creates a fundamental tension for any OSS initiative attempting to scale: interventions are needed to solve these coordination issues, but they must be implemented with great care to protect the very unstructured and autonomy-driven approach that generates the core advantages of the open-source model.

The Challenging Development at Blender

This fundamental tension of open source is also visible in large-scale projects like Blender. It has evolved from a small software into a large and more formalized organization. Yet Blender retains much of the DNA of its origins, following an online-first principle where every design proposal, mockup, decision, and discussion is documented digitally to allow for remote and community involvement. This structure still reflects processes seen in early-stage open-source projects.

A visible example of the paradox for Blender is its reliance on separate modules for development, where developers work on different aspects of Blender (e.g. modelling, animation, rendering). While dividing the software into isolated modules is efficient for managing complex code (allowing developers to work without breaking other parts), it is challenging for design. Code benefits from isolation; design is holistic.

At its core, Blender's development process is inherently dynamic and difficult to manage precisely because it is built on a community-focused model where anyone can be involved and contribute. In turn, contributions naturally vary widely in quality, intent, and commitment, making coordination a continuous challenge.

When this community coordination challenge is combined with Blender's complex ecosystem, a mix of professional developers, a globally distributed community of volunteer contributors, a non-profit governance model reliant on donations, and the integration of decisions and innovation from both the professional core and the broader community, makes its development process not merely a technical task but a complex systemic design problem.

The Impact

Viewing large-scale open source as a systemic design problem is important since the consequences of its internal challenges are not simply contained within the development team. OSS development is an inherently complex, multi-step process where technical, creative, and organizational decisions intersect. Developers, designers, contributors, users, and the larger organization all play interconnected roles, forming a socio-technical ecosystem.

As organizations like Blender scale, this ecosystem becomes increasingly vulnerable to critical coordination challenges. With geographically separated teams working asynchronously, issues like misaligned priorities, communication breakdowns, conflicting contributions, and the difficulty of maintaining a consistent quality and vision become magnified. In this distributed environment, effective structure, processes and coordination is absolutely essential (Hunsen et al., 2020; Alsaqqa et al., 2020).

Crucially, these process-related hurdles do not just affect internal workflows, they also impact the final software product and degrade the user experience. Poor development practices born from a lack of coordination inevitably lead to increased software complexity and lower quality. This, in turn, results in lower user satisfaction, rising maintenance costs, and even potential financial loss, demonstrating that solving these internal process challenges is essential for the long-term health and success of the project (Ghanbari et al., 2019).

To better understand how we can possibly create value in this complicated environment, the next section explains the unique perspective on value creation and innovation in OSS.

→

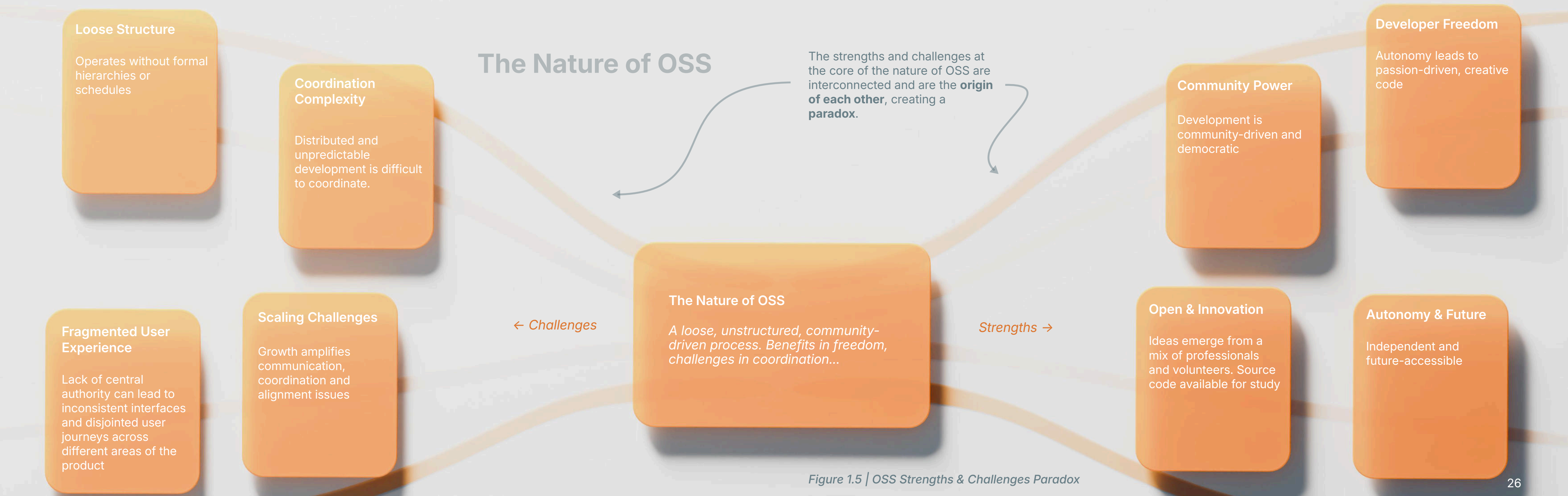


Figure 1.5 | OSS Strengths & Challenges Paradox

1.4 The Design & Development Workflow

A workflow are the steps taken to get to a result. Blender's development process consists of two main workflows: design and development. The design process translates problems into proposals that developers can then implement (Ozkaya, 2020). In essence, the Design process focuses on the "what" and "why". while Technical Development focusses on the "how".

Development Workflow Defined

In this thesis, development is defined as the technical building and implementation process that translates a proposal into a fully functional and operational product. (Yas et al., 2023; Alsaqqa et al., 2020). Typically, this involves tasks such as coding, bug-fixing, deployment and maintenance. Blender's software developers and external contributors are the main stakeholders concerned with these tasks.

Design Workflow Defined

Generally, design is understood as a user-centered problem-solving process, which integrates constraints and insights to shape the final qualities of a product or service (Best, 2006; Kim et al., 2017).

Specifically in the context of this thesis, design is the process that translates and conceptualizes a user problem/need into an executable solution proposal (Yas et al., 2023). For Blender, these proposals are concrete visuals such as high-fidelity UI mockups and text descriptions explaining the mockup.

The design workflow is the sequence of steps a designer executes within the design process: their tasks and tools. During this design workflow, a holistic and systemic perspective is considered, and takes a user-centered approach while considering the broader context, diverse stakeholders involved, and the overarching vision of the organization. The primary stakeholders involved in this process are Blender's designers and strategy roles, but everyone takes on design tasks in their own projects.

To facilitate this understanding, the visual below (Figure 1.6) provides an overview of the entire process. This will serve as a reference for Chapter 3, where a detailed map of Blender's specific development process including challenges is introduced.

We Need Both

Although design and development are separated linearly for simplicity, in reality the process is far more complex, iterative, and unpredictable, with frequent back-and-forth between the two. Hence, many process visualization methods can be used to illustrate its linear-cyclical nature (van Boeijen, A. et al., 2013; Roozenburg & Eekels, 1998).

For example, a design may prove difficult to implement, requiring a return to the design phase with the developer, or a development solution may fail to capture the original design intent. Solutions aimed at connecting the two therefore require flexibility and adaptability throughout the process.

Ultimately, it is crucial to recognize that both design and development are integral components. They should be employed collaboratively to tackle complex challenges, creating a synchronized and mutually supporting relationship between the two.

The Focus of This Thesis

In OSS to accelerate Product Innovation, we have to improve the process. This thesis operates at this meta-design level: designing the process so that others can design the product.

Key Takeaway

While the start of the design process is mostly taken up by user research and strategic problem-solving, the intervention proposed in this thesis focuses specifically on improving the second half of the design process: the creation of solution proposals (mockups). Justification for this can be found in Chapter 3.

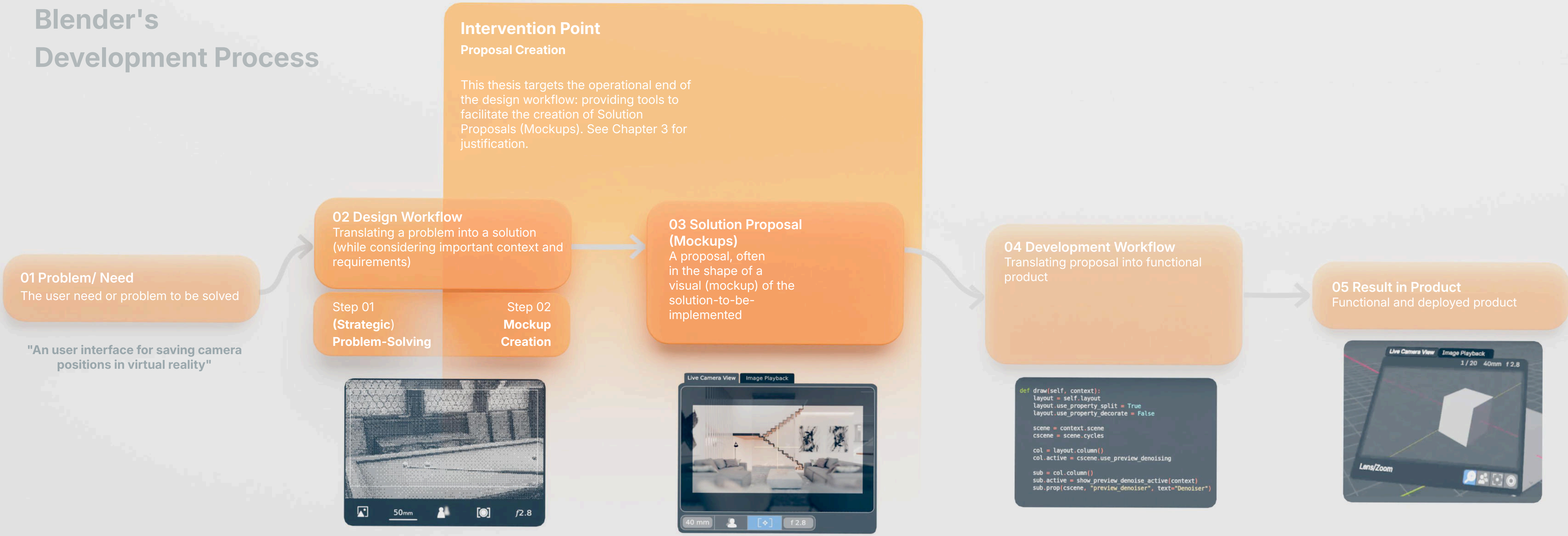


Figure 1.6 | Blender's Development Process overview. I'm using this specific example since I helped by creating the mockup for this feature

1.5 The Three Layers of Design

Design's purpose in OSS

Having established the challenges inherent to open-source development, the opportunity design presents can be explored. Successful adoption of open source relies more on social, cultural, and leadership principles than on tools or technical changes (Dinkelacker et al., 2002). Therefore, design presents a clear opportunity in OSS development to facilitate this socio-cultural alignment by improving communication, aligning stakeholders, and making collaboration more effective.

In the context of open source, understanding design is less about the code or technical outcomes and more about the interactions between people, their discussions, and the shared technical artefacts that shape the process (Sack et al., 2006), as illustrated with the designer-developer gap. Viewed through this lens, design methodologies can provide the frameworks and boundary objects needed to address coordination challenges. Integrating design methodologies in OSS enhances development by creating inclusive, collaborative environments that drive innovation and user-centric solutions (Raghunathan, 2024). They offer a way to organize contributions, coordinate tasks, maintain coherence, and support collective decision-making in these distributed, loosely structured teams (Sack et al., 2006).

Defining a Framework for Design

To accurately analyze the role of design within this complex ecosystem, it is necessary to differentiate between its various levels of operation. Since the goal of the thesis is to use systemic insights to drive practical interventions, we differentiate the multifaceted discipline of design in its abstractness. Design management literature provides a clear framework for this, categorizing design's influence at three interconnected levels: strategic, tactical (or process), and operational (or implementation). This hierarchical model is central to the discipline, as exemplified by the foundational structure of Kathryn Best's Design Management (2006). These levels are also reflected in Blender's own internal coordination efforts (Blender Foundation, 2021).

Layer 01

Strategic Design (The "Why")

This is the most abstract level, where design initiatives are conceived and aligned with the organization's core mission. As Best (2006) describes, this stage is focused on "identifying opportunities for design, interpreting the needs of its customers, and looking at how design contributes to the overall business." It answers the question, "Why are we doing this?" by setting the vision, principles, and long-term goals that guide all subsequent design and development work.

Layer 02

Tactical Design (The "How")

This crucial middle layer translates abstract strategy into actionable plans and systems. Best (2006) frames this as "Managing the Design Process," where strategy is made "visible and tangible through design." This level answers, "How will we achieve our strategic goals?" It is concerned with structuring the collaborative workflows, team structures, and enabling frameworks, such as design systems and development methodologies, that provide the organizational infrastructure for consistent and efficient execution.

This is the key layer this thesis will act on.

Layer 03

Operational Design (The "What")

This is the most tangible level, focused on the day-to-day execution and delivery of design outcomes. Referred to by Best (2006) as "Managing the Design Implementation," this stage is where the tangible "products, services and experiences" are created. It answers the question, "What are we making?" This encompasses the specific creative decisions and artifacts of UI/UX design, such as mockups, prototypes, and user journeys - the concrete proposals that are handed off for technical implementation.

Where to Solve The Problem?

By distinguishing these levels, this thesis can precisely define its intervention point. The core argument is that to solve the persistent challenges observed at the operational level (inefficient handoffs, inconsistent UI), interventions must be made at the tactical level by improving processes and tools. These tactical solutions, in turn, must be guided by a clear understanding of the strategic level to ensure they effectively contribute to Blender's overarching goals for innovation and value creation.

Key Takeaway

To improve the challenges on the operational level, the tactical level is the area of intervention, guided by the strategic level.

Three Layers of Design

"Why are we doing this?"

Defining the mission, vision, and long-term goals. Identifying opportunities and understanding needs to set the overall direction for design.

Vision, Principles, Core Mission, Long-term Goals.

"How do we do it and achieve our strategic goals?"

Translates strategy into actionable plans and systems. Creating infrastructure, structuring workflows and teams

Actionable Plans, Workflows, Team Structures, Frameworks

"What are we making?"

Design and development execution. Design artifacts.

Day-to-day Execution, Mockups, Prototypes

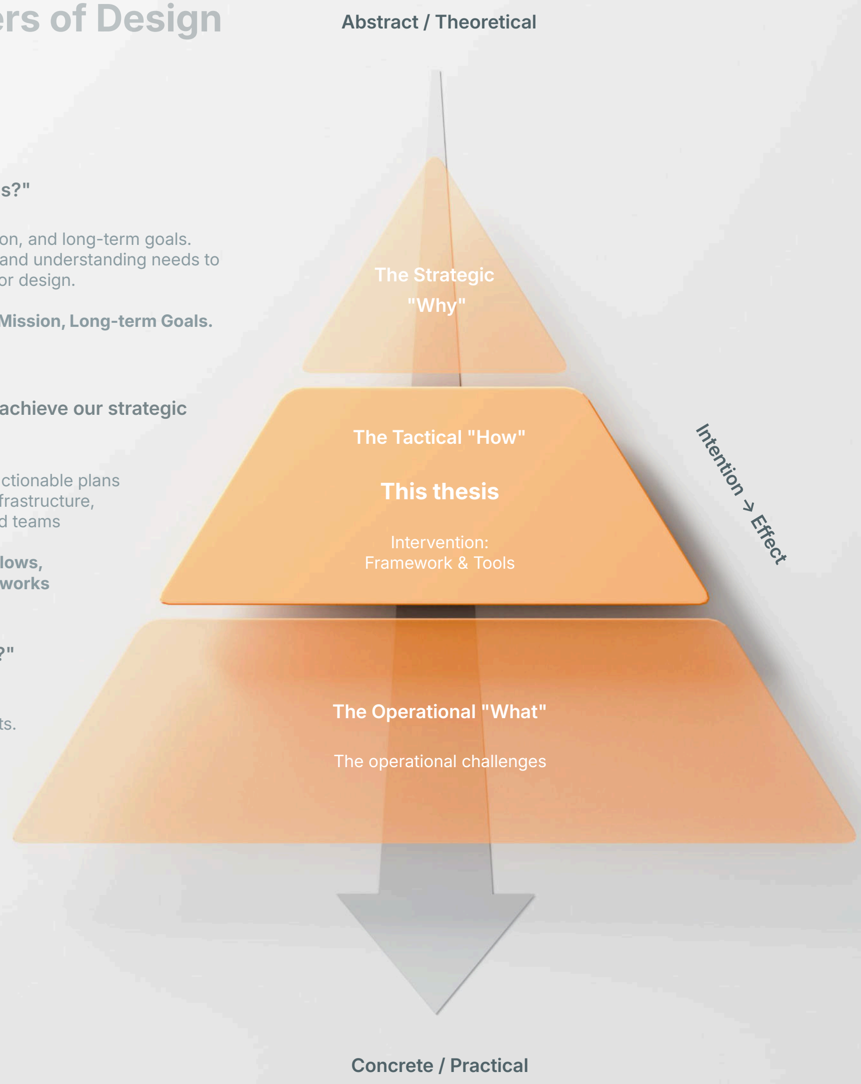


Figure 1.7 | Illustration Hierarchy 3 Levels of Design Types

1.6 The Problem, Research Question & Objectives

Despite Blender's success as a decentralized ecosystem, the design expertise experiences significant friction. While technical (developer) contribution has a clear path, design contribution lacks the tactical infrastructure to be efficiently used. Valuable design expertise is lost in translation or blocked by barriers, leading to missed opportunities for innovation.

Fragmented coordination, differing contributor priorities, informal governance, and the absence of shared design frameworks often results in inefficiencies, misalignments, and a reactive approach to design. The existing infrastructure and culture supports technical expertise contributions, but integrating design to leverage opportunities remains difficult: external designers face barriers due to limited knowledge accessibility and the dominance of technically driven processes, causing valuable design expertise to fail to scale into actionable improvements and leading to missed opportunities for innovation and value creation.

Research Objective & Gap

While the general challenges and characteristics of OSS organizations are explored, there is a gap in actionable insights on how to develop and implement solutions to emphasize design with the goal to create a more balanced design and development relationship.

Design is important in OSS as it largely shapes the user experience, which in turn drives the adoption and viability of the software. **While the technical development in OSS is essential for functionality, design provides the integrative perspective that connects the technical features into a usable, unified product.** To realize these important outcomes, the development process itself must be optimized to effectively integrate this design expertise (Akbar et al., 2018).

Thus, the research aims to address the imbalance between design and development within Blender. It investigates how design tools and processes ('tactical' infrastructure) can be integrated into a historically developer-centric culture to empower contributors and effectively leverage the (limited) design expertise.

Explanation & Approach

Currently, Blender's growth has exposed an imbalance. While the ways to contribute technically are well established, the ways to capture and use design expertise are still underdeveloped. This misalignment leads to challenges where valuable design skill is lost or difficult to implement (see Section 3.4-3.6). To address this, the research first analyzes why this difference exists within Blender's specific community-driven context. It argues that simply increasing the number of designers is insufficient without first resolving the underlying workflow barriers (see Section 3.7 & 4.4). Crucially, to succeed within Blender's culture of autonomy, an intervention cannot be top-down or forced. So to align and ensure adoption, the proposed solution must be a practical workflow tool that is empowering and optional (see Section 1.2).

Definition of Successful Project

The project is considered to be a success when the design workflow at Blender is improved by a tool and process that empowers contributors to design and increases design's influence in the development process - validated by user adoption and positive user experiences.

Research Question

To achieve this objective, this thesis will be guided by a central research question, supported by three sub-questions:

Can a practical intervention improve the design workflow of the core Blender team, to increase Blender's ability to leverage their design expertise, increase efficiency and improve working experience?

Sub-questions (SQ's)

SQ1: Diagnosis
What is the current state of the development process within Blender's ecosystem, and where are the primary points of friction and misalignment between different phases and stakeholders?

SQ2: Analysis
What are the key barriers and opportunities for leveraging design interventions within Blender's existing community-driven, open-source culture?

SQ3: Synthesis
What actionable frameworks and design tools can be proposed to effectively bridge the gap between systemic insights and practical solutions, while preserving the unique open-source work culture and its benefits when implementing design-led interventions?

1.7 Scope: The Area of Intervention

To effectively answer the research questions, this thesis will specifically focus its investigation on a critical domain within the open-source ecosystem:

The design process and workflow inside Blender's feature development process. The phase that translates a problem into an initial conception of a new functionality to its final solution proposal and refinement.

This scope represents the critical area where design expertise is translated into product value: concrete software functionality. By targeting this specific phase, the intervention can directly address the workflow inefficiencies that currently hinder Blender's ability to fully leverage its design talent.

It is within this process that all design expertise contributions occur. As has been previously identified, effective integration of these skills is essential for ensuring the software quality and usability critical to a project's success. Although this issue is not new, anecdotal evidence still highlights persistent challenges (Zhang et al., 2025). The landscape of software development is constantly evolving and new tools such as Penpot and Figma have emerged to improve cross-disciplinary workflows. Since Blender also faces these challenges, it makes for a highly relevant scope for this study.

Contributions

Within this defined scope, "contributions" are understood as any effort made with the intention of improving a project's functionality, performance, or overall value. In OSS, contributions take many forms beyond core software development, including documentation, testing, community management, and user support. However, to maintain a clear focus, this research will limit its analysis to the software-related human contributions that directly influence the feature development of the final product.

Activities such as hardware contributions or optimizations, external consultancy, and other peripheral contributions fall outside the scope of this study.

By narrowing the focus to the feature development lifecycle, this thesis can conduct a deep and meaningful analysis of the most critical interactions, providing a solid foundation for its findings and recommendations.

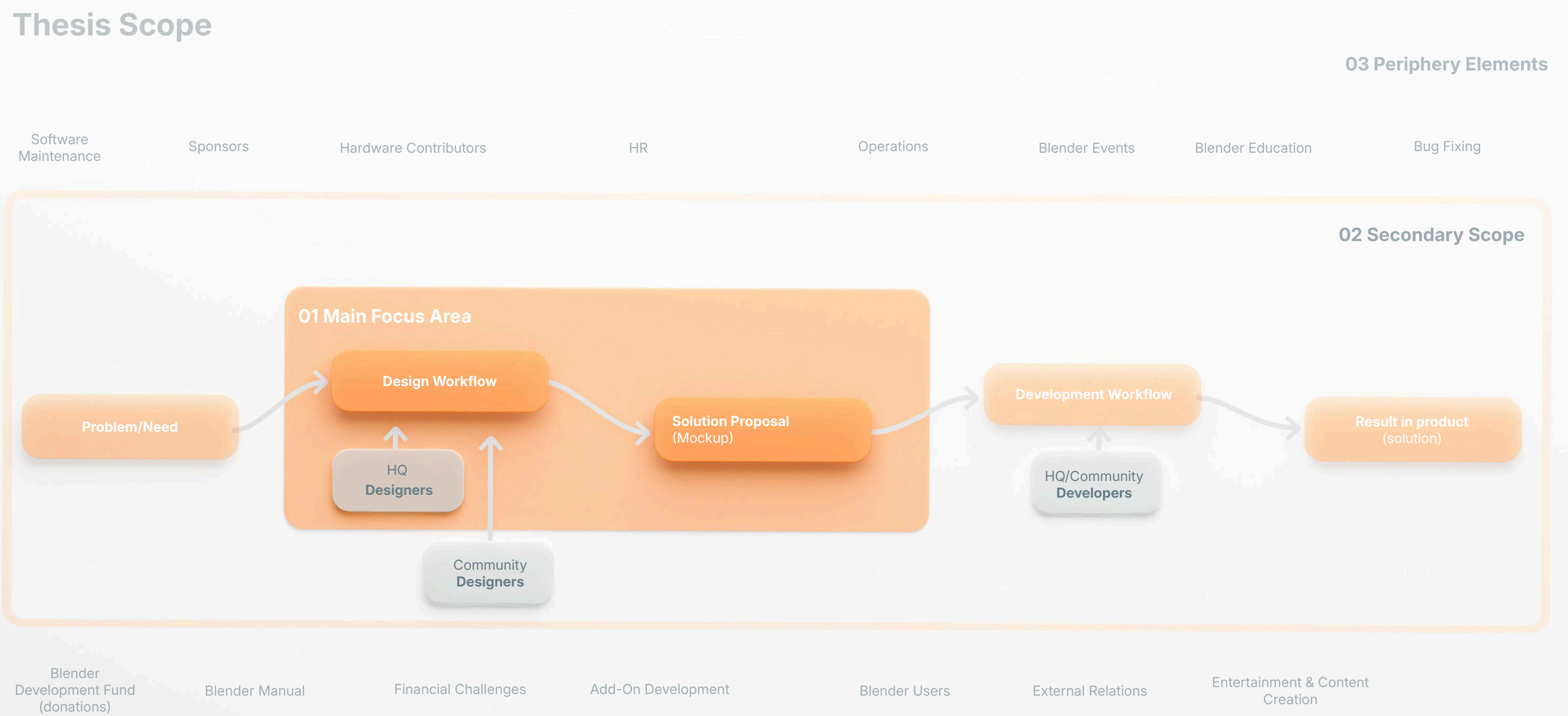


Figure 1.8 | Scope overview map

02

Methodology

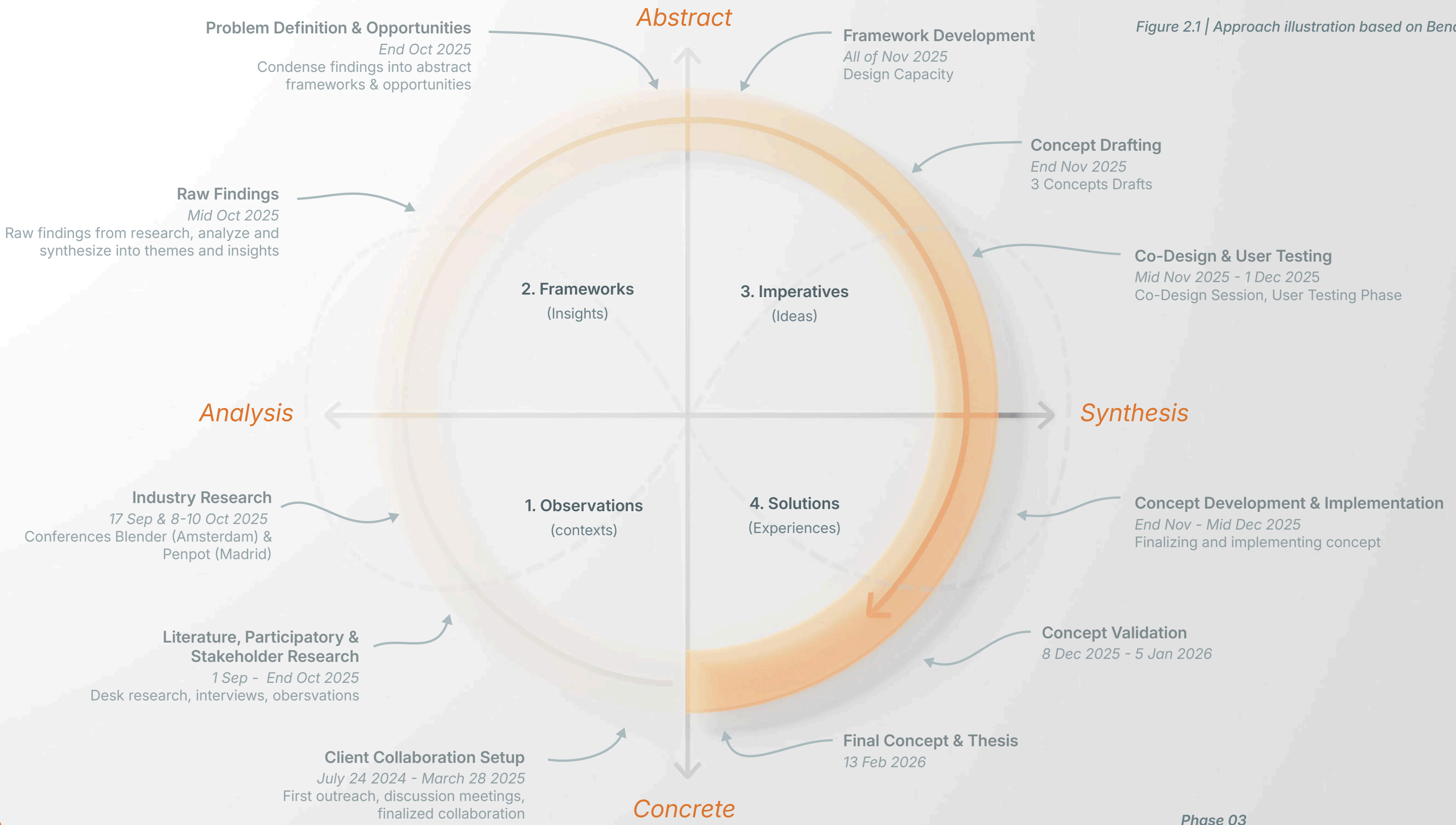


Figure 2.1 | Approach illustration based on Bender-Salazar (2023)

2.1 Project Approach

This research is guided by an **iterative, abstract-concrete approach** adapted from the work of Bender-Salazar (2023). This approach provides a structured yet iterative process for investigating complex problems and developing concrete solutions. It fundamentally combines divergent thinking to explore a wide range of possibilities with convergent thinking to isolate and refine the most viable solutions.

The selection of an abstract-concrete approach was a deliberate choice, driven by the nature of the research problem itself. The challenges within Blender's ecosystem are not purely technical but are also rooted in social dynamics, community culture, and collaborative workflows. The research goal is not only to understand the abstract system level but also to translate those insights into concrete, actionable improvements. The selected approach is well suited for this task, as it provides a robust methodology for bridging the gap from abstract, systemic understanding to concrete, practical interventions. This aligns with the objective of developing interventions that are both impactful and sensitive to the existing open-source culture.

The structure of this project follows the four distinct yet overlapping phases of this hybrid model, as visualized above (Figure 2.1). While this process somewhat reflects the divergent and convergent stages of the well-known Double Diamond model, this thesis will, for the purpose of academic rigor, adopt the specific terminology from the cited framework, which labels the four phases based on their primary outputs:

Phase 01
Observations (Contexts)

The initial phase involved an extensive exploration of the problem space to gather concrete contextual data. This was achieved through three primary activities: a comprehensive literature review of open-source software dynamics, a series of semi-structured interviews with key stakeholders within the Blender ecosystem, and participatory observations while contributing to the development environment at the Blender office. The goal was to understand the environment, identify key stakeholders and their pain points, and gather observations about the development process

Phase 02
Frameworks (Insights)

In this phase, the raw data and observations were synthesized to create abstract understanding and frame the core challenges. This involved a thematic analysis of the interview transcripts and literature, allowing for the identification of recurring patterns, systemic frictions, and key insights. The output was a clear problem definition and the analytical frameworks (such as the challenge map presented later in this chapter) that articulate the primary points of misalignment.

Phase 03
Imperatives (Ideas)

Shifting from analysis to synthesis, this phase focused on abstract ideation to conceptualize potential solutions. Drawing from the established frameworks and problem definition, this creative phase was used to generate strategic principles and high-level concepts for practical design-led interventions that could address the identified inefficiencies in Blender's development process.

Phase 04
Solutions (Experiences)

The final phase focused on translating the imperatives into concrete and actionable recommendations. This represents the tangible output of the research: a coherent set of validated proposals, including process frameworks and design tools. These solutions are designed for implementation to improve collaboration, efficiency, and value creation within the Blender ecosystem.

2.2 Research Strategy

To answer the questions at the heart of this research, a qualitative case study strategy was employed. The approach is qualitative because the research goal is to gain a deep, contextualized understanding of a complex social phenomenon. The leverage of design within an open-source ecosystem is a complex socio-technical process that cannot be understood by studying variables in isolation. A case study allows for a holistic and richly detailed examination of the interactions, workflows, and cultural dynamics that shape innovation, providing the depth necessary to develop contextually relevant insights and solutions.

Blender was selected as the case for this investigation for several reasons:

- **Representativeness:** As a globally successful and mature open-source project, Blender serves as a prime example of a large-scale OSS ecosystem. The coordination and integration challenges it faces are representative of those encountered by other major community-driven projects, making the findings from this study potentially transferable to a broader context.
- **Problem Relevance:** The Blender development process directly exhibits the central tension of this research: the friction between a historically technology-driven culture and the growing need for integrated, (systemic) design expertise. This makes it an ideal environment in which to study the research problem directly.
- **Access and Richness of Data:** The direct collaboration with Blender during this thesis provided an invaluable level of access. This enabled direct engagement with key stakeholders, observation and even consistent participation in internal processes, and a nuanced understanding of the organizational culture that would not be possible through external analysis.

To ensure the investigation remained focused and feasible, the study was carefully scoped. As outlined in the Scope (1.7), the primary unit of analysis for this case study is not the entire Blender organization, but specifically design leverage related to feature development. This scope was intentionally selected because it is the critical area where the critical design interactions occur and where the core challenges of coordination, communication, and knowledge sharing manifest most clearly.

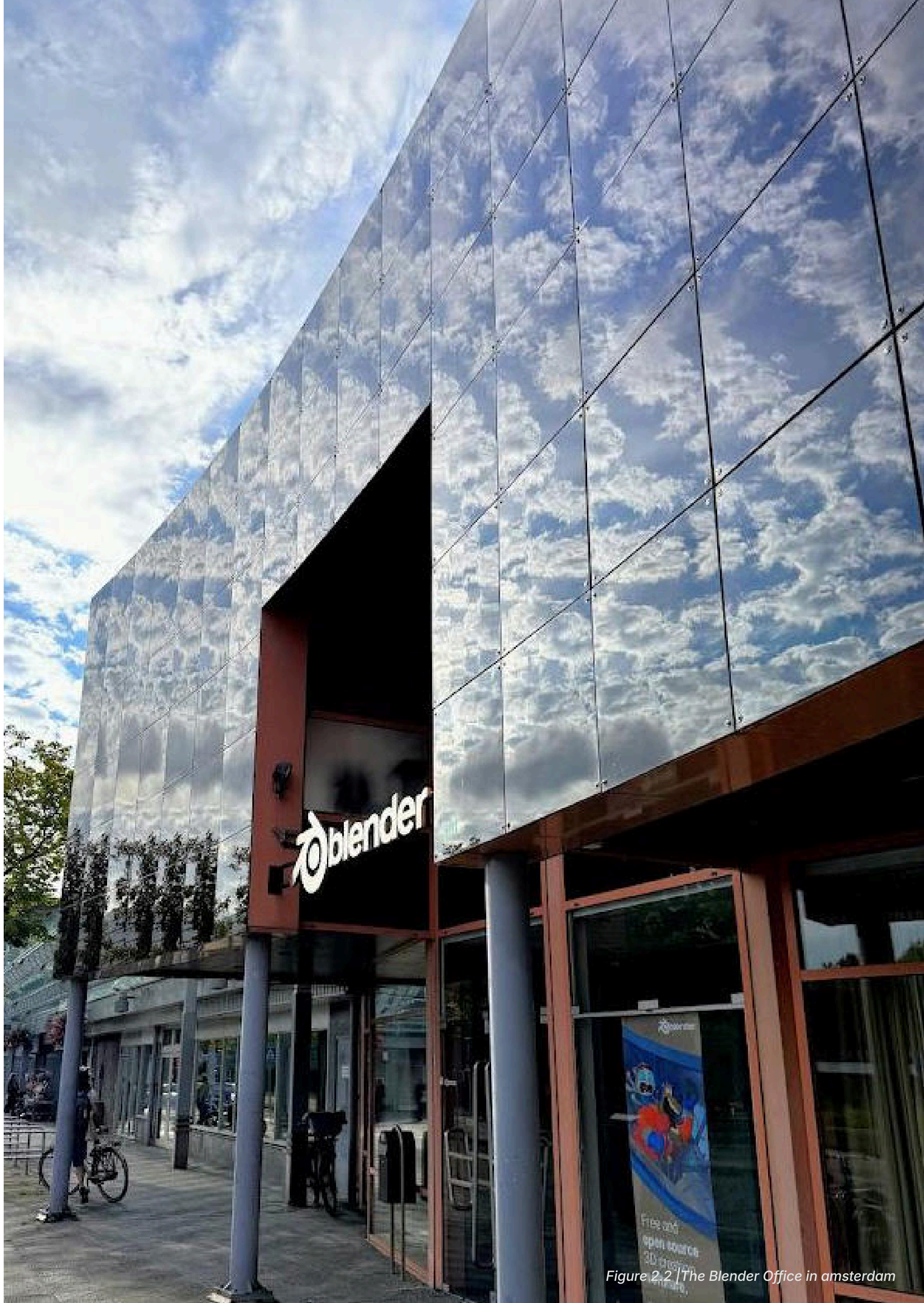


Figure 2.2 | The Blender Office in amsterdam

2.3 Data Collection Methods

To construct a rich and multi-faceted understanding of the Blender development ecosystem, this research employed a strategy of data triangulation, integrating data from four primary qualitative sources. This approach ensures that the findings are robust and well-grounded, as insights from one method could be disproven, corroborated or contextualized by others. Listed below are the different data collection methods:

Data Collection Method 01

Literature Review

A comprehensive review of existing literature and documentation was conducted to establish a strong theoretical foundation for the research. This review served two primary purposes: first, to understand the established discourse on open-source software (OSS) development, community-driven innovation, and designer-developer collaboration; and second, to identify the specific research gap concerning the leverage of design in large-scale OSS projects. The review included a total of 70 academic journals, literature articles, relevant books, and official Blender documentation.

Data Collection Method 02

Semi-Structured Interviews

The primary source of empirical data was a series of 10 in-depth, semi-structured interviews with key Blender stakeholders, averaging one hour in duration. This format was chosen for its ability to combine a consistent line of inquiry with the flexibility to probe deeper into emergent themes and individual experiences. The interviews were set up to capture the "actual experience" of those involved in the development process, providing rich, first-hand experiences of workflows, communication practices, and perceived points of friction. More information about the interviews can be found in Appendix B.

- Sampling: Participants were selected to intentionally include a diverse range of roles and perspectives. This ensured the data was comprehensive and directly relevant to the research questions. The anonymized roles of the participants and their relevance to the research are detailed in Table 2.1.
- Process: The interviews were conducted in person between September 2025 and January 2026. Most sessions were audio-recorded with the explicit consent of the participant and were later transcribed verbatim to ensure accuracy and supported cross referencing during the analysis phase.

Throughout the remainder of this project, references to interview sources will be made using the general term “participant” without any further details regarding their function and experience to ensure anonymity.

Figure 2.3 | Word cloud of word frequency during interviews, where larger words appeared more frequently

Participant	Role Description - Discussion Topics	Organization	Length & Format	Relevance to Research
P1	(Product) Designer - key Blender design knowledge, UI, UX	Blender	1-hour in person	On-the-ground design/ development
P2	Head of Operations - people at Blender, values, people challenges	Blender	1-hour in person	Relationships and work culture, values
P3	Head of Product - Blender design development process, strategy implementation	Blender	1-hour in person	Implementation of high-level strategy, dev decision making
P4	Current CEO - Blender's philosophy, design principles	Blender	Numerous conversations	High-level strategic perspective, history, founding values
P5	Soon to be CEO/ current COO - Strategy, Blender's challenges, opportunities	Blender	1-hour in person	High-level strategic perspective
P6	Software Developer - Challenges in development	Blender	1-hour in person	On-the-ground development
P7	HR Consultant - Work culture, people, workflows, Blender's challenges	External Consultancy	1-hour in person	External unbiased perspective, relationships and work culture
P8	IT & Infrastructure Manager - Strategy to practise translation	Blender	1-hour in person	On-the-ground development
P9	Development Manager - development coordination	Blender	Half-hour in person	Mid-Level strategy, coordination
P10	Graphic/Systems Designer - Design workflows in dev, design systems, process	External Design Studio	1-hour in person	External design/dev workflow perspectives

Table 2.1 | Participant details of interview sessions

Data Collection Method 03

Participatory Observation

To gain a deeper, more tactile understanding of the ecosystem that goes beyond what participants might articulate in an interview, a method of participatory observation was employed throughout the duration of the project (September 2025 to February 2026). This involved being actively embedded within and contributing to the development environment, allowing for the observation of the difference between formulated experiences (interviews) and actions.

The participations included:

- **Meetings:** Attending scheduled weekly design discussion, brainstorming and problem-solving meetings where ongoing projects and topics were actively discussed and problem-solved.
- **Projects:** Contributing directly to design tasks, such as creating mockups and solution proposals for developers and publishing them to the official Blender projects platform.
- **Creating and testing improvements:** Developing workflow tools, including design systems, to support active and future development. This is the ultimate goal of this thesis.

This ethnographic approach provided valuable contextual data. Field notes were systematically taken to document observations of meetings, online forum discussions, and informal interactions, directly capturing data about the collaborative process.

Figure 2.4 | Me observing work culture and team interactions at the Blender HQ office

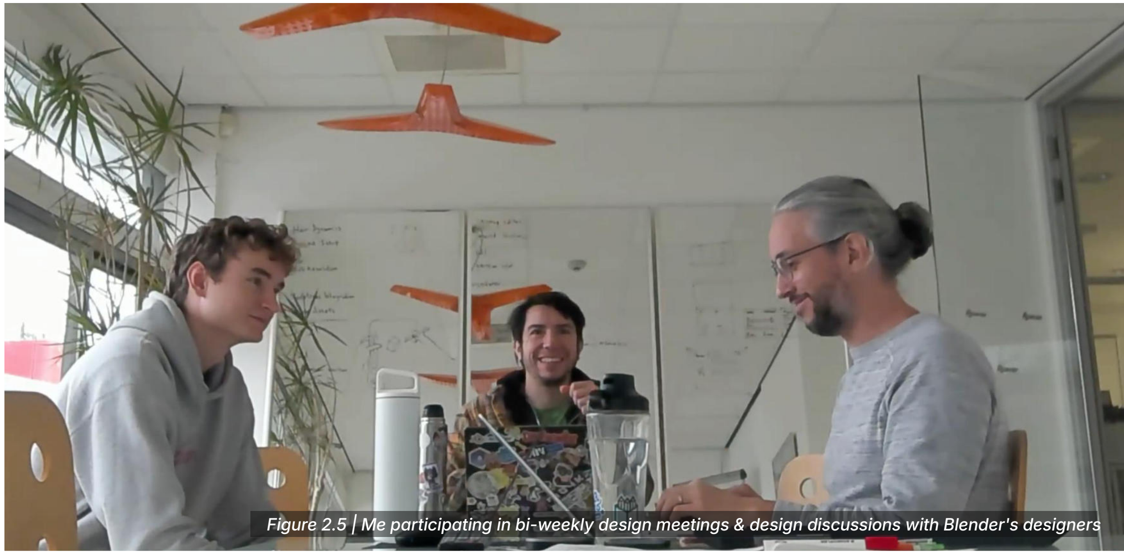


Figure 2.5 | Me participating in bi-weekly design meetings & design discussions with Blender's designers

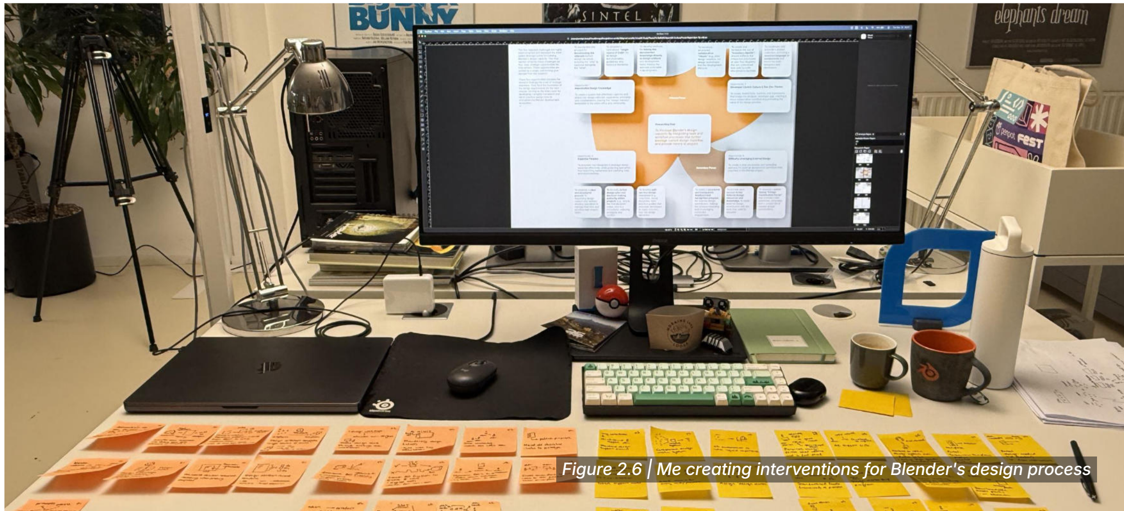


Figure 2.6 | Me creating interventions for Blender's design process

Figure 2.7 | Me and Blender's founder in conversation

Data Collection Method 04

Professional Engagement and Field Research

To situate the case study within the broader professional discourse and to validate its findings against the state-of-the-art in the field, the research was enriched through active professional engagement. Attending two key industry conferences served as a crucial method for gathering contextual data, understanding the challenges and solutions emerging across the open-source landscape, and providing a macro-level perspective that a single-case investigation alone cannot offer.

