Beyond the Double Diamond: Establishing Design Observability in Professional Practice

LENNARD LUKAS SCHMIDT✉, Google, Germany
JONAS GUNKLACH, Google, Germany
IVO BENKE, Google, Germany
CLARA KLIMAN-SILVER, Google, USA
FRANK BENTLEY, Google, USA

Human-Computer Interaction research and design pedagogy rely on idealized process models, such as the Double Diamond, to describe how user experiences are designed. These models assume an orderly, linear process that provides a useful macro-structure but fails to capture the highly iterative micro-loops of collaborative daily practice. While qualitative studies have successfully captured this complexity, analyzing these non-linear trajectories in real time and at a systemic scale remains a challenge. To understand how design unfolds in real products, we analyzed historical snapshots of 102 Figma files from Google and investigated the true trajectories of the design process at scale. Our analysis reveals that while the established process models might be applicable, the operational details are highly non-linear. Rather than a straight line from ideation toward completion, design advances are repeatedly reset to the ideation stage as feedback is received. We argue that by treating design files as operational telemetry, the industry can move beyond abstract frameworks to build practices and collaborative tools that support the non-linear realities of modern product development.

CCS Concepts: • **Human-centered computing → Empirical studies in collaborative and social computing**; **Collaborative and social computing systems and tools**.

ACM Reference Format:
Lennard Lukas Schmidt, Jonas Gunklach, Ivo Benke, Clara Kliman-Silver, and Frank Bentley. 2026. Beyond the Double Diamond: Establishing Design Observability in Professional Practice. In *Proceedings of The 29th ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW '26).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

For decades, HCI and CSCW scholars have investigated the design process to inform the development of superior design education, tools, and systems [13]. Process models such as the Five Planes of UX [6], the Double Diamond [3], or Design Thinking [7] are widely utilized across the design discipline. These models aim at describing design in a linear, predictable process from ideation to evaluation that, while easy to understand, only partially reflects the lived realities of professional design work [8], limiting its applicability to understanding design operations at scale and ultimately building tools that are sensitive to the varying requirements throughout the process.

Authors' Contact Information: Lennard Lukas Schmidt, lennardschmidt@google.com, Google, Munich, Germany; Jonas Gunklach, jonasgunklach@me.com, Google, Munich, Germany; Ivo Benke, ivobenke@google.com, Google, Munich, Germany; Clara Kliman-Silver, klimansilver@google.com, Google, New York, New York, USA; Frank Bentley, fbentley@google.com, Google, San Francisco, California, USA.

Manuscript submitted to ACM

Past research has shown that real-world design work rarely unfolds cleanly. Studies of professional design practice repeatedly reveal complexity, iteration, and backtracking as central features [e.g., 10, 11, 16]. Even experienced teams work in non-linear ways, using prototypes not only for evaluation but for reframing problems and exploring alternative directions [4]. Consequently, the initial