The Blender Conference 2025 (Amsterdam)

The first event was a three-day attendance at The Blender Conference, the primary annual gathering for the entire Blender community. This conference provided opportunities for informal ethnographic research within the case study's specific environment. Unlike formal interviews, this setting facilitated numerous spontaneous and candid discussions with a wide spectrum of stakeholders - from casual end-users to core developers and professional designers. These conversations offered unfiltered insights into community sentiment, developer priorities, and the perception of Blender's development process. This direct engagement was invaluable for capturing the collective sentiment of the community and gathering rich, spontaneous data that provided extra industry context next to the more structured findings from the interviews and observations.

Figure 2.8 | Keynote during Blender Conference 2025

Figure 2.9 | Penpot Fest 2025, Madrid

Figure 2.10 | Me interviewing and discussing with Penpot's CEO and other key stakeholders during Penpot Fest 2025

Penpot Fest (Madrid)

For the second event I visited the two-and-a-half-day Penpot Fest, a conference specifically dedicated to the intersection of design and open source. This experience was strategically vital for contextualizing and validating the research. While the Blender Conference offered depth within the case, Penpot Fest provided breadth across the field.

Attending keynotes and participating in discussions with professionals from a wide array of OSS projects focused directly on the core themes of this thesis: designer-developer interaction, collaborative workflows, and the implementation of processes to improve development. This engagement confirmed that the challenges observed within Blender are not unique but are part of a significant, industry-wide conversation. Furthermore, it provided valuable exposure to emerging best practices and innovative tools being pioneered in other projects, directly informing the recommendations of this study.

Figure 2.11 | Me interviewing and discussing with stakeholders at Penpot Fest conference, Madrid

Figure 2.12 | Engaging in Design in Open Source workshops

Figure 2.13 | Me and Blender Head of Product asking questions during Q&A sessions

Figure 1.14 | Me engaging in Keynote Q&A's

Figure 1.15 | Me and Blender Team discussing findings

2.4 Data Analysis Techniques

The qualitative data gathered from the interviews, participatory observations, and field research required a systematic process to identify meaningful patterns and synthesize them into actionable insights. To achieve this, a thematic analysis approach was used. Thematic analysis is a well-established method for identifying, analyzing, and reporting patterns (or "themes") within qualitative data. It provides a flexible yet structured process for moving from a large body of unstructured data to a rich, detailed, and insightful interpretation (Braun & Clarke, 2006).

The analysis was conducted in a series of iterative phases using a digital whiteboard to allow for visual clustering. The process followed these specific steps:

1. Familiarization and Data Extraction: During this phase the key data relevant to the research questions was extracted from the source data. This involved collecting significant quotes, statements, and observational notes representing a friction, a belief, or a workflow bottleneck.

- **Example:** A participant quote of "[For design knowledge,] You should ask the person that worked on it" was extracted since it represented a potential friction of inaccessible knowledge.
- **Example:** An observation of "a designer manually editing screenshots" was extracted since it can represent a workflow inefficiency.

2. Clustering and Theme Formation: The extracted data points were placed on a virtual canvas. These data points were then iteratively clustered based on their similarity, dissimilarity and conceptual relationships. This bottom-up process allowed themes to emerge from the data itself. Later this bottom-up approach was combined with the top-down insights from the literature review for a complete result. Initial clusters were formed around recurring issues, such as communication, designer-developer interaction, and inaccessible design knowledge.

- **Process Example:** I noticed a pattern of data points related to "justifying design decisions" and "coding is considered to be the real work" These were initially grouped under a working cluster titled "Attitude/Culture towards Design."

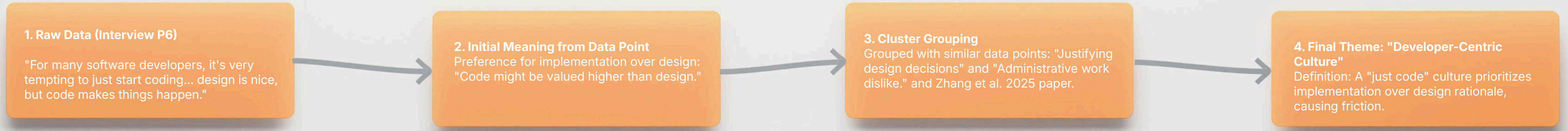
3. Reviewing and Refining Themes: The initial clusters were reviewed against data from the literature review and participatory observation to ensure validity (Triangulation). I applied the MECE principle (Mutually Exclusive, Collectively Exhaustive) to sharpen the definitions.
To ensure the analytical rigor and mitigate researcher bias, this iterative review also included a search for opposing evidence. In this iterative phase, some clusters were combined to form broader, more substantive themes, while others were split. The "challenge map" in Section 3.3 is a direct output of this process.

- **Refinement Example:** The initial cluster "Attitude/Culture towards Design" was cross-referenced with recent literature on Open Source culture (Zhang et al., 2025). Using these literature insights, it was refined and renamed to "Developer-Centric Culture" (See Figure 2.16). Conversely, vague clusters regarding "time zones" and "remote work" were merged into the broader "Communication and Coordination" theme, as they resulted in the same friction: distinct silos.

4. Defining and Naming Themes: The final analysis resulted in 10 distinct themes/challenges. Each challenge was given a descriptive definition. These challenges were then categorized into four categories: Coordination, Workflow & Culture, Strategy & Overview and Community. They directly inform the problem definition in the next chapter. For the result see the overview in Figure 2.16 below.

Furthermore, the recurrence and weight of data points within the 'Developer-Centric Culture' cluster identified it as a foundational root cause theme, distinguishing it from operational symptoms like 'screenshot editing.' This distinction was crucial for structuring the Problem Definition in Chapter 3.

Example of Analysis Progression



Themes (Challenges) Tree

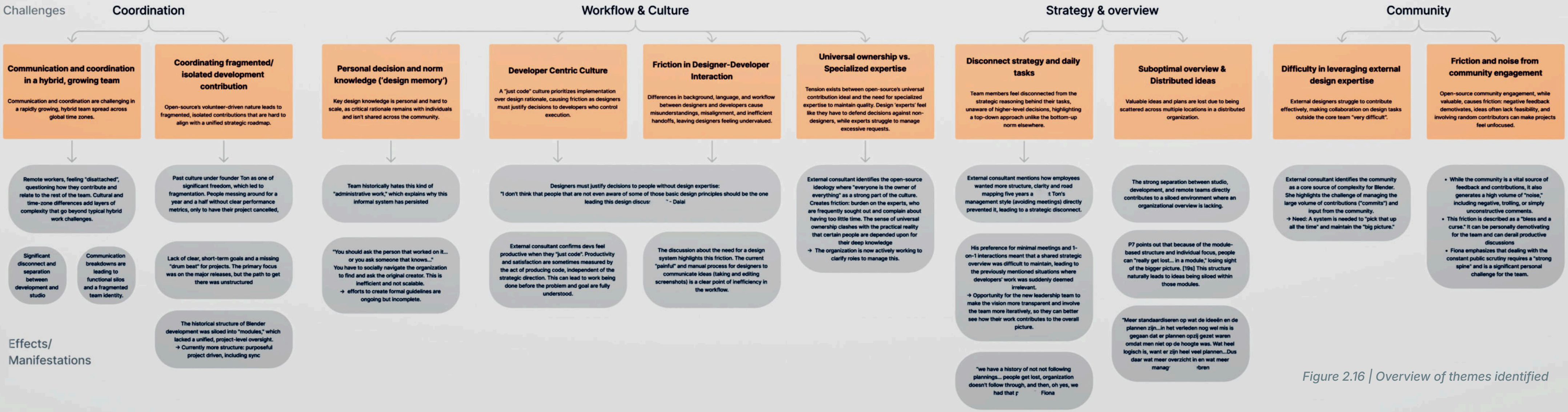


Figure 2.16 | Overview of themes identified

2.5 Research Quality and Limitations

To ensure the integrity of the research, several measures were taken to protect its quality and trustworthiness. Simultaneously, it is important to acknowledge the inherent limitations of the chosen methodology to properly contextualize the findings.

Research Quality and Trustworthiness

The trustworthiness of this qualitative research was established through two primary strategies:

- **Triangulation:** As detailed throughout this chapter, the study was designed around the principle of triangulation. By integrating data from four distinct sources, the research does not rely on a single perspective. This approach allowed for the cross-verification of findings, which mitigated the risk of individual bias and contributed to a more holistic, robust, and credible understanding of the phenomenon.
- **Prolonged Engagement:** I was actively embedded in the research context for the entire six-month duration of the project. This prolonged engagement facilitated the building of trust with participants and provided a deep understanding of the organizational culture that short-term observation could not achieve. This ensured that the data collected was grounded in the reality of the Blender development ecosystem.

Limitations of the Study

Despite the rigorous intentions of the research design, several limitations must be acknowledged:

- **Single-Case Study Design:** The primary limitation is the focus on a single case. While Blender was chosen for its representativeness of a large-scale open-source project, it possesses its own unique history, governance model, and culture. Therefore, while the findings offer valuable and transferable insights for similar contexts, they are not statistically generalizable and should not be treated as universal laws applicable to all OSS projects, but rather as evidence-based, environment-specific findings that can potentially inform a wider context.
- **Participant & Researcher Subjectivity:** In qualitative inquiry, I was the primary instrument of data collection and analysis. As a participatory observer actively contributing to design tasks, my own experiences and interpretations inevitably shaped the inquiry. While triangulation and a systematic analysis process were employed to ensure objectivity, the potential for subjective influence cannot be entirely eliminated. This limitation is also inherent to the structure of a Master's thesis, which is fundamentally an individual project. Although every effort was made to discuss emerging findings and involve participants and supervisors in a dialogical process, the final synthesis and interpretation ultimately represent the work of a single person.
- **Sample Scope:** The sample of 10 interviewees provided rich, in-depth data but is, by design, not a large-scale representative sample. The insights reflect the deep experiences of these key stakeholders in the research environment, but do not capture the full spectrum of opinions across the entire global Blender community.

03

Research Results

3.1 Results Introduction

Following the methodological approach established in the previous chapter, this chapter presents the primary findings. It frames Blender as a unique open-source ecosystem, mapping the complex web of contributors and stakeholders. The chapter then details ten key challenges, categorized into broad Organizational issues and specific Development Process frictions. Finally, these findings are distilled into a core Problem Definition, rooting the issue in a historical lack of design expertise, and reframed as actionable Opportunities for Intervention to guide the concept development that follows in the Chapter 4.

3.2 Mapping the Blender Ecosystem

Setting the Context

Before detailing the specific challenges, it is essential to first map the landscape in which they occur. Figure 3.1 below provides a visual overview of the key stakeholders and the relationships that connect them. The map distinguishes between the core organizational bodies and the wider community of external contributors and users. It visualizes the critical flows of funding, development work, feedback, and influence that define the system.

This ecosystem serves as the foundational context for this chapter. It is at the interfaces between these actors, between the Blender Institute, community contributors and between developers and users, that the frictions and challenges arise.

Designers and Developers

Defining clear roles within the Blender team can be somewhat misrepresentative as team members' roles are fluid. However, Blender only counts **2 dedicated designers** alongside **26 developers**. The developers also perform design activities for their modules (the areas Blender's development is divided into), creating some role and responsibility ambiguity. This 1:13 ratio falls short of the 1:10 industry benchmark (Nielsen Norman Group, 2025). This ratio illustrates challenges detailed in sections 3.3 and 3.4.

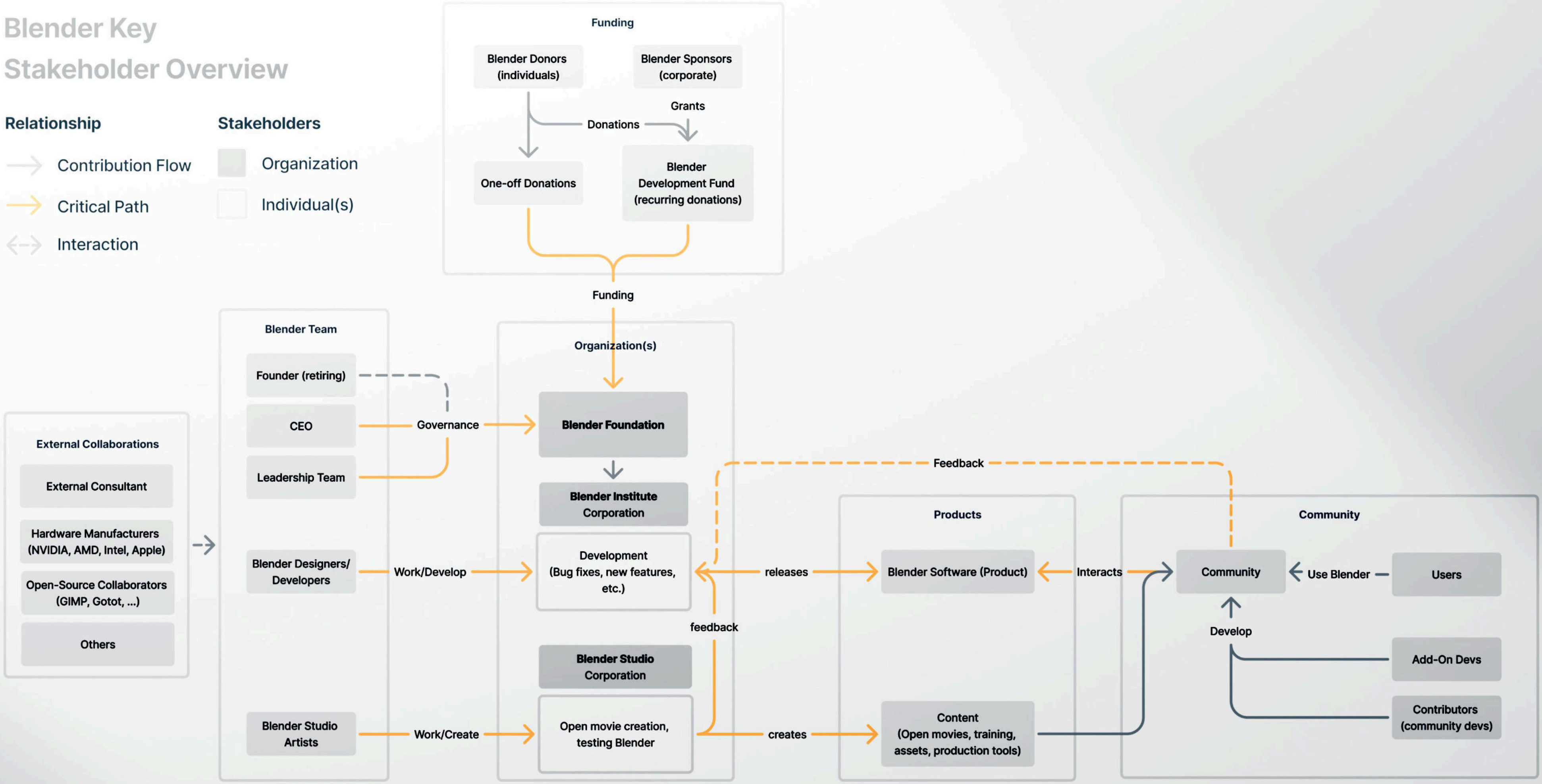


Figure 3.1 | Stakeholder interaction map of Blender Ecosystem

Stakeholder Hierarchy

To understand Blender's specific contributor structure, it is helpful to first look at the general evolution of open-source projects. As visualized in the Open Source Ecosystem Evolution model below (Figure 3.2), projects typically begin with a small group of core developers who are also the primary users. As the project grows in scale and over time, it attracts new layers of contributors who move from the periphery (e.g., passive users) toward the core (e.g., active contributors and developers), creating a multi-layered ecosystem.

Contributor Classification

Applying this theoretical model to Blender's current state allows for the identification of four distinct contributor levels, **classified in hierarchy of influence to the product**:

- **C1: Blender HQ Office Contributors:** This is the project's core, comprising the full-time designers, developers, executives, and operational staff based at the Blender Institute HQ. This group holds the most direct and significant influence over the project's strategic direction and daily development.
- **C2: Remote Dev-Grant Developers:** This group consists of developers who are not based at the HQ but work directly for Blender, often funded by development grants (salaries). They are highly integrated into the core development process and also have a high degree of influence.
- **C3: External Contributors (The Community):** This is the community composed of community members worldwide who voluntarily contribute to Blender. Their contributions primarily take the form of development tasks, code patches, and technical discussions.
- **C4: Users:** This is the largest and most diverse group, including all individuals who use Blender but do not actively contribute. Their influence is indirect, channeled through feedback, bug reports, and community forum participation.

Key Takeaway

Blender's contributor structure is a hierarchy of influence defined by four distinct levels (C1-C4). While C1 and C2 hold the primary power to shape the product, they represent a small fraction of the ecosystem. The vast potential for design contributions lies in C3 and C4, but without a way to channel their design input, this potential remains largely untapped.

Open Source Ecosystem Evolution Map

The different rings represent the different hierarchical levels of influence to the product, with the central ring having the most influence. The various stakeholders are positioned in the rings along with their contributions to Blender.

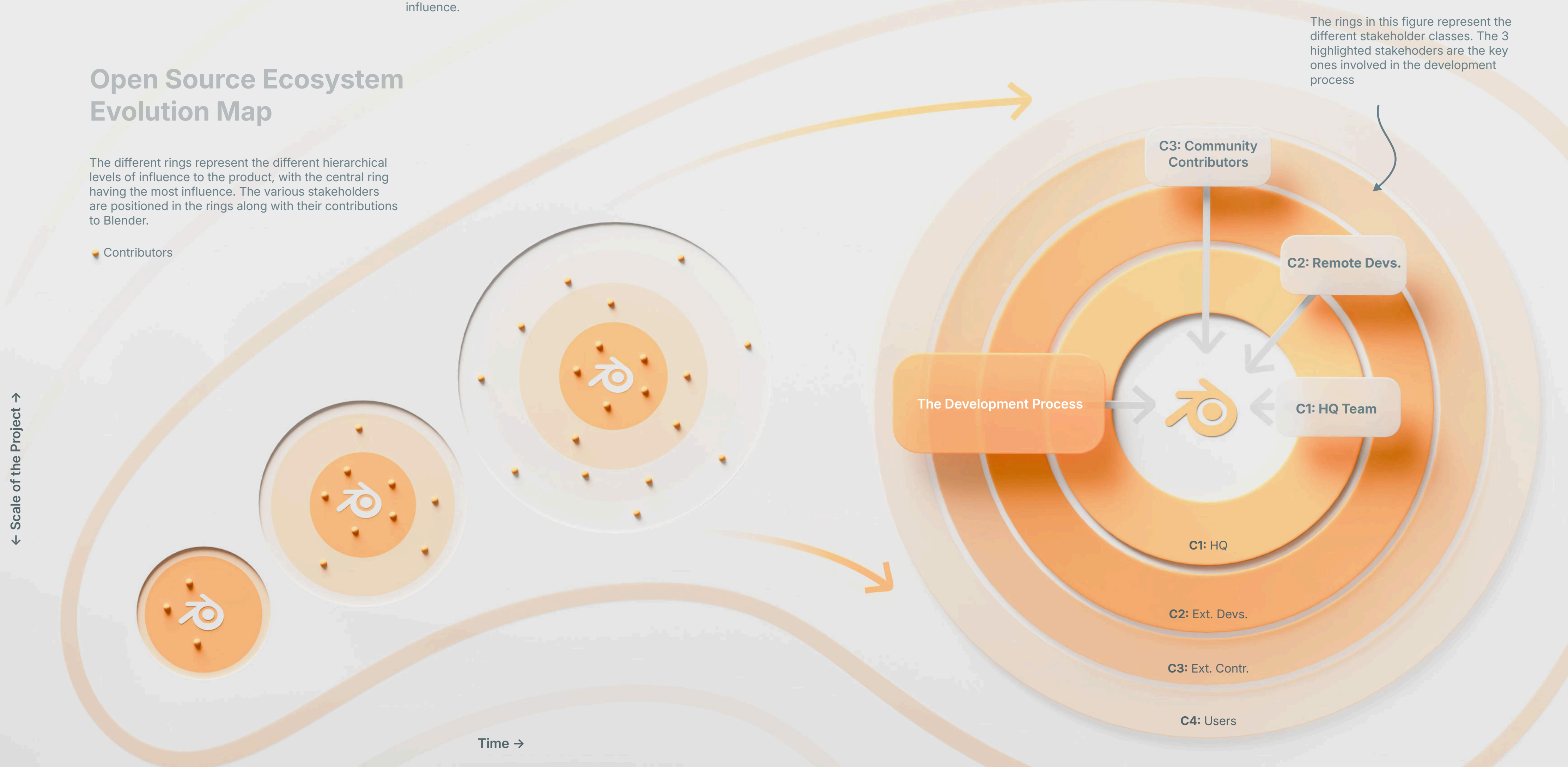


Figure 3.2 | Illustation of OSS Ecosystem Evolution including Blender

Stakeholder Asymmetry Tensions

While the classification of contributors identifies who is in the ecosystem, it is their dynamic interplay and asymmetrical influence that reveals the core tensions. A critical finding is the asymmetrical influence, particularly concerning the distinct processes of design and development. This dynamic is visualized in the Stakeholder Influence Overview on the right (Figure 3.3), which maps contributor groups based on their relative influence on design versus their influence on implementation (development). The map reveals two tensions:

- **Asymmetrical Influence:** HQ Developers (C1) have influence over design decisions, often more than HQ Designers (C1) have over the final technical implementation. This dynamic is a primary source of friction and directly points to the developer-centric culture that will be detailed later. As one designer mentioned:

"I don't think that people that are not even aware of some basic design principles should be the one leading these design discussions..."

Interview data confirms that the many technical contributors can drive design decisions despite 'lacking the expertise'. This was vividly illustrated by a participant: when a developer argued for a visually 'noisy' implementation based on technical logic, forcing the lead designer to intervene with an 'executive decision.' This creates a heavy burden of proof, where designers must 'convince everyone' of basic design principles against technical arguments.

- **An Unclear Path for Design:** While External Contributors (C3) have a well-established infrastructure and clear pathways to contribute to development, a similar accessible and effective pathway for contributing to design is largely absent. As one stakeholder noted, it is

"[It's] very hard for them to find out how to contribute to design."

Key Takeaway

Somewhat inherent to the inequal Designer-Developer ratio mentioned earlier, the Blender ecosystem is defined by an influence asymmetry.

Internally, developers often have more influence than designers do. Externally, it's easy for community developers to help, while it's hard for community designers.

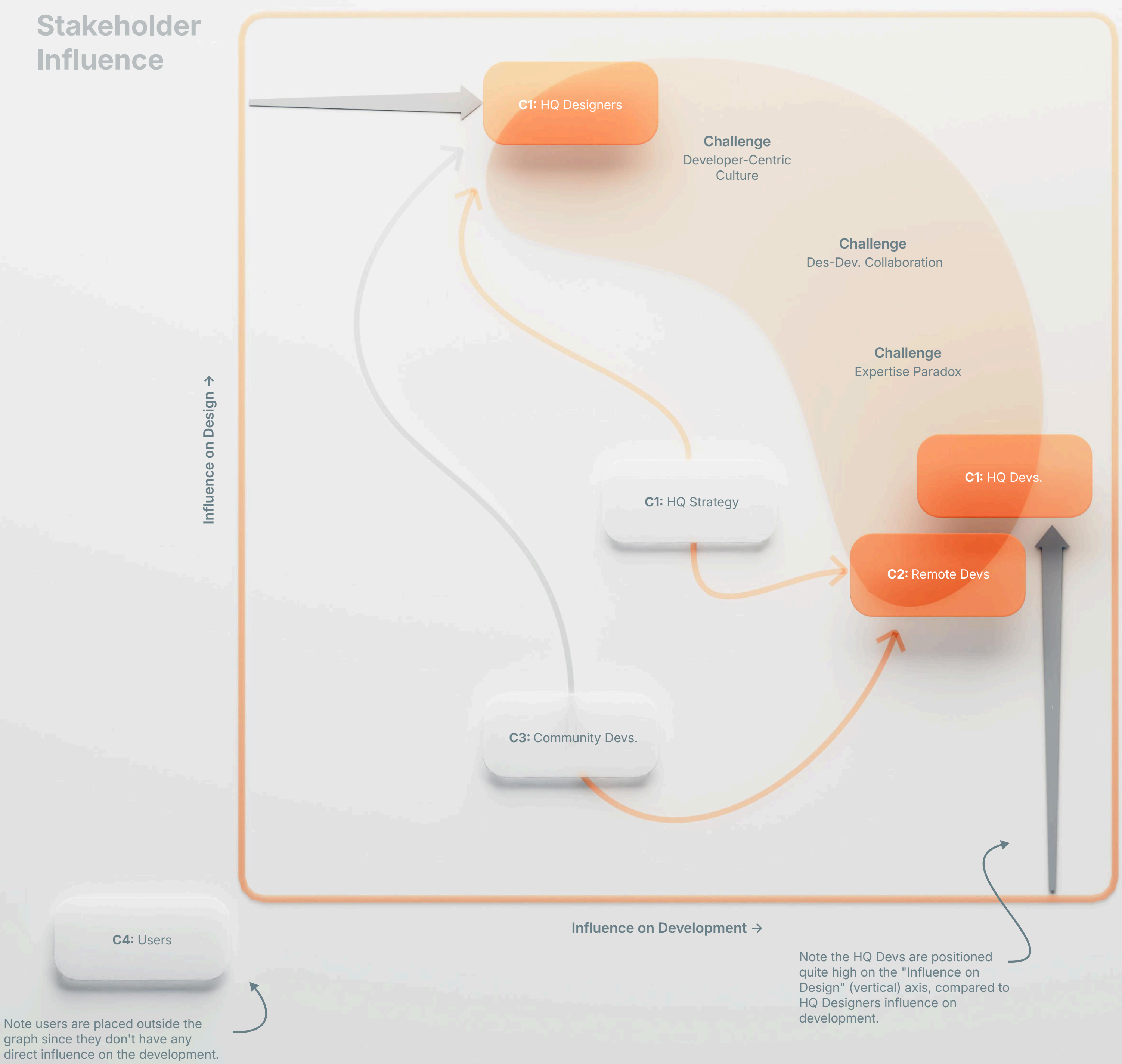


Figure 3.3 | Overview of stakeholder dynamics and influence

3.3 Key Challenges in The Blender Development Process

Introducing The 10 Challenges

The research revealed ten distinct yet interconnected and recurring challenges that manifest throughout the organization and its development process. They impact the efficiency, innovation potential, and collaborative health of Blender's development process.

Challenge 01
Communication

Challenge 02
Coordinating Fragmented Contributors

Challenge 03
Inaccessible Design Knowledge

Challenge 04
Developer-Centric Culture

Challenge 05
Too Little Design Expertise

Challenge 06
Friction in Des-Dev Interaction

Challenge 07
Expertise Paradox

Challenge 08
Scattered Ideas & Strategy

Challenge 09
Difficulty in Leveraging External Design Expertise

Challenge 10
Friction and Noise from Community Engagement

Challenges Positioned in The Process

Having identified the key actors in the ecosystem, the analysis now turns to the design and development process through which they collaborate. To provide a foundation for the challenges discussed in this chapter, it is essential to first understand the current 'as-is' feature development workflow within Blender. The process map below positions the 10 identified challenges within the process (Figure 3.4), which will be individually detailed in Section 3.4.

The goal of this map is to provide the sequential context needed to understand precisely where in the workflow the challenges occur. These challenges which will be summarized in another map on the next spread that illustrates from which stakeholders challenges originate and which other stakeholders they affect.

Critical Path

The orange critical path presented in the map below illustrates the value creation throughout the feature development process where every step contributes to creating the final result.

Key Note

The proposals are intentionally located in between the office and the community, indicating Blender's 'Development-in-the-open', where all their work is publicly available online.

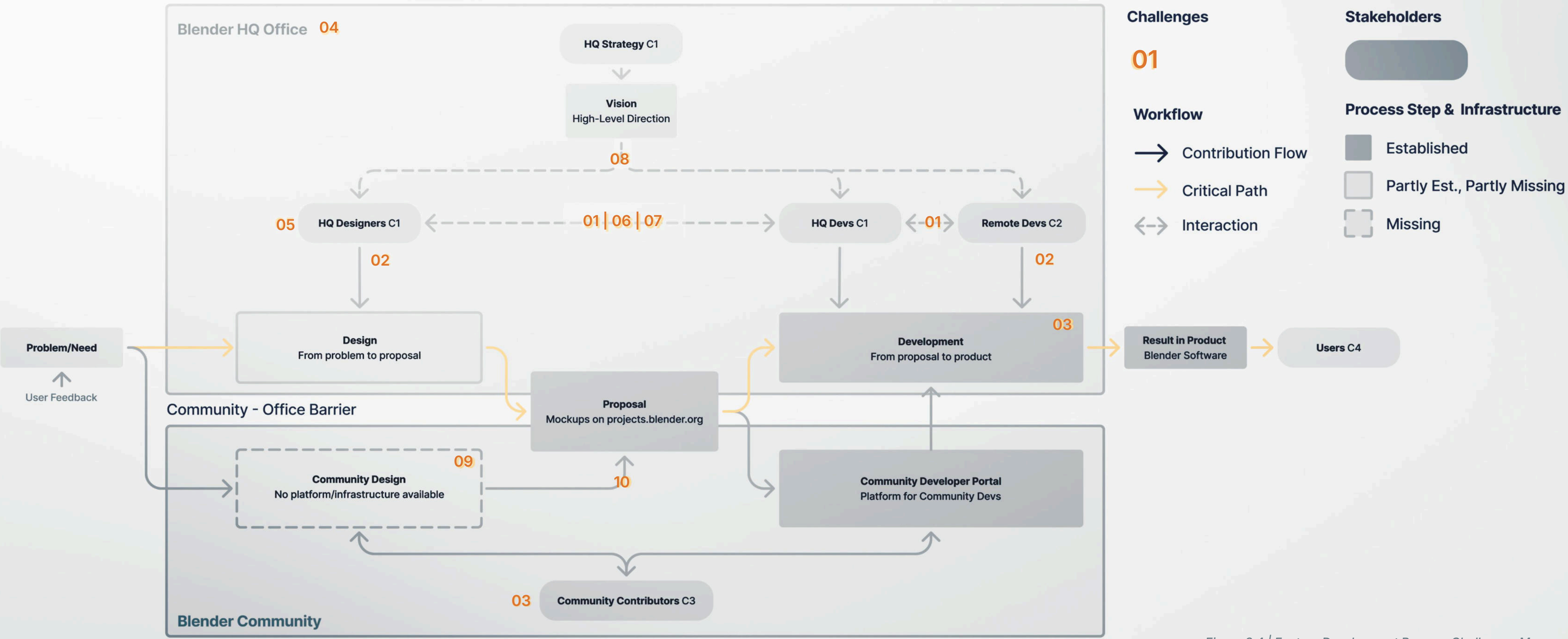


Figure 3.4 | Feature Development Process Challenges Map

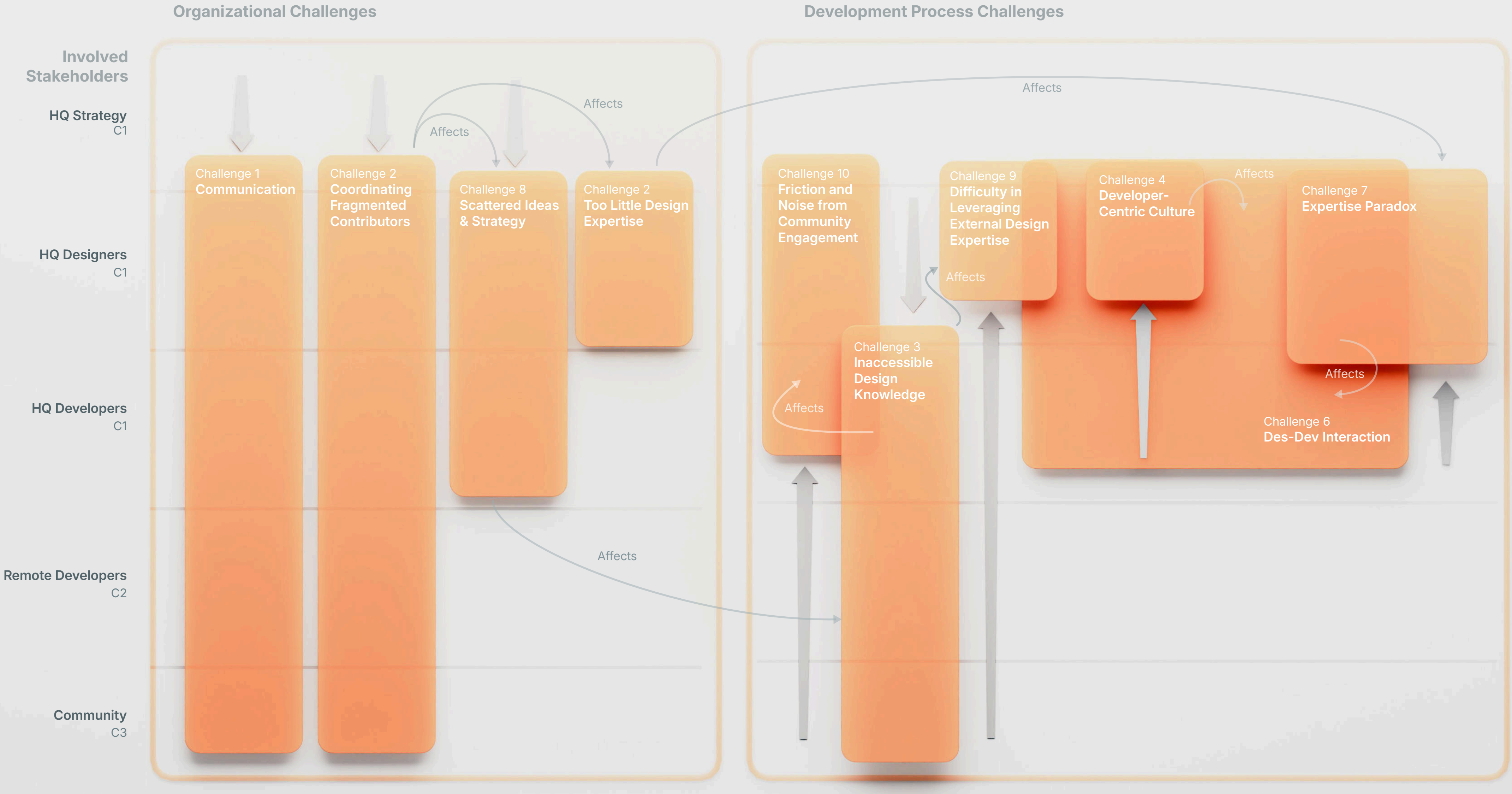
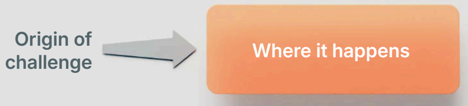


Figure 3.5 | Challenges in Process vs. Stakeholders Matrix



Ecosystem Stakeholder Challenges Map

The research reveals an ecosystem with inherent structural tensions. The goal of this map is to illustrate that these are not isolated problems but an interconnected system of issues, as visualized above (Figure 3.5). This map illustrates that some challenges are broad and Organizational, affecting the entire system and all stakeholders, while others are specific to the Development Process itself.

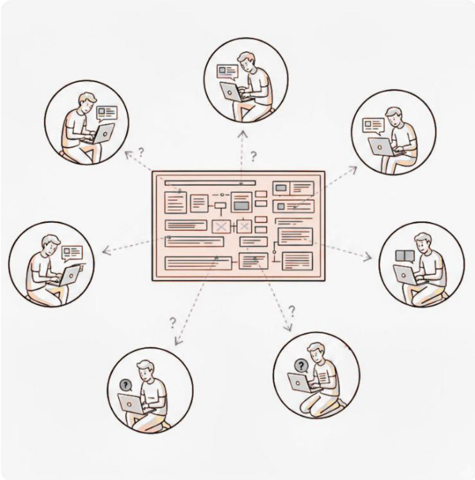
This map visualizes the position of the identified challenges across the development process and highlights which stakeholders are involved in a specific challenge, and where the challenge originated and manifests.

The following section (3.4) will detail each of these ten challenges, while providing the direct evidence from interviews and observations.
→

3.4 Challenges in Detail

01 Coordination Challenges

The most foundational category of challenges identified relates to coordination. In a rapidly growing, globally distributed team that combines full-time staff with a vast network of volunteers, maintaining alignment is a persistent struggle.



Challenge 1
Communication

The first and most pervasive challenge is the inherent difficulty of communication in a hybrid team spread across global time zones. The rapid growth of the team, combined with its hybrid and globally distributed nature, has introduced communication frictions.

Participants described a "disconnect" between (remote and office) team members, leading to functional silos and a fragmented team identity. This is increased by the complex mix of internal staff, remote developers, and external volunteers. A team member noted the feeling of being "disattached" that remote workers experience, questioning how their individual work relates to the team's broader goals.

One participant explained, this particularly affects remote workers:

"Remote workers, feeling 'disattached', questioning how they contribute and relate to the team"

This communication gap slows decision-making and makes it difficult to maintain a shared context across the organization.

Challenge 2
Coordinating Fragmented Contributors

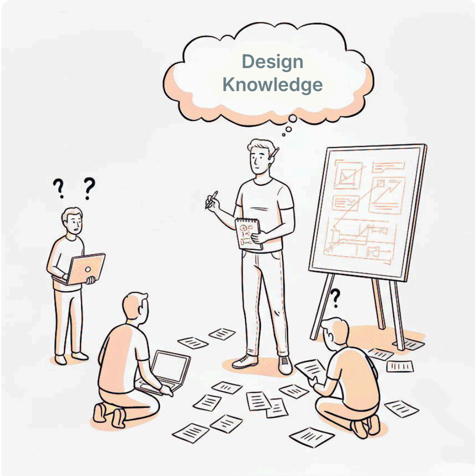
The open-source, volunteer-driven nature of Blender, while a strength, also leads to fragmented and isolated contributions. Historically, the structure of Blender's development was siloed into "modules," which inherently challenge a unified, project-level oversight. While this is currently improving with a more project-driven approach, the legacy of this structure persists.

As one participant explained, the primary focus is often on the major releases, but the path to get there can feel unstructured, without a clear "drum beat" for projects. This lack of a consistent rhythm makes it difficult to align the numerous, smaller volunteer contributions with the larger strategic roadmap, sometimes leading to duplicated efforts or work that is ultimately discarded.
A participant described this past inefficiency:

"People messing around for a year and a half without clear performance metrics, only to have their project cancelled"

02 Workflow & Culture Challenges

This category represents the most deeply rooted challenges, stemming from the cultural history of the project and affecting the daily interactions between designers and developers.



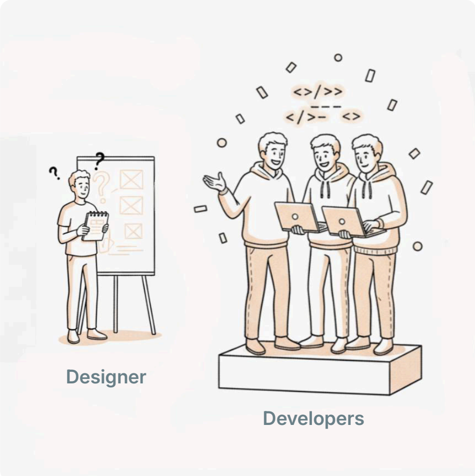


Figure 3.6 | Sketches describing challenges

Challenge 3
Inaccessible Design Knowledge

Key design knowledge often remains personal and tacit, held by the individuals who made the decisions or created the knowledge. This "design memory" is not systematically documented or shared, making it difficult to scale and inaccessible to new or external contributors.

This inefficiency was vividly described by one participant:

"You should ask the person that worked on it... or you ask someone that knows..."

Challenge 4
Developer-Centric Culture

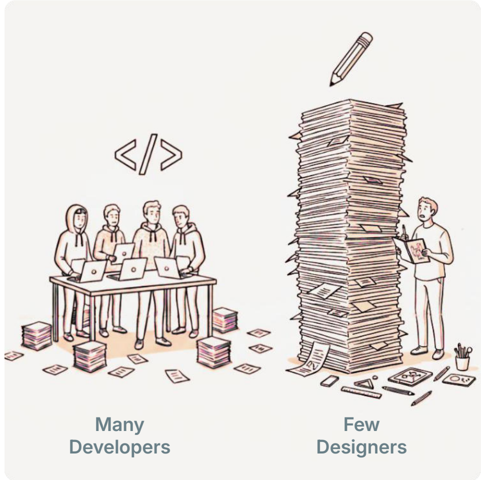
The data consistently pointed to a dominant "developer-centric" culture where the act of coding and implementation is often experienced as more productive or rewarding than the preceding design and problem-framing work. This creates a dynamic where design is sometimes seen as a secondary, rather than a foundational, activity. Multiple participants described how they perceive this culture:

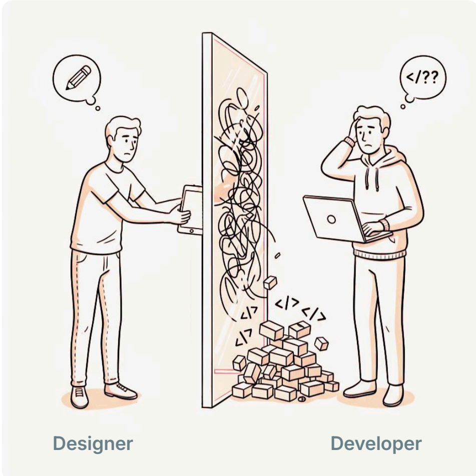
"Blender has a history of a strong engineering culture."

"For many software developers, it's very tempting to just start coding. Not thinking too much about design. Just happily writing code, because it feels like you're being productive. Code comes out, and it makes things happen. That's fantastic"

"Design, yeah nice... we talk about that, but actually we just have to code, right? We have to code there. That is the work."

"Those [technical] aspects they sometimes can even tend to take over ... the whole process and it becomes technology first more than anything else"





Challenge 5
Too Little Design Expertise

As noted by participants, the developer-centric culture and the demographic tendency of contributors to be technically-oriented has resulted in an environment where there is simply too little design expertise:

"It's really capacity. Just simply there are way more challenges out there than can be tackled by the people involved right now at all levels. It's almost a chronic lack of people with the skill and the focus to tackle the issues at this level."

Challenge 6
Friction in Designer-Developer Interaction

A direct consequence of the developer-centric culture is significant friction in the daily workflow between designers and developers. This friction stems from differences in background, language, and tools, leading to misunderstandings and inefficient handoffs.

This was directly observed in the participatory research, where the primary method for designers to communicate UI changes was a manual process of taking, editing, and annotating screenshots, highlighting a lack of shared tools and integrated workflows.

The ongoing discussions among Blender's designers about the need for a design system highlights this friction.

This challenge is well documented in the literature (Zhang et al., 2025). The friction between designers and developers is often rooted in a fundamental disconnect in their professional languages and artifacts, a finding consistent throughout the interview data. This communication gap leads to misunderstandings and inefficient handoffs.

This challenge is not unique to Blender and reflects a well-understood pain point across the software industry. When asking about the single most crucial factor for improving the designer-developer workflow at the 2025 Penpot Fest Conference, industry experts emphasized the need for a 'shared vocabulary' (G. Langreo, personal communication, October 2025). This underscores the critical need for a common language to bridge the disciplinary gap.

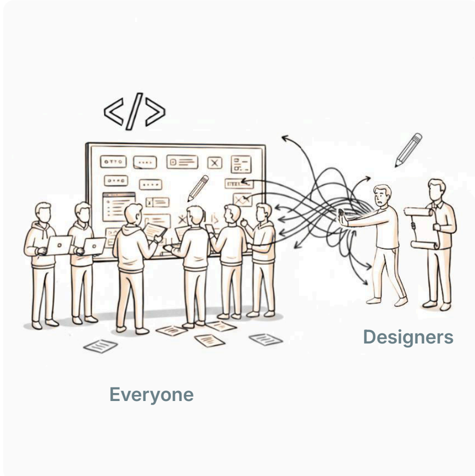


Figure 3.7 | Sketches describing challenges

Challenge 7
The Expertise Paradox: Universal Ownership vs. Specialized Expertise

A core tension exists between the open-source ideal of "universal ownership" where everyone can contribute to anything, and the practical need for specialized expertise to maintain quality. Because of Blender's module-based structure, design tasks are executed across every contributor, regardless of whether they are a designer. Since everyone is effectively 'doing design,' the lack of shared standards means non-designers are forced to invent design solutions from scratch, often leading to inconsistency.

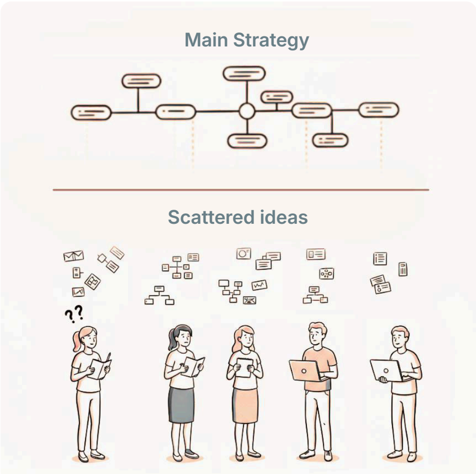
This paradox creates two problems: designers feel they must constantly defend their decisions to non-designers, while team members with specialized expertise themselves also become bottlenecks, struggling to manage excessive requests.

As noted by a designer: "I don't think that people that are not even aware of some of those basic design principles should be the one leading these design discussions..."

As a participant noted, the culture of "everyone can contribute to everything" clashes with the reality that "certain people are depended upon for their deep knowledge."

03 Strategy & Overview Challenges

This challenge relates to a lack of shared strategic context and organizational overview, leading to lost work and misaligned efforts.



Challenge 8
Scattered Ideas & Strategy

In a distributed organization with a history of siloed modules, valuable vision, ideas, plans, and design rationale are frequently lost. Information is scattered across multiple online platforms, personal documents, and individual memory. Individuals noted to feel disconnected from a more holistic strategy.

A participant summarized the problem and the need for a solution:

"What has gone wrong in the past is that plans were put aside because people were not aware of them... So [I would suggest] more overview and more management in that."

"It's difficult when everyone puts their plans down somewhere else."

04 Community Challenges

These challenges arise at the interface between the core Blender team and its vast external community of contributors and users.

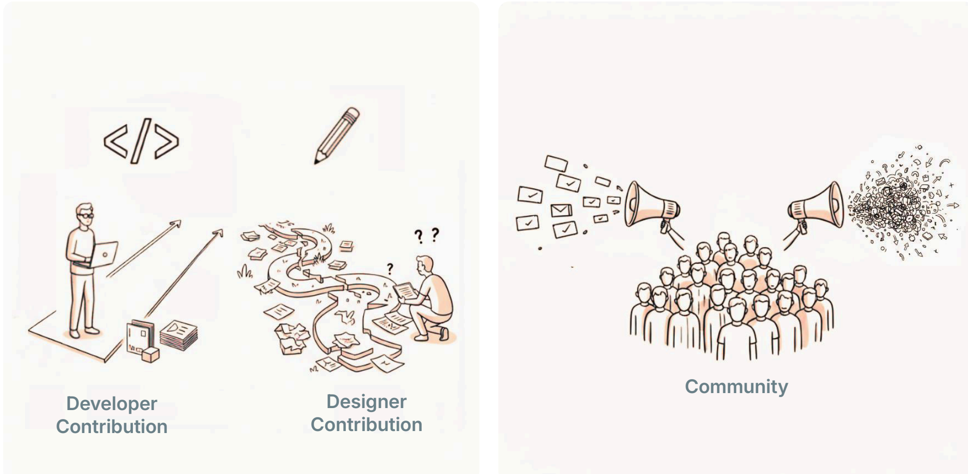


Figure 3.8 | Sketches describing challenges

Challenge 9
Difficulty in Leveraging External Design Expertise

While Blender successfully leverages external contributions for development, the same is not true for design. The lack of clear guidelines, accessible documentation, and dedicated tools makes it, in the words of one participant, "very difficult" for external designers to contribute effectively. The existing infrastructure provides clear pathways for external developers to contribute, but a similar accessible and effective pathway for external designers is absent.

"Why doesn't Blender have a design platform? Why is design not visible, and there is only development? Well, because 'development is what we do.' "

"But within the developer website, is there any design section? There is some stuff scattered around..."

Challenge 10
Friction and Noise from Community Engagement

While community engagement is vital, it also generates significant friction and "noise." Participants described the personal toll of public scrutiny and negative feedback, which can be demotivating. Furthermore, the high volume of unstructured community input, often lacking technical feasibility or strategic alignment, can derail productive discussions and make projects feel unfocused.

Multiple participants described the interaction and dynamic with the community as both a "blessing and a curse."

"Community ideas often lack the systems-thinking required".

These individual challenges illustrate the full breadth of the issues at play. To create an effective and focused intervention, however, it is necessary to narrow our focus to the challenges most relevant to this thesis's specific scope: the designer-developer interaction in the feature development process.

→

3.5 Three Key Findings

From the detailed challenges described above, three overarching findings emerge that form the core argument of this research.

When analyzing the challenges, they form an interconnected pattern that reveals a deeper issue. From this evidence, three overarching findings emerge that form the core argument of this research and explain why the Blender development process experiences these specific frictions.

These findings are supported by the data gathered from stakeholder interviews, participatory observation and field research, but most importantly, are also widely recognized in recent literature (Zhang et al., 2025).

These three findings are causally linked: the foundational tech-focused culture (Finding 3) creates and perpetuates the operational Des-Dev Gap (Finding 2), which in turn results in design being a critical but underleveraged asset (Finding 1). Together, these findings point directly to a single, underlying problem that the recommendations of this thesis will address.

→

Figure 3.9 | Overview of three key findings

3.6 The Problem Definition

The three key findings from Section 3.5 allow the core problem can be broken down into a clear cause-and-effect chain:

- **Origin (The Culture):** Blender's open-source history has cultivated a successful and dominating tech-focused culture. This has led to a "chronic scarcity of design expertise contributions" as formulated by a participant.
- **Direct Effect (The Gap):** This culture has created and sustained a recognized lack of infrastructure and cultural norms for design, where workflows, tools, and priorities are not naturally aligned.
- **Manifestation (The Consequence):** The persistence of this gap means that design remains a critical yet underleveraged component of the feature development process, with its full potential for improving UX, quality, and innovation unrealized.

The Risk: So What?

At Blender's current scale and complexity, the design-development imbalance is a risk. **It limits the organization's ability to address increasingly complex challenges that require solid design capacity alongside its already strong technical development.** Addressing this core problem is therefore about "leveling the playing field" as noted by a participant.

Why is the dominance of development over design a critical problem? Because developers and designers solve different problems. Development focuses on functionality: "how do we make it work?". Design focuses on usability & alignment: "Does this fit the user? Does it make sense within the whole system?". We need both disciplines to create a both powerful and usable product.

If development takes over completely, Blender risks becoming a collection of technical functionalities that lack ease of use or coherence. This leads to steep learning curves, inconsistent interfaces (e.g., five different ways to change one setting), and 'bloated' software.

The Goal

The goal is not for design to dominate development, but to provide the integrative perspective. While development can work in specific modules (silos), design works horizontally across modules to ensure consistency. The ideal is a balance where both disciplines are most effective together at solving the problems.

> **Key Takeaway**
>
> *Blender experiences underleveraged design expertise due to a historically tech-focused culture. Without the integrative perspective of design, the software risks becoming a collection of isolated features rather than a cohesive product.*

Problem Definition

Cause — Blender's open-source development has cultivated a technology-first development culture,

Effect — leading to a chronic scarcity of design expertise contributions.

Problem — This has left it underrepresented, and manifested into an environment with little current infrastructure, tools and capacity for integrating design contribution/thinking. This makes both internal and external contributions inefficient and inaccessible.

Importance (Risk) — At Blender's current scale, this imbalance limits the ability to address increasingly complex challenges. It risks creating powerful but fragmented software, where technical features fail to translate into a cohesive user experience.

01 — Blender's tech-focused culture

02 — Led to a "scarcity of design expertise contributions"

03 — Which resulted in an environment with **little tools and infrastructure for integrating design**

04 — Which resulted in an inefficiency in leveraging design contributions

05 — This imbalance limits the ability to address increasingly complex challenges. It risks creating powerful but fragmented software, where technical features fail to translate into a cohesive user experience.

Figure 3.10 | Overview of Problem Definition and Cause and Effect chain

3.7 Design Capacity: A New Framework for Design in OSS Development

Why a new Framework?

To fully understand how to leverage design expertise in the identified problem space, the conceptual lens of Design Capacity was developed.

The academic field of design management has long sought to define how organizations effectively leverage design to create value with two key concepts: Design Maturity and Design Capability. Design Maturity, often described using models like the Danish Design Ladder, classifies an organization's use of design on a spectrum from a superficial afterthought to a core strategic driver (Doherty et al., 2014; Kretzschmar, 2003). Design Capability, on the other hand, is framed as a strategic, firm-level resource: the ability to deploy design processes and skills to achieve competitive advantage and drive innovation (Marzia Mortari et al., 2014; Stephenson, 1998).

While these concepts are invaluable, they are tailored to traditional, hierarchical corporate environments where resources are significantly centrally managed. As Marzia Mortati et al. (2014) describes: "the term capability is often associated to firms rather than individuals". The decentralized, community-driven, individually autonomous, and non-commercial nature of a large-scale open-source project like Blender presents a unique context that these models do not fully capture.

A New Framework

The decentralized framework of Design Capacity is more fitting within the OSS context. It is distinct from both Design Maturity and Design Capability in its focus and application. While Design Maturity models, such as the Danish Design Ladder, assess an organization's strategic progression on a linear scale, and Design Capability refers to the firm-level possession of design skills and resources, Design Capacity measures the underlying systemic infrastructure that makes both effective.

Therefore, an ecosystem may possess a mature, design-centric vision (high maturity) and access to skilled designers (high capability), yet still fail to consistently deliver value if it lacks the workflows, tools, and knowledge systems to leverage them, as discovered with Blender. Design Capacity ultimately determines whether that design potential is effectively leveraged or not.

If Design Maturity represents the strategic 'why' and Design Capability the skilled 'who,' then Design Capacity constitutes the essential 'how': the holistic, enabling system of workflows, tools, culture, and knowledge systems. Therefore, to provide a more fitting framework, this thesis introduces and defines the concept of Design Capacity:

Distinct from both Design Maturity and Design Capability, Design Capacity is the measure of an ecosystem's systemic infrastructure effectiveness to integrate (capture) and leverage (accelerate) design capability potential into consistent organizational output.

It considers both the number of designers an ecosystem has (quantity) and a measure of the ability to convert that expertise to effective outcomes (quality). It manifests through four interdependent components:

- **Infrastructure & Tools:** The shared software, platforms, and artifacts that enable collaborative design work.
- **Workflows & Processes:** The established, repeatable methods for how design is conducted, communicated, and integrated with development.
- **Cultural Norms:** The shared values, priorities, and attitudes within the community regarding the role and importance of design.
- **Knowledge Accessibility:** The ease with which design rationale, history, and principles can be stored, found, and utilized by all contributors.

The Term

The term Design Capacity is deliberately chosen for its specific connotations in systems thinking and strategic management. The term frames the challenge as a structural and measurable property of the organization, similar to "production capacity" or "network capacity." It implies a potential that can be diagnosed, invested in, and systematically increased. By framing the issue as one of building capacity, this thesis focuses on creating the concrete, actionable design workflow infrastructure required to transform potential into consistent, scalable, and effective organizational output.

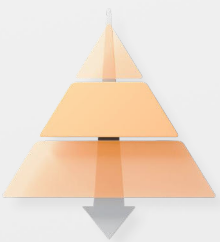
Framework Value

The value of this framework is twofold. First, it provides a precise diagnostic lens: the systemic issue within Blender can now be clearly identified as a deficit in organizational design capacity, particularly regarding its infrastructure, workflows, and knowledge accessibility. Second, it guides the outcome of this thesis. The objective is to propose a strategic intervention aimed at building Blender's long-term design capacity, thereby enhancing its overall potential for user-centered innovation and value in the end-product.

A practical intervention, without a guiding framework, is just a simple tool, but with the framework, it becomes an argued, enabling intervention.

Tactical Layer

This analysis makes it clear that the problem, a deficit in organizational Design Capacity, requires a capacity-building solution. Building a robust tactical layer is the most critical and highest-leverage action to take. The desired outcome is to translate the systemic insights from the research into concrete, actionable interventions within this specific workflow.



Systemic Insights (Abstract)
Research

Design Capacity Framework
Translation: problem to opportunities

Implementation: Tools (Concrete)
Design Workflow Intervention

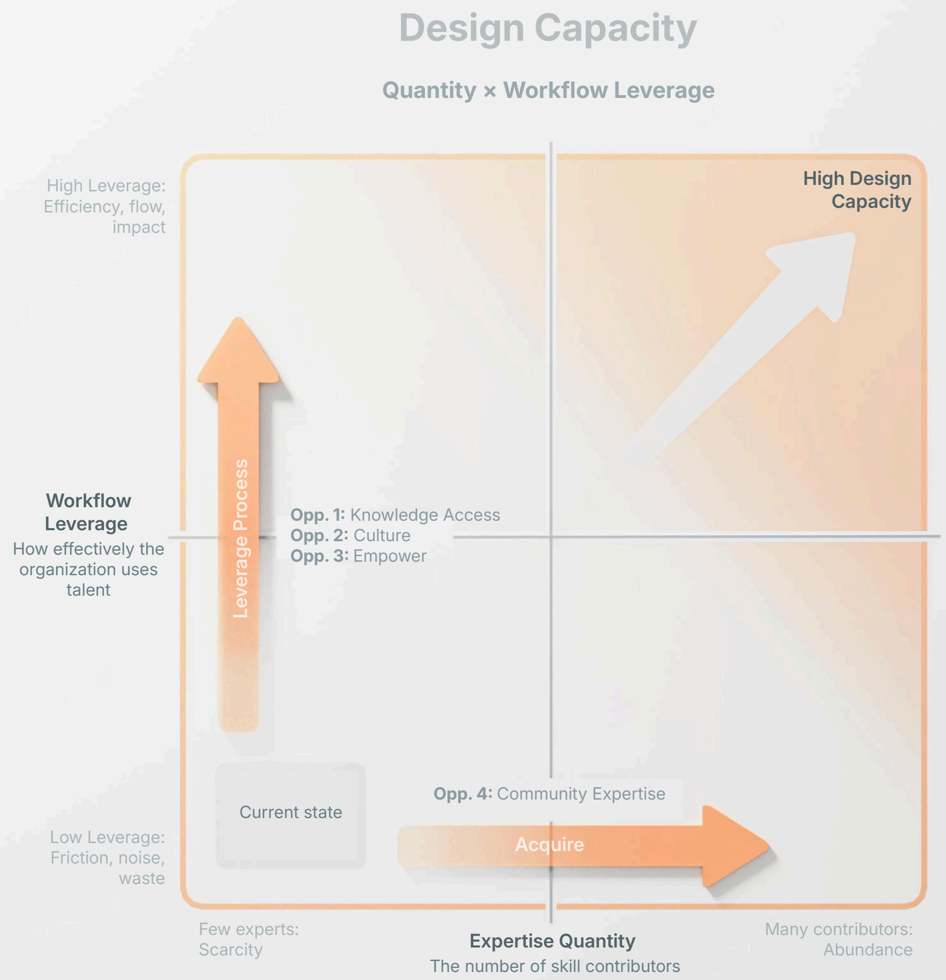


Figure 3.11 | Design Capacity Framework, displaying Process Quality and Expertise Quantity Levers

3.8 Applying Design Capacity: The Concept Focus

The problem definition reveals a systemic issue, rooted in culture and history, that manifests across 10 diverse challenges. However, an effective intervention requires strategic focus. The goal is not to solve every organizational issue, but to target the areas that offer the most direct leverage for improvement within the scope of leveraging design expertise.

Strategic Focus: Acquiring vs. Workflow Leveraging

To determine the direction of the intervention, we first apply the Design Capacity framework (from 3.7). Capacity can be increased by two levers: acquiring more expertise (Quantity) or better utilizing existing expertise (Leverage).

A logical management strategy to address the Scarcity of Design Expertise (Challenge 5) would be to simply recruit more designers. However, this approach reveals critical flaws:

- **Systemic Inefficiency:** Adding people to a system with suboptimal workflows yields diminishing returns. New designers would face the same inaccessible Design Memory (Challenge 3) and Friction in Handoffs (Challenge 6). Without addressing the tactical process (how) layer first, hiring is an inefficient use of resources. It potentially could even worsen these issues by creating a larger, more isolated design team, potentially widening the cultural gap.
- **Financial Sustainability:** Given Blender's non-profit and donation-funded nature, optimizing the process for the current team is a significantly more sustainable multiplier than considerably increasing salary costs.

Recruitment is an Operational solution to a Tactical problem. While recruiting more design talent is a necessary long-term goal, as a standalone solution it is insufficient. Therefore, this thesis focuses on the second lever: improving the workflow quality.

Once this workflow foundation is laid, future recruitment efforts (addressed in the Recommendations, Chapter 5) can yield even higher value.

Selecting High-Leverage Challenges

With the strategic focus on Workflow Leverage, we can identify which specific challenges offer the greatest potential for improvement.

To facilitate this strategic selection, the ten identified challenges were mapped to Figure 3.12 below using the C-Box method (van Boeijen et al., 2013). The First Priority challenges have the largest leverage, followed by the Secondary Priority challenges.

While broader organizational issues like hybrid-team communication are important, these Tertiary challenges are less unique to the OSS design and development context being investigated.

When reframing these challenges not as problems but as opportunities, they provide a clear and actionable direction for the concept creation phase. The next section translates the four most important challenges into concrete opportunities for concept development.

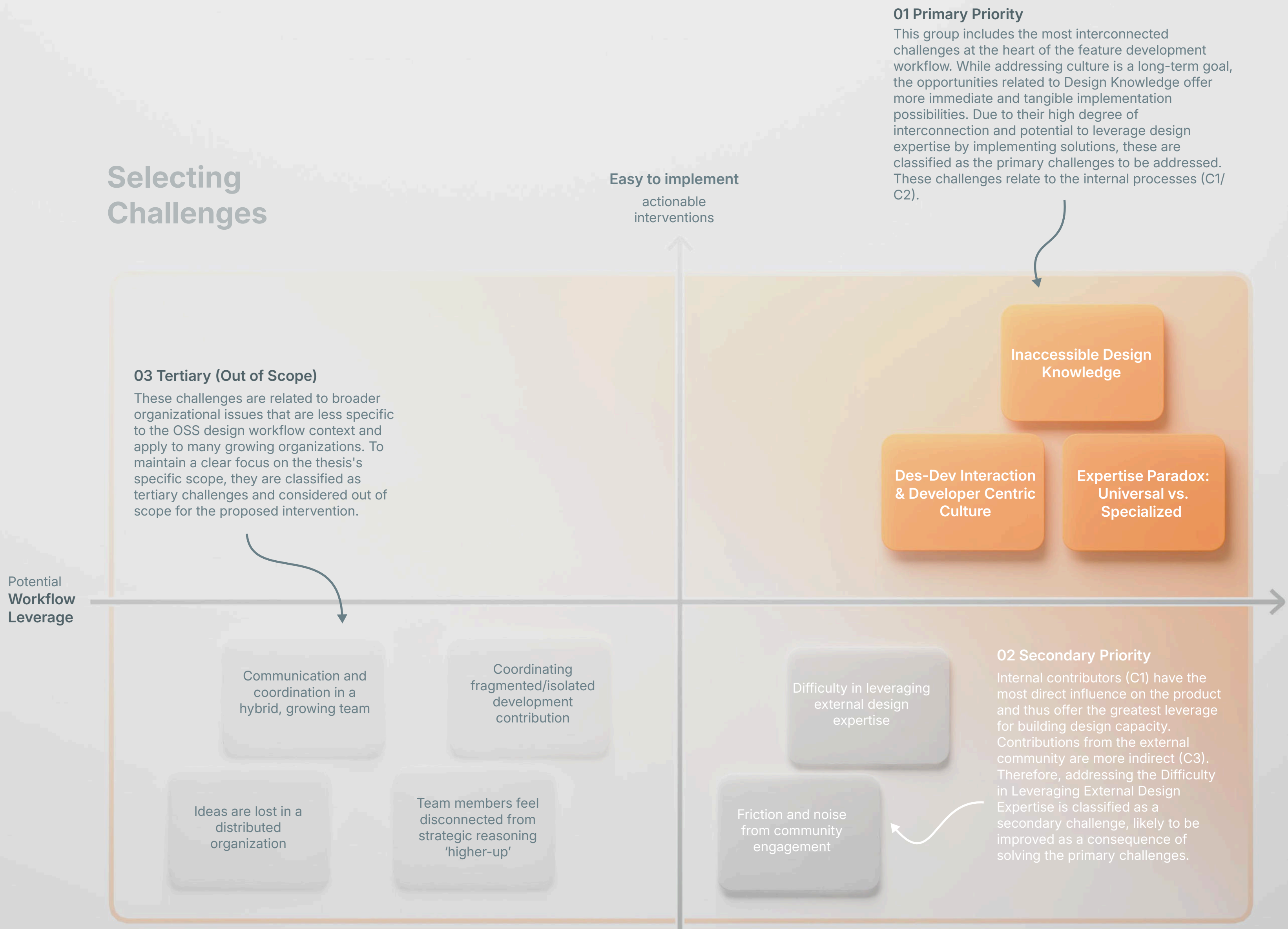


Figure 3.12 | Overview of selected challenges to focus on

3.9 The Opportunities for Design Intervention

The four selected challenges are highly interconnected and represent the most potent leverage points for building Blender's design capacity. This final section reframes these challenges as four clear, strategic opportunities for intervention. These opportunities are unified by a single, overarching goal derived from the research.

These four opportunities translate the research findings into a set of strategic intentions. They form the foundation of the design requirements for the next chapter, serving as the direct input for developing a tangible framework and set of practical design tools to strengthen the Blender development ecosystem.
→

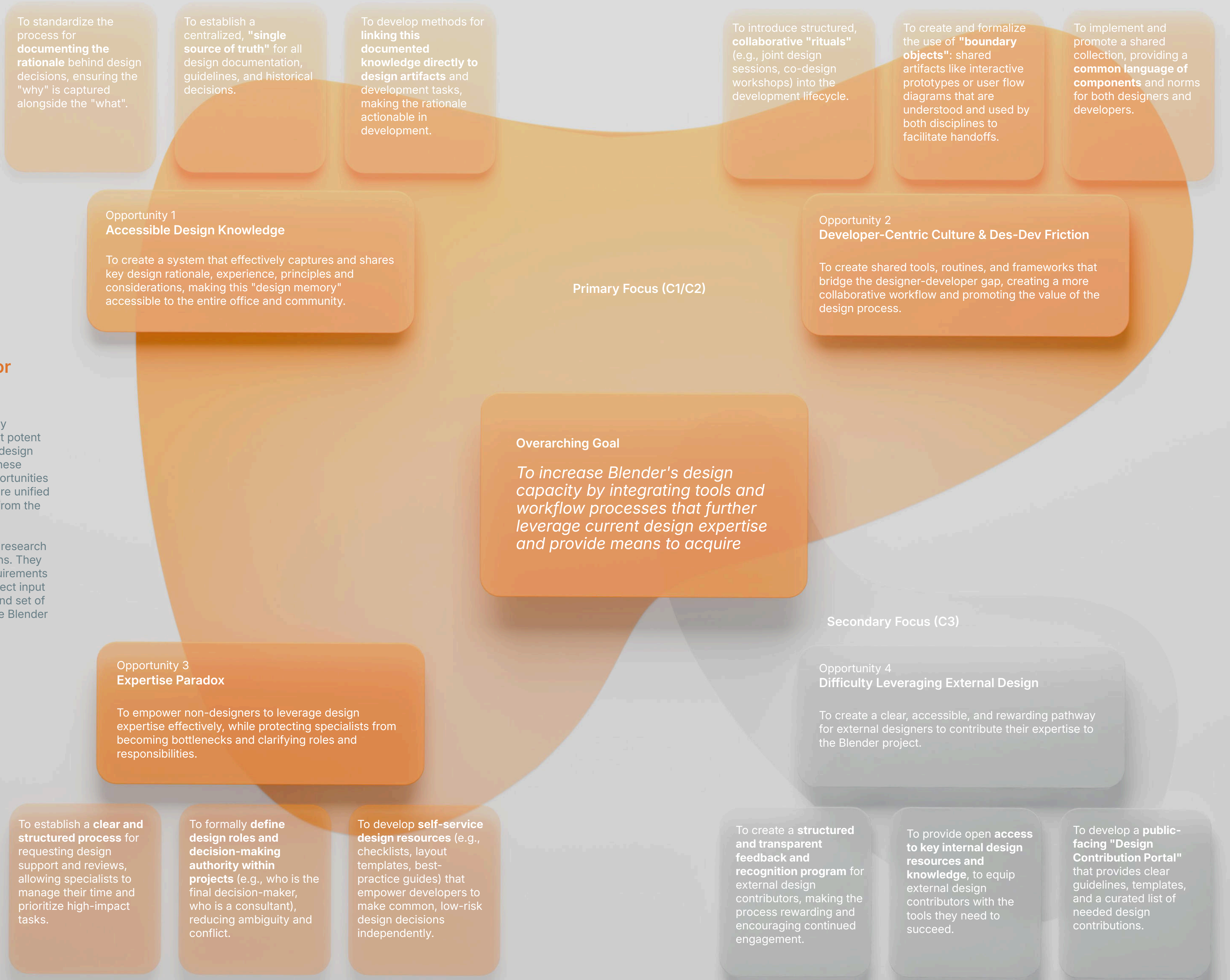


Figure 3.13 | Overview of opportunities

3.10 List of Requirements

Primary Design Criteria

To ensure the concept addresses the challenges identified, these requirements must be met for the solution to be considered successful:

General

- The solution should be digital (as opposed to physical), and must be compatible with an asynchronous work culture (online first)
- The solution must be entirely created using open-source software
- The solution must be compatible with any operating system (OS), including Windows, Linux and MacOS
- The solution is built completely in the English language

Design Capacity

- Req 0.1: The solution must improve the design workflow and act on the workflow quality lever to build Design Capacity

Opportunity 1: Accessible Design Knowledge

- **Req 1.1:** The solution must capture and establish a single library to act as a "single source of truth" for design artefacts (UI elements)
- **Req 1.2:** The solution must provide a place to document Blender's most important design principles (the "established" design style)
- **Req 1.3:** The solution must be integratable in Blender's current internal development process and should be optional (non-essential)
- **Req 1.4:** The solution should support rich media (images, videos, interactive prototypes) to facilitate high-fidelity proposal creation
- **Req 1.5:** The solution should be searchable by typing in keywords or search criteria, allowing users to easily find what they're looking for
- **Req 1.6:** The solution must be distributable, allowing for sharing over the online ecosystem. And therefore, only contain elements and components that can be distributed (no copyrighted elements etc.)

Opportunity 2: Collaborative Designer-Developer Workflow

- **Req 2.1:** The solution must include a shared, accessible library of components, terms and knowledge to serve as a common language
- **Req 2.2:** The solution should allow for real-time collaboration inside the digital solution
- **Req 2.3:** The solution must standardize the format and look of key "boundary objects" (e.g., most used UI elements for proposals) for handoffs to ensure clarity
- **Req 2.4:** The solution should use version control and allowing to go back to previous versions and create new ones
- **Req 2.5:** The solution should be usable by and valuable to both designers and developers

Opportunity 3: Leveraging and Protecting Expertise

- **Req 3.1:** The solution must provide self-service design resources (e.g. templates) to empower non-designers to handle common, low-risk design tasks without direct designer help
- **Req 3.2:** The solution must be intuitive and reasonably self-explanatory (as far as can be expected and realized) and usable by any contributor, internal or external

Secondary Design Criteria

These criteria are supporting and future-oriented, since this opportunity has second priority. They mostly apply to Concept 03, which will be defined in Chapter 4.

Opportunity 4: External Design Contribution

- **Req 4.1:** The solution should define a single, clear, and public-facing entry point for external designers
- **Req 4.2:** The proposed solution should inform, guide and educate an external contributor about Blender's design principles and philosophy

Wishes

These wishes apply to all concepts and are supporting all previous requirements.

- The solution is available for offline use (downloadable)
- The solution fully supports interactive UI prototyping (e.g. clicking button opens new page)

Out of Scope

- Physical product solutions (due to Blender's digital work culture, online transparency and distribution and online-first principle)

This structured list of requirements concludes the analytical phase of this research. It forms the direct and actionable foundation for the upcoming Concepting Chapter, where these criteria will be used to guide ideation and evaluate potential solutions.
→

04

Concept Development

4.1 Concept Development Introduction

Having defined the core challenges hindering Blender's design capacity, this chapter transitions from analysis to creation. It begins by exploring a range of ideas and potential intervention strategies before converging on a final, justified solution. The goal is to identify the most effective and culturally fitting path forward to address the four challenges using an actionable tool.

Guiding Philosophy

With the list of requirements serving as the foundation, the ideation process moved the project from the analytical to the generative.

The solution should empower the individual, be distributable, and translate the systemic insights into a practical intervention that leverages the expertise in the development process. Hence, the decision to develop a digital tool that leverages design expertise followed naturally from the nature and characteristics of Blender's Ecosystem.

The guiding philosophy was to design an actionable tool that felt like an invitation and enabler, not an imposition. Recognizing Blender's unique culture of autonomy and open contribution, any proposed solution had to be a lightweight and adaptable proposal that could enhance and support, rather than replace, existing workflows.

The core tension to be resolved was between providing necessary structure and preserving creative freedom. The solution needed to offer clear, structured pathways for collaboration while still allowing for the informal, dynamic interactions that define open source.

A central design driver was the need for an asynchronous-native approach. Given a globally distributed team, the framework had to prioritize tools and processes that did not depend on real-time meetings. This led to a focus on creating a "single source of truth" - a living repository for design knowledge that could serve as a shared context for all contributors to be available anytime, regardless of their location or time zone. The challenge, therefore, was to design a system that was both a stable archive for this "design memory" and a dynamic collaboration space for active projects.

In addition, the focus of the solution had to be less on technical details and more about enabling collaboration through collective creation. It's about creating something that sparks the conversation and creates and facilitates interactions. Collaboration is what ultimately improves workflows, and making developers use and appreciate design tools works better than making designers code, as mentioned by an executive from Penpot and a design director at GitHub in my conversations with them (P. Ruiz-Múzquiz & G. Langreo, personal communication, October 9, 2025). Therefore, the solution should be usable and adoptable by developers, empowering them to design as well.

Vision Statements

"Through a system of design tools and processes that is flawlessly integrated into Blender's development that can adapt to the contributors' workflows, they feel empowered to design and make design decisions. Organizational design capacity grows, and a synergetic balance between design and development takes place."

"Blender uses a contributor-focused portal and process that allows anyone to learn about Blender's design principles and meaningfully contribute. Blender's design expertise quantity grows significantly since many more people can and want to contribute."

Figure 4.1 | Photo of concept ideation

4.2 Ideating Opportunity Solutions

Based on the opportunities and ideation philosophy, a divergent ideation process was undertaken to explore a wide array of potential solutions. This creative phase began by focusing on the four core opportunities for intervention while considering how to increase Blender's design capacity. For each opportunity, ideas were brainstormed on sticky notes. Inspiration was taken from research, interviews, professional engagement, and industry standard practices and emerging tools. This method allowed for a rapid creation of ideas with little friction. The full overview of ideas can be found in Appendix D.

Thematic Clustering

The best ideas were clustered to synthesize the information and identify overarching themes, helping to move from a collection of individual concepts to a structured overview of potential intervention strategies. After several iterations of grouping and labeling, three distinct high-level categories emerged from the themes, indicating areas where interventions could be focused, as can be seen in Figure 4.2 on the right.

01 - Internal Expertise Leveraging: Process Tools

This category relates to internal processes and consolidated themes centered on improving the day-to-day operational workflows. The most significant theme here was Design Artefact Optimization, which included ideas for creating standardized templates, a "single source of truth" for design elements, and dynamic tools to make design mockups more interactive and useful for developers. Another key theme was the creation of better Boundary Objects: shared artifacts like interactive prototypes that can be understood and used by both designers and developers to bridge the communication gap.

02 - Internal Expertise Leveraging: Process Methods (Strategy, Teams, Roles)

The second category also relates to internal processes, but grouped ideas related to the organizational and strategic aspects of design integration. Prominent themes included clarifying Team & Roles by formally assigning design responsibilities and decision-making authority, and establishing design as a Strategic Force with dedicated budgets and checkpoints. Ideas around Bridging Disciplines through structured workshops and Process Standardization were also prevalent, highlighting ideas for more formalized collaborative routines.

03 - External Expertise Leveraging: Community Outreach

The final category focused on better integrating the external Blender community into the design process. The main themes revolved around Leveraging Community & Open Distribution by creating a public-facing design platform. This was complemented by ideas for Lowering Barriers for the Community, such as providing clear guidelines, "get started" packages, and self-service resources. Finally, a clear theme emerged around the need to Incentivize Contributors with ideas related to community challenges and structured feedback to make contributing a more rewarding experience.

Next, these thematic ideas were condensed into three comprehensive concepts.
→

Condensing Thematic Ideas to Concepts

These clusters naturally formed three complementary concepts that address the problem from different perspectives (tools, methods, outreach).

Based on observational findings, interview insights and participatory observation, the combined ideas were further developed into 3 concepts.
→

Key Takeaways From This Process

The top ideas solved multiple challenges at the same time, while leveraging both internal and external processes and expertise

Since the scope of creating an actionable and digital tool is quite specific, it was quite straightforward to combine the best ideas into concepts, similar to adding features to a software.

The different ideas and themes reflect the different stakeholder classifications (internal C1/ C2, and external C3) and will thus have varying importance for the first implementation of a solution.

Figure 4.2 | Thematic clustering of best ideas generated

4.3 The Three Concepts: A Roadmap

The research results in Chapter 3 revealed that Blender’s challenges are systemic and connected. They occur across different levels: a lack of infrastructure (tools), a lack of defined process (methods), and a barrier to entry (access). To address these different layers, a Systemic Roadmap was developed, combining three concepts. While traditional product design often selects a single 'winner,' the complexity of Blender's ecosystem requires a layered approach: establishing infrastructure (Leverage) before expanding community access (Quantity) (read 4.4). In addition, a roadmap is the most valuable form of deliverable for Blender, as it extends the scope of the project since it can be used and implemented even after this project is finished.

This strategic decision emerged from the ideation phase in the previous section (4.2), where it became clear that the ideas were not competing alternatives, but complementary layers of a cohesive, necessary infrastructure. It appeared impossible to make one concept 'win' over another since both are equally necessary. This is also the reason why the ideas within the overarching themes naturally allowed for combining the ideas into larger, comprehensive concepts.

Rather than viewing the following concepts as mutually exclusive competitors, they are designed as sequential interventions of a holistic implementation strategy. An Intervention Roadmap consisting of three interdependent interventions is proposed that can be implemented sequentially to provide the most value to the case study organization.

The goal of this chapter is to define this roadmap by outlining the 3 concepts and to identify the foundational prerequisite concept that must be developed first to enable the others.

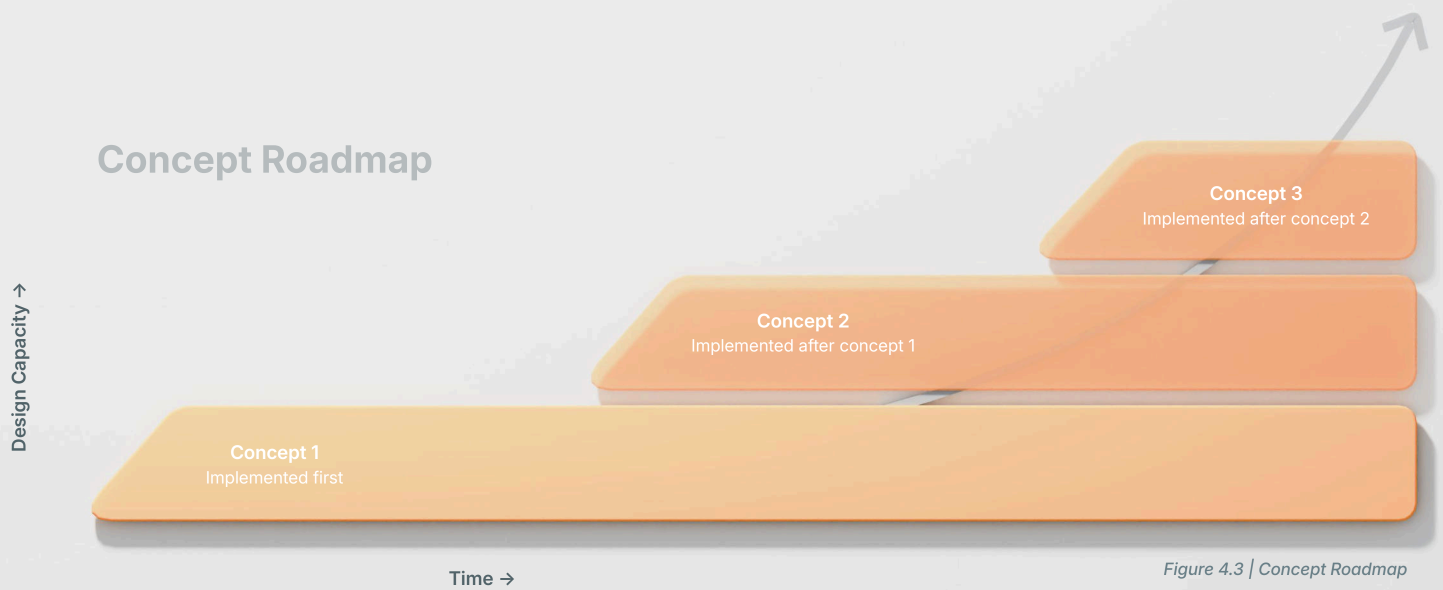


Figure 4.3 | Concept Roadmap

Concept Generation Criteria

The previously listed findings of the research require that any successful intervention meet a specific set of criteria. The intervention must be:

- **Must be a Systemic and Tactical Intervention:** The research has pinpointed it must address the underlying infrastructure and processes, the tactical "How" layer. Therefore, any viable solution must be fundamentally Systemic, addressing the underlying processes and infrastructure that connect strategy to execution. It must provide concrete tools and workflows to resolve the day-to-day operational frictions observed between designers and developers.
- **Culturally Compatible:** A solution that is not Culturally Resonant will be rejected by the organization. This resonance is a direct reflection of alignment with the organization's core values and mission: the strategic-level "Why". The intervention must therefore support Blender's established culture of autonomy, transparency, and community-driven contribution.
- **Holistic:** It must ensure that strategic intent ("Why") is effectively translated through tactical processes ("How") to produce superior and more efficient Operational ("What") outcomes, such as higher-quality UI mockups and smoother developer handoffs. It must address the interconnected nature of the challenges, from inaccessible knowledge to the developer-centric culture, rather than providing a point-solution for a single issue.

These criteria will now be used to create the three primary concepts to realize the most feasible, viable and desirable path forward.

→

Figure 4.4 | Concept 01 Visualization

Concept 01 - Blender's Design System

Based on the insights from the internal process-focused ideation theme and the need for process tools, the concept of a Design System was developed. This solution is widely recognized in the industry as the best practice for emphasizing and facilitating design in software development organizations and is also supported by recent academic literature (Zhang et al., 2025). Furthermore, this concept directly addresses the needs explicitly mentioned by key internal stakeholders during the interview phase.

1. Concept Definition (The "What")

This concept proposes the creation of a mockup creation tool serving as the "single source of truth" for Blender's product development. It consists of a centralized, accessible library containing three layers: Tokens (design tokens for colors, typography, and spacing), Components (reusable UI elements like buttons and menus), and Guidelines (usage rules and design principles). Together, these elements form a shared language and toolkit that enables both designers and developers to build consistent, high-quality user experiences with greater speed, efficiency and accuracy. The Design System will first target stakeholders closest to the challenges (C1) and then expand outwards towards C2 (Remote Devs) and C3 (Community) stakeholders (see Concept 3).

2. Tactical Mechanism (The "How")

Functionally, the Design System serves as the boundary object between the two disciplines. Instead of relying on static image mockups or ambiguous screenshots, the system utilizes design tokens (variables) to create a literal shared vocabulary: a change in a visual property (e.g., an 'active.blue.color' token) in the design file can map directly to code variables in the development environment. This mechanism creates a "living" infrastructure that empowers developers to utilize self-service resources for common UI tasks without needing constant designer supervision, while ensuring that complex design decisions are manifested in pre-made UI templates and documented in the guidelines.

3. Value: Feasibility, Viability & Desirability

- **Desirability:** This intervention is highly desirable for the internal team (C1/C2) as it directly resolves the daily friction of "Inaccessible Knowledge." It empowers non-designers (developers) to make confident design decisions independently (solving the "Expertise Paradox") and frees designers up from the burden of being asked for support.
- **Viability:** The concept is the primary driver of Leverage within the Design Capacity framework. Unlike the Portal (Concept 03, which increases Quantity), the Design System drastically increases the quality and efficiency of the existing workforce. By reducing the time spent on inefficient UI mockup creation and bug fixes, it multiplies the output of the current team.
- **Feasibility:** Technically, the concept is highly feasible and established using open-source tools like Penpot, which align with Blender's licensing values. Culturally, while standardization and structuring can sometimes be met with resistance in open source, a design system functions as an inspiring enabler (a toolkit in the toolbox) rather than a constraint (a rulebook), ensuring higher adoption by providing immediate practical value to contributors.

Crucially, feasibility has been validated during this thesis. Utilizing my position within the Blender Team, the Design System was developed and user-tested in a real production environment from day one, confirming its alignment with key stakeholders.

4. Challenges Tackled & Stakeholders

- **Challenges Tackled:** Directly targets Challenge 3 (Inaccessible Design Knowledge) by turning the hidden knowledge into an accessible library, Challenge 6 (Friction in Designer-Developer Interaction) by creating a shared language, clear guidelines, reusable components and the direct connection of code and design (tokens & code inspection) and Challenge 7 (The Expertise Paradox) by enabling self-service UI creation.
- **Stakeholders:** Primary focus on C1 (HQ Designers/ Developers) and C2 (Remote Developers), who handle the core implementation work.

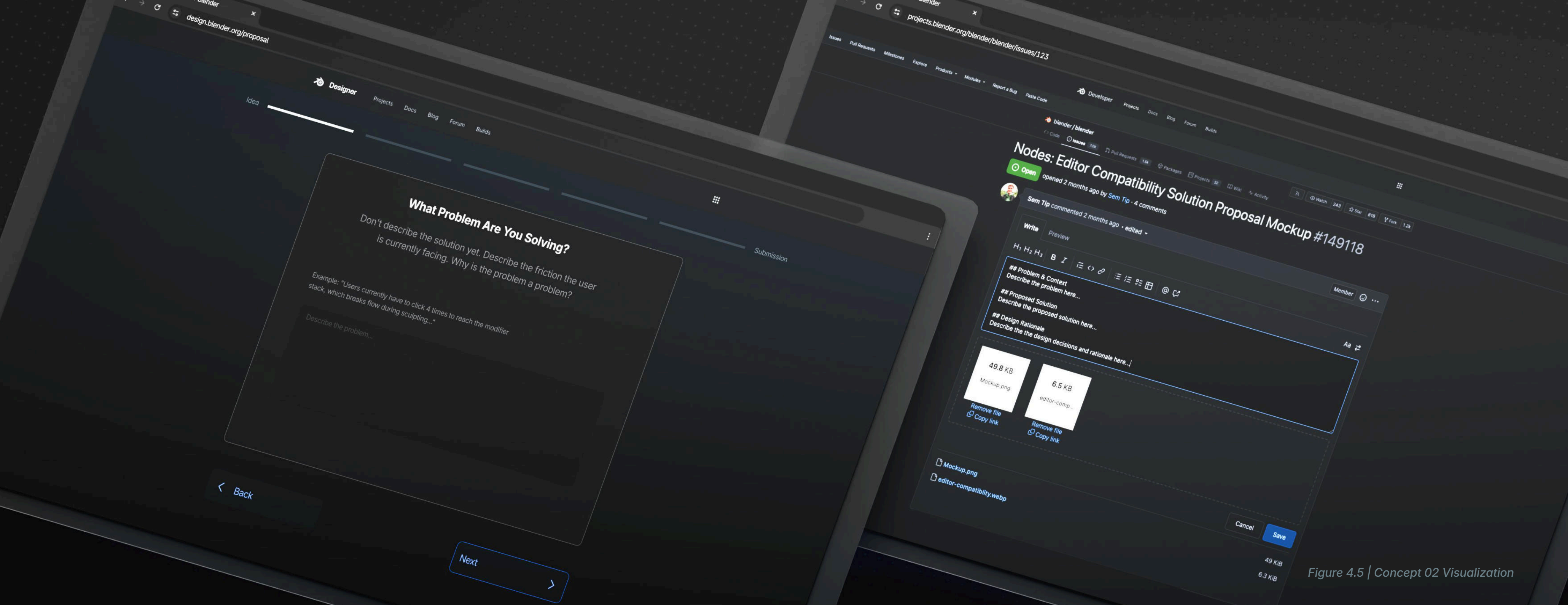


Figure 4.5 | Concept 02 Visualization

Concept 02- Design Proposal Guide

While the Design System (Concept 01) provides the necessary tool, relying on infrastructure alone also creates a significant risk. When expanding the design system to external stakeholders (C3) later, if the community is given powerful design assets without a guiding methodology, the result may simply be "prettier bad ideas", or worse, more convincing bad ideas: high-fidelity UI mockups that appear polished but lack structural logic or strategic alignment. To prevent this, the tool must be paired with a method. Concept 02 serves as this necessary guide and intellectual filter, teaching contributors how to think about the problem and define the rationale, while using the Design System to execute and present the solution.

"For the community, a lot of the time, they come up with just an idea and without thinking about how that would be implemented, or ... how that affects the other existing features." - Interview participant

1. Concept Definition (The "What")

This concept proposes the implementation of a design proposal guide, a standardized methodology and flowchart webpage for defining and documenting design problems before implementation begins. It is a sequential form and a supporting decision-making flowchart, guiding the contributor to better design considerations. Unlike purely technical tasks, this framework requires contributors to explicitly articulate the User Problem, Context, and Design Rationale alongside the visual solution. This directly addresses the core friction identified by Zhang et al. (2025), where the use of disparate artifacts leads to a "misalignment between the initial design and the eventual deliverable" (p. 14).

2. Tactical Mechanism (The "How")

Functionally, this operates as a "pre-flight check" integrated into Blender's existing project management tools. Before a community proposal even gets reviewed, the contributor utilizes the guide to structure their intent through two distinct stages:

- **The Flowchart:** Acting as a filter before design work begins, this visual guide helps contributors understand their wishes and guides them to better consider the context and right design approach. It prompts them to determine the scope: whether a problem requires a simple UI tweak (directing them to specific components) or a fundamental workflow rethink (directing them to Blender's design principles). This ensures that the contributor is solving the right problem before they start executing.
- **The Template:** Serving as the standardized output of the process, this proposal template is integrated directly into the project management system (Gitea). It helps contributors to articulate the logic behind their visuals, explicitly defining the user problem, edge cases, and rejected alternatives. This creates a necessary bridge between disciplines; as noted by Zhang et al. (2025), while designers rely on visuals, "developers primarily depend on text... to elucidate information about their implementations" (p. 14). By prioritizing this textual logic, the template functions as an effective "boundary object" that supports "subsequent interactions and establishes shared understanding" (p. 17).

Participatory observation confirmed this friction firsthand: when drafting proposals myself, the absence of examples and structure made it difficult to determine what rationale was needed alongside the visuals. This validates that a guide can help (new) contributors through the technical considerations they would otherwise overlook, the unknown unknowns.

3. Value: Feasibility, Viability & Desirability

- **Desirability:** It addresses the "Developer-Centric" culture (Challenge 4) by framing design not as "making things pretty," but as "logic and problem solving", a language developers understand respect. It is a way to address the "noise from community" (Challenge 10), by structuring and increasing the quality of their proposals.
- **Viability:** This creates Design Capacity leverage by stopping "bad ideas" early. By forcing problem definition upstream, it prevents wasted development hours on features that don't fit the big picture. It ensures that "developers can gain a more comprehensive grasp of the UX vision" (Zhang et al., 2025, p. 16) before code is written.
- **Feasibility:** It is culturally feasible because it mimics the existing Technical Proposal workflows already native to Open Source: it is a document-driven (async-native) process that is close in (digital) location to the actual work done, instead of being a separate document, making it robust for a distributed team.

4. Challenges Tackled & Stakeholders

- **Challenges Tackled:** Directly targets Challenge 10 (Noise from Community) Challenge 3 (Inaccessible Design Knowledge) by manifesting documentation of rationale and making it directly accessible, and Challenge 4 (Developer-Centric Culture) by shifting focus from "Implementation first" to "Definition first."
- **Stakeholders:** Primary focus on C1/C2 (Internal & Core Contributors) to establish the standard and content for the form, which then serves as the model for C3 (Community) via the Portal.

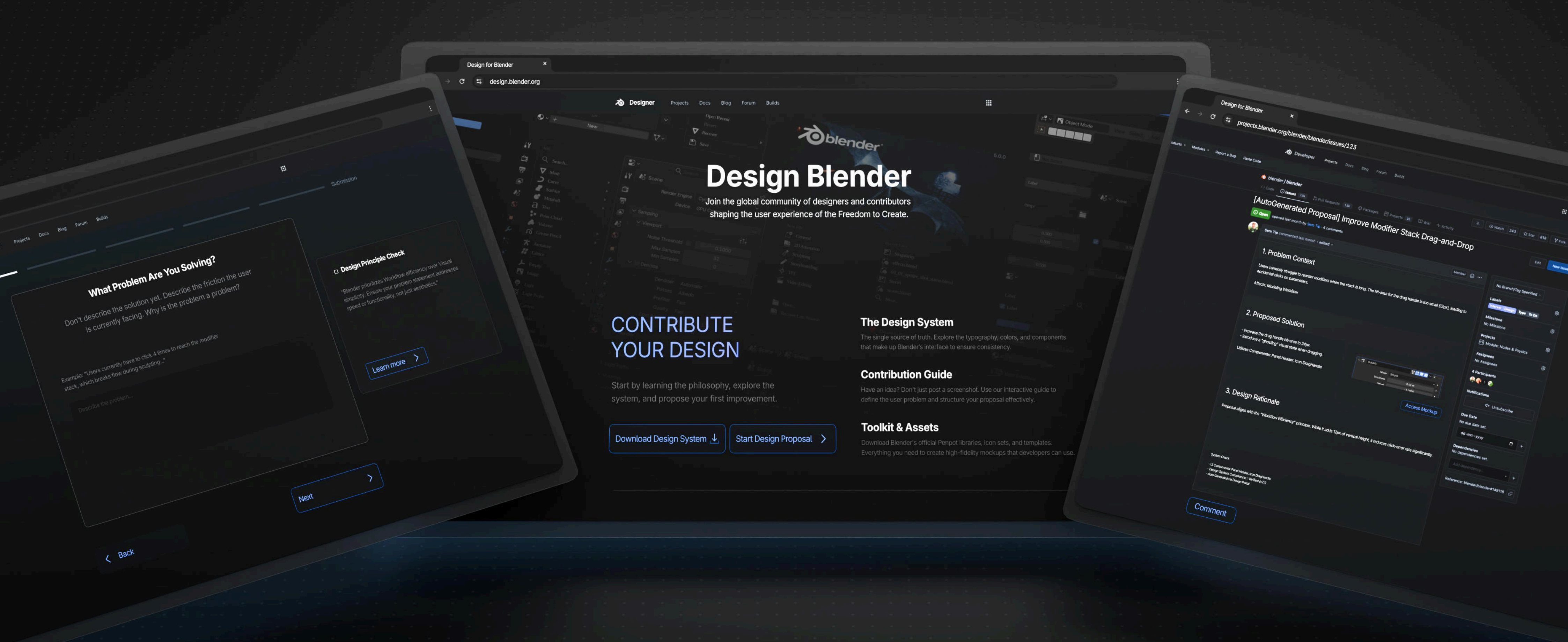


Figure 4.6 | Concept 03 Visualization

Concept 03 - Accessible Design Portal

Based on the insights from the community-focused ideation theme, the concept of an Accessible Design Portal was developed. The initial idea was discussed and supported during the interviews with Blender's executives. Appendix D contains a user interaction flow.

Concept Definition (The "What")

This concept extends Concept 02 and proposes the creation of a dedicated, public-facing Blender Design Portal serving as the "single source of truth" for design within the Blender ecosystem. Analogous to the existing Developer Portal, it creates a centralized webpage that aggregates guidelines, downloadable asset libraries (Design Systems), and dashboards for active projects. The core feature is the Design Proposal Guide (Concept 02).

2. Tactical Mechanism (The "How")

Instead of scattered forum posts, contributors are visually guided through a structured questionnaire to analyze context and define problems before proposing solutions. This mechanism functions as a structuring funnel, generating standardized proposal templates that move design discussions out of disparate technical forums (such as projects.blender.org and devtalk.blender.org) into a design-focused environment. The portal autogenerates proposal submissions to Blender's established design principles, making sure that submissions are not just isolated ideas but well-reasoned proposals that core developers can easily review, understand, and further develop.

3. Value: Feasibility, Viability & Desirability

- **Desirability:** This intervention creates value and is desirable for both the community and the core team since it transforms the current frustration of unstructured feedback into actionable assets, lowering the barrier to entry for external designers while reducing "noise" for developers.
- **Viability:** The concept directly builds the organization's Design Capacity: it expands the quantity of available expertise by unlocking the previously inaccessible external design expertise pool (C3), while simultaneously increasing leverage by providing the missing infrastructure to capture their contributions effectively. It aligns well with the open-source principle of empowering the community and the goal of innovation by attracting a wider pool of talent and elevating the discourse around design in open source.

"A design thinking 101 flowchart. I think that that would be super interesting... you are helping everyone to think in terms of design, even non-designers, just a user." - Interview Participant

- **Feasibility:** Culturally, this proposal is highly compatible with Blender's open-source ethos of transparency and "online-first" collaboration. The Design Portal naturally extends the success of the well-established and proven code-contribution Developer Portal to the design domain. Technically, this also makes the portal highly feasible as it can be built upon Blender's existing web infrastructure. However, the ongoing viability depends on the core team's ability to develop the first content and maintain the portal to ensure it remains a living resource rather than static documentation.

During this thesis and together with the Blender Team, great first steps have been made in emphasizing design at Blender which resulted in the recent launch of design.blender.org which currently hosts the Design System - with discussions to potentially expand this initiative into the comprehensive portal proposed here.

4. Challenges Tackled & Stakeholders

- **Challenges Tackled:** Directly targets Challenge 9 (Difficulty Leveraging External Design Expertise) by providing a clear entry point, and Challenge 10 (Friction and Noise from Community) by structuring and filtering input.
- **Stakeholders:** Primary focus on C3 (External Contributors) and C4 (Users), while aiming to reduce the filtering burden on C1 (HQ Developers/Designers).

4.4 Concept Evaluation & Prioritization

Having defined the three components of the roadmap, they are evaluated not as competing alternatives, but as sequential parts of a roadmap. This section analyzes how these concepts address the opportunity of increasing Design Capacity and determine the sequence for implementation.

1. Validation: Design Capacity Impact

Section 3.8 established that increasing Leverage is the strategic priority. To validate that the proposed roadmap aligns and effectively solves the challenges, the three concepts the three concepts are mapped onto the framework:

- **Concept 01 (Design System) & Concept 02 (Guide):** These interventions focus primarily on the Leverage axis. By providing shared tools and standardized processes, they increase the efficiency and quality of the existing team's output.
- **Concept 03 (Portal):** This intervention focuses on the Quantity axis. It leverages external talent by increasing the volume of contributors.

Thus, the roadmap addresses both axes. However, attempting to increase Quantity (Concept 3) without first securing Leverage (Concept 1 & 2) would likely worsen existing coordination issues, leading to more noise rather than more value. As established in Section 3.8, the leverage axis has to be addressed first.

The research identified that the primary bottleneck is the internal development process (C1/C2). Improving community interaction (C3) is a valuable 'downstream' goal, but it is an optimization, not a foundational fix.

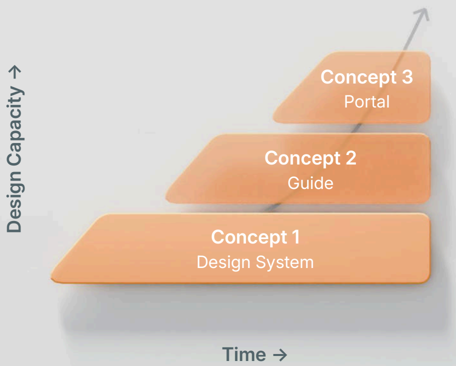
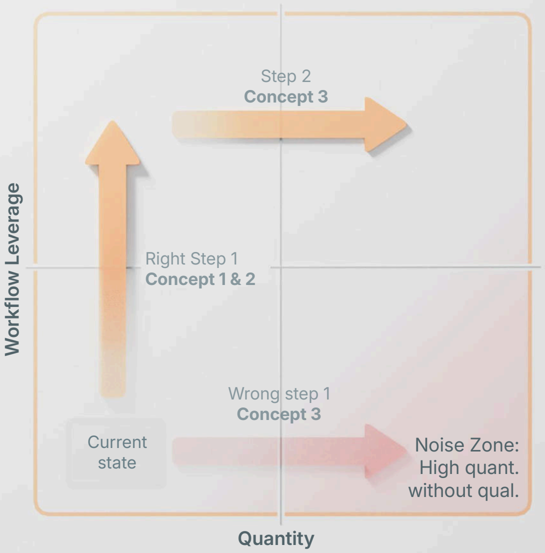


Figure 4.8 | Concept Roadmap

Figure 4.7 | Design Capacity Matrix



2. Interdependence & Implementation Order

While all three concepts are desirable, they have to be implemented sequentially due to the 'leverage first' principle (Section 3.8) and because the concepts depend on each other:

- **The Portal (Concept 3) uses the Design System (Concept 1):** Guidance and consistency are important when involving the community. The distributable design system consisting of shared libraries, guidelines, or templates to download is an essential element of the concept.

- **The Method (Concept 2) depends on the Design System (C1):** The attempt to propose a design process without providing the enabling infrastructure is risky to be perceived as strategic force. Without necessary tools, developers and designers would lack the common artifacts and vocabulary required to bridge their disciplinary gap. The pragmatic and autonomous work culture at Blender will appreciate practical manifestations (tools) to relate it to. A process method provides the rules but no shared equipment or space for interaction.

- **The System (Concept 1) is independent and can be created today:** A Design System creates immediate value for the internal team (C1/C2 Contributors) even without the other concepts.

3. Selection of the Primary Intervention

Based on this dependency analysis,

Concept 01 (The Design System) is selected as the foundational prerequisite for this roadmap and is thus selected to be developed first. Concepts 2 and 3 are therefore designated as future implementation phases and are mentioned in the "Future Implementation Strategy" (Chapter 5).

Design System Value Related to Roadmap

- **Immediate return on investment(ROI):** It solves the acute internal pain of "Inaccessible Knowledge" (Challenge 3) immediately.
- **Feasibility:** It can be built now by the core team without requiring 'forced' cultural behavior change first. The design system will function as a natural 'invite' to change.
- **Enabling future potential:** It creates the necessary assets to eventually enable the Process Method (Concept 2) and build the Design Portal (Concept 3).

Design System Value Related to Blender's Design Workflow

- **Facilitates & accelerates creative processes:** The design system should feel as fluid and quick as sketching, facilitating creative thinking and creating - at a higher fidelity and speed by eliminating repetitive parts (addressing challenge 4: Developer Culture)
- **Enables non-designer autonomy:** Since every contributor does design work within their specific module, the system empowers these non-designers to make high-quality proposals independently. The pre-built component library allows them to construct accurate, high-fidelity UIs without relying on designers (Addressing Challenge 7: Expertise Paradox).
- **Centralizes the truth:** It replaces scattered wiki pages and chat logs with a single, version-controlled library that acts as the definitive reference for everyone (addressing challenge 3: Design Knowledge).
- **Shared language:** It establishes a shared visual vocabulary where design tokens map 1:1 to code variables, creating a more direct connection and reducing friction during handoffs (Addressing Challenge 6: Des-Dev Friction).

4. The Strategic Evidence

Addressing the Systemic Challenges

The research reveals that the primary obstacles are a "scarcity of design expertise contributions" (as defined in 3.6) and an infrastructure that struggles to leverage them effectively. Therefore, the most impactful solution is to create a system that enhances Blender's overall Design Capacity (as defined in 3.7).

An operational-level ("what") fix would fail to address the underlying issues of inaccessible knowledge, developer-centric culture, and the friction in designer-developer handoffs that would inevitably compromise the next project (as defined in 1.6).

Validation: Industry & Internal

The most valuable and sustainable intervention is one at the tactical-level ("how") that builds a foundation by improving the organization's systemic ability to produce high-quality, consistent design proposals efficiently. This requires a systemic solution, and the convergent evidence from both industry standards and academic research points to a design system (Zhang et al., 2025; G. Langreo & P. Ruiz-Múzquiz, personal communication, 8-10 October 2025).

- **Industry Standard:** as identified during the interviews and conference visits, it is the industry standard adopted best practice for these exact problems.
- **Internal Need:** there is a clear internal need from Blender's own stakeholders for it (interviews & participatory research).
- **Academic Support:** and most critically, the academic literature directly points to the functional components of a design system as the solution (Zhang et al., 2025).

Academic Support

The comprehensive literature review by Zhang et al. (2025) supports this direction, identifying best practices that are describing a functional definition of a design system. The paper highlights the severe inefficiency caused by teams using disparate artifacts - "'static and textual' versus 'dynamic and visual' designs" - which have both been identified during the interviews and participatory research phase. This leads to a "misalignment between the initial design and the eventual deliverable" (Zhang et al., 2025, p. 14).

To counter this, the research recommends "establishing more effective boundary objects" (p. 17) and a "shared vocabulary" (p. 18) to "guarantee effective communication with minimal confusion." This 'shared language' is a key concept that has been further supported during the interviews and professional engagement research phase.

Key Takeaway

Concept 01 (The Design System) is selected as the primary intervention for development in this thesis. The remainder of this chapter will therefore focus exclusively on the structural definition, detailed design, and validation of Blender's foundational Design System.
→

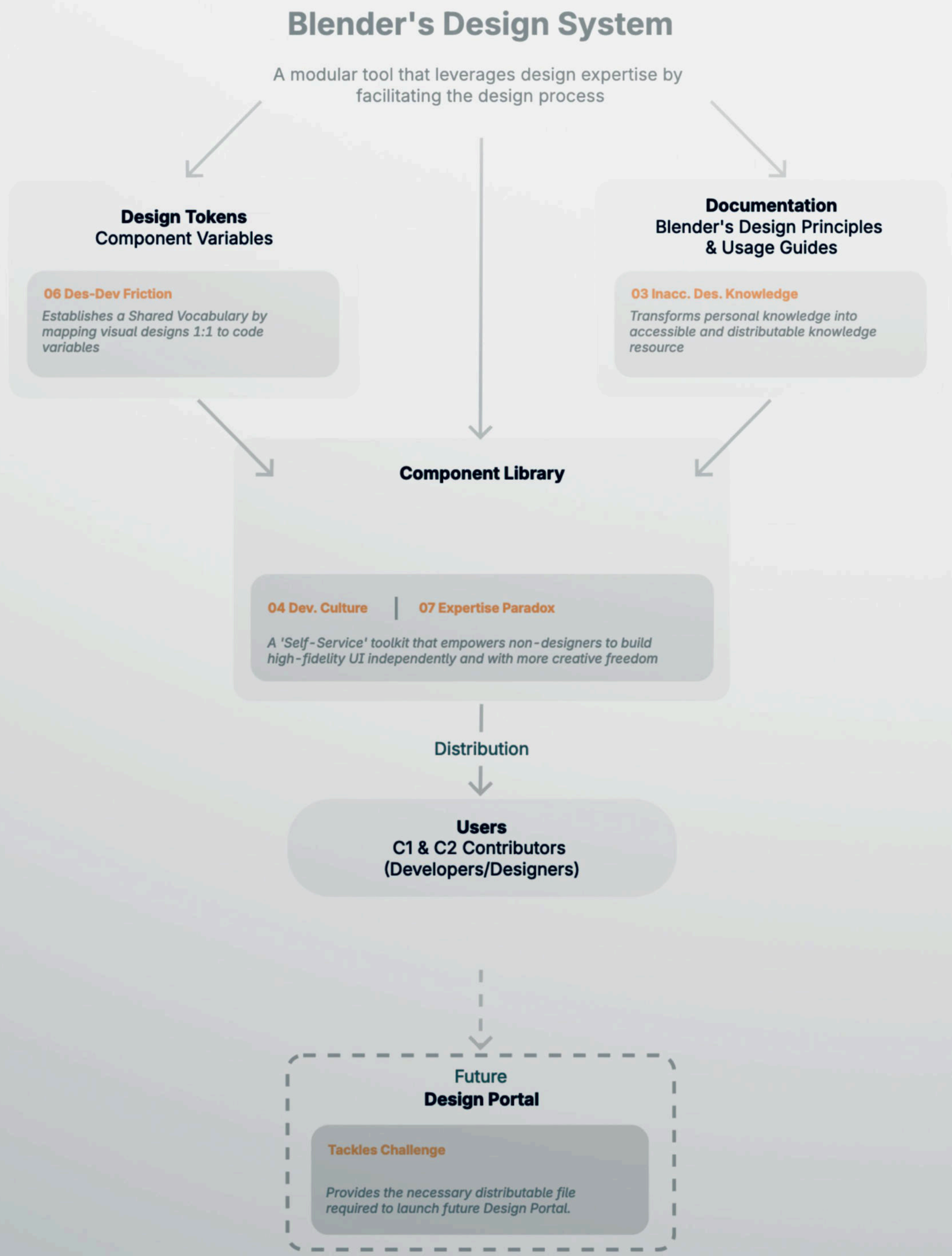


Figure 4.9 | Strategic architecture of Design System concept

4.5 Building Blender's Design System

Strategic System Architecture

To solve the identified challenges and deliver the value of autonomy, a single truth, and accelerating creative processes mentioned in the evaluation phase (4.4), the design system requires the following elements. The system is organized as visualized in Figure 4.9 on the left, where each element addresses a specific systemic challenge.

Operational Risks & Mitigation

A concern with the Design System is that it can become overly intricate or rigid, slowing down the workflow rather than accelerating it. To mitigate this, extremely early user testing was conducted and the proposed system was designed around a "Puzzle Piece" (component) philosophy. It creates a deliberate balance between constraint (providing accurate, pre-built UI blocks that ensure consistency) and freedom (allowing these blocks to be dragged, assembled, and modified without restriction).

Software Selection: Penpot

To build design systems, many organizations favor and use 'established' proprietary software like Figma. But in contrast, Penpot was selected, an open-source UI/UX design tool. The choice was primarily driven by two non-negotiable requirements identified in the research:

- **Open-Source Alignment:** Since Penpot is open-source, it aligns with Blender's philosophy of using open-source tools to make Blender. It also removes licensing barriers and simplifies distribution (addressing Challenge 3 & 9).
- **Accessibility:** Open source means no financial cost, no data sharing and no artificial usability limits such as a maximum number of files. In contrast, an own version of the digital tool can even be located internally at Blender (self-hosting) allowing for endless customization, freedom and safety. During the creation of the design system, Blender's self-hosting of Penpot launched and is now live on design.blender.org. After creating the first versions of the design system on Penpot's own hosting, this is where the design system lives now.

Even though Penpot has these advantages, it's still relatively young and early in development. There are plenty of bugs (errors), stability issues and missing features that made it very challenging to work with.

During the participatory research I identified that Blender has already started adopting Penpot in their workflow. This decision builds on that.

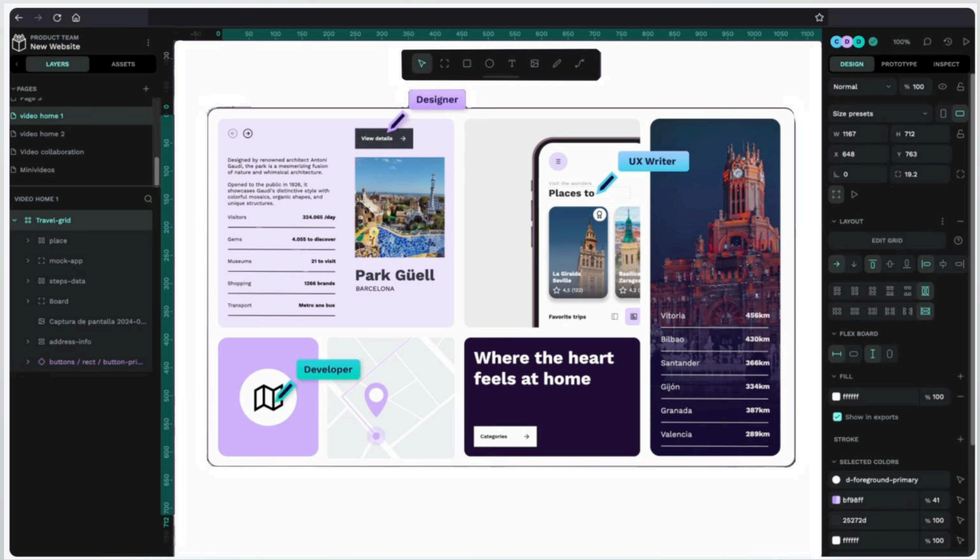
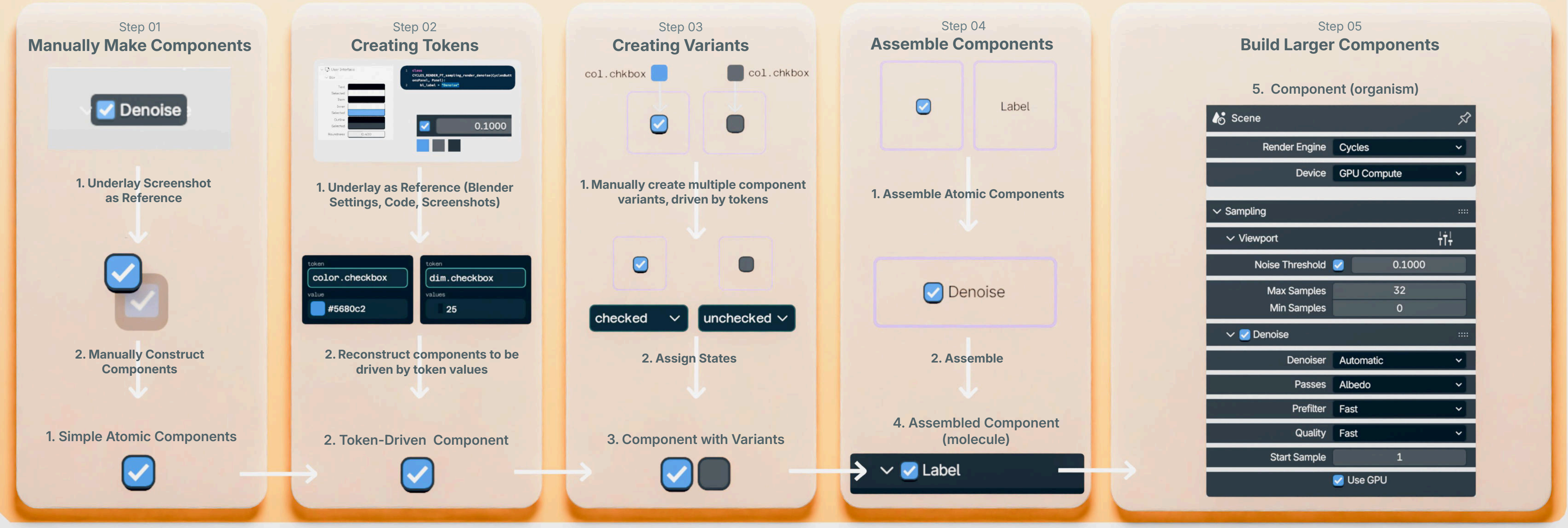
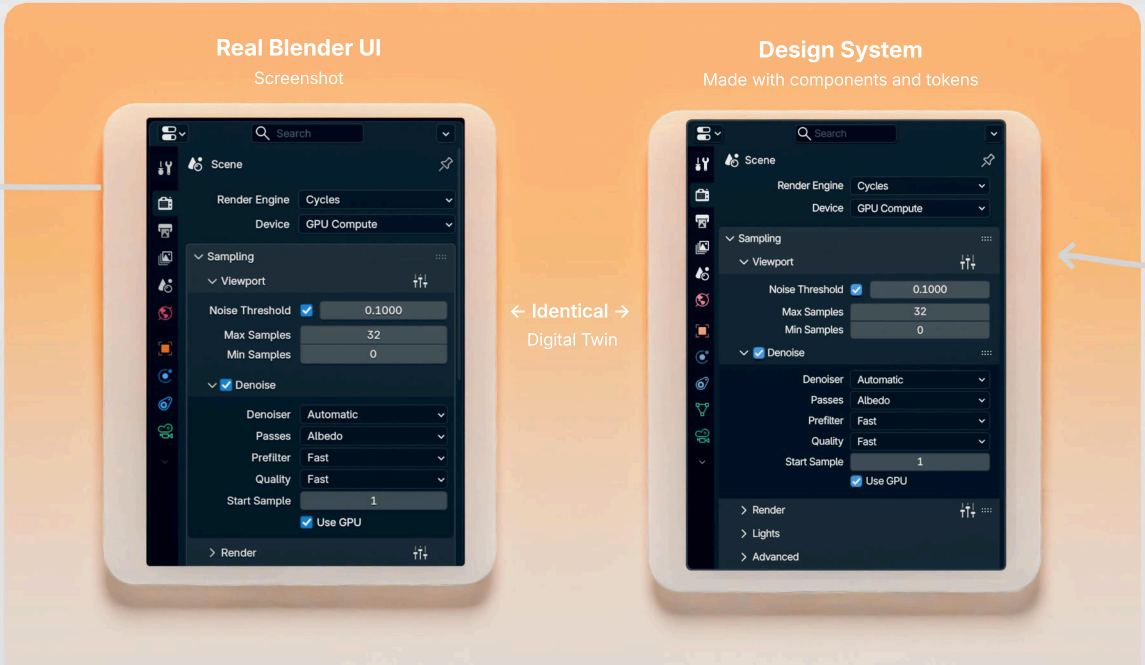


Figure 4.10 | Penpot's User Interface (Penpot.app, 2025)

The Process of Creating Blender's Digital Twin





The Process of Creating a Modular Twin

The objective was to reverse engineer Blender's existing UI. To do this, many design systems were studied. Unlike typical design system applications (websites, apps), Blender is an advanced 3D software with a more complex UI with many different states, modes, inputs and customizability. This makes it challenging to reflect accurately in the design system.

The system was built by first studying and reverse-engineering Blender's existing interface into as atomic (smallest) components as possible. Next, using the real UI as an underlay, all design components were constructed individually to match the exact size, states, and behaviors of the live software. This included creating different variants of components, such as checked and unchecked. This process resulted in a design system that creates results that are visually identical to the real Blender software.

Design System First Version

The result of this process was a design system draft that was ready for first user testing. After first tests and fundamental improvements, such as switching from a grid to a row layout which turned out to be more intuitive, the design system was ready for the co-creation workshop where it was tested and evaluated by key stakeholders. →

Figure 4.11 | Process visual of Design System creation

4.6 Co-Design Session: Requirement Validation

To align, test, and validate the Design System against user needs and expectations, I organized and led a co-design session with key C1/C2 stakeholders.

Purpose

The goal was to validate the stakeholder's requirements and test the tactical utility of the draft system, validating stakeholder requirements and refining the user experience in preparation for real-world production

Session Setup & Methods

The 3-hour session alternated between context presentation and interactive activities. To collect feedback, rose-thorn-bud (for feedback), dot voting (for prioritization), and alignment checks (thumbs up/side/down) were used. See Appendix E for the workshop slides and the full activity results.

Design System Demo

To test actual utility, participants were given access to the Alpha version of the Design System and tasked with specific, realistic challenges. Screen recordings were captured to analyze cursor movements, hesitation points, and overall workflow.

- Prepared slide-deck with 32 slides for interactive session & 6 interactive activities including design system alpha demo

Participants

A cross-discipline group of 4 internal contributors, representing the core users (C1 & C2): 1 HQ Developer, 2 Product Designers, and 1 Blender Studio Artist (Blender user). Including designers and non-designers was a strategic choice to represent the challenges identified in Chapter 3.

Desired Outcomes

- **Validated Requirements**
- **Shared understanding purpose:** collectively what problems the DS should solve and create a group awareness
- **Capabilities:** define what the DS should do (using a component tree), and how it will do that
- **User needs:** what experience do the stakeholders want (priorities)
- **User-testing & feedback:** usability and intuitiveness of design system
- **Actionable plan:** next steps in DS development and implementation

Key Insights

- **The "Sketching" feeling & creative flow:** "Using the system should feel as quick and fluid as sketching", validating the "Puzzle Piece" risk mitigation strategy (defined in 4.4). The components need to snap together loosely and fast enough to allow for rapid ideation and creative flow
- **It should be fun!:** participants noted that using the design system should be enjoyable enough to encourage use and curiosity to learn about Blender's design
- **User experience and speed as the primary metric:** Reducing friction is key. Participants reported that dragging and dropping pre-made UI blocks felt easier than starting from scratch, shifting the cognitive load from 'graphic design' to 'problem-solving,' effectively indicating the design system's ability to increase leverage.
- **Custom mockup theme:** A critical new outcome of the session was the decision to create a dedicated Blender mockup theme. With the aim to prevent high-fidelity prototypes from being mistaken for real software, previously having led to feedback on a too detailed level. Instead of prompting detailed visual feedback, this theme keeps correct Blender dimensions and layout but applies a simplified, clearly work-in-progress style to steer viewer attention toward proposal functionality rather than details. After the session, a Blender designer developed the theme, and I integrated it into the design system.
- **Feedback on usability:** Certain components and workflows of the design system weren't intuitive. Limitations in components and components names were confusing. Specific default property visibility and inconsistent spacing were mentioned. Variants and flexibility highly valued and appreciated.
- **Aligning & Deciding:** 7 statements were presented to discuss and align on the key requirements for the design system. See the next spread for an overview of the decisions made.

Key Takeaway

The co-creation session validated the proposed requirements and gathered critical insights into the structure and usability of the tool. Most importantly, it defined the 'Sketch-like' experience required for adoption. These insights were directly integrated into the final system detailed in the following section.

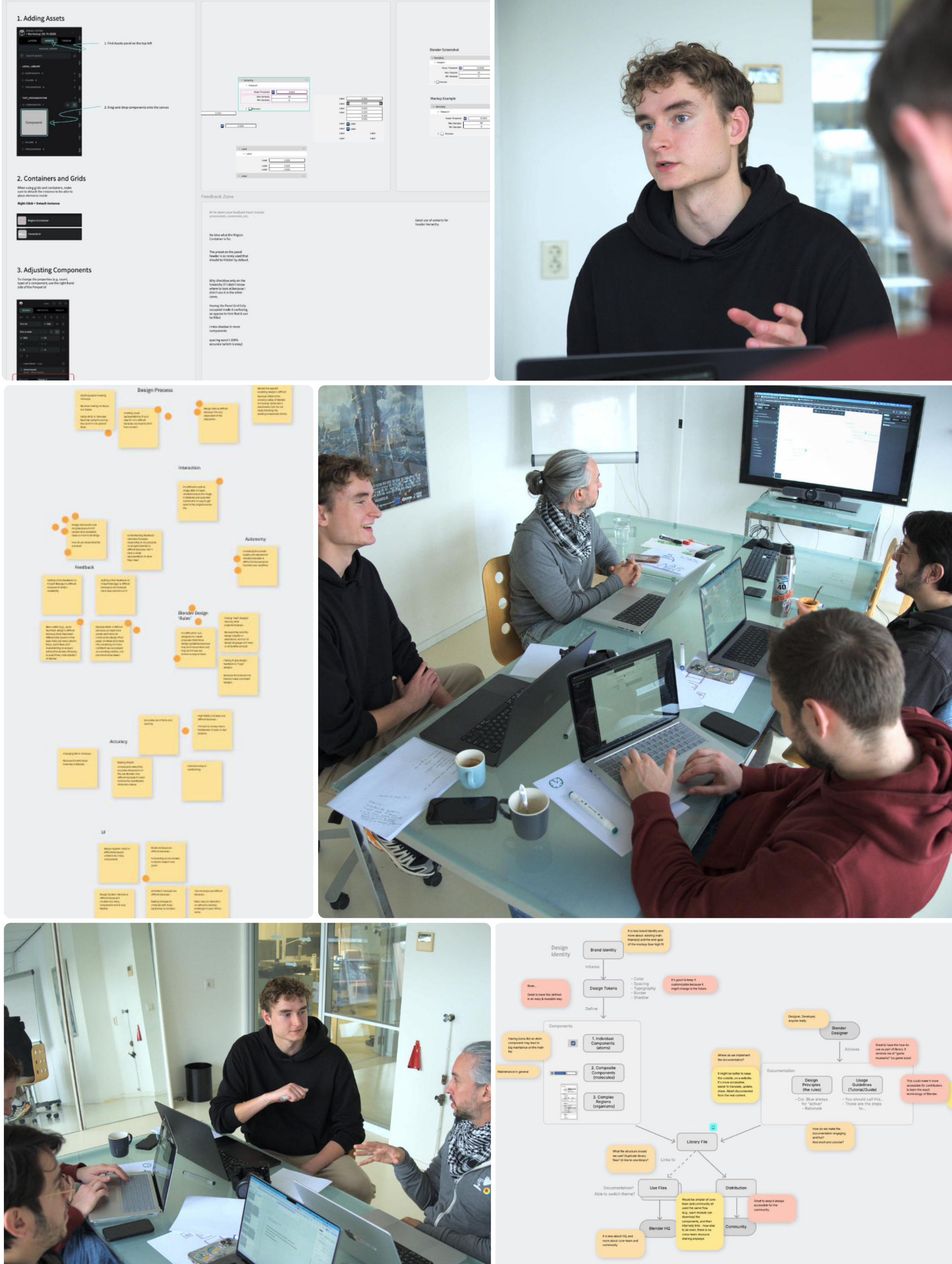
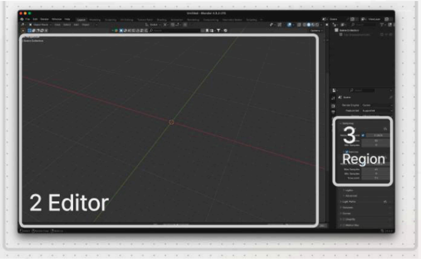


Figure 4.12 | Photo's and activities (from top to bottom): Design System Demo, Dot Voting on requirements, Design System architecture tree

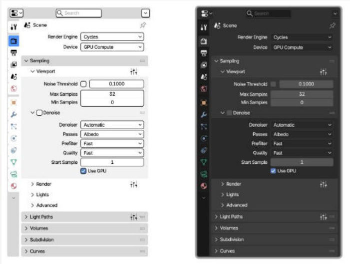
Validated Decisions

After the first ideation phase, How-To's and the preceding research, it became clear that a few elements are key to the concept's success. Below is a brief overview of the most important principles that were identified by presenting controversial statements during the Co-Design Session. These will form the foundation for further development.



01 - Demand-Driven Expansion

Ideally, the entire Blender UI would be available as pre-built components to maximize utility. However, given the massive scale of the software, releasing every module at once is resource-intensive and hard to maintain. Therefore, the system will be built out on a current-demand basis, prioritizing the UI elements most frequently requested by the core team. This approach has been validated in User Testing (4.8).



02 - Low and High-Fidelity Themes

Stakeholders emphasized the strong need for distinct visual modes to control how their designs are interpreted. A High-Fidelity Theme (exact match) is needed for final work, while a dedicated Low-Fidelity "Mockup" Theme is crucial for early-stage proposals to signal that the design is a concept, not a finished product. This approach also future-proofs the system against changes in Blender's theming.

Design System publicly available

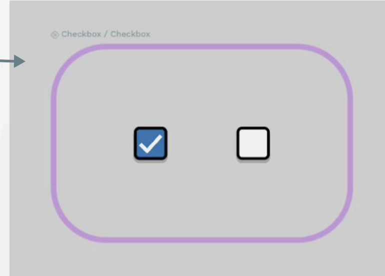
03 - Internal First, But Community Later!

While the long-term vision is full community access, the consensus was to prioritize the immediate needs of the internal team (C1/C2) first. Releasing an underdeveloped tool risks creating confusion rather than clarity for the wider community. Therefore, the system will first be developed for and validated by the core team to resolve internal workflow friction.

04 - Embedded Documentation (Guides)

Guidance should not be hidden in a separate PDF. Usage guidelines and documentation should be embedded directly within the Design System file itself (inspired by Blender's best practices from VS Code). This ensures that the instructions are always accessible at the point of use.

User is able to check and uncheck the checkbox



05 - Dynamic States (variants)

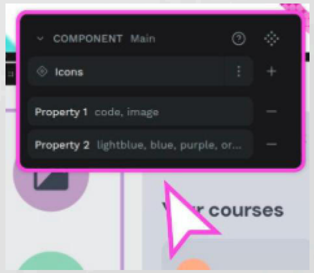
Components should definitely support dynamic states (e.g., hover, active, disabled). However, only where it adds significant value. Over-engineering every possible state adds unnecessary complexity. The guiding principle is "Deliberate Granularity": fewer, more flexible components are better than a vast library of rigid, hyper-specific ones.

Rejected Proposals

These were tested and discarded based on stakeholder feedback:

06 - Size Adjustable

Why rejected: While technically impressive, making every component infinitely resizable added massive complexity to the system (and makes it more fragile) without adding proportional value. Standardized sizes are sufficient for 99% of use cases. In rare cases sizing can be adjusted manually. Different size presets (variants) were created as a middle ground.



07 - Mockups Editable After Sharing

Why rejected: The idea of allowing others to edit a shared mockup PNG/file was considered as a "nice to have" but not a direct priority. The extra effort and complexity of implementing a collaborative "edit" feature outweighed the immediate benefit for the initial release. Perhaps a feature for later.

Figure 4.13 | Overview of validated decisions and rejected proposals

4.7 Concept Architecture

The Implemented System Architecture

To recreate the complex UI of Blender, the system is organized into the following parts:

- **Foundations (Tokens)**: This layer captures Blender's theme data, colors, typography, spacing, and corner radii as "Design Tokens" (variables). These values drive the components and make the entire system procedural. This ensures that a change in the design file (e.g., switching a theme) can propagate mathematically to every file that uses it, mirroring how Blender's code handles theming and UI definitions.
- **Components (The Interface Blocks):** Reusable, functional elements such as Buttons, Text Inputs, Checkboxes, and Sliders. These are built using the tokens and represent the "building blocks" of the interface.
- **Templates (Organisms):** Complex assemblies of components that form complete UI regions, such as editors and menus. These patterns define how components behave together in layout contexts. These are pre-built templates.
- **Documentation (Guidelines):** The design system is grounded in a clear set of design principles. These are the fundamental philosophies and goals that guide all design and development decisions, ensuring that every part of the product feels cohesive.

A Get Started Guide has been implemented into the concept, the two other documentation parts outlined in Figure 4.15 are next steps for future development.

Figure 4.14 | The System Architecture explained in layers

Concept Architecture

Figure 4.15 | System Architecture Overview

4.8 User Testing & Concept Refinement

To further improve the design system, the concept was implemented in the real-world production environment where it was stress tested by real ongoing Blender projects and key stakeholders over the course of 6 weeks.

Testing Goal & Method

The most important thing for the design system to have the desired effect and solve the problem is that it actually gets used during the real production work. If the concept doesn't get adopted or doesn't provide value during the actual work it won't solve the challenges. Aligning it with stakeholder needs in its actual use case is key. Thus, the testing was executed to improve and further tailor the concept using continuous user feedback from real use.

To ensure the Design System effectively addresses the challenges, its success fundamentally depends on voluntary adoption within the real-world production environment. A controlled, task-based test could improve the usability of the concept, but it wouldn't reflect the complex, non-linear nature of Blender's development workflows. Therefore, the testing utilized a longitudinal field deployment. Unlike isolated and controlled testing, this method evaluates the concept within the complex, unscripted reality of Blender's actual production environment. This allowed for the observation of how the tool creates value, or friction, during actual project work. By embedding the system into live projects, the concept could be improved and tailored based on continuous feedback derived from genuine stakeholder needs rather than hypothetical scenarios.

Over a duration of six weeks, the Design System was integrated into the daily workflows of C1 Blender Designer stakeholders (lead users), allowing for the observation of friction points and unexpected emergent behaviors (see next spread) that only arise during genuine tasks with deadlines.

On top of Blender designers as test participants, I used the design system myself also for participatory tasks such as contributing to ongoing projects. This way I got to experience the workflows firsthand and have better empathy and understanding for the users.

Data Collection

Qualitative feedback data was collected through triangulation, cross-referencing three different sources:

- **In-Situ Feedback (Asynchronous):** Real-time annotation of friction points via Penpot's commenting system and ad-hoc discussions in Blender's internal chat, to capture immediate reactions during the "flow of work."
- **Retrospective Analysis (Synchronous):** Bi-weekly review meetings to reflect on the process, discuss friction points, and prioritize improvements.
- **Artifact Analysis:** A review of the actual mockups (results) created during the testing period to assess the technical accuracy and fidelity of the output compared to the intended output.

Testing Criteria

Reflecting the original thesis research question (1.7), and the value proposition (4.4), the user testing specifically targeted Blender's design workflow and the creation of mockups for proposal presentation. The main testing criteria were:

- **Efficiency:** Does the system accelerate the speed of mockup creation?
- **Facilitates creative flow:** Does it facilitate creative flow by removing the burden of repetitive UI construction?
- **User experience:** Is the tool enjoyable to use, and do stakeholders voluntarily adopt it for new tasks?

Other criteria, such as Autonomy and Des-Dev Collaboration, are validated in the final concept validation (Chapter 6).

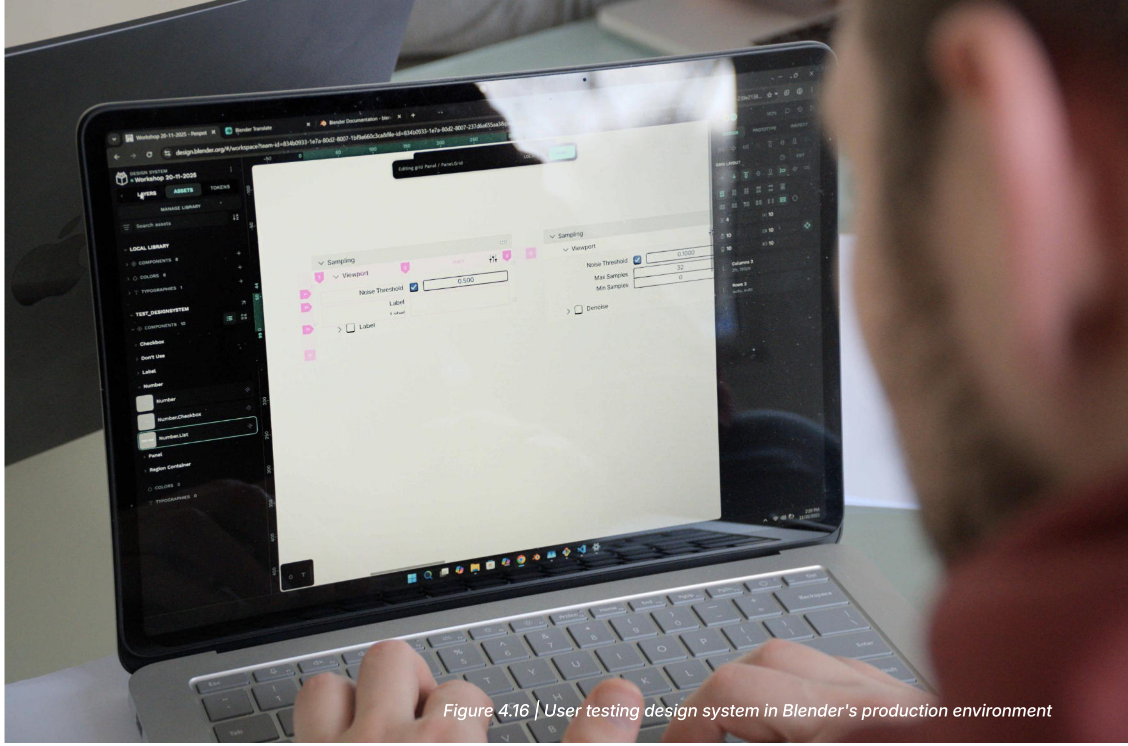


Figure 4.16 | User testing design system in Blender's production environment

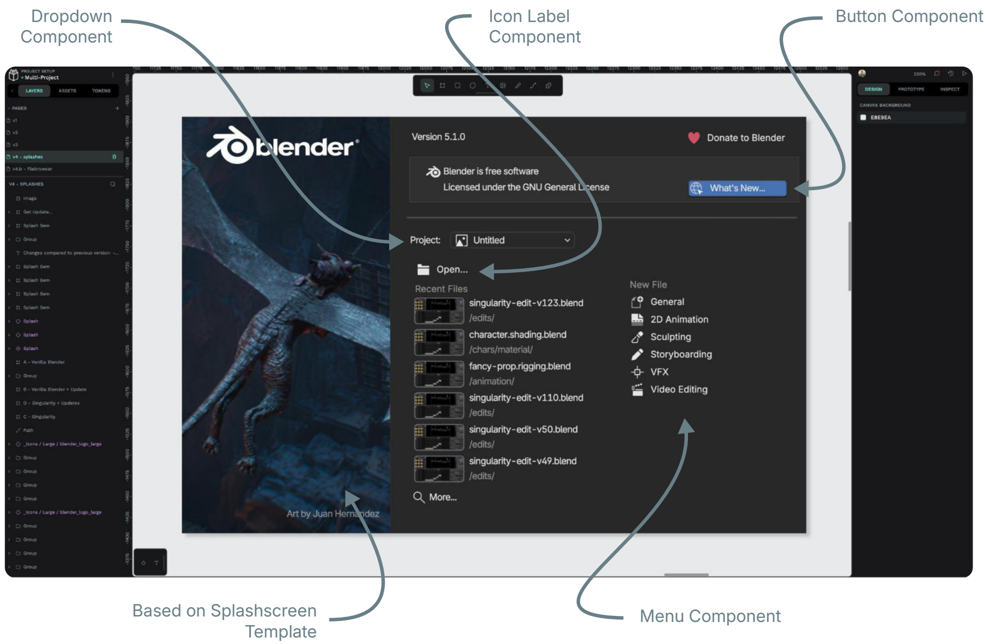


Figure 4.17 | New Blender Splashscreen: one of the many mockups created by a Blender designer (C1) using the design system during user testing

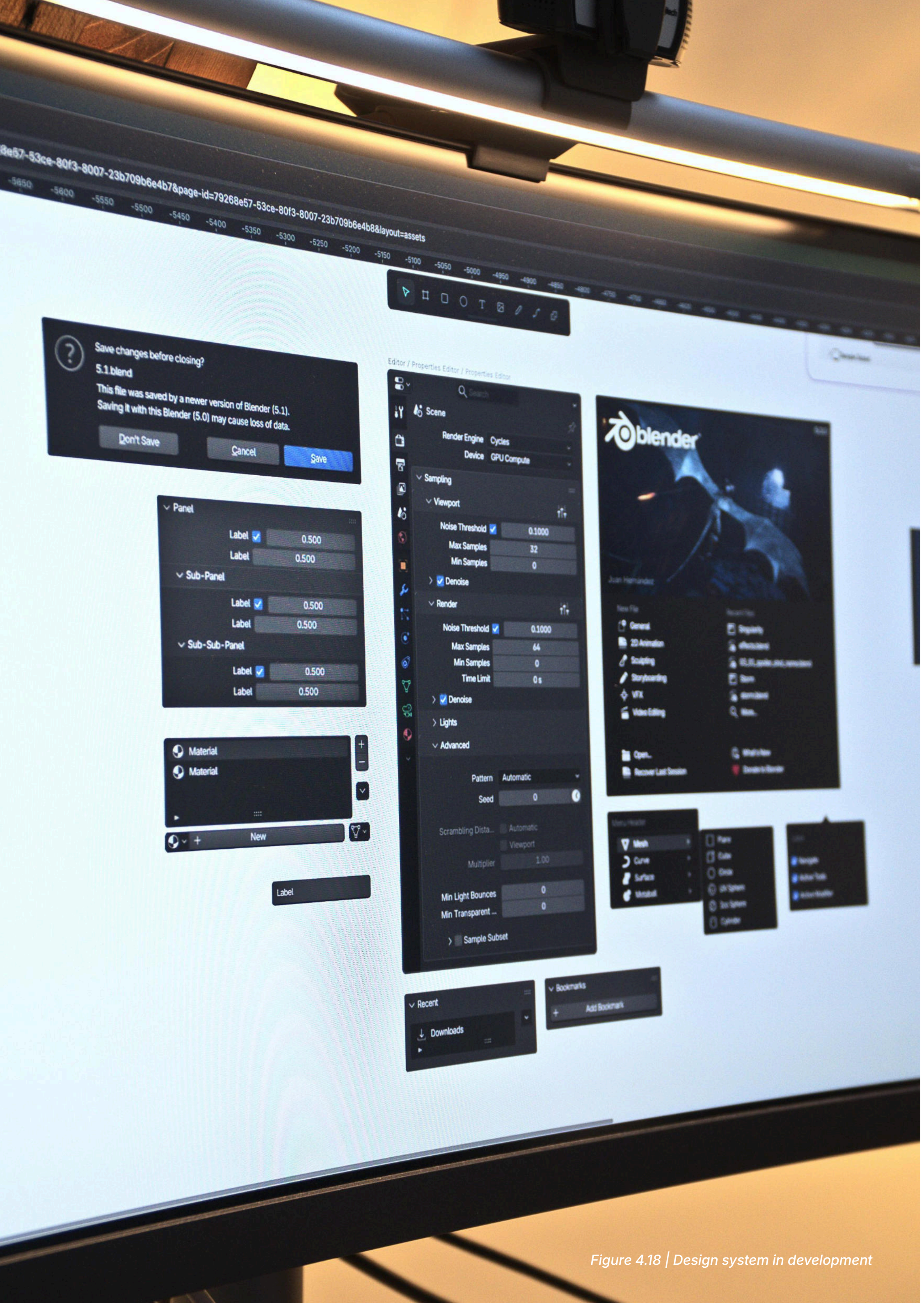


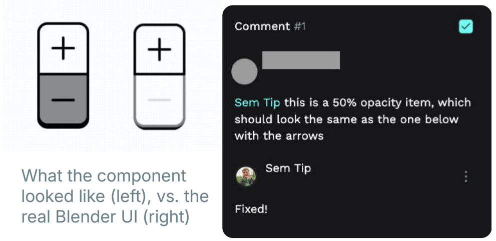
Figure 4.18 | Design system in development

Key Findings for Concept Improvement

- The 6 week production stress test revealed critical insights that helped improve the system. While the core architecture and setup of the system was validated immediately, with users reporting increased speed and enjoyment, several important friction points emerged that required refinement:

01 - Most Common Issue: Visual Inconsistencies

- **The Issue:** Newly created components often failed to match the pixel-perfect reality of Blender, breaking the "Digital Twin" promise. This happened most often with complex, multi-state components where colors (tokens), dimensions and other details were inconsistent. This resulted from a mix of my limited familiarity with certain Blender UI nuances and early-version visual issues that surfaced during component testing.
- **The Fix:** These problems were solved easily and quickly by updating the components using the real UI as a reference together with user feedback. Also new variants and tokens were added to solve the issues. The commenting system in Penpot facilitated this feedback process.

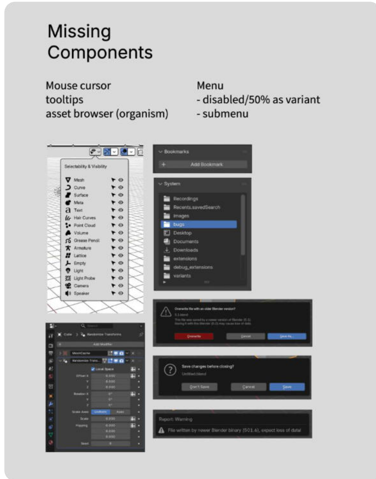


What the component looked like (left), vs. the real Blender UI (right)

02 - The "Missing Component" Trade-off

- **The Issue:** Using the pre-built templates was experienced as very quick. However, users frequently hit "dead ends" where specific components (e.g., specific menus & asset browser) were missing from the library. This forced a decision: stop investing time to build the component (slowing down the immediate task), or to simply fake the component manually (breaking/bypassing the system).
- **The Refinement:** To address this, the "Demand-Driven Expansion" approach was used. Rather than trying to build 100% of Blender's UI upfront (an impossible task), a "Missing Components" board was created. Besides directly validating demand, this made the current demand visible and allowed the system to grow organically based on active project needs, ensuring resources were invested where they added the most immediate value.

"The question is whether to solve the current need, which the system definitely does, or to invest in future needs that might come up. It's a balance." - Test Participant, Blender Designer



03 - Technical Constraints: Theme Switching

- **The Issue:** Users expected to toggle between low and high-fidelity themes instantly within their project files. However, Penpot's technical limitations initially restricted theme switching to the main library file, frustrating users who wanted to preview high-fidelity versions in linked project files.
- **The Fix:** A manual token export/import workflow was used as an interim solution, allowing users to import the design system tokens into their specific project files. This technical workaround worked perfectly and ensured the dual-theme requirement could be met. This issue will probably be fixed with future Penpot updates.

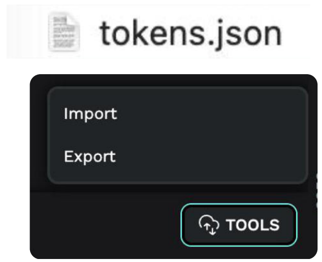


Figure 4.19 | Visuals of concept improvements

Unexpected Validation: A Hybrid Workflow

During the testing, an unexpected but highly valuable workflow emerged. Designers began mixing raw screenshots of the existing Blender interface with components from the Design System.

This "Hybrid Workflow" validates the design system reaching the Digital Twin goal. Because the system's components matched the live software's rendering with pixel-perfect accuracy, designers could seamlessly overlay new UI elements onto old screenshots without a visible "seam."

This ability significantly increases leverage. It allows contributors to propose custom and complex changes without needing to build the entire surrounding interface, drastically lowering the time-cost of creating high-fidelity mockups. Below, Figure 4.20 shows an example of a resulting mockup of this workflow:

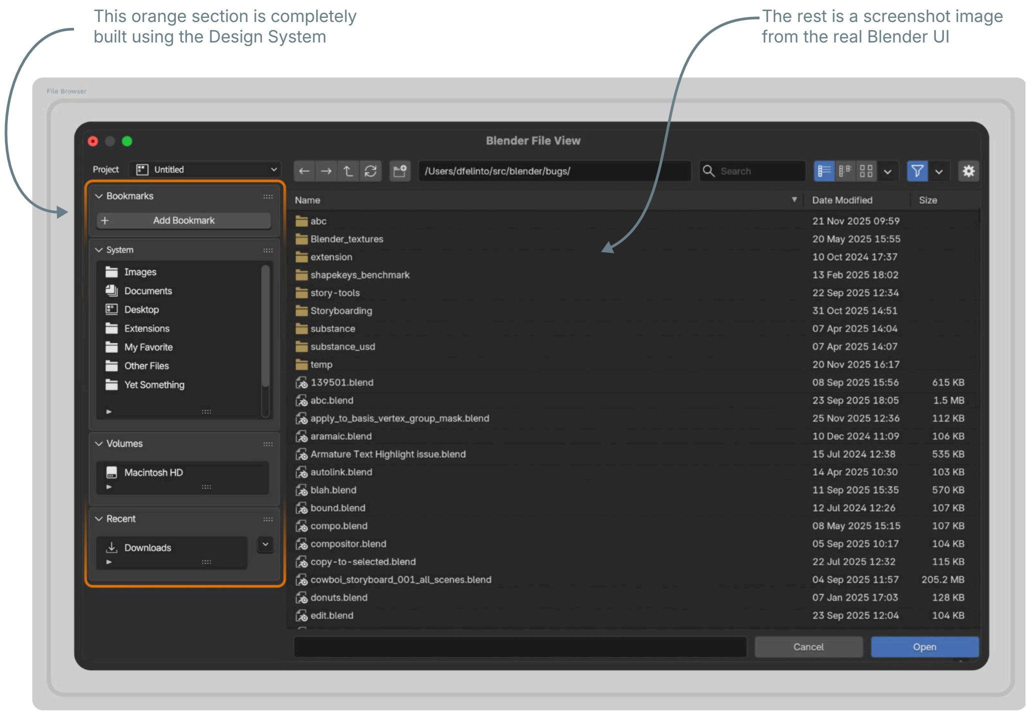


Figure 4.20 Mockup created by Blender designer during user testing, illustrating digital twin concept and new hybrid workflows that speed up the design process. The accuracy of the system allows components to be blended directly with screenshots, indistinguishable from the real software.

Dedicated screenshot components were added to the design system to further support this hybrid workflow:

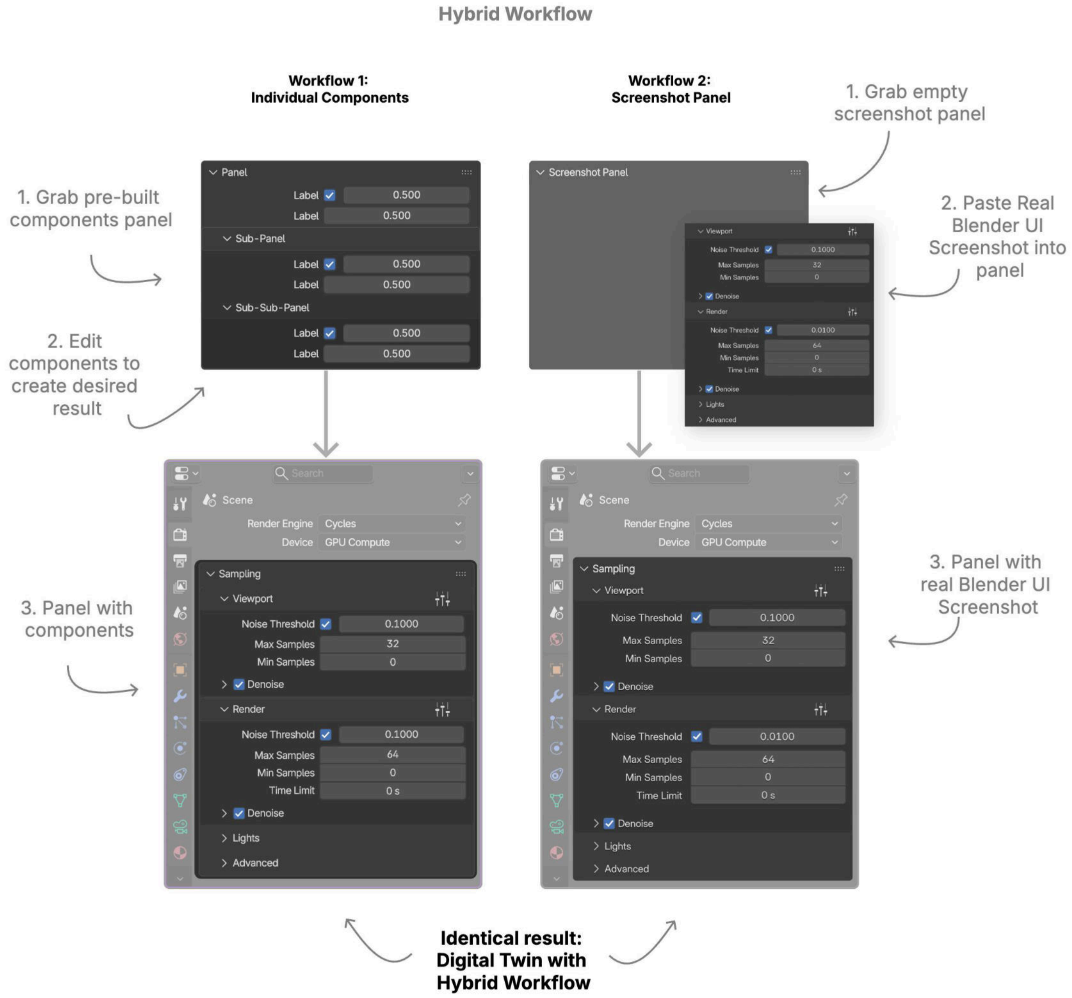


Figure 4.21 | Regular component vs. Hybrid Workflow process steps

This marks the completion of the iterative development phase. Chapter 5 presents the final Design System, its strategic evaluation, and the roadmap for implementation. →

05

The Final Concept

5.1 The Blender Design System: Infrastructure for Leverage

Introducing the infrastructure for Blender's design future

The Design System is a mockup tool designed to optimize a specific part in the Blender development process: the creation of mockup proposals. It bridges the gap between creative design and technical development, serving as a library that transforms inaccessible knowledge into an explicit, shared tool. It is accessible today at design.blender.org.

Democratizing Design

The system addresses the Expertise Paradox by decoupling design quality from individual graphic skills. Blender's Design System lowers the barrier to entry, empowering the broader contributor base, both core developers or community members, to generate high-fidelity proposals. It ensures that the lack of design tools is no longer an excuse for a lack of solving problems.

Use it Today

Already proven in production, the system is hosted live on design.blender.org. On an independently live version of Penpot, it aligns with Blender's open-source values and data sovereignty. It combines a Component Library with embedded documentation, ensuring that design rationale is accessible exactly where the work happens, reducing the reliance on others.

The Hybrid Workflow: Reducing Friction

The system eliminates the cognitive load of the previous , manual "screenshot-editing" workflow. It enables a Hybrid Workflow where real Blender screenshots serve as the canvas, and the Design System's editable components are overlaid. Because the system acts as a pixel-perfect Digital Twin match, any edits blend seamlessly, making the workflow fast and dynamic.

Systemic Value Creation

- **For the Designers (Leverage):** No more repetition. Focus your energy on solving complex UX challenges. Tell better stories with adaptive themes and present your ideas in interactive prototypes that feel and act like the real Blender.
- **For the Developers (Precision):** Removes ambiguity. The system acts as a shared language. Tokens can map 1:1 to code variables. Developers receive proposals that are not just images, but technically feasible specifications. Create your own high-quality designs without needing any help.
- **For Blender (Scale):** A unified, scalable infrastructure that accelerates the development cycle and creates a consistent visual identity without requiring constant supervision, while opening the door for a new generation of contributors.
- **For the Community (Access):** A seat at the table. By publishing the same professional toolkit used by the core team, the Design System levels the playing field, ensuring external contributions are judged on their logic, not their graphic design skills.

It's Just Like Blender: A Digital Twin

Buttons, sliders, and menus have been reverse-engineered from the real Blender UI. When you design here, you are working with components that match the live software with pixel-perfect accuracy. What you see is exactly what developers will be able to build.

Blender Designer Experience: "Have you seen the system?! I've been showing some of my mockups already to colleagues and their reaction is often like "this is just Blender?!" "

Tell Your Story with Adaptive Themes

Communicate the right intent at the right time. Use the High-Fidelity Theme for final specs, or switch instantly to a custom developed "Mockup Mode", a simplified, low-fidelity style that helps you discuss concepts and flows without getting distracted by details.

85 Smart Components

No need to start from scratch. 85 responsive, state-variant components are ready to drag and drop, with more coming soon. They handle the complexity of colors, spacing, and layouts automatically, so you can focus on problem solving.

Figure 5.1 | Final Design System Concept

The Design System Interface

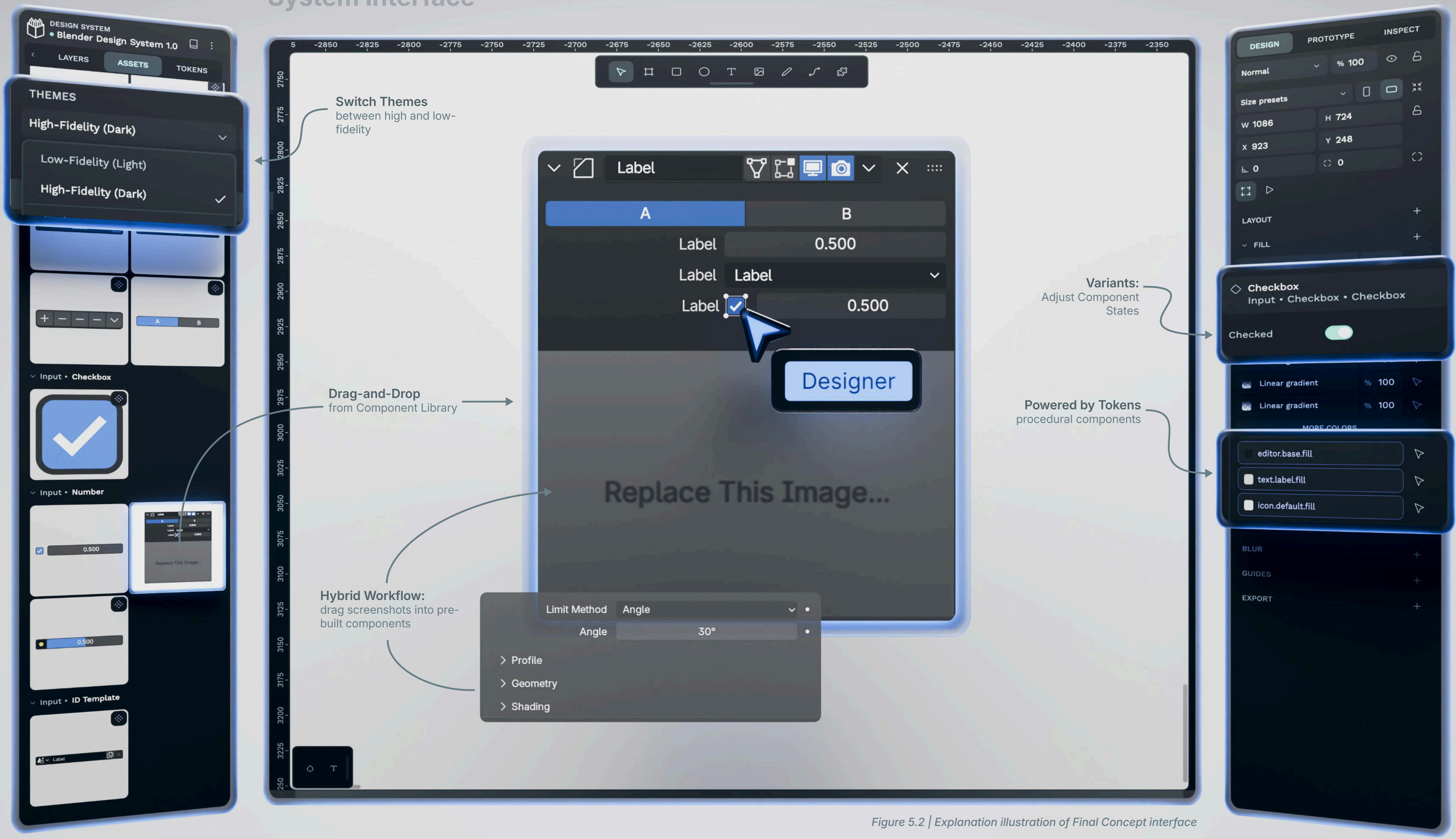


Figure 5.2 | Explanation illustration of Final Concept interface

85 Unique Components Available

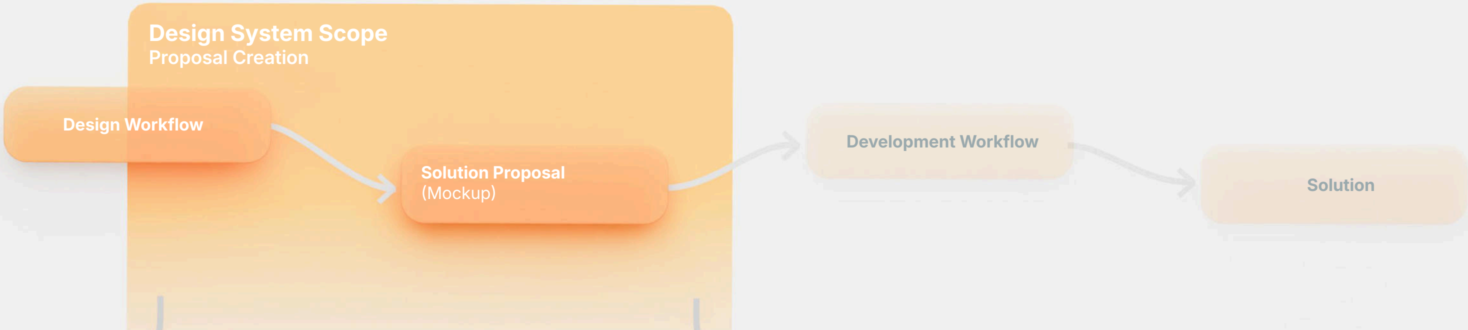
Currently, the Design System contains 85 individual components, with 100 unique variants. Every component is token-driven, fully customizable and compatible with other components. The system is designed to scale, with new components added regularly based on demand.

Figure 5.3 | Overview of individual components developed for Blender's Design System

5.2 Workflow Impact: Optimizing Mockup Creation

While the design workflow encompasses everything from strategy to final proposal, the Design System functions specifically as a production tool for the mockup creation phase. It replaces the inefficiency of manual image editing with a structured, component-based workflow.

The Design System acts as a specific tool within the proposal creation design phase (towards the end). It does not replace strategic problem framing or technical implementation (development). Blender's Designers consider it (and should continue to do so) to be "one tool in the toolbox".



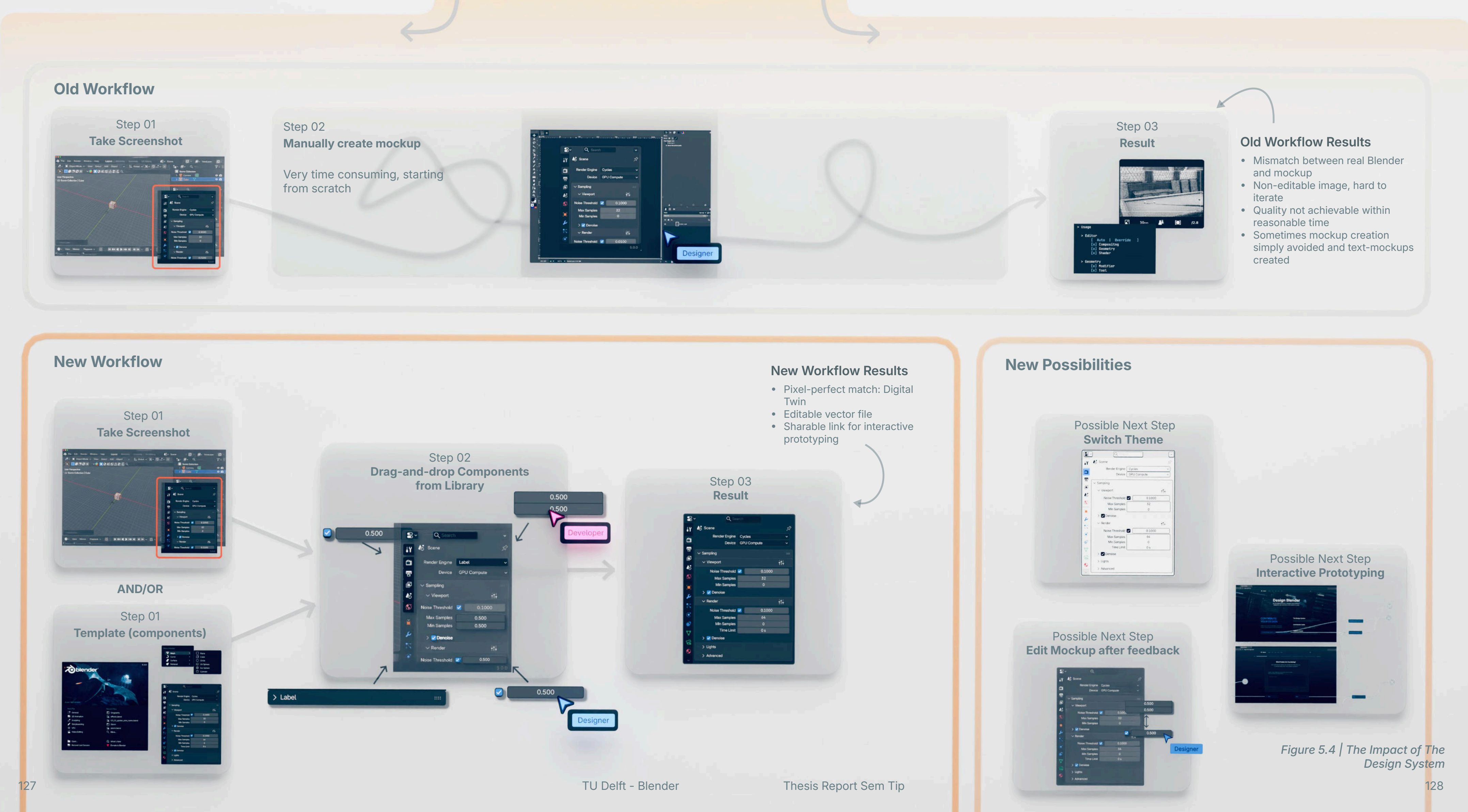


Figure 5.4 | The Impact of The Design System

5.3 Implementation Strategy

Following the insights from 4.4 Concept Prioritization, the roadmap below in Figure 5.5 was created. This roadmap is meant to define priorities and offer guidance. The timing isn't meant to be exact, it's a flexible plan that can be adjusted as needed.

This sequence is strategically critical: as explained in 4.4 and validated by Blender's leadership (see 6.3 Viability), the organization must secure internal Workflow Leverage first by establishing shared tools and infrastructure, before attempting to manage Expertise Quantity from the external community. The roadmap guarantees that the ecosystem is ready to scale efficiently when the Community Portal (Phase 3) finally opens the doors for external input by first strengthening the basis with the Design System and subsequent Proposal Guide (Phase 2).

The Phased Approach

- **Phase 1 (Immediate):** Adoption of Concept 1 (Design System) by the Core Team (C1/C2).
- **Phase 2 (Mid-term):** Implementation of Concept 2 (The Design Proposal Guide/Method).
- **Phase 3 (Long-term):** Launch of Concept 3 (The Community Portal) to open the floodgates for external contribution.

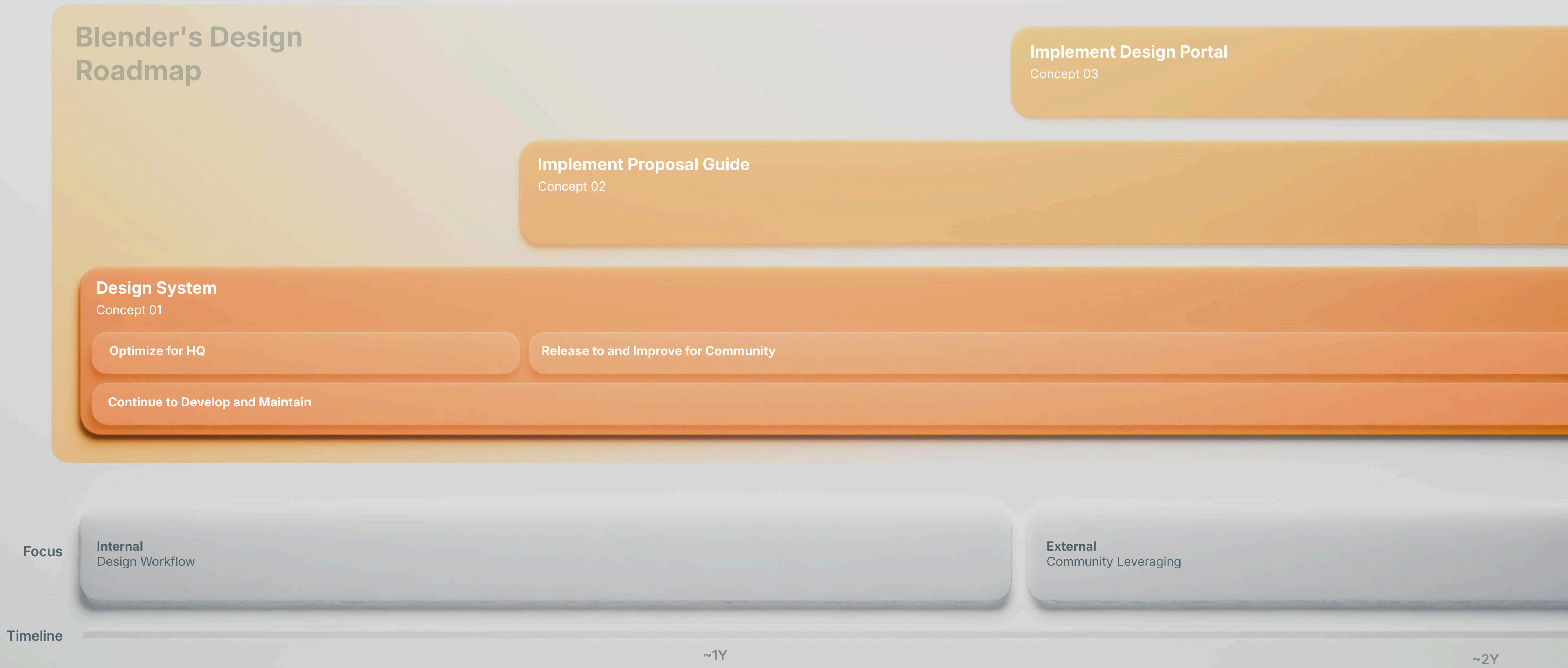


Figure 5.5 | Implementation Roadmap

5.4 Concluding Recommendations

This overview presents the final recommendations for Blender based on the research of this thesis. The recommendations listed here also don't necessarily mean that Blender isn't working on similar elements themselves. In case they do, this section serves as further confirmation.

Elevate Design Culture

Developer-Centric Culture and the Expertise Paradox

01 Emphasize and Standardize Design Practices

Infrastructure alone cannot shift a historically developer-centric culture, design must become visible in Blender's daily rhythm.

- **Recommendation:** Create a stronger design culture, identity and norm at Blender. Make design visible in regular and dedicated design discussions. Set a standard for design and create accessible documentation where anyone can find Blender's design philosophy and guidelines. Use dedicated design communication channels and issue tags, to separate design rationale discussions from technical implementation details. (During this thesis Blender has made great progress in this!)
- **Value:** Create design-development synergy by tackling current lack of design. Shifts design from an ad-hoc task to a cultural norm. It improves the shared understanding needed to solve complex challenges.

02 Make Design More Accessible

The Expertise Paradox creates a bottleneck where non-designers want to contribute to design but are blocked by Inaccessible Design Knowledge and a lack of skills. They need help from designers who can't always help. Meanwile, designers feel they must defend their decisions against non-designers.

- **Recommendation:** Empower non-designers (developers) to design. They do not need to be designers, they need autonomy. By providing pre-built, logic-based components, you remove the barrier to entry and allow non-designers to contribute higher quality proposals independently.
- **Value:** Allows non-designers to contribute high-quality proposals independently, granting them autonomy and reducing load on designers.

Improve Design Workflow

Inaccessible Knowledge and Designer-Developer Interaction

03 Improve Design Workflow with Tools

The current reliance on manual and unstandardized processes creates friction and inefficiency, preventing contributors to spend time on high-value problem solving. User testing revealed that using the design system is a massive leverage point for efficiency.

- **Recommendation:** Let the workflow create space for user-centric problem-solving. Standardize the workflow of design and raise the workflow quality by implementing workflow tools that support and guide. Make the design workflow and the tools customizable (hybrid workflow) and highly autonomous, and eventually externally accessible.
- **Value:** Improve efficiency and autonomy by shifting the focus from the manual mockup building to more important problem solving. Standardizing flexible methods like the "hybrid workflow" allows non-designers to communicate design intent clearly and accurately while staying in creative flow.

04 Maintain Design System (and Other Design Tools)

The Design System is the Single Source of Truth. It must remain as close to a 1:1 reflection of the live Blender version as possible. Then it provides the most value. Assign a specific maintainer to ensure stays aligned.

- **Recommendation:** Assign clear ownership of the Design System to a specific internal stakeholder (or rotating role). This individual is responsible for building needed components, updating tokens and ensuring the "Digital Twin" remains accurate.
- **Value:** Prevents the inaccessiblility challenges from returning and ensures the tool remains a reliable single source of truth.

Expertise Scaling

Scarcity of design expertise, Design Capacity Framework

05 Increase Amount of Designers, But Hire For Leverage First

After having implemented the Design System as a stable, efficient infrastructure, Blender will eventually reach a point where optimized workflows alone can no longer scale design presence and output. To tackle this, the next step is to increase the amount of designers to address the scarcity of design contributions.

- **Recommendation:** Recruit dedicated Product Designers who can now immediately be productive because the workflow is optimized. They can further scale the significance of Blender's design workflow and design culture.
- **Value:** This capitalizes on the investment made with the Design System. The combination of increasing the amount of skilled contributors (Quantity) operating within a high-leverage workflow (Leverage) is the best recipe for design success.

06 Leverage Community Design Expertise

Blender's greatest asset is its massive community. Leverage the scale of their expertise once the internal process is solid.

- **Recommendation:** Once ready, use the Design System as a self-service onboarding kit to guide and filter noise. Implement the Design Portal (Concept 3) only when the HQ team has the capacity to manage the influx of extra contributions and when Blender's the only limiting factor is the amount of contirbutors. By embedding rules and guidelines directly into the tool, the learning curve for contributing design is drastically lowered.
- **Value:** This scales the "Quantity" lever of Design Capacity. It changes the community from a source of friction into a scalable workforce of design contributors, effectively democratizing the design process alongside the development process.

06

Concept Validation

6.1 Validation Setup

Validation Objective

To conclusively validate the effectiveness of solving the challenges, fitting the requirements and addressing the stakeholder's needs, the concept has been validated. A dedicated concept evaluation session was conducted with key stakeholders. The goal was to validate the impact on the workflow against the design criteria established in Chapter 3 and to ultimately answer the question whether the design system and the proposed roadmap improve the design workflow.

Validation Criteria

Thus, following the research and insights from Chapter 3 and 4, the validation measured success against four core requirements:

- **Does it solve the challenges:** Does it empower design capabilities (especially for non-designers)? Does it improve collaborative workflows?
- **Does it build Design Capacity:** Is the new workflow measurably improved? Does the concept improve Design Capacity through workflow leverage?
- **Technical feasibility:** Does the concept hold up in reality?
- **Is the concept viable:** does it align with Blender's mission? Will the concept last or be able to adapt in the future?

Validation Setup & Limitations

Validation Participants

The test focused on C1 and C2 contributors (Designers and Developers) as they are the primary users of the concept. Since the to-be-validated criteria are subjective, the methodology utilized a mixed-methods approach to quantify user perception. The validation counted 5 total participants, 4 of which took part in the questionnaire and interview, and 1 took part in a single dedicated validation interview. **The interview information can be found in Appendix B & F. In the charts, participant 1 (P1) and P2 are non-designers, while P3 and P4 are designers.**

Validation Method & Data

The testing approach prioritized qualitative depth supported by indicative quantitative data. Quantitative data was collected using Likert Scale Questionnaires where Participants rated 12 statements (S1-S12) on a 1-7 scale from strongly disagree (1) to strongly agree (7). **The full overview of the statements and raw data (all scores and quotes) can be found in Appendix F.** Following each statement, a brief semi-structured interview was conducted to collect qualitative data and gain a deeper understanding. The results mean score was M = 6,25 with a standard deviation of SD = 0,88.

Limitations

Limitations include the small participant sample size (n=5) and the short timeframe of the validation (~2 weeks). While n=5 is statistically small, these participants represent the entirety of the active HQ design/development core team. Although the participants represented the core target users, the data does not guarantee broad representativeness across the entire Blender team, and neither does it predict long-term retention and adoption of the concept.

Figure 6.1 | Blender Designers using the developed Design System for a new feature in Blender

6.2 Feasibility

Definition

Can it be done?

To improve the design workflow, it must be feasible for the design system to realize the digital twin promise and be compatible with the autonomous, open-source work culture.

Insights

The Design System has been built and used for many ongoing projects, confirming its technical feasibility. The technical feasibility of independently hosting and maintaining the tool was proven during the thesis timeline. Blender successfully deployed its own self-hosted version at design.blender.org,

The quality of the Design System results are high enough to blend seamlessly with real Blender software screenshots, validating the 1:1 'Digital Twin' accuracy. As a designer and Design System user wrote:

"Have you seen the system? I've been showing some of my mockups already to colleagues and their reaction is often like 'this is just Blender?!'" (suggesting that Design System mockups look exactly like the real Blender software) (P3)

Participant P3 explicitly validated this workflow innovation:

"Components like the 'Screenshot Panel' mix the best of both worlds: a hacky quick screenshot interleaved with new or existing components." (P3)

Feasibility Limitations & Concerns

The main technical risk is the dependency on Penpot. As a designer noted:

"There is also the tiny gloomy scenario of the underlying technology (Penpot) to not continue to be developed... also, bugs directly impact the experience". (P3)

To address this, Blender has already been self-hosting Penpot on their own servers and is actively contributing back to the Penpot open-source project to fix bugs to secure the Design Systems future.

Feasibility Conclusion

The concept is technically feasible, proven by its deployment on design.blender.org and the extensive use of the concept. The 1:1 alignment with the real Blender software confirmed the realization of the Digital Twin promise.

6.3 Viability

Definition

Will it survive on a longer term?

To ensure the concept survives it should strategically align with and advance Blender's mission and provide sustainable value without imposing an unmanageable maintenance burden.

Quantitative Results

Figure 6.2 | Statement 10/11 scores per participant - See Appendix F for all raw data

Insight 01 - Strategic Alignment & Adoption

Statement 10 ("I'm certain the concept will continue to be used/developed") scored an average of **6,25/7** (see Figure 6.2), with participants emphasizing they think the concept has a lot of potential, but also faces maintenance concerns. This score is supported by the fact that the system has been adopted and integrated into the production workflow during the thesis timeline The qualitative data indicates a reason why this happened: the tool could serve as a practical example of a strategic desire to elevate design. A designer noted that the tool is a cultural signal:

"Investing time on [design] tooling shows that we are willing to ... treat design with as much importance as we want everyone else to treat it." (P3)

Crucially, this bottom-up adoption is matched by top-down strategic validation. During the validation interview, Blender's leadership confirmed that the roadmap aligns with their vision. The CEO explicitly validated the decision to prioritize internal infrastructure (Design System) over community expansion:

"We should work on first even setting a workflow standard at Blender and then improve it." (CEO)

Interpretation

With the Design System already integrated into the team's workflow, it has removed (or having already largely overcome) the potential barrier of "adoption friction". Participants expressed confidence in the tool's longevity, while also being aware of concerns (see Viability Limitations). Besides being a useful tool, the data indicates that the concept is viable because it connects the team's desire for better design standards and the leadership's vision.

Insight 02 - Financial Sustainability vs. Maintenance Cost

The developed concept relies on existing open-source infrastructure (Penpot) and requires zero licenses or fees, eliminating a primary financial risk of unjustifiable costs in Blender's OSS environment. **However,** while financial costs are zero, human costs are not. The validation interviews highlighted a tension between using the tool and maintaining it. While users enjoy the speed (See 6.4 Desirability), they recognize the hidden cost of maintianing the concept. A designer highlighted this operational risk:

"It (the design system) can sometimes take time away from doing the actual work"

Another participant expressed concern about the reliance on specific individuals:

"My concerns are with maintenance... if this becomes a tool only for the design team, how much it will be worth spending on maintaining and expanding it? I I think it may still be worth it, it is always a balance between investing on tooling and doing the creative work of design" (P3)

In contrast, the data suggests that this "maintenance cost" might be offset by efficiency gains in speed (See 6.4 Desirability). In addition, statement 11 ("Community proposals will be easier to review") scored an average of **5,75/7**, indicating that the team expects the system to reduce at least some of the burden of reviewing external contributions. One participant explained this clearly:

"I think the design system allows for good proposals to be made better, thus easier to review and consequently implement. But it doesn't make a bad proposal better necessarily. It may help to more easily dismiss an incomplete proposal." (P3)

Interpretation

The data indicates that concept creates a shift in where time (the main cost) is spent. While it requires time to potentially expand or maintain the library (a new cost), the system speeds up the design workflow (reduce an existing cost, see 6.4) and is expected to speed up community reviews (potentially reduce an existing cost). The concept is financially viable forever due to its open-source nature. However, its operational viability depends on the approach and investment in maintenance. This connects to the recommendation (5.3) to assign a maintainer role, to make sure the "human cost" is managed well.

Viability Limitations & Concerns

As mentioned, the most significant viability risk is maintenance. It's important to assign clear maintenance responsibility (see 5.3). Additionally, regarding community contributions, there is a concern that the system might prioritize visual looks over other critical design considerations. Yet, this is somewhat contradicted with some expectations that bad reviews would be easier to dismiss (see Insight 02 on the left).

Finally, the short validation timeframe (6 weeks) cannot guarantee permanent adoption or prevent a potential reversion to old workflows. However, the system's and Penpot's flexibility, described by a designer as a "sandbox", mitigates the risk of obsolescence by allowing it to evolve alongside Blender's changing needs.

Viability Conclusion

The Design System is expected to be viable. Strategically, it aligns with leadership's vision to "raise the quality bar," validated by its immediate adoption and continued use in production. Operationally, while maintaining the system introduces a new time cost, the data suggests this is offset by the immediate efficiency gained from faster workflows (see 6.4 Desirability), higher-quality proposals and smoother community reviews.

6.4 Desirability

Definition

Does it address the stakeholder's values and needs?

To improve the design workflow, a desirable design system must enable critical "problem-solving" and take care of the less critical. To be desirable, the workflow should be faster and a more enjoyable experience. To solve the expertise challenges the concept should empower non-designers.

Quantitative Results

Figure 6.3 | S1-3, S5, S8, S9 scores per participant - See Appendix F for all raw data

Insight 01 - Shifting Focus from Construction to Solution (Efficiency)

Statement 01 ("Faster than old method") and Statement 02 ("Focus on creative problem solving") both scored high agreement: avg. **6,0/7** and **6,25/7** respectively (see Figure 6.3). All developers rated the speed (S1) a perfect **7/7**, with designers rating it a **5/7**. The qualitative feedback highlights why this speed matters, it is about "flow". A developer noted that the pre-built library removes the barrier of starting from scratch and the distraction of technical details, while a designer noted the Design System doesn't break flow:

"Without a design system I have to start from scratch, and spend a lot of time on basic things and details." (P2, Developer)

"It doesn't break the flow of creating and re-iterating." (P3, Designer)

Simultaneously, a designer noted that this speed changes the nature of the work itself, moving it away from a handover process:

*"It actually contributes with helping making mockups ... and it *really* helps with the iteration process. This makes the design less of a mockup handover process and more a collaborative exploration." (P3)*

Interpretation

The convergence of high quantitative scores for speed with qualitative reports of "collaborative exploration" indicates that the system is perceived to accelerate the workflow. The concept is expected to reduce the load of manual mockup construction, allowing users to stay in a the important problem-solving flow. The results indicate that the system addresses the disciplinary Des-Dev gap by making the design process more "collaborative". Non-designers experience the most significant speed improvement. This could be the case because it supports their absence of design skills, and because they were less involved in developing the Design System and thus didn't perceive that time cost.

Insight 02 - Democratizing Design (Autonomy)

Statement 03 ("I feel confident constructing UI without help") scored an average of **6,0/7**. Notably, regarding the quality of the output, non-designers (P1 & P2) gave a perfect score of **7/7** for Statement 05 ("I feel like I'm able to create a higher quality proposal"). The qualitative feedback indicates that this increased confidence stems from the system's ability to separate design intent from graphic design skills. By providing pre-built components, the system guarantees visual quality regardless of the user's skill. A designer (P3) noted that this accuracy actually improves decision-making capabilities, too:

"The high-fidelity mockups produced with this system are super close to Blender's UI. In fact, they allow for fine-tuned design decisions that are typically constrained by Blender's existing UI toolkit." (P3)

This theoretical confidence translated into immediate practical adoption. During the test period, stakeholders voluntarily applied the system to real-world tasks outside of the test scope. As one designer concluded regarding the system's utility:

"It's the question is whether it solves the current need, which it definitely does." (P3)

Participant P1 (Non-designer) offered a realistic view that the system isn't magic, but a guide:

"Making mistakes with UI layout [is still possible]... but the Design System nudges them in the right direction." (P1)

Interpretation

The combination of **7/7** fidelity scores from non-designers and the immediate voluntary adoption by the team indicates that the Design System provides real utility. It also reduces non-designer's dependence (waiting for a designer) and increases their autonomy (non-designers can do it themselves). Therefore, the data suggests it might address the Expertise Paradox (Challenge 7). By embedding design rules into the tool, the results indicate that the system democratizes the ability to contribute to the design workflow, thereby probably increasing the total Design Capacity without requiring additional headcount.

Yet, a critical nuance must be made. Providing tools does not replace design skill. There is still risk that non-designers might create technically feasible but user-unfriendly layouts. However, The Design System attempts to mitigate this by having the pre-approved components available as gaurdrails. It shifts their contribution from 'inventing UI' (high risk) to 'assembling pre-approved UI' (lower risk), but it does not remove the need for a final review by a skilled designer.

Insight 03 - The "Sketching" Experience (Engagement)

Statement 08 ("The process feels more like sketching than engineering") received an average score of **6,0/7**. Statement 09 ("The process... makes me enthusiastic") scored a near-perfect **6,75/7**. The qualitative responses indicate that this enthusiasm stems from reducing the technical friction usually experienced with translating an idea into a digital mockup. One designer (P3) described how the high fidelity of the components provides immediate gratification, and maintains the creative momentum usually lost:

"The only downside is that it doesn't have the same fun of doodling on a white board. But the moment the initial ideas are sketched out, then the computer step is less of a dread and almost just as fun." (P3)

The designer also emphasized that the speed and polish of the system changes the emotional relationship with the task:

"It feels like you are dealing with a finished piece of software... There is something magical about being able to quickly build an UI, and adjust the naming... It is overall very exciting and fun to use." (P3)

Interpretation

The data suggests a positive shift in the work experience. The high score on fun and enthusiasm indicates that the tool potentially acts as a motivator rather than just a utility. This potential is critical for the context of Open Source Software. As mentioned in Chapter 01, OSS contributors are primarily driven by intrinsic motivation and "flow." The results might show that the system changes the design workflow into an engaging, "sketch-like" activity, thereby aligning the design process with the core cultural drivers of the Blender ecosystem, potentially increasing the likelihood of sustained voluntary (design) contribution.

Desirability Limitations & Concerns

While quantitative data on time-savings is missing, qualitative feedback indicates improvements in speed, problem-solving, enjoyment, and output quality.

Participants also noted that "in practice there are also design challenges that go far beyond graphic design and mockup creation" (e.g. strategy). The concept might not be able to directly help with those tasks, or worse, make the team unintentionally focus too much on mockups. Awareness of this possibility is the first step in mitigation.

A concern raised by multiple participants is the risk of "Inside the Box" thinking. Users might only design what is currently easy or possible within the library, focusing less on new explorations. To mitigate this, it is encouraged to mix the Design System workflow with 'old' workflows including sketching to maintain the speed of the system without losing the flexibility to "break" the rules when innovation is required. Designers reported that they see the Design System as "just one tool in the toolbox", indicating that they also rely on other methods.

Desirability Conclusion

The data shows the Design System is highly desirable, as it changes mockup creation from technical friction into an enabler of creative flow. By reducing the barrier to entry, particularly for non-designers, it addresses the Expertise Paradox, empowering the wider team to produce high-fidelity proposals autonomously. The immediate adoption and high satisfaction scores demonstrate that the system creates the necessary Workflow Leverage to shift the team's focus from manual construction to valuable problem-solving.

6.5 Validation Conclusion

Concept Impact

Satisfying the concept requirements set in Section 3.10, the validation confirms that the Design System is an effective, feasible, and desirable intervention that has been quickly adopted. The concept improves the mockup design workflow and successfully helps shift the focus from technical execution to creative problem-solving. It is expected to address the Expertise Paradox challenge by empowering non-designers and accelerating their workflow (rated **7/7** by non-designers).

The validation confirms that the Design System not only delivers value for the design workflow itself, but also feels like a shift towards a more desired, design-focused work culture, addressing the "scarcity of design expertise" problem. Statement 12 ("The system positively contributes value") received an average score of **6,5/7**. Furthermore, leadership identified the system as a mechanism that "raises the quality bar" and shifts Blender's historically engineering-driven culture toward a more balanced design one. As a designer articulated:

"Investing time on [design] tooling shows that we are willing to ... treat design with as much importance as we want everyone else to treat it." (P3)

Acknowledgement

It is important to acknowledge that the Design System is a workflow tool, not a replacement for design strategy. It solves the inefficiency of how design proposals are made, but it does not magically solve what gets designed. It eliminates the technical barrier to entry for contributors, but the broader challenge of defining a long-term product vision remains a strategic task for Blender's leadership. This tool is an enabler for that future work, not the solution to it.

Validation Conclusion

Answering the initial research question, the intervention successfully showed potential to increase Blender's ability to leverage design expertise through empowerment, increased speed and output quality and by increasing accessibility. Crucially, it improved the working experience and enthusiasm towards design, with the team describing the new process as "exciting" and "more like sketching than engineering." It shifts the focus from "graphic design" to "problem-solving". However, the primary concerns are the future maintenance burden and the risk of 'inside the box' thinking. Ultimately, the validation suggests the concept's ability to increase Design Capacity leverage within the team.

07

Discussion & Conclusion

7.1 Discussion

Summary of Key Findings

This thesis addressed the scarcity of design expertise within the Blender ecosystem, an issue rooted in a historically technology-first culture. It resulted in inaccessible design knowledge and friction between designers and developers. The thesis investigated whether a practical intervention could resolve these inefficiencies and enhance the organization's overall Design Capacity by leveraging design expertise.

The findings demonstrate that the implementation of a Design System serves as a successful workflow solution. The results validate that by prioritizing Workflow Leverage, specifically through a Design System that empowers non-designers to independently create high-fidelity proposals, the organization can address the Expertise Paradox challenge and shift the culture from developer-centric to a more balanced, design emphasized one.

Interpretation of Findings

Patterns and Relationships

The validation data reveals a clear relationship between the availability of structured design infrastructure and the autonomy of non-designers. The introduction of the Design System correlated with a reduction in the graphical cognitive load required to create UI proposals and more mental space to focus on problem solving. By creating ready to use templates, the system functioned as a successful tool for non-designers, which reported an increased perception of autonomy. Addressing the Expertise Paradox, the design of the concept with modular components provided the necessary consistency without compromising individual agency. It proves that in an open-source environment, structure does not have to come at the cost of autonomy, when implemented as enabling infrastructure, it actually enhances it.

Furthermore, the findings confirm the expected relationship between Workflow Leverage and Design Capacity: the data shows that improving the workflow (Design System) improved the use of expertise of existing contributors, through an increase in speed and working experience. This illustrates the framework's suggestion that for organizations like Blender, Workflow Leverage is a prerequisite for Expertise Quantity.

Results Expectations

The results largely met the initial research expectations, particularly regarding the hypothesis that a Digital Twin approach would facilitate adoption in a developer-centric culture. As anticipated in the Concept Development phase, providing pre-built, pixel-perfect components empowered developers to create high-fidelity proposals independently, indicating that the prior scarcity of design contributions could have also been caused by an infrastructure problem rather than purely a lack of capability (their own skill). The immediate adoption of the system into the daily workflows confirmed the expectation that a tool respecting the technical reality of the software would be embraced rather than rejected by the team. By reframing design assets as systematic, token-based logic, essentially making design feel 'code-like', the system bypassed potential cultural resistance. It made design respectable and accessible to a OSS developer-centric culture that values distributability, modularity and logic.

Unexpected Results

While the adoption of the system was expected, the spontaneous emergence of a hybrid workflow merging the old and new design workflows was a significant, unanticipated finding. During testing, users began overlaying components (new method) directly onto screenshots of the existing Blender interface (old method) to create seamless mockups. This behavior was not prescribed in the initial concept but highlights an important characteristic of designing for software: speed and creative flow are often prioritized over purity. This unexpected usage pattern serves as a strong validation of the system's technical fidelity, since the components blended invisibly with screenshots, and suggests that flexibility in tool usage is important for adoption in complex, established OSS projects.

Alternative Explanations

While the improved workflow efficiency points to the Design System, it is important to consider alternative explanations for the high adoption rate and validated user experiences:

- **My Involvement:** As a participant-observer who was actively embedded in the team, the positive reception may be partially influenced by the direct support and enthusiasm provided by me, rather than the standalone merit of the tool (The Hawthorne Effect).
- **Novelty Effect:** The enthusiasm might reflect the novelty of having any dedicated design tool after years of having none, rather than the specific superiority of this particular system. However, the specific validation insights citing "speed" and "reduced friction" suggests the system solved genuine structural problems beyond simple novelty.

Ultimately, regardless of the precise influence of these external factors, the resulting shift towards a more efficient and autonomous design workflow represents a positive improvement for the organization.

Figure 7.1 | All the Blender splashscreens over the years, the Blender office in Amsterdam

Implications

Relevance and Implications

The implications of this research extend beyond the immediate improvements observed at Blender. By successfully implementing a Design System in a historically developer-centric environment, this thesis demonstrates that design expertise does not need to be centrally structured to be effective. For Blender, the implications are operational and strategic: the organization now has the infrastructure to scale its design output without having to hire. On a larger scale, this thesis can function as a blueprint for other OSS organizations. It suggests that the path to integrating design in OSS lies not in forcing corporate design processes onto contributors, but in democratizing the tools of design and transforming design from a specialized service into an accessible, infrastructural capability.

Contextualization within Existing Literature

These findings align closely with recent literature on software development in open-source ecosystems, while adding specific depth to the understanding of design integration. Specifically, the success of the Design System confirms the work of Zhang et al. (2025), who argue that establishing shared vocabularies is critical for mitigating the misalignment between static design artifacts and dynamic code. By implementing the design system, the thesis demonstrates it functions as the connecting object to address this friction.

The research also supports Sack et al. (2006) regarding the social construction of influence in OSS. Sack argues that influence is derived from technical expertise, this thesis demonstrates that by embedding design decisions and expertise into a technical tool (the Design System), design work gained legitimacy in the organization, perhaps functioning as a form of 'soft governance'. The intervention elevated the status of design within the technical hierarchy, bypassing the cultural resistance often faced by purely aesthetic contributions.

However, this research adds nuance to the general understanding of design workflows. While general theory often assumes a linear design process, the research insights and developed concept suggests that in OSS ecosystems, effective integration is not about replacing existing processes with clean design methodologies, but about layering flexible design tools over the existing reality.

Finally, this thesis diverges from traditional design management theories (e.g., Best, 2006) which often prioritize top-down strategy to explain design maturity. Instead, the Design Capacity Framework suggests that in decentralized OSS cultures, top-down mandates are less effective. Instead, the concept implementation shows that bottom-up infrastructure is a key driver of organizational change. In an environment without a command and strong hierarchy (OSS), perhaps infrastructure and workflow tooling becomes the de facto management strategy.

Contributions

This thesis made contributions on three levels:

- **Theoretically:** It introduces the Design Capacity Framework, a contribution to organizational theory in decentralized ecosystems. Traditional models like the Danish Design Ladder or Design Maturity matrices rely on hierarchical decision-making structures that do not exist in decentralized OSS ecosystems. While those traditional models assume top-down management to drive design integration, this thesis shows that in OSS, the critical driver is bottom-up infrastructure. This thesis introduces a new, decentralized lens for analyzing and improving OSS ecosystems across the two axes of Expertise Quantity and Workflow Leverage. By analyzing an organization's current position, deliberate steps can be made towards improvement. It offers a new insight: that in decentralized ecosystems with a partly structured core team in an office, solving problems in Workflow Leverage (infrastructure) is a necessary precursor to successfully scale Expertise Quantity (external community contributions).
- **Practically:** The thesis delivered a fully functional, open-source Design System now implemented within the Blender organization and used daily. In addition, the thesis analyzed the Blender organization and provided insights and knowledge into how they operate with concrete recommendations for improvement. Lastly, while studying the workflow and building the implementation, concrete contributions to ongoing projects were made to experience and test design workflows.
- **Regarding the general Open Source domain:** This thesis offers the insight of OSS organizations to reach their full potential, they must evolve beyond code-first contribution. This thesis shows that democratizing the design workflow tools is the distributable and culturally-aligned way for democratizing the design process itself and opening the doors to the larger community.

Limitations

This study concludes that a Design System effectively improves workflow leverage within the specific context of Blender, a large and well-resourced open-source organization. However, as a single-case study, these findings are deeply rooted in Blender's unique structure, which includes a centralized office and paid core developers. Therefore, the results may not be fully generalizable to smaller, decentralized, or purely volunteer-driven open-source projects that lack these organizational resources structure.

While the immediate impact on the core Blender team has been shown, this thesis represents a single case study over a six-week testing period. The long-term sustainability of the Design System relies on continued maintenance and cultural commitment, which cannot yet be fully measured. Furthermore, the secondary goal of leveraging the external community (C3) through the proposed Design Portal remains a future phase of the roadmap, dependent on the stability of the foundation built here.

My role as a participant-observer can introduce potential biases which may have influenced the high adoption rate. In addition, the concept's dependence on the relatively young and to-be-industry-adopted Penpot software introduces some uncertainties regarding future availability and development.

Despite these constraints, the results remain valid for answering the research question. The spontaneous emergence of new workflows building further on the concept demonstrate that it provided genuine utility beyond my influence. The validated increase in autonomy, efficiency and the accuracy of the components confirm that the intervention successfully addressed the most important inefficiencies, regardless of the limitations.

Recommendations

To sustain the gains in Workflow Leverage, Blender must transition the Design System from a validated experiment to an embedded tool. For Blender to transition from a historically developer-centric culture to a scalable design organization, a phased implementation roadmap is recommended. In addition, the recommendations from section 5.3 propose a three-pillar strategy: Elevating Design Culture by standardizing practices and empowering non-designers with accessible tools; Improving Design Workflow through a maintained Design System that acts as a single source of truth and is an accurate digital twin; and Scaling Expertise by involving external community integration when internal design workflows are solid.

Future Research

Expanding the scope and tackling the limitations of this thesis, future studies should investigate the long-term impact of such interventions on organizational culture through longitudinal research, assessing whether the "developer-centric" mindset fundamentally shifts over a period of years rather than weeks. Additionally, researchers should test the Design Capacity Framework in smaller, purely volunteer-based open-source projects to determine how well it applies beyond organizations like Blender. Lastly, future research could further explore the Expertise Paradox, researching the extent to which developers should be empowered to design, or that a clear distinction and division in roles is more beneficial to distributed OSS organizations. Perhaps there is no best way and this differs per organization and has to emerge naturally and perhaps exactly that is the strongest aspect of OSS.

Personal Note

Going into this project, I wanted to do a project that includes both strategy and practical implementation. This thesis showed me that a single industrial designer can really both deliver strategic plans and build practical solutions. It was great to see that this thesis intersected both and it's satisfying to see that the resulting concept is grounded in research and strategic reasoning. I see myself both as a designer that likes to think broadly and strategically about things and considers the broader context, relevance and importance of the environment, but then also really makes it very concrete and practical and directly valuable.

Working within Blender's unique ecosystem was very new and insightful. While the lack of basically any formalized structure was initially surprising to someone who values organized planning, I learnt that this chaos isn't a flaw, but a sign of open-source resilience. It creates an autonomy that drives the team's dedication. Ultimately, I learned that for organizational analysis, soft skills like observation, empathy, and discussion are just as critical as hard design skills. My role as a designer in this context was not just as a creator of visuals, but as a connector who made implicit workflows visible and improvable.

More personal thoughts can be read in 7.3 Personal Reflection.

7.2 Final Conclusion

This thesis set out to answer if a practical intervention could improve the design workflow of the core Blender team. The goal was to increase Blender's ability to leverage their design expertise and balance the technical-heavy environment. The research found Blender has a long-term scarcity of design expertise, which originated due to a historically developer-centric culture. This created an environment with little tools (infrastructure) and standards for design, making current design contributions challenging and workflows unoptimized.

Answering the research question, the developed Design System functions as a tool that improves and streamlines the design workflow. The tool increases Design Capacity by bridging the cultural and technical gap between designers and developers and transforms inaccessible knowledge into an accessible toolkit.

The validation confirms that building a Design System improves Blender's workflow and connects designers and developers. By providing a digital twin of the software interface, the tool empowers non-designers to independently create accurate, higher quality design proposals. This addresses the Expertise Paradox and lowers the barrier to adoption by supporting different workflows (hybrid workflow). User testing and validation proved that the system shifts the workflow focus from the manual mockups building to the higher-value activity of problem-solving. The team at Blender is currently building further on the delivered Design System and actively developing extra features to further improve the workflow, indicating the potential for the Design system to outlive the duration of this thesis project.

For Blender, the priority is to maintain the momentum in emphasizing and optimizing for design. The path forward is detailed in the Intervention Roadmap: having optimized the internal workflow with the Design System, Blender can consider acquiring more design expertise. Either by hiring designers or by implementing the Community Portal.

For Open Source to stay relevant in the future, OSS organizations must evolve beyond traditional code-first innovation. Building Design Capacity enables better solutions by making the design of our 3D tools as efficient, open, and collaborative as the code that powers them.

Figure 7.2 | The Blender Office in Amsterdam

Figure 7.3 | The Oscar won by Flow in 2025 (a movie made entirely in Blender) which I got to hold during the Blender Conference

7.3 Personal Reflection

Balancing the academic expectations of TU Delft with the practical reality at Blender was honestly quite a challenge, but it's what I signed up for. It often felt like I was trying to merge two different worlds, connecting all these abstract ideas for the report while simultaneously trying to understand an organization by being part of it and building a concrete Design System that actually worked. It wasn't always easy to keep a clear overview or even figure out where my specific contribution fit in, especially since there is already so much work happening at the office every day. I really had to push myself to capture and especially really understand the nuance of what has already been there before my project. Not to mention to accurately position my own work within that.

On the flip side, I was positively surprised by how quickly I became part of the team. I wasn't just observing from a corner but I really got to participate, share my perspective, and see how open everyone was to new suggestions and workflows. The Blender team is wonderful and very openly welcoming. Looking back, I'm really happy that I took a proactive approach, whether it was setting up interviews, building relationships with the developers, or making the trip to Spain happen. It felt rewarding to go beyond just writing a thesis and actually make things happen within the organization and see the mostly positive, but of course sometimes also constructive reactions to my work.

Discovering how things are made always takes away some of the magic. I have always looked up to Blender for the free tool they provide and what it has allowed me to do. Being embedded in the office, however, was a complex experience. As someone who likes the style of structure and formalized plans, I was initially a bit conflicted to find that Blender operates differently than what I'm used to. It was confronting to see the reality behind the curtain. Yet, as the project concluded, this actually turned into a deeper kind of admiration. I learned to respect that the lack of standardization is the very thing that makes the organization resilient and welcoming to contributors. My personal lesson was one of letting go of my own wish for neatness and control, and learning to facilitate the team's existing strengths. I am perhaps more impressed now than I was before. They manage to build an impactful product with a globally decentralized team.

Looking back at the learning goals I set for myself at the start of the project, I achieved most of them. My ambition was to create value within a complex, multi-stakeholder organization not by finding a perfect, textbook solution, but by experiencing the reality of an open-source organization. My goal to "balance diverse stakeholder needs" was realized through the handling of merging academic reporting with the practical demands of participating and analyzing the organization and building a functional concept. I learned how to "bridge technical solutions with strategic decision-making" by actually doing the work, prototyping, debating, and iterating alongside the designers and developers. This hands-on experience gave me a far deeper understanding of how the digital creation industry operates than any theoretical study could have provided. I discovered that creating positive change in an organization like Blender isn't about imposing anything, but about learning to provide options, suggestions and navigate and respect its unique culture to create something that truly fits.

Lastly, while encouraged to make the project my own, the weight of my own expectations and the amount of report writing often felt like a distraction from the core "thinking and doing" process that I value most. Luckily, this struggle was very clarifying for my future. While I deeply enjoyed the hard skills of building the Design System and the soft skills of navigating the social dynamics of the office, I realized that my energy lies in understanding, entrepreneurship and building solutions rather than academic research and documentation. This thesis felt research-heavy to me and was a validation of the type of designer I want to be. A designer who also prioritizes doing next to abstract thinking, and practical value, experimentation and human connection over theoretical perfection and correctness.

References

Akbar, M. A., Sang, J., Khan, A. A., Fazal-E-Amin, Nasrullah, Shafiq, M., Hussain, S., Hu, H., Elahi, M., & Xiang, H. (2018). Improving the Quality of Software Development Process by Introducing a New Methodology–AZ-Model. IEEE Access, 6, 4811–4823. https://doi.org/10.1109/ACCESS.2017.2787981

Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies (iJIM), 14(11), 246. https://doi.org/10.3991/ijim.v14i11.13269

Bach, P. M., DeLine, R., & Carroll, J. M. (2009). Designers wanted: Participation and the user experience in open source software development. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 985–994. https://doi.org/10.1145/1518701.1518852

Badreddin, O., Hamou-Lhadj, W., Abdelzad, V., Khandoker, R., & Elassar, M. (2018). Collaborative Software Design and Modeling in Open Source Systems. In F. Khendek & R. Gotzhein (Eds.), System Analysis and Modeling. Languages, Methods, and Tools for Systems Engineering (Vol. 11150, pp. 219–228). Springer International Publishing. https://doi.org/10.1007/978-3-030-01042-3_13

Balka, K., Raasch, C., & Herstatt, C. (2010). How Open is Open Source? – Software and Beyond. Creativity and Innovation Management, 19(3), 248–256. https://doi.org/10.1111/j.1467-8691.2010.00569.x

Bender-Salazar, R. (2023a). Design thinking as an effective method for problem-setting and needfinding for entrepreneurial teams addressing wicked problems. Journal of Innovation and Entrepreneurship, 12(1), 24. https://doi.org/10.1186/s13731-023-00291-2

Bender-Salazar, R. (2023b). Design thinking as an effective method for problem-setting and needfinding for entrepreneurial teams addressing wicked problems. Journal of Innovation and Entrepreneurship, 12(1), 24. https://doi.org/10.1186/s13731-023-00291-2

Blender Foundation. (n.d.-a). Blender Foundation. Blender. Retrieved September 5, 2025, from https://www.blender.org/about/foundation/

Blender Foundation. (n.d.-b). Blender's History. Blender. Retrieved September 9, 2025, from https://www.blender.org/about/history/

Blender Foundation. (n.d.-c). Blender's License. Blender. Retrieved September 9, 2025, from https://www.blender.org/about/license/

Blender Foundation. (n.d.-d). Blender's License. Blender. Retrieved September 5, 2025, from https://www.blender.org/about/license/

Blender Foundation. (n.d.-e). FAQ. Blender. Retrieved September 9, 2025, from https://www.blender.org/support/faq/

Blender Foundation. (n.d.-f). FAQ Blender License. Blender. Retrieved September 5, 2025, from https://www.blender.org/support/faq/

Blender Foundation. (n.d.-g). History. Blender. Retrieved September 5, 2025, from https://www.blender.org/about/history/

Blender Foundation. (n.d.-h). Roles. Blender. Retrieved September 5, 2025, from https://www.blender.org/about/roles/

Blender Foundation. (2021a, February 16). Module teams for core Blender development. Blender Developers Blog. https://code.blender.org/2021/02/module-teams-for-core-blender-development/

Blender Foundation. (2021b, February 16). Module teams for core Blender development. Blender Developers Blog. https://code.blender.org/2021/02/module-teams-for-core-blender-development/

Blender Foundation. (2025a). Blender Development Fund. Blender Development Fund. https://fund.blender.org/

Blender Foundation. (2025b). In Wikipedia. https://en.wikipedia.org/w/index.php?title=Blender_Foundation&oldid=1303004398

Boisseau, É., Omhover, J.-F., & Bouchard, C. (2018). Open-design: A state of the art review. Design Science, 4, e3. https://doi.org/10.1017/dsj.2017.25

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Capra, E., Francalanci, C., & Merlo, F. (2008). An Empirical Study on the Relationship Between Software Design Quality, Development Effort and Governance in Open Source Projects. IEEE Transactions on Software Engineering, 34(6), 765–782. https://doi.org/10.1109/TSE.2008.68

Castro, H., Putnik, G., Castro, A., & Bosco Fontana, R. D. (2019). Open Design initiatives: An evaluation of CAD Open Source Software. Procedia CIRP, 84, 1116–1119. https://doi.org/10.1016/j.procir.2019.08.001

Design Paradigms—Blender Developer Documentation. (n.d.-a). Retrieved September 9, 2025, from https://developer.blender.org/docs/features/interface/human_interface_guidelines/paradigms/#non-overlapping

Design Paradigms—Blender Developer Documentation. (n.d.-b). Retrieved September 5, 2025, from https://developer.blender.org/docs/features/interface/human_interface_guidelines/

Dinkelacker, J., Garg, P. K., Miller, R., & Nelson, D. (2002). Progressive open source. Proceedings of the 24th International Conference on Software Engineering - ICSE '02, 177. https://doi.org/10.1145/581360.581363

Doherty, R., Wrigley, C., Matthews, J., & Bucolo, S. (2014). Climbing the Design Ladder: Step by Step.

Gacek, C., & Arief, B. (2004). The many meanings of open source. IEEE Software, 21(1), 34–40. https://doi.org/10.1109/MS.2004.1259206

Ghanbari, H., Vartiainen, T., & Siponen, M. (2019). Omission of Quality Software Development Practices: A Systematic Literature Review. ACM Computing Surveys, 51(2), 1–27. https://doi.org/10.1145/3177746

GNU in a Nutshell—GNU Project—Free Software Foundation. (n.d.). Retrieved September 5, 2025, from https://www.gnu.org/gnu/about-gnu.html

GPL2-license.txt. (n.d.). Retrieved September 5, 2025, from https://download.blender.org/release/GPL-license.txt

GPL3-license.txt. (n.d.). Retrieved September 5, 2025, from https://download.blender.org/release/GPL3-license.txt

Hahsler, M. (2005). A Quantitative Study of the Adoption of Design Patterns by Open Source Software Developers: In S. Koch (Ed.), Free/Open Source Software Development (pp. 103–124). IGI Global. https://doi.org/10.4018/978-1-59140-369-2.ch005

Hippel, E. V., & Krogh, G. V. (2003). Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. Organization Science, 14(2), 209–223. https://doi.org/10.1287/orsc.14.2.209.14992

Hunsen, C., Siegmund, J., & Apel, S. (2020). On the fulfillment of coordination requirements in open-source software projects: An exploratory study. Empirical Software Engineering, 25(6), 4379–4426. https://doi.org/10.1007/s10664-020-09833-8

J Stephenson,. (1998). The Concept of Capability and its importance. https://doi.org/10.4324/9781315042046

Kathryn Best. (2006). Design Management: Managing Design Strategy, Process and Implementation.

Kim, G. Y., Nam, K.-Y., & Borja De Mozota, B. (2019). Design Management Knowledge. Conference Proceedings of the Academy for Design Innovation Management, 1(1). https://doi.org/10.33114/adim.2017.98

Kretzschmar, A. (2003). The economic effects of design. National Agency for Enterprise and Housing, Cope.

Lakhani, K. R., & Wolf, R. G. (2005). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects.

Marzia Mortati, Beatrice Villari, & Stefano Maffei. (2014). Design Capabilities for Value Creation.

Müller, M., Vorraber, W., & Slany, W. (2019). Open Principles in New Business Models for Information Systems. Journal of Open Innovation: Technology, Market, and Complexity, 5(1), 6. https://doi.org/10.3390/joitmc5010006

Nielsen Norman Group. (2025). Typical Designer–to–Developer and Researcher–to–Designer Ratios. Nielsen Norman Group. https://www.nngroup.com/articles/ux-developer-ratio/

Norbert Roozenburg & Johannes Eekels. (1998). Productontwerpen, structuur en methoden.

Oberloier, S., & Pearce, J. (2017). General Design Procedure for Free and Open-Source Hardware for Scientific Equipment. Designs, 2(1), 2. https://doi.org/10.3390/designs2010002

Open Source Design. (n.d.). Open Source Design. Retrieved October 17, 2025, from https://opensourcedesign.net

Open-design movement. (2025). In Wikipedia. https://en.wikipedia.org/w/index.php?title=Open-design_movement&oldid=1302938239

Ozkaya, I. (2020). Building Blocks of Software Design. IEEE Software, 37(2), 3–5. https://doi.org/10.1109/MS.2019.2959049

Penpot: The Design Tool for Design & Code Collaboration. (2025, December). https://penpot.app/

Polaris Market Research. (2025). Digital Content Creation Market: Size & Global Outlook, 2034. Polaris. https://www.polarismarketresearch.com/industry-analysis/digital-content-creation-market

Raghunathan, S. (2024). Community-Led Development and Participatory Design in Open Source: Empowering Collaboration for Sustainable Solutions. International Journal of Computing and Engineering, 5(2), 45–55. https://doi.org/10.47941/ijce.1803

Rahad, K., Badreddin, O., & Reza, S. (2021). Characterization of Software Design and Collaborative Modeling in Open Source Projects: Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development, 254–261. https://doi.org/10.5220/0010266802540261

Ruiz-Múzquiz, P. (2025). AI Whitepaper Penpot.

Sack, W., Détienne, F., Ducheneaut, N., Burkhardt, J.-M., Mahendran, D., & Barcellini, F. (2006). A Methodological Framework for Socio-Cognitive Analyses of Collaborative Design of Open Source Software. Computer Supported Cooperative Work (CSCW), 15(2–3), 229–250. https://doi.org/10.1007/s10606-006-9020-5

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. (2006). Understanding Free/Open Source Software Development Processes. Software Process: Improvement and Practice, 11(2), 95–105. https://doi.org/10.1002/spip.255

Shanker, A. (2012). A Customer Value Creation Framework for Businesses That Generate Revenue with Open Source Software.

Spinellis, D., & Szyperski, C. (2004). How is open source affecting software development? IEEE Software, 21(1), 28–33. https://doi.org/10.1109/MS.2004.1259204

The Humane Interface. (2025). In Wikipedia. https://en.wikipedia.org/w/index.php?title=The_Humane_Interface&oldid=1294117125

The Open Source Definition. (n.d.). Open Source Initiative. Retrieved October 20, 2025, from https://opensource.org/osd/

van Boeijen, A., Daalhuizen, J. J., van der Schoor, R., & Zijlstra, J. (2013). Delft Design Guide: Design strategies and methods. BIS Publishers.

Wynn, D. C., & Clarkson, P. J. (2018). Process models in design and development. Research in Engineering Design, 29(2), 161–202. https://doi.org/10.1007/s00163-017-0262-7

Yas, Q., Alazzawi, A., & Rahmatullah, B. (2023). A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions. Iraqi Journal for Computer Science and Mathematics, 173–190. https://doi.org/10.52866/ijcsm.2023.04.04.014

Yin, R. K. (2018). Case Study Research and Applications: Design and Methods.

Zhang, S., Zhang, T., Cheng, J., & Zhou, S. (2025). Who is to Blame: A Comprehensive Review of Challenges and Opportunities in Designer-Developer Collaboration. Proceedings of the ACM on Human-Computer Interaction, 9(2), 1–32. https://doi.org/10.1145/3711105

Appendix

Appendix A: Personal Project Brief

TU Delft

DESIGN FOR our future

IDE Master Graduation Project

Project team, procedural checks and Personal Project Brief

In this document the agreements made between student and supervisory team about the student's IDE Master Graduation Project are set out. This document may also include involvement of an external client, however does not cover any legal matters student and client (might) agree upon. Next to that, this document facilitates the required procedural checks:

- Student defines the team, what the student is going to do/deliver and how that will come about
- Chair of the supervisory team signs, to formally approve the project's setup / Project brief
- SSC E&SA (Shared Service Centre, Education & Student Affairs) report on the student's registration and study progress
- IDE's Board of Examiners confirms the proposed supervisory team on their eligibility, and whether the student is allowed to start the Graduation Project

STUDENT DATA & MASTER PROGRAMME
Complete all fields and indicate which master(s) you are in

Family name	Tip	IDE master(s)	IPD ✓ Dfl SPD
Initials	S.W.	2nd non-IDE master	
Given name	Sem	Individual programme (date of approval)	
Student number	5295068	Medisign	
		HPM	

SUPERVISORY TEAM
Fill in he required information of supervisory team members. If applicable, company mentor is added as 2nd mentor

Chair	Jan Willem Hoftijzer	dept./section	HCD, HICD
mentor	Marina Bos-De Vos	dept./section	DOS, DIVE
2nd mentor	Dalai Felinto, Head of Product @ Blender		
client:	Blender Institute		
city:	Amsterdam	country:	The Netherlands
optional comments			

! Ensure a heterogeneous team. In case you wish to include team members from the same section, explain why.

! Chair should request the IDE Board of Examiners for approval when a non IDE mentor is proposed. Include CV and motivation letter.

! 2nd mentor only applies when a client is involved.

APPROVAL OF CHAIR on PROJECT PROPOSAL / PROJECT BRIEF -> to be filled in by the Chair of the supervisory team

Sign for approval (Chair)

Name Jan Willem Hoftijzer Date 1 okt. 2025 Signature

Personal Project Brief – IDE Master Graduation Project

Name student **Student number**

PROJECT TITLE, INTRODUCTION, PROBLEM DEFINITION and ASSIGNMENT
Complete all fields, keep information clear, specific and concise

Project title

Please state the title of your graduation project (above). Keep the title compact and simple. Do not use abbreviations. The remainder of this document allows you to define and clarify your graduation project.

Introduction

Describe the context of your project here; What is the domain in which your project takes place? Who are the main stakeholders and what interests are at stake? Describe the opportunities (and limitations) in this domain to better serve the stakeholder interests. (max 250 words)

3D creation software is essential to digital content production across diverse industries. Among the tools, Blender is free and open-source (FOSS), has a foundation organisation structure, community-contributing development, and large global user base, unique features rarely found in other 3D creation software.
However, the community-oriented, non-profit nature of Blender inherently introduces challenges such as: financial sustainability, coordination of fragmented contributions (community development), and addressing diverse user needs with limited resources.
This thesis applies an integral product design (IPD) approach to investigate how design methodology and expertise (both theoretical and practical) can enhance the development processes (organizational) and the resulting final Blender product (the software) by proposing a set of actionable design interventions that integrate design principles into the core development process. In this context, design refers to a holistic methodology that integrates user-centered, organizational, and strategic principles to shape both the development process and the resulting product, creating value for all stakeholders.

The Blender software product is influenced by the foundation structure and the objectives and factors mentioned. By understanding how the foundation currently influences the process and thus the final product, this project aims to enhance the development process by exploring the role and influence of design within it. Therefore, the project will analyse how to leverage design to drive process/product innovation and create value (for the foundation, the team and end-users) within open-source 3D creation software organisations.
Key stakeholders include the Blender Foundation, the Blender team, external contributors (such as developers and donors), designers, and end users. The central research question is: What is the role of design in shaping innovation, the development process and value creation to optimize the final product in open-source 3D creation software ecosystems? By identifying barriers and opportunities in development, this project aims to use IPD principles to propose solutions for design integration and contribute to the broader discourse and literature on integration focused design to improve products in complex and multi-stakeholder socio-technical organisations.

➔ space available for images / figures on next page

Personal Project Brief – IDE Master Graduation Project

Problem Definition

What problem do you want to solve in the context described in the introduction, and within the available time frame of 100 working days? (= Master Graduation Project of 30 EC). What opportunities do you see to create added value for the described stakeholders? Substantiate your choice.
(max 200 words)

Assignment

This is the most important part of the project brief because it will give a clear direction of what you are heading for. Formulate an assignment to yourself regarding what you expect to deliver as result at the end of your project. (1 sentence)
As you graduate as an industrial design engineer, your assignment will start with a verb (Design/Investigate/Validate/Create), and you may use the green text format:

Then explain your project approach to carrying out your graduation project and what research and design methods you plan to use to generate your design solution (max 150 words)

To carry out this graduation project, I will use a design-centric and systems-thinking and integral design (IPD) approach, combining qualitative research and design science/literature. First, to learn about the context, I will observe and conduct stakeholder interviews and user research to identify pain points and opportunities for design integration within Blender's ecosystem. I will analyze findings and compare them to existing literature (open source, innovation ecosystems, design in software, etc.). Guided by design methodology, I will iteratively prototype and test interventions in the development process with team/community feedback loops and focus on strategies that enhance process/product innovation, the development process, and value creation. The process will culminate in validated recommendations, actionable solutions, and frameworks for sustainable design integration in open-source 3D software organizations.

Project planning and key moments

To make visible how you plan to spend your time, you must make a planning for the full project. You are advised to use a Gantt chart format to show the different phases of your project, deliverables you have in mind, meetings and in-between deadlines. Keep in mind that all activities should fit within the given run time of 100 working days. Your planning should include a ***kick-off meeting***, ***mid-term evaluation meeting***, ***green light meeting*** *and* ***graduation ceremony***. *Please indicate periods of part-time activities and/or periods of not spending time on your graduation project, if any (for instance because of holidays or parallel course activities).*

Make sure to attach the full plan to this project brief.
The four key moment dates must be filled in below

Kick off meeting ______

Mid-term evaluation ______

Green light meeting ______

Graduation ceremony ______

In exceptional cases (part of) the Graduation Project may need to be scheduled part-time. Indicate here if such applies to your project

Part of project scheduled part-time	
For how many project weeks	
Number of project days per week	

Comments:

Motivation and personal ambitions

Explain why you wish to start this project, what competencies you want to prove or develop (e.g. competencies acquired in your MSc programme, electives, extra-curricular activities or other).

Optionally, describe whether you have some personal learning ambitions which you explicitly want to address in this project, on top of the learning objectives of the Graduation Project itself. You might think of e.g. acquiring in depth knowledge on a specific subject, broadening your competencies or experimenting with a specific tool or methodology. Personal learning ambitions are limited to a maximum number of five.
(200 words max)

I want to deepen my ability to create solutions that add value for organizations, particularly within complex, multi-stakeholder systems. I want to explore design's practical role in the industry. How does it drive value? I also want to learn how to balance diverse stakeholder needs while navigating real-world organizational constraints and frictions. This project is an opportunity to see how much positive change I can actually create in an organization like Blender, whose tools I have personally used for years.

Beyond the project's core goals, I have several personal learning ambitions. I want to gain hands-on experience applying design to strategic and organizational challenges, and to understand how a globally recognized open-source company like Blender operates behind the scenes. I aim to further develop my ability to bridge technical, practical solutions with broader strategic decision-making. Additionally, I am eager to learn how organizations in the digital creation (CG) industry function and how to position myself in this, as I aspire to work in this field in the future.

This thesis will allow me to pursue these ambitions while tackling a unique challenge and delivering meaningful value to an organization I admire.

Appendix B: Interviews

Note: Reference Table 2.1 on Page 42 for participant information.

Interview Preparation

Introduction & Ethics (Standard for all interviews)

- Briefing on research goal and topic and current phase
- Permission for recording and confirmation of anonymity

Data collection

- Notes taken & audio recording (with permission from participant). Transcribed audio to transcript.

Interview Topics

Interview Theme	Rationale	Sample Probing Question
A. The Design-Dev Workflow	Map the "as-is" state of feature development.	Walk me through the path from a feature idea to merged code. Where does 'design' happen?
B. Friction & Communication	Identify the root of the "Des-Dev Gap" (Finding 2)	How effortless does the handoff to developers feel? When a design proposal is misunderstood by a developer, what is usually the cause?
C. Design Memory & Documentation	Explore "Knowledge Accessibility" and silos	If a new contributor wants to know why a button looks a certain way, where do they look? Who do they ask?
D. Culture & Power Dynamics	Investigate the "Developer-Centric Culture" (Finding 3)	Who has the final say on decisions? How much does 'technical feasibility' affect 'user experience'?
E. Scalability & Community	Look at the "Expertise Paradox" (C3 contributions)	What is the biggest barrier for a community member who wants to contribute a design vs. code?

Note: Detailed questions were created based on the participant's role (Developer vs. Designer vs. Executive).

Interview Topics, Questions, Results & Impact on project

Note: The questions below don't include all the questions discussed, below is a selection of the most important ones.

ID	Topic & Main Questions Asked (doesn't necessarily include all questions asked)	Key Results	Impact on Project
P1 Des-igner	• Topic A, B, C, D, E • How are past design decisions, failures, and successes remembered, communicated, and used to inform future work, especially when the contributors are constantly changing? • From your perspective, what is the value of design in an open-source context like Blender? What is its purpose? What does design mean? • How do you define your unique role and measure your impact in an environment where 'everyone' is a designer to some extent? • What do you think about a designer's role to facilitate design instead of design themselves? • What does the typical design process at Blender look like? • Is innovation in Blender primarily driven top-down (by the Foundation) or bottom-up (by the community)? How does this dynamic work? • How would you explain Blender's work culture and pace of work? • What are the main challenges Blender is facing, and which ones are related to design?	• Multifaceted Role: Manages UX, branding, and graphics; acts as a bridge between core dev and community needs. • Integration Friction: High difficulty in creating a cohesive UI across siloed modules (e.g., Compositor vs. Video editor). • Review Burden: Significant time is spent manually reviewing community input rather than proactive feature design. • Hardware Constraints: Design must remain adaptable for older hardware, adding technical complexity.	• Validated Challenge 7 (Expertise Paradox): Showed how a single expert becomes a bottleneck for community reviews. • Informed Req 1.1: Proved the need for a "Single Source of Truth" to unify disparate modules. • Design Capacity: Highlighted that fast release cycles (3-4 months) require high-leverage tools to maintain quality.
P2 HR	• Topic B, D & E • How would you explain Blender's work culture and pace of work? • How do you experience hierarchy in the Blender team? • What are challenges that are experienced in the team? • What does the typical design process at Blender look like? • What do you think Blender's values are? What does the organization stand for? • What do you think the Team's values are? What do people find important? • What is the teams and your opinion on the community? • Is innovation in Blender primarily driven top-down (by the Foundation) or bottom-up (by the community)? How does this dynamic work? • What are the main challenges Blender is facing?	• Culture: Laid-back and autonomous, but faces "remote disconnect." • Values: Care, sharing, and freedom are central; however, there is a historical "lack of management." • Friction: Difficulty balancing commercial needs with open-source development.	• Validated the need for the intervention to be autonomous and optional (Req 1.3) to fit the 'not managing people' history • Informed Challenge 1 (Communication) regarding remote developer silos.
P3 Des-igner	• Topics A, B, C & D • What does a typical week look like for you? What kind of tasks do you spend most of your time on? • Who do you communicate & collaborate with most frequently? • What does the typical design & development process at Blender look like? Which steps are involved? - developers handbook • From your perspective, what is the value of design in an open-source context like Blender? What is its purpose? What does design mean? • If you could change or add one thing to the development process, what would it be? • What opportunities for design do you see in the development process?	• Ownership Culture: Highly decentralized; anyone can find solutions, but there is a constant need for a "bridge" to ensure individual work fits the "big picture." • The Process Paradox: Describes a "Two Diamond" model where implementation often forces the design "diamond" to reopen. Emphasizes that user testing must happen as early as possible to prevent late-stage friction. • Community Noise: Notes that external contributors often propose ideas without considering implementation implications or holistic feature impact ("just an idea without thinking how it would be implemented"). • Admits to making "executive decisions" on design when discussions become "noisy" or are led by people lacking design skills	• Validated Finding 3 (Tech-Focused Culture) • Directly Informed Opportunity 3 (Expertise Paradox) • Refined Concept 02 (Proposal Guide) • Digital Twin Requirement: focus on the "Two Diamonds" intersecting during implementation justifies why the Design System must be a "Digital Twin" used during the implementation phase

ID	Topic & Main Questions Asked (doesn't necessarily include all questions asked)	Key Results	Impact on Project
P4 Founder	• Topics B, E • Several conversations on: • Philosophy of Design at Blender • Core UI Principles • Legal/IP Strategy • Innovation process in open source	• Generalized Design: Features must be designed fundamentally and generally to benefit the whole ecosystem; rejects the isolated developer approach. • Blender DNA: UI must remain non-overlapping, non-blocking, and non-modal (Influenced by Raskin's The Humane Interface). • Legal Protection: Membership in the Open Invention Network (OIN) allows Blender to share ideas freely with allies (other organizations). • Strategic Publication:"Timestamping" ideas publicly on blender.org serves as a defensive patent strategy. • Legal Privacy: Discussions regarding competitor features (e.g., Adobe) must	• Validated Design System Architecture: Confirmed that components must be "fundamental and general" (Atomic Design) • Informed Req 1.2: Established the core principles • Informed Strategic Value of Concept 03
P5 CEO	• Topics B, D & E • From your perspective, looking at not just UI aesthetics, but the entire user experience and development process - what is the role of 'design' in achieving Blender's long-term strategic goals? • How do you perceive the current integration of design practices within the development cycle at Blender? Are there areas where you feel this integration could be better or more effective? • From your perspective on organizational growth, does this 'improve the internal process first' sequence feel like the right path, or do you see a strategic risk in delaying the public-facing component? • Do you believe that a Design Portal structured around education and templates (as proposed) is the right way to help the community design, or are there other ideas with more potential? • Where do you see challenges and the most promising opportunities to introduce or enhance design methods, tools, or processes?	• Design as the "Glue": Defines design for Blender as the act of combining engineering, UI, and communication to ensure a product is usable and efficient; it is the "vision-making" phase • Tech-First Tension: Acknowledges that technical aspects often "take over" the process, leading to a "technology-first" culture that can alienate the user base if not mediated by design • Roadmap Order: Explicitly validated the sequence: Internal Design System first, then Community Portal. "Improve the internal workflow standard before involving external contributions." • The "Educational Funnel": Believes the biggest value of a Design Portal is increasing the "contextual understanding" of contributors to move them from "just an idea" to "systems-thinking."	• Strategic Validation of Roadmap • Refined Concept 03 (Design Portal) • Validated Finding 3 (Culture): Provided executive-level confirmation that Blender suffers from a lack of design-centric individuals due to its technical/engineering history.
P6 Developer	• Topics A, B, C, D • Can you walk me through the journey of a recent feature you worked on? Where did the initial idea come from, and what were the key steps to get it into the hands of users? • When you began coding that feature, what information or guidance was most helpful? For example, did you have mockups, a technical spec, user stories, or direct access to a designer or product lead? • From your perspective as a developer, what makes a new feature in Blender 'successful'? • How does feedback from the community actually influence your work on a feature? Can you think of a specific example? • If you could change or add one thing to the development process before a feature gets to you for coding, what would it be? What is the biggest opportunity to improve the process? • Is there one aspect of how Blender is developed that you think most outsiders misunderstand or underestimate?	• The "Just Start Coding" Trap: Admits that coding immediately is tempting and feels productive • Scattered Planning: Identifies a historical issue where valuable plans were "pushed aside" because they weren't visible to leadership • Remote Friction: Notes that collaborating on design with remote contributors who aren't part of the "daily team" is very difficult • Value of Sketches: Highlights that physical/paper (design) sketching workshops were "super handy" for resolving complex UI problems	• Finding 2 (Des-Dev Gap): Confirmed the "remote disconnect" as a primary source of friction • Informed Challenge 8 (Scattered Ideas): Provided a direct quote/evidence for "plans being put aside because people were not aware of them," justifying the Single Source of Truth requirement. • Informed Req 1.3: His preference for "sketches" over rigid specs informed the "Sketch-like" experience requirement for the Design System to maintain developer creative flow.

ID	Topic & Main Questions Asked (doesn't necessarily include all questions asked)	Key Results	Impact on Project
P7 External consultant	Topics A, B, D, E • Kun je kort je rol en de aard van je betrokkenheid bij Blender beschrijven? Hoe ben je bij ze terechtgekomen? • Kun je Blender's organisatiestructuur en werkcultuur toelichten, en wat hen uniek maakt in vergelijking met andere bedrijven waar je hebt gewerkt? • Wat denk je dat de waarden van het team zijn? Wat vinden mensen belangrijk? • Wat zijn de belangrijkste uitdagingen die je binnen het team ziet? • Welke verbeter kansen zie je voor Blender gebaseerd op wat je hebt ervaren in jullie samenwerking?	• The "Just Code" Trap: Confirmed developers feel "productive" writing volume of code, often at the expense of zooming out to ask "what is the goal?" • Lack of "Drum Beat": Historically, people could "mess around" for 1.5 years without feedback because the founder was a visionary leader, not a manager. • Ancient Workflows: Expressed shock that a software company uses manual "screenshot-editing" and Photoshop for mockups, calling it an "ancient mindset" (oudheid idee). • Remote Disconnect: Identified a significant divide where remote developers feel "disattached" and siloed from the Studio's daily rhythm. • Structure as Freedom: Argues that "freedom without structure is not freedom" because people don't know the boundaries of their autonomy.	• Validated Finding 3 (Root Cause): Provided high-level confirmation about Blender's tech-focused culture • Justified Concept 01 (Design System): Her observation of "ancient" manual workflows provides the evidence for why a modern UI toOlkit is an opportunity • Informed Req 3.2 (Intuitive/Self-Explanatory) the Design System must feel like a "toy box" (speeldoos) rather than a "manual." • Confirmed that the current leadership transition creates the "perfect window" for new interventions
P8 IT and digital systems	• Topics B, D, E • Can you walk me through a recent project, its origin, and the key steps to completion? • What information or guidance (e.g., mockups, specs, or direct contact) was most helpful at the start of that task? • What is the single biggest opportunity for improvement in the current development process? • What aspect of Blender's development is most misunderstood or underestimated by those outside the organization?	• The "Why" Gap: Emphasizes the need to understand not just the "What" (pragmatics) but the "Why" (semantics). Without the "Why," it is impossible to find the "golden zone" of feasibility and budget. • Management Information Silo: Receives high expectations and responsibility but is excluded from the management information loops that provide long-term vision/context. • Timezone Friction: Direct collaboration is limited by a 3-hour daily overlap, making sparring and alignment difficult.	• Validated Des-Dev Gap: Proved that information silos don't just affect UI, but other areas (IT) too • Informed Opp. 1 (Accessible Knowledge): His plea for "Semantics" (the Why) inspires the requirement to document Design Rationale alongside visual mockups. • Informed Req 2.5: Highlighted that asynchronous (due to timezones) is a non-negotiable constraint for any new workflow tool.
P9 Development Manager	• Topics A, B, C • Having worked in other professional environments, how would you describe the level of structure in Blender's development process compared to more traditional software companies? • In your experience at Blender, at what stage does a feature's design typically become "clear"? Does this happen before coding begins, or does it emerge during the implementation process? • Within the current module-based structure, how much autonomy do module owners have over UI/UX decisions, and how often are these decisions made without the core design team's direct awareness?	• Emergent Design: Unlike industry standards where design is clear before coding, at Blender, "clarity emerges during execution," leading to less predictable timelines. • Decentralized UI: Module owners frequently make independent UI decisions without the core design team being aware. • Review Bottleneck: The core designers want to be involved but lack the "capacity to review everything," creating an awareness gap. • Multi-tasking Strain: Developers are often stretched across multiple unrelated projects simultaneously, reducing focus. • Short-termism: Acknowledge a cultural preference for "running with ideas" (speed) over adhering to a long-term unified design vision.	• Validated Expertise Paradox: specialists are currently bottleneck for design quality. • Des-Dev Friction: Showed that friction is caused by a lack of "visibility" into what module owners are building. • Justified Design System: Proved the need for a system that acts as an "automated guardrail", allowing developers to "run with ideas" while automatically staying within design standards • Informed Req 1.3: Validated that any new tool must not "slow down", or it will be rejected in favor of short-term speed.
P10 Graphic Designer (design system's developer)	• How important is it that the design system looks "exactly" like the software? Does a low-fidelity system risk losing the interest of the development team? • Can a new set of tools actually change the "Developer-Centric" culture of an organization, or is it purely a technical improvement? • How does an interactive design system change the dynamic of a design meeting?	• Design system is an industry standard; without it, too much time is wasted on manual work • Stresses that a DS must match the live software's fidelity exactly to be useful. • Recommends building from single elements into complete assemblies • Believes you can shift a company's culture by showing them a working tool. Interactive features (like hover effects) are crucial for team engagement and real-time meeting iteration.	• Validated Design System as solid and proven solution option • Justified the "Atomic" approach of building from Foundations to Organisms. • Design System can be a "Tactical Intervention" that leads to a "Cultural Shift" (Section 1.5).

Appendix C: Conference Visits

Key Takeaways from Penpot Fest

1. Shared Language & Collaboration

- The most crucial thing to get right to bridge design and code is speaking the same language (communication, shared understanding and group awareness)
- True creativity happens between roles, not within them
- Collaboration should focus on sparking conversations and interactions, not just technical implementation (Wikimedia workshop).

2. Integrating Developers into the Design Space

- Making developers use and appreciate design tools works better than making designers code - this is what Penpot tries to do
- The goal is to create design tools for developers, not the other way around

3. Encouraging Design Contributions

- Ask designers for help with a challenge, don't present the solution space to them - this approach makes them more curious and creative.
- Designers' problems are "wicked" (vague) while devs' are "tame" (clear). Feedback and solution is not as satisfying for designers as how satisfying it is for the devs — recognize this difference in collaboration. (Eriol Fox, Superbloom).
- Designers often hesitate to contribute until they understand the full system — onboarding and environment setup are essential.
- Research into designer motivation in OSS is still limited and valuable.
- Resource: sosdesign.sustainoss.org

4. Systems, Components & Consistency

- Implement guidelines to ensure consistent use of components and build a common design language (Wikimedia workshop).
- Focus of design system: When starting new ideas with components, focus less technical details and more on enabling collaboration through creation. It's not really about the technical details but more about creating something that sparks the conversation and interactions: collaboration (Wikimedia workshop).

5. Other

- Creativity begins where the client brief ends — that's where questions and exploration start
- Competitiveness isn't in technology itself, but in what you create with it
- The value of (artificial) creative constraints
- Resource: Penpot AI Whitepaper

Key Questions Asked During Blender Conference

For Artists/Users:

- Think about a recent feature you were excited about. How well did the final implementation meet your expectations?
- If you could give the Blender development team one piece of feedback about their process, what would it be?"
- Innovation
- (Objective: Understand the sources and drivers of innovation in the ecosystem)
-

For All Groups:

- Where do you see the biggest opportunities for Blender to innovate in the next few years?
- How does Blender's open-source nature help or hinder genuine innovation compared to other 3D software?
- What's the most innovative thing you've seen come out of Blender (or its community) in the last year? What made it so special?
- What is the biggest friction point in your daily workflow with Blender?
- If you had a magic wand to change one thing about Blender's user interface or workflow, what would it be and why?

For Studio/Professional Users:

- "Besides Blender being free, what is the single biggest value Blender provides to your studio's pipeline?
- For Independent Artists/Hobbyists:
- What makes you choose Blender for your personal projects? What part of the 'Blender experience' is most valuable to you?
- Challenges & Opportunities (Pain Points)
- (Objective: Identify the specific friction points that your design interventions could address)

Appendix D: Ideation & Analysis

Initial Thematic Clustering (Discovery Phase)

Following the stakeholder interviews, raw data points were clustered into broad themes such as Community, Organizational Structure, and Design Process. This step identified the high-level friction points in the ecosystem.

Analysis Phase 01

Key Themes & Insights

Community

Difficult to manage fragmented community contributors
- maintaining alignment, language barriers, cultural differences, timezones

Sem

Friction from Community Feedback
- community feedback can be hard for team
- collecting feedback is informal
- community often creates ideas, not argued concepts
- goal is not to assess popularity but to find blind-spots (Dalai)
- Danger of opening up to random people (comm.) can lead to project feeling lost (Dalai)

Sem

Challenge to leverage external design expertise
- harder to contribute design compared to code (dalai)
- collab w external designers very difficult (Sybren)

Sem

Distributed Ecosystem: Community is significant part of Blender
- remote devs
- excitement & active involvement

→ still visible in Blender's work culture

Sem

Highly open and transparent development process
- community has access to almost everything

→ quick feedback (CLD)

Sem

Organizational Structure and Culture

Institute vs. Studio
- Studio project crunch institute continuous but release deadlines
- desire from devs to have tighter connection w studio

Sem

Informal Design memory
- critical design rationale held by individuals (social memory)

Sem

New Hierarchy
- hierarchy is new, necessary because of growth

Sem

Disconnect Between Strategy and Daily Tasks
- unawareness of knowledge, decisions and plans (strategic reasoning)
- coordinating fragmented development

Sem

People

Work culture based on:
- Freedom
- Autonomy/Ownership
- Flat
- Informal

→ module isolation
→ No external motivation (investors) sets own pace and priority
→ informal work culture (design memory)

Sem

Contributors have High Intrinsic Motivation & Strong Values
- freedom
- trust
- autonomy
- sustainability (timelessness)

Sem

Development Process and the Role of Design

Universal ownership vs. dedicated expertise
- conflict between OS ideal that anyone can contribute ideas and need for specialized design expertise for quality and coherence
- Designers often have to defend their decisions against critique from non-designers.

Sem

Devs culture
- 'just code' culture: jumping to implementation (less upfront thinking)

Sem

Friction in Designer-Dev interaction
- Different backgrounds, terminologies, and workflows between UX designers and software developers. Misunderstandings, misaligned expectations, and an inefficient handoff process.
- Designers feel their work is sometimes de-emphasized.

Sem

Suboptimal planning and overview
- Good ideas and plans get lost because they are noted down in different locations across a large, distributed organization.
- Lack of a centralized overview and management of the innovation pipeline.

Sem

Innovation

Hybrid top-down and bottom-up approach to innovation
- Strategy from office and contributions form community combine - productive tension
- challenge lies in coordinating these fragmented contributions and ensuring they align with the overall vision

Sem

Balancing innovation with stability
- LTS

Sem

Sustainability (Financial) is challenging

Sem

Defining success
- user satisfaction
- metrics (although little)

Sem

Challenge & Opportunity Matrix (Synthesis Phase)

The initial themes were refined into the core challenges presented in Chapter 3. Here, data points (quotes) are directly mapped to specific tensions and potential opportunities, forming the basis for the design interventions.

Section 2

Analysis Phase 02

Key Challenges & Tensions

Coordination

Challenge: Communication and coordination in a hybrid, growing team

- With a team that has grown significantly and operating in a hybrid model across global timezones, communication and coordination is a difficult, making collaboration with the office challenging.

Challenge Evidence

P7
- P7 explicitly identifies a significant disconnect and separation between different parts of the organization. She states, "...between development and studio and remote, there is actually still too much separation." [12s]
- This separation leads to employees, particularly remote workers, feeling "disattached," questioning how they contribute and relate to the rest of the team. [12s]
- She sees a major opportunity in unifying these disparate parts (remote, studio, developers) into a single, cohesive system to increase impact. [19s] The challenge is to "bundle the forces" more effectively. [19s]

P2
Fiona explicitly states that maintaining alignment and a shared context with remote employees is a major difficulty. The cultural and time-zone differences add layers of complexity that go beyond typical hybrid work challenges.
- "I mean, I think communication is, is a big one always, in sense of, is everyone aware of what I'm doing? ... For me, personally, really the link with the remote people that I know in development is a big topic. It's keeping people in the loop, keeping people committed, keeping people also talking the same language. That's hard, yeah."

- The combination of rapid team growth, a hybrid work model, and a highly international team creates significant communication overhead.
- Fiona explicitly states that maintaining alignment and a shared context with remote employees is a major difficulty. The cultural and time-zone differences add layers of complexity that go beyond typical hybrid work challenges.
- P7 identifies a tangible "disattached" feeling among different parts of the organization—specifically between the studio, development, and remote workers. This indicates that communication breakdowns are leading to functional silos and a fragmented team identity.

Challenge: Coordinating fragmented development/contribution

- The bottom-up volunteer-driven nature of open source contributions leads to high volume of fragmented contributions that are difficult to align with a coherent, strategic roadmap

Opportunity: Implement design frameworks for cohesion
- standardized design briefs, public roadmaps, user journey maps

→ help guide community contributions to be more aligned with strategic goals without stifling creativity - to provide a clear, shared context

Challenge Evidence

P7
This is proven with a vivid, multi-layered story about people working for a year and a half "rommelen" (messing around) only to have their project cancelled, demonstrating a profound lack of coordination and feedback. [10s]

- She describes the past culture under founder Ton Roosendaal as one of significant freedom, which led to fragmentation. Employees could "rommelen" (mess around) for a year or even a year and a half without clear performance metrics. [10s]
- This freedom resulted in a lack of clear, short-term goals and a missing "drum beat" for projects. The primary focus was on the major releases, but the path to get there was unstructured. [11s]
- A critical issue was the lack of continuous feedback. P7 gives a powerful example of people working on a project for over a year, only to be told by Ton that it was pointless and they had to stop, causing immense frustration. [10s]

P6
"Maar het idee is wel dat er wat meer projectmatig gewerkt wordt, zodat je iets hebt wat een... beperkte scope heeft, zowel in features als in tijd, eh waar mensen op gezet kunnen worden..."

P1
So since we moved to more project driven... since a couple of years or so, we tried to focus more on projects... we've been trying to have more every week we have like a sync of all these projects. Okay, where are all this project going?"

- The historical structure of Blender development was siloed into "modules," which lacked a unified, project-level oversight.
- Multiple employees confirm that Blender is deliberately shifting towards a more "project-based" or "project-driven" workflow. This is a direct strategic response to the inefficiencies of fragmented development.
- This new approach aims to establish clear scope, timelines, and dedicated teams for initiatives, improving predictability and coordination.

Challenge: Personal decision and norm knowledge ('design memory')

- Personal design knowledge is hard to accelerate with growth. Critical design rationale is held by individuals and inaccessible to the community

Opportunity: Establish a centralized knowledge base
- introduce design methodologies and governance frameworks, formal design system for UI consistency, standardized design brief templates, public roadmaps
- formalize and document design decisions in a shareable environment

→ ensure consistency and efficiency during development process
→ empower community by increasing accessibility to higher-level design expertise (created elements) that is already approved by Blender's designers
→ create a durable, accessible record that improves onboarding, consistency, makes design process more transparent and scalable

Challenge Evidence

P7
- She acknowledges that the team historically hates this kind of "administrative work," which explains why this informal system has persisted. [32s]

P1
"...for like everyday work... you should ask the person that worked on it... You just go to the chat you ask the person, or probably somewhere here, or you ask someone that knows..."

- Crucial information about past design decisions, failures, and successes is not formally documented but resides in the memory of the individuals who were involved. This is explicitly referred to as the "design memory."
- The standard procedure for understanding why something was built a certain way is to socially navigate the organization to find and ask the original creator. This is inefficient and not scalable.
- Pablo confirms this is the de facto process, while acknowledging that efforts to create formal guidelines are ongoing but incomplete.

Workflow & Culture

Challenge: Developer Centric Culture vs Design Informed

- The 'just code' cultural tendency to prioritize coding over upfront design rationale - creates friction where designers may have to defend their decisions against developers who hold the 'keys' to implementation

Opportunity: Elevate and Integrate design
- frame 'upfront design thinking' as a crucial, value-adding part of the development process. Create tools/processes to realize this.

→ create a development process that balances and synergizes both design and development. Empower development by solid design fundamentals
→ make the hand-off process smoother

Opportunity: Empower Design authority while fostering collaboration
- establish principles where final design decisions are made by designated design experts
- use design systems and clear documentation to educate and empower developers, enabling them to make informed design choices within the established framework

→ respecting both expertise and the collaborative spirit

Opportunity: Frame 'upfront thinking' as a producive part of development
- introduce design methodologies (mockups, defining problems) as a formal and valued part of the development lifecycle
- a design system can empower devs with tools to quickly create high-fidelity representations of their ideas

→make the design phase feel more tangible and productive

Challenge Evidence

P7
This is proven with a direct quote. When you say developers feel productive when they "just code," she replies, "But that is echt precies wat je zegt eigenlijk" (But that is really precisely what you are saying, actually). [31s] This is undeniable proof.

- When I describe the developer culture of "just code," where productivity is felt by producing a high volume of code, P7 emphatically agrees, stating, "But that is exactly what you are saying, actually." [31s]
- She elaborates on this by contrasting it with a more strategic approach: "thinking about maybe one rule is enough if you do it right." [31s] This confirms her observation of a culture that prioritizes the act of coding over strategic design thinking.

P3
"I don't think that people that are not even aware of some of those basic design principles should be the one leading this design discussions..."

- A "just code" mentality is present, where developer productivity and satisfaction are sometimes measured by the act of producing code, independent of the strategic direction. This can lead to work being done before the problem and goal are fully understood.
- Dalai articulates a clear struggle for design to have a decisive voice. He notes that design discussions are sometimes led by developers who may lack deep design expertise, forcing designers to defend fundamental principles.
- The culture of developers feeling productive when they "just make a lot" is explicitly acknowledged.

Challenge: Friction in Designer-Developer Interaction

- Different backgrounds, terminologies, and workflows between UX designers and software developers. Misunderstandings, misaligned expectations, and an inefficient handoff process.
- Designers feel their work is sometimes de-emphasized.

Opportunity: Establish shared processes and boundary objects
- promote a more integrated process where devs are involved early in design, and designers remain involved during implementation - design systems can serve as boundary objects

→ shared language and toolset allowing both groups to communicate clearly and collaborate more effectively

Challenge Evidence

P7
- P7 recognizes the underlying cultural tension. She notes that giving design a proper place is a challenge because the prevailing mindset is, "...design, yes, nice, we talk about it, but actually we just have to code, right? We have to code. That's the work." [33s] This implicitly confirms the friction and cultural divide between the disciplines.
- The discussion about the need for a design system highlights this friction. The current "painful" and manual process for designers to communicate ideas (taking and editing screenshots) is a clear point of inefficiency and misunderstanding in the workflow. [47s, 48s]

Challenge: Universal ownership vs. Specialized expertise

- a core conflict exists between the open-source ideal that anyone can contribute, and the need for specialized expertise to ensure quality and coherence
- Designers feel like they have to defend their decisions against critique from non-designers.
- Some experts feel like they don't have time to keep up with the many requests they get

Challenge Evidence

P7
This is proven by her identifying the core ideology ("everyone is the owner of everything") and immediately contrasting it with the reality that specific experts (Dalaj, Sergei) are bottlenecks because they are constantly sought out. [33s]

- P7 identifies the open-source ideology where "everyone is the owner of everything" as a strong part of the culture. [33s]
- However, she immediately points out the friction this creates by highlighting that specific individuals have recognized, specialized expertise (e.g., "a Dalaj... is much more adept in design... and a Sergei is very good at implementing, at coding"). The difficulty is in how this all comes together. [33s]
- This creates a burden on the experts, who are frequently sought out and complain about having too little time. The sense of universal ownership clashes with the practical reality that certain people are depended upon for their deep knowledge. [34s]

- The open-source ethos of "universal ownership," where everyone can contribute to everything, is a powerful part of the culture.
- However, this clashes with the reality that certain areas require deep, specialized expertise (e.g., in design or specific codebases). This creates bottlenecks, as the team becomes reliant on a few key experts.
- P7 points out that while everyone is an "owner," the organization often has to turn to specific people like Dalai or Sergey, which creates tension. The organization is now actively working to clarify roles to manage this.

Strategy & overview

Challenge: Disconnect strategy and daily tasks

- People feel disconnected between their assigned tasks and the high-level strategic reasoning behind them.
- Unawareness of made decisions or knowledge 'higher-up'
- This is a top-down model, in contrast to bottom-up which is found elsewhere.

Opportunity: Formalize design expertise
- involve key staff as consultants before making decisions

→ leverage their expertise and make sure their work aligns with the strategic vision

Challenge Evidence

P7
This is proven by her recounting how employees wanted more road mapping five years ago, but Ton's management style (avoiding meetings) directly prevented it, leading to a strategic disconnect. [10s]

- P7 notes that even five years ago, there was a clear need among employees for more structure, clarity, and road mapping, but the founder, Ton, was resistant to it. [10s]
- His preference for minimal meetings and 1-on-1 interactions meant that a shared strategic overview was difficult to maintain, leading to the previously mentioned situations where developers' work was suddenly deemed irrelevant. [10s]
- She sees a significant opportunity for the new leadership team to work more iteratively on vision and plans, making them more transparent and involving people more, so they can see the "overall picture." [23s]
- "Ik denk dat het leiderschapsteam ehm meer iteratief met mensen kan gaan werken op visie en plannen zeg maar... waardoor ze makkelijker in kunnen stappen en dat ze het overall ook plaatje zien."

P2
"we have a history of not not following plannings... people get lost, organization doesn't follow through, and then, oh yes, we had that project. Oh, we need to do it well."

- There is a recognized gap between the high-level vision of the Foundation/leadership and the day-to-day work of the team.
- Historically, plans were not always followed, causing projects to get "lost" and creating a disconnect between daily tasks and strategic intent.
- Employees see a clear opportunity for the new leadership team to make the vision more transparent and involve the team more iteratively, so they can better see how their work contributes to the overall picture.

Challenge: Suboptimal overview & Distributed ideas

- good ideas and plans get lost because they are noted down in different locations across a distributed organization

Opportunity: Standardize process for ideas and plans
- implement a standardized process and overview for submitting, reviewing and tracking ideas and plans

→ leverage expertise and make sure their work aligns with the strategic vision

Challenge Evidence

P7
- The strong separation between studio, development, and remote teams directly contributes to a siloed environment where an organizational overview is lacking. [12s]
- P7 points out that because of the module-based structure and individual focus, people can "really get lost... in a module," losing sight of the bigger picture. [19s] This structure naturally leads to ideas being siloed within those modules.

P6
(when asked what one thing he would change in the process):
"Oh. Qua proces denk ik gewoon meer, eh, wat meer standaardiseren op wat de ideeën en de plannen zijn."
"Eh, daar vind ik dat het in het verleden nog wel mis is gegaan dat er, eh, ja gewoon plannen opzij gezet waren omdat men die niet vanaf de hoogte las."
"Wat heel logisch is, want er zijn heel veel plannen... Ik snap heel goed dat het lastig is als iedereen het ergens plannetjes ergens anders neerzet. Dus daar wat meer, eh, wat meer overzicht in en wat meer management..."

Community

Challenge: Difficulty in leveraging external design expertise

- it is significantly harder for external designers to contribute meaningfully compared to devs. Collaborating on "design things" with people outside the daily team is describes as "very difficult"

Opportunity: Develop a Design System as a contribution gateway
- a robust and accessible design system can act as a bridge. It would allow external designers to contribute in a structured way (e.g. by creating mockups with established components) and lower the barrier to entry for design-focused contributors

→ leverage external design expertise

Challenge Evidence

P7
P7 identifies the community as a core source of complexity for Blender. She states, "what makes it very complex for Blender... is the community." [7s]
- She highlights the challenge of managing the large volume of contributions ("commits") and input from the community. A system is needed to "pick that up all the time" and maintain the "big picture." [7s]
- This process is fraught with the risk of contributions being small, valuable edits that might inadvertently break the larger whole. [7s] She describes the entire process of managing community input as "best complex" (quite complex). [7s] While not specifying "design" expertise, her description of the process as a whole confirms it's painful and inefficient for all types of contributions.

Challenge: Friction and noise from community engagement

- The OS model is a major strength, but also a source of significant friction. Negative community feedback can be demotivating, while raw community ideas often lack the technical or systems thinking required for implementation. Opening up projects to "random people" (community) can make the project feel "lost"

Opportunity: Formalize design expertise
- structured (formalized) process and channels for community contributions

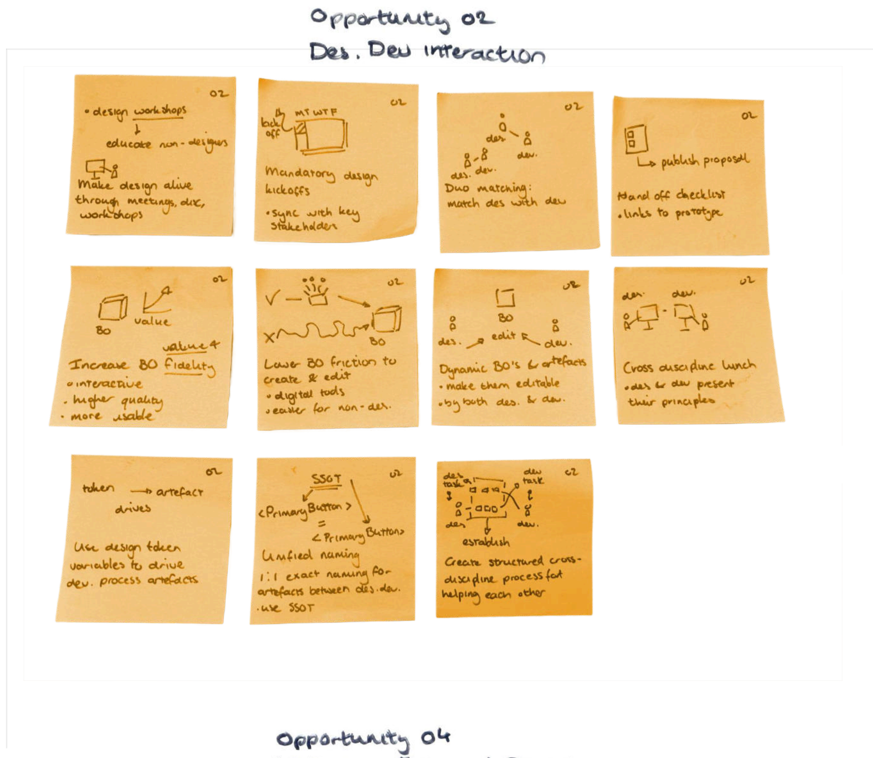
→ by establishing a more formal design process Blender can better guide contributors to translate raw ideas into feasible concepts - (ensure decisions are guided by design fundamentals, not just popular opinion)
→ empower Blender's designers to make confident decisions when reviewing community 'noise'
→ make community or user testing better align with designer's goals and find the "blind spots"

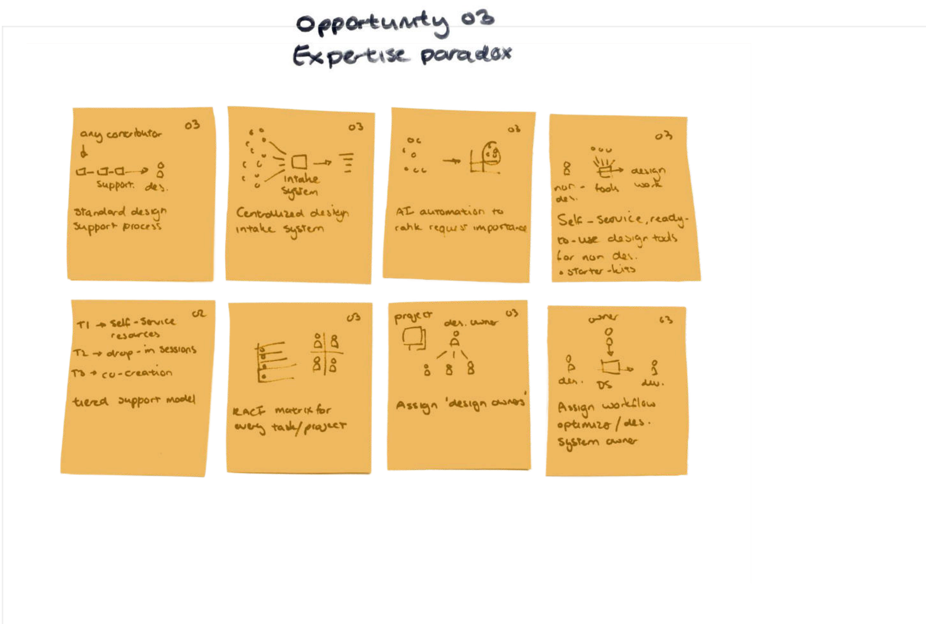
Challenge Evidence

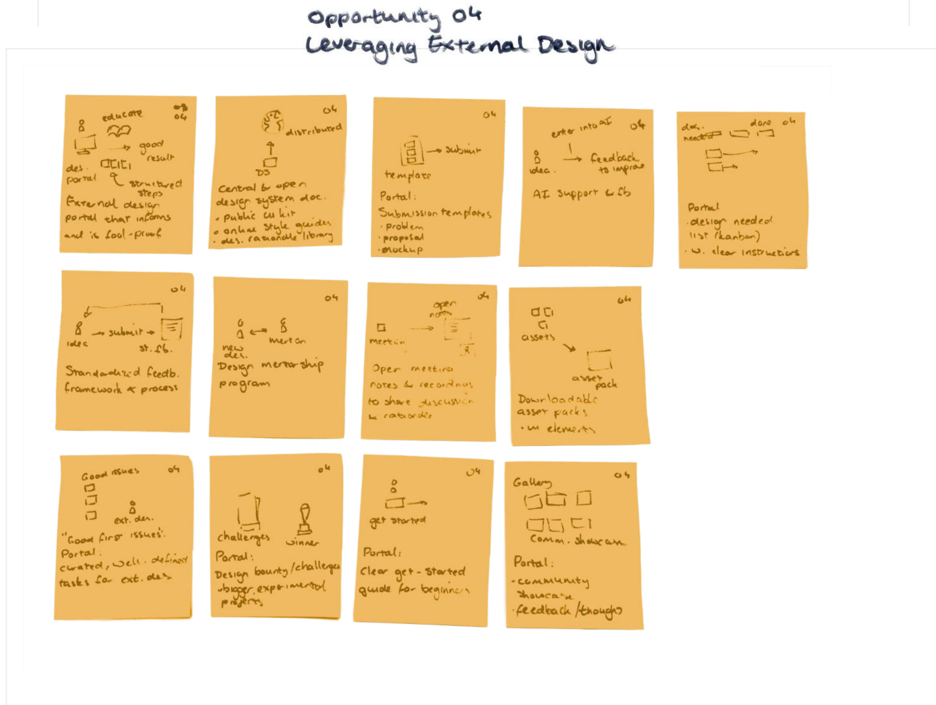
- **P1:** "...it's a bless and a curse is the living dealing with the community... there is this few other ones that are just like negative or trolls or whatever, I don't know, and it's just just demotivating..."
- **P2:** "...it can be very challenging on the personal level, because of all the communication, all the openness, scrutiny you can face... You got to have a strong spine, yeah?"

- While the community is a vital source of feedback and contributions, it also generates a high volume of "noise," including negative, trolling, or simply unconstructive comments.
- This friction is described as a "bless and a curse." It can be personally demotivating for the team and can derail productive discussions, sometimes even halting the development of new features.
- Fiona emphasizes that dealing with the constant public scrutiny requires a "strong spine" and is a significant personal challenge for the team.

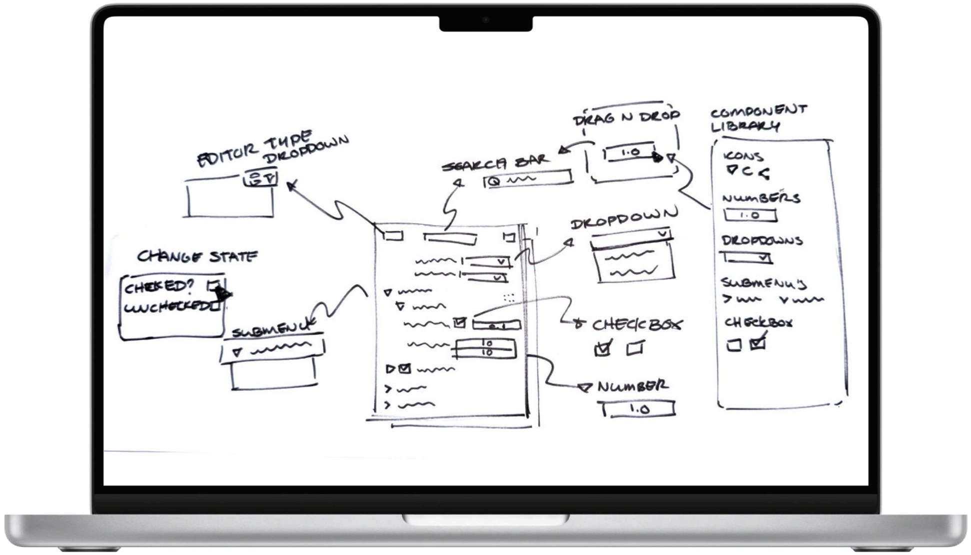
Overview of Idea Generation

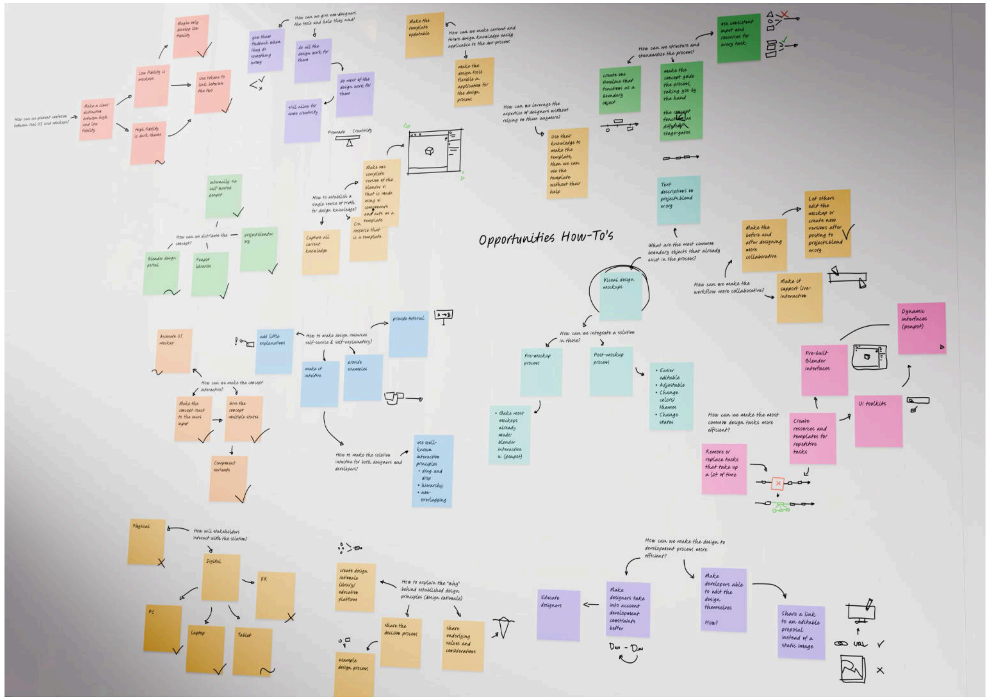




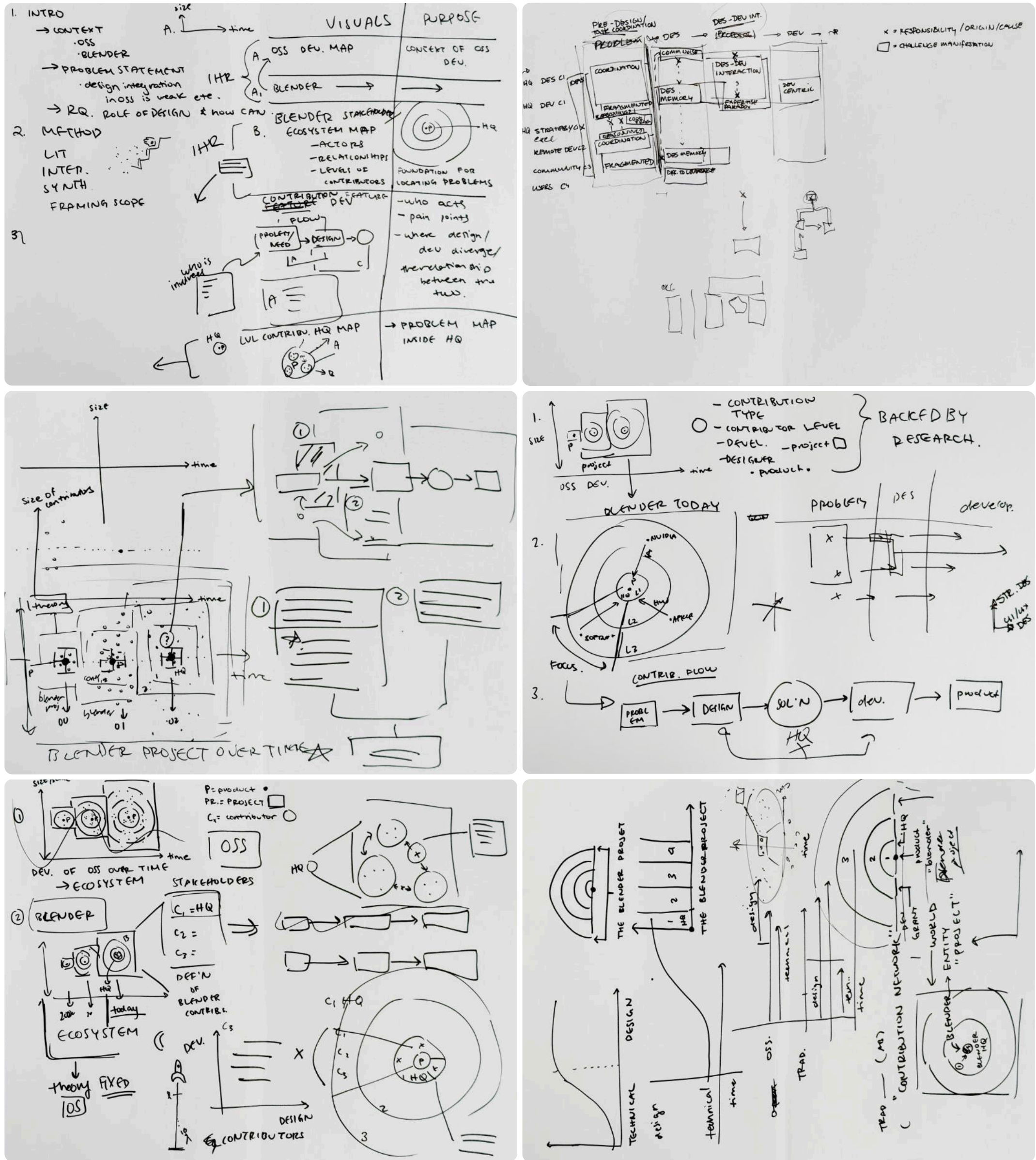


First Ideation Design System



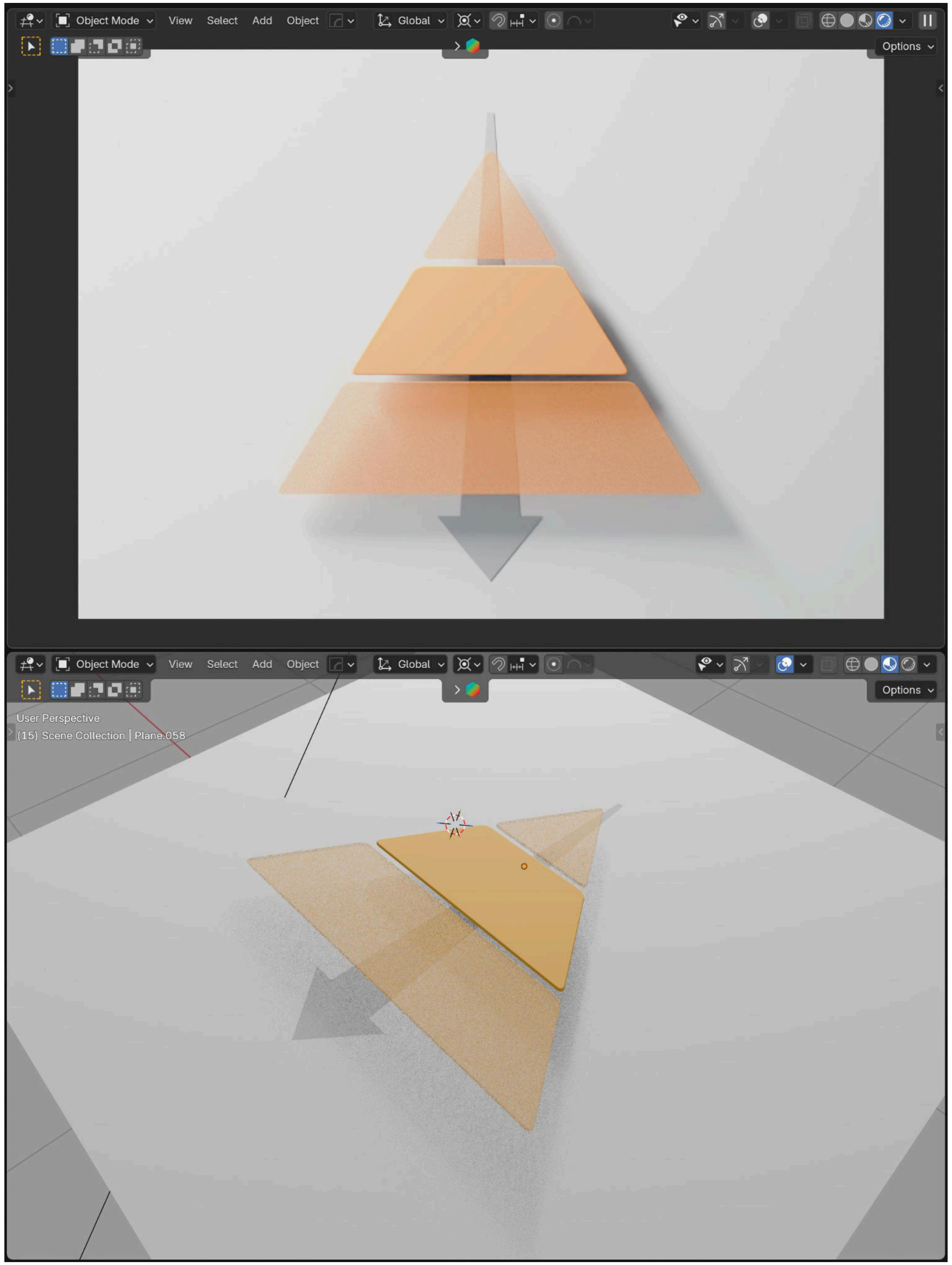


Brainstorm Ideation of Many Report Visuals



Report Visuals Creation

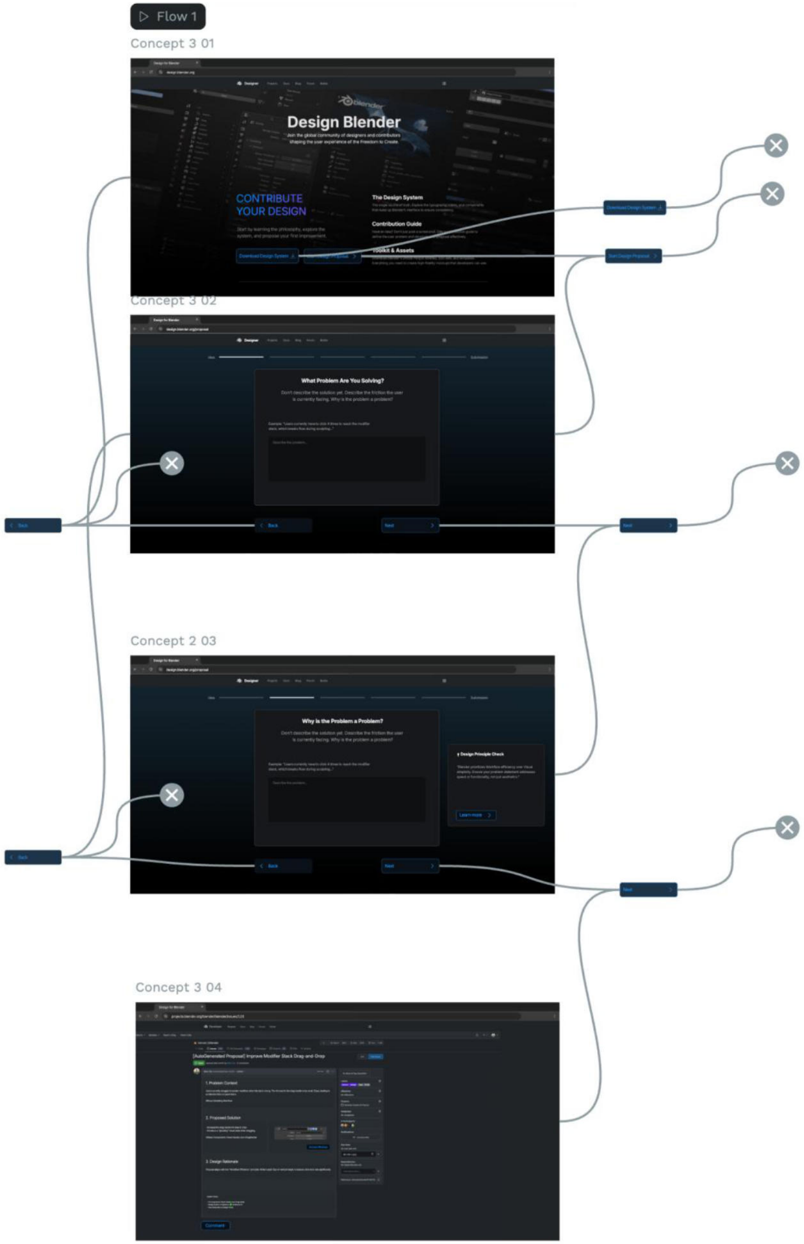
The report visuals were all modeled and rendered in Blender. Below is an example of the camera view (top) and the 3D scene (bottom).

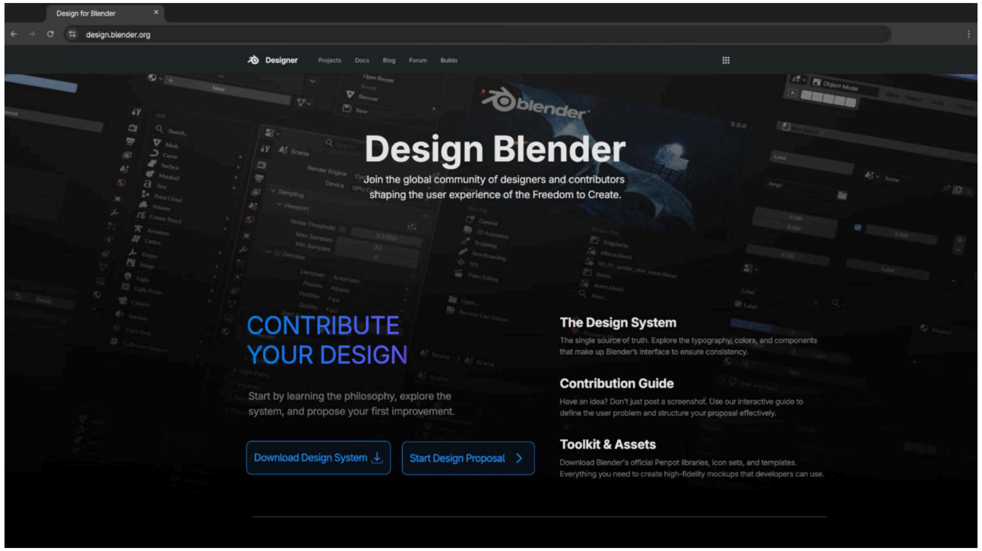


Appendix D: Extra Concept Materials

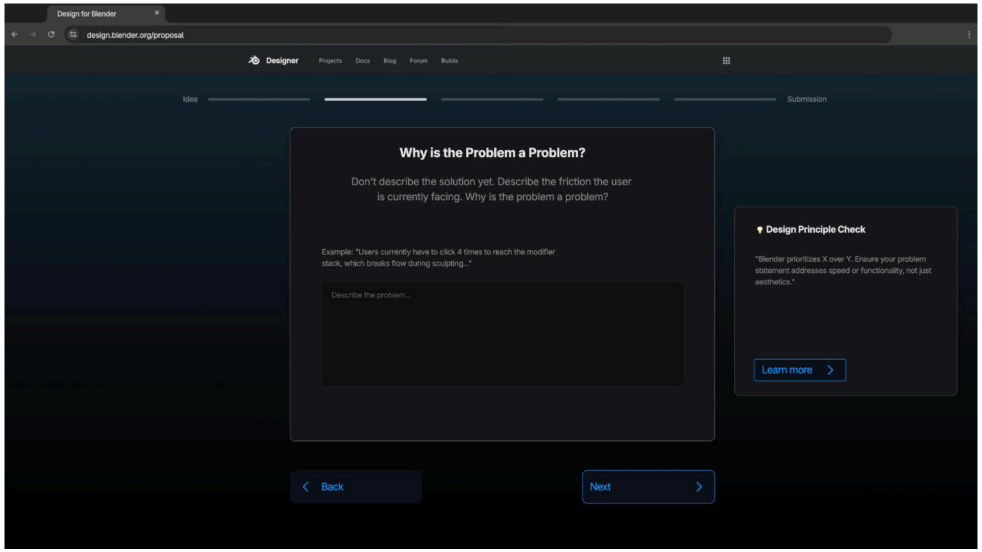
Concept 2 & 3 Interactive Prototyping - User Flow

Here, an interactive prototype was built where clicking buttons navigate the user to a different page, just like on regular websites. This was done to further detail and explore what the concepts could be.





Design.blender.org website landing page designed for Concept 03.



Concept 02/03 guided design proposal experience visualisation

Appendix E: Co-Design Workshop Materials

Workshop Slides

Below is a small selection from the 32 prepared workshop slides

Introduction

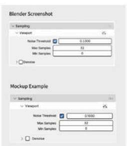
Agenda & Outcomes

01 "Why"
Shared Understanding Purpose
Problems a DS can solve → How it will solve them (goals)

02 "What"
Capabilities
What should it do? (features)

03 "How"
User Needs & Testing
What we want & feedback on current proposal

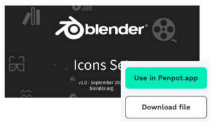
04 "Next Steps"
Priorities & Next Steps
Priorities & Dev, Impl. planning



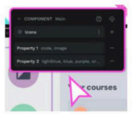
Mockup vs. Real UI
Have to look different (distinguishable)

All of UI Available
Every editor is in the DS, usable by any module

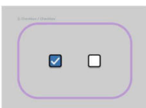
Guidelines & Tutorial
Onboarding help, self explanatory



The DS Should be directly usable by community
Create template file ready to distribute



Mockups Editable
After sharing with someone else, mockups should easily be editable (as opp. to image)



Dynamic States
Multi-state components (variants)

Multiple Themes
Be able to create results in both Light & Dark mode

Size adjustable
Components and boards should be adjustable in size (high impl. cost & fragile)

Introduction

Super Quick Warm-Up

60"

Quickly sketch the one object or moment that represents your biggest frustration in the design-to-development workflow.

Why a Design System?

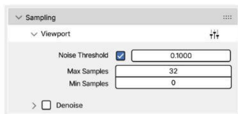
Theory: Design Capacity

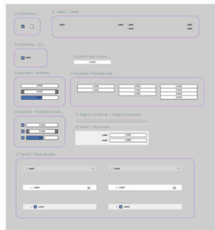
How can we build a shared system - a bridge - to close this gap, make our collaboration smoother, and unlock Blender's full design potential together?

→ through building infrastructure

Making it Real

The Proposed Design System Draft





Next Steps

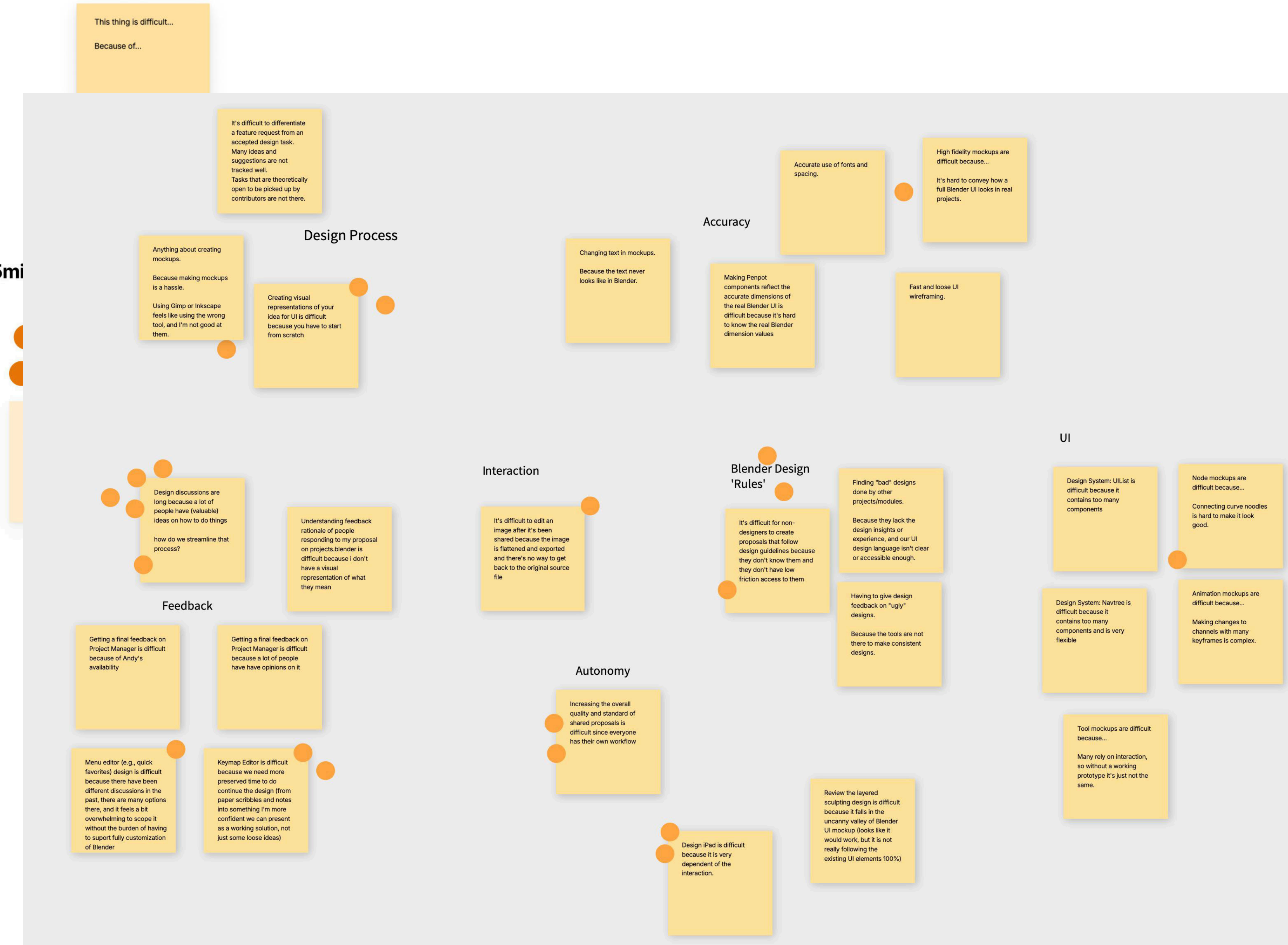
Next Steps

Defining Capabilities

Activity: Component Brainstorm (5min)

What are the most common, repeated, or frustrating UI components or tasks you have to design or build?

What makes that so frustrating?



Defining Capabilities

Activity: Component Prioritization (5mi

Which ones are most critical?
→ Dot voting

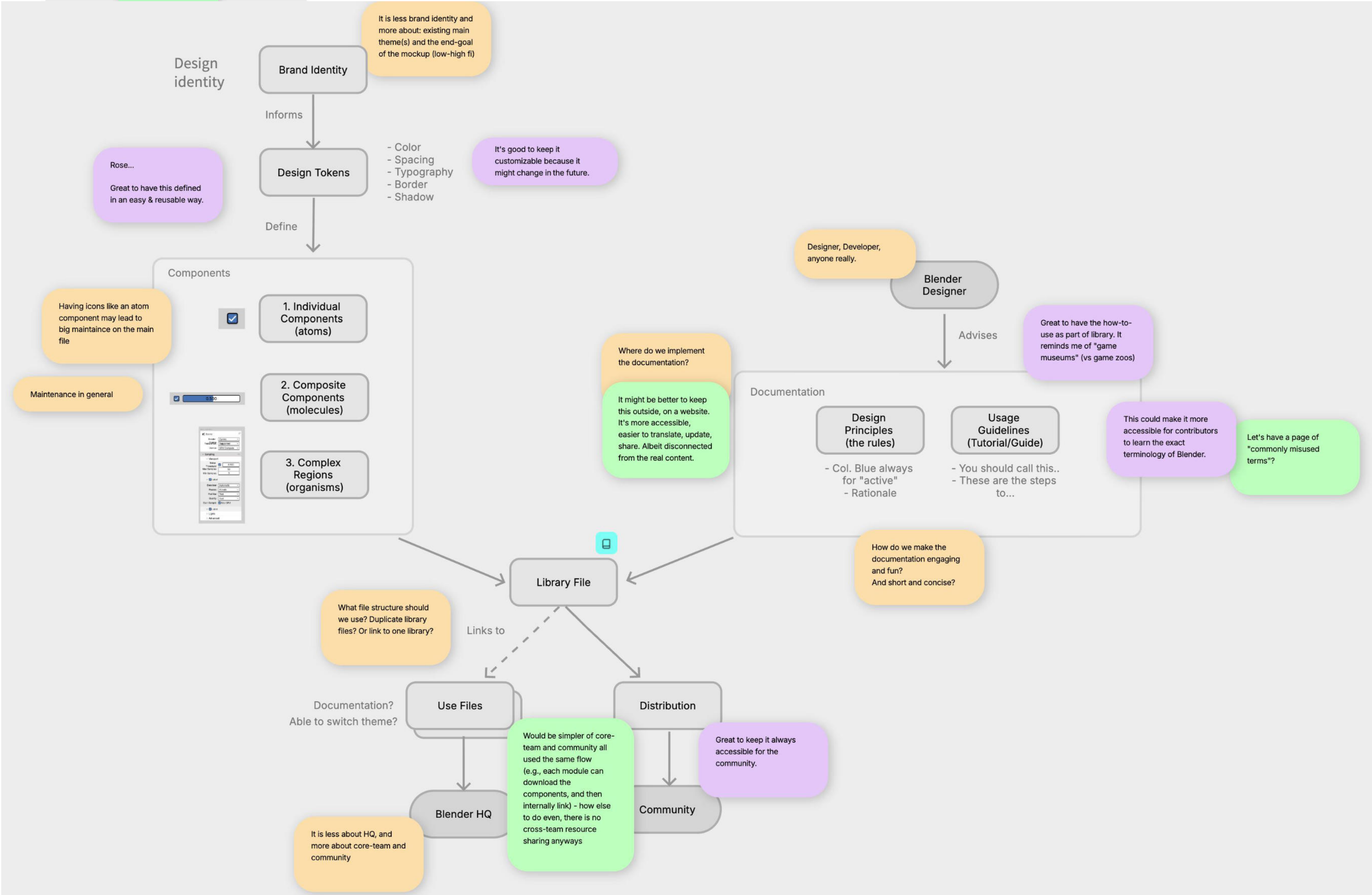
Dot Voting

Defining Capabilities

Activity: Rose, Thorn, Buds

- Rose (What's Good/Clear) → Keep this
- Thorn (What's Confusing) → Concern
- Bud (What's Missing/An Idea) → Potential

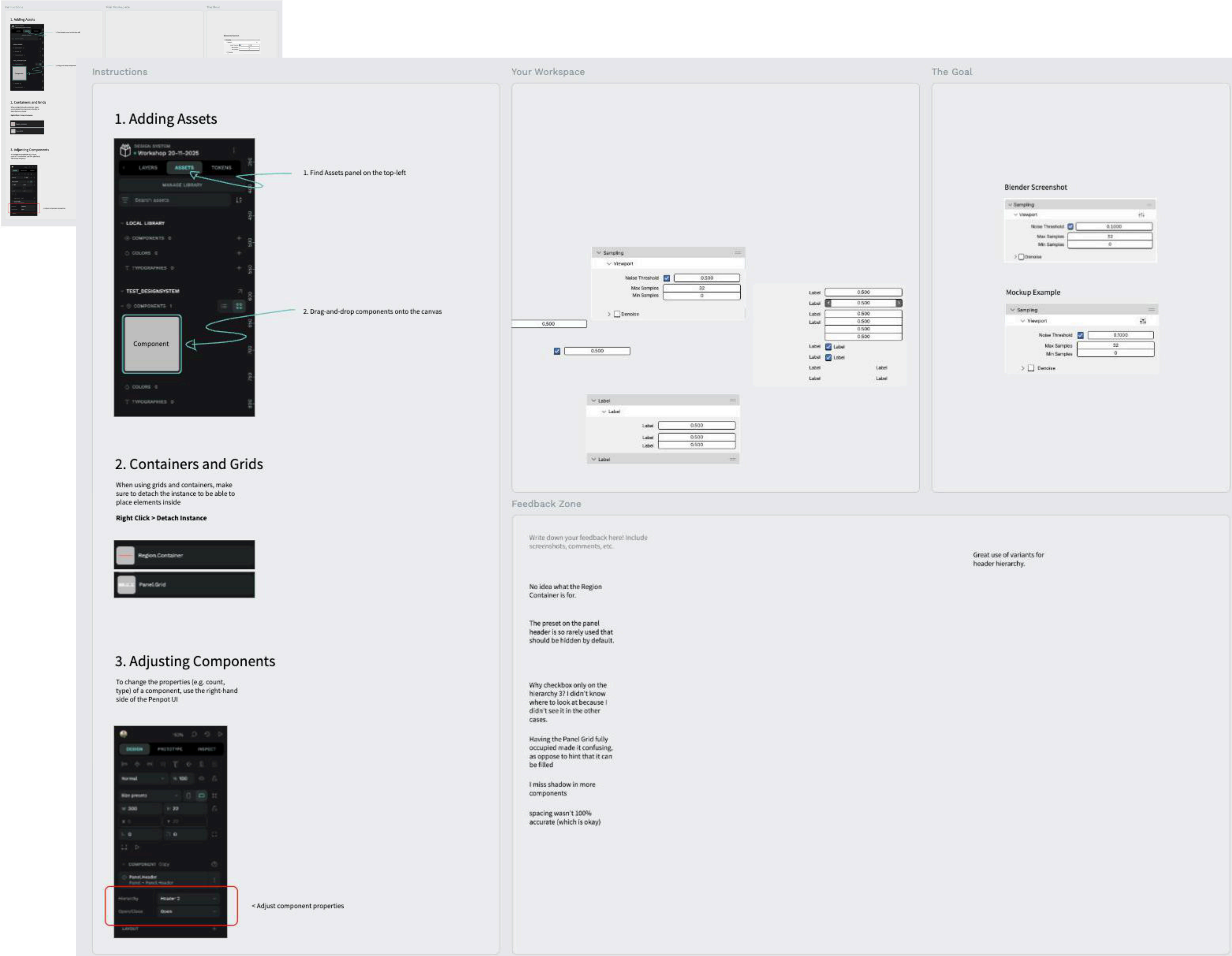
Rose, Thorn, Buds



Making it Real

Activity: Using The Design System (20 min)

Draft file-setup in Penpot, let's use it!



Appendix F: Validation Data

Questionnaire Statements & Raw Results

The validation questionnaire contained 12 statements (S1-S12) where participants could respond with a 1-7 score from strongly disagree (1), to strongly agree (7). Below is an overview of the statements and raw data of the results.

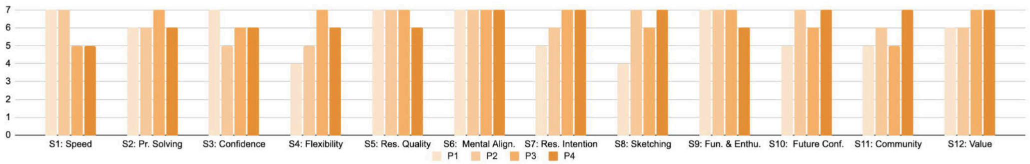


Table A.1 | Statement scores per participant

P1 & P2 are non-designers (developer & Blender artist), P3 & P4 are designers

Statement	P1	P2	P3	P4	Avg.
Section 1: Workflow Effectiveness &					
S1: "Using this system is significantly faster than my old method of creating UI mockups (e.g., screenshots/image editing/other)."	7	7	5	5	6
S2: "The design system allows me to focus more on creative problem solving and coming up with solutions. It makes me focus less on the technical details and graphic design of creating mockups"	6	6	7	6	6,25
Section 2: Autonomy					
S3: "I feel confident constructing a complex UI using these components, even without help."	7	5	6	6	6
S4: "The library provides enough flexibility to express my ideas without feeling restricted or overwhelmed"	4	5	7	6	5,5
S5: "I feel like I'm able to create a higher fidelity design proposal (mockup) by using the design system compared to not using it"	7	7	7	6	6,75
Section 3: Alignment & Shared Language					
S6: "The component names (e.g., layout.panel) align with my mental model of how Blender is built/coded."	7	7	7	7	7
S7: "If I received a proposal (a mockup) created with this system, I would have a clear understanding of the intended implementation."	5	6	7	7	6,25
Section 4: Working Experience					
S8: "The process of using this system feels more like 'sketching' than 'engineering'."	4	7	6	7	6
S9: "The process of using the design system feels fun and makes me enthusiastic to design for blender."	7	7	7	6	6,75
Section 5: Future Development & Expansion To Community					
S10: "I'm certain that the design system will continue to be used and further developed after Sem's thesis ends in February 2026."	5	7	6	7	6,25
S11: "If community contributors use this system, I expect their proposals will be easier to review and implement."	5	6	5	7	5,75
Section 6: Benefit To Blender					
S12: "I think that the design system positively contributes and adds value to the design workflow at Blender."	6	6	7	7	6,5
Avg. per Participant	5,83	6,33	6,42	6,42	

Table A.2 | Raw statement scores and averages

Qualitative Rationale Quotes

Below each statement participants had a text field where they could optionally enter their rationale about how they scored the statement. Below is an overview of these explanations per statement.

Note: some rationale explanations are missing since they were communicated verbally during the validation questionnaire interview, or simply not entered by the participant (when they didn't have a comment).

Section 1: Workflow Effectiveness & Speed

S1: "Using this system is significantly faster than my old method of creating UI mockups (e.g., screenshots/image editing/other)."

- "Because without a design system I have to start from scratch, and spend a lot of time on basic things and details." (P2)
- "The Design System is complete enough that I spend most of the time using it instead of building new required components. It is faster to re-iterate over the first design pass." (P3)
- "It's faster for certain tasks, but at the end of the day I'd probably still mix screenshots of the current UI, some of it changed via Python, and some with the design system, for the parts of the UI that are hard/impossible to do via Design System." (P4)

S2: "The design system allows me to focus more on creative problem solving and coming up with solutions. It makes me focus less on the technical details and graphic design of creating mockups"

- "high fildelity, speed of creating something completely new. low fid mockups would try another method: sketching whiteboard" (P1)
- "It definitely helps a lot. But in practice there are also design challenges that go far beyond graphic design and mockup cration." (P2)
- "It doesn't break the flow of creating and re-iterating. I can do it very easily and quickly with the design (assuming a design system that is complete enough). Even for special cases that rely on components that yet need to be built, I think preparing those components ahead of the design phase will help to keep the design phase fluid and continuos. I will also make an active efffort to only worry about making the then new components a part of the final collection after the design process is fully done." (P3)

Section 2: Autonomy

S3: "I feel confident constructing a complex UI using these components, even without help."

- "Would probably get stuck a few times when learning how to use DS" (P2)
- "I have to use them to build said confidence. So far we are still exploring the limits of what the system can do, and iteractively improving it accordingly. Looking back, I do feel confident I can build the complex UIs I needed before. Those cases are already incorporated in the system, and for those, I can work by myself. I'm yet to educate myself on how some of those components were built. I hope that once I do it (and once this is documented) I will feel confident to explore even the unknown new complex UIs with component I could build myself. "(P3)
- "It's clearly labeled and follows the naming conventions Blender already uses." (P4)

S4: "The library provides enough flexibility to express my ideas without feeling restricted or overwhelmed"

- "DS helps a lot to get things done with existing design language, but there might in reality be things you wnat to try that are not in the existing DS or language. Balance in how restrictive it should be." (P2)
- "Components like the "Screenshot Panel" mix the best of both worlds: a hacky quick screenshot interleaved with new or existing components. Besides that, Penpot itself is a sandbox creation tool, which allows me to "mix and match" the design system with native tools (e.g., it is easy to import a new icon from inkscape into penpot, and use it as an icon to a Design System component)." (P3)

S5: "I feel like I'm able to create a higher fidelity design proposal (mockup) by using the design system compared to not using it"

- "people online construct mockups with screenshots but it looks cheap and not right, DS helps a lot" (P1)
- "Without DS have to spend a lot of time on creating basic elements that make up the UI" (P2)
- "Have you seen the system? I've been showing some of my mockups already to collegues and their reaction is often like "this is just Blender?!" More than that, the prototyping (interactions) possible with Penpot are fully compatible with the Design System. This combination allows for unprecedent levels of high fidelity interactions not possible before. "(P3)

Section 3: Alignment & Shared Language

S6: "The component names (e.g., layout.panel) align with my mental model of how Blender is built/coded."

- "people familiar with terminology would work well, as well as new people since there's a visual right next to the label" (P1)
- "I'm biased because I was involved in some of those decisions. All in all I'm glad we did build a system that prioritizes the existing UI-building mental-model over some more stablished design names." (P3)

S7: "If I received a proposal (a mockup) created with this system, I would have a clear understanding of the intended implementation."

- "concern: making mistakes with UI layout, up to the person making it, DS helps with it though. Can be challenging with higher level blender ui mistakes (e.g. properties operators) - but it can also nudge them in the right direction" (P1)
- "For most basic layouts yes, but Still have to figure out how to hack around default layout. Unsure how much flexibility DS gives" (P2)
- "The high-fidelity mockups produced with this system are super close to Blender's UI. In fact, they allow for fine-tuned design decisions that are typically constrained by Blender's existing UI toolkit. Being able to mockup at that level allows for grounded proposals that can be built to match the look and feel of the mockups.Specially for the cases where the proposal includes the interaction flow. Which gives even less room for miscommunication." (P3)

Section 4: Working Experience

S8: "The process of using this system feels more like 'sketching' than 'engineering'."
- "would have to use it more to experience it" (P1)
- "It is fun to do. And I feel like I'm design something (as oppose to just put a UI together)." (P3)

S9: "The process of using the design system feels fun and makes me enthusiastic to design for blender."
- "There is something magical about being able to quickly build a UI, and adjust the naming, fine tune some things. Then switch between low/high fidelity to build some distance (to avoid the "falling in love with your design" problem), and going all the way to prototyping. It feels like you are dealing with a finished piece of software, that all it needs now is to magically be built :) It is overall very exciting and fun to use.(the only downside is that it doesn't have the same fun of doodling on a white board. But the moment the initial ideas are sketched out, then the computer step is less of a dread and almost just as fun)." (P3)

Section 5: Future Development & Expansion To Community

S10: "I'm certain that the design system will continue to be used and further developed after Sem's thesis ends in February 2026."
- "people using and putting effort into it" (P2)
- "I do hope that it is the case. My concerns are with: Maintainance (how will the system look in 2 years). Easy of use / adoption (how accessible is this to the team at large). Related to that, if this becomes a tool only for the design team how much it will be worth spending on maintaining and expanding it (I think it may still be worth it, but the team being small, it is always a balance between investing on tooling and doing the creative work of design). There is also the tiny gloomy scenario of the underlying technologuy (Penpot) not continuing to be developed. At the moment there are already some hidrances (bugs) that directly impact the experience. If this gets worst, it may get on the way." (P3)
- "It's already a huge help so it will only get better as we keep polishing it." (P4)

S11: "If community contributors use this system, I expect their proposals will be easier to review and implement."
- "depends on the person that creates mockup" (P1)
- "in case people propose designs that fall outside of existing lanauge" (P2)
- "The system (if used on its full potential) allows for a well-rounded design proposal. It forces the designer to think of a lot of the implications of the design (how will a feature affect other areas? how will this functionality be exposed to users, ...). In that sense, any design proposal with that much thought is easier to be reviewed. And I take that a proposal using this system would more naturally be more complete. That said, there is more to a design proposal than mockups. And this is not covered by the Design System at the moment (e.g., design principles, definition of target audience, framing of the problem, scope/deliverables breakdown, ...). Besides this, I think the design system allows for good proposals to be made better (thus easier to review and consequently implement). But it doesn't make a bad proposal better necessarily. It may help to more easily dismiss an incomplete proposal. But I'm yet to see how it will be adopted by the community." (P3)

Section 6: Benefit To Blender

S12: "I think that the design system positively contributes and adds value to the design workflow at Blender."
- "concern: everyone starts making same kind of mockups and starts thinknig 'inside the box'." (P2)
- "Blender has a history of a strong engineering culture. I think it is important to level the play-field for the design process. Investing time on tooling for the design process in Blender reflects the space we (I?) want design to take on the development process. The design system helps with: Sending a clear message to the team that we are raising the quality bar on the design mockups. Similarly, it shows that we are willing to "walk the walk" and treat design with as much importance as we want everyone else to treat. Finally it actually contributes with helping making mockups. It helps to make them more fun, and it *really* helps with the iteration process. This makes the design less of a mockup handover process and more a collaborative exploration." (P3)

Open Question

This open question followed at the end of the questionnaire after the 12 statements.

"What are your main concerns with using a design system compared to not using one?"
- "I hope people using the design system (me included) don't get too stuck on having to use it. It may sound paradoxical, but there are times that a simple sketch on a paper, or an edit screenshot may be the shortest path to a final design. There is also an implicit risk of being constraint by what is already possible, and neglect thinking a bit out-of-the box of the existing UI solutions. Specially given how much more convenient it is to mockup existing components. This is a bias developers have already, and it will be detrimental if designers run into this as well." (P3)
- "No concerns. It might make it harder to think outside the box to design new widgets, but we haven't had any new widgets in ages so I don't think it's something that happens too often. It's better to have this starting point anyway. It's awesome!" (P4)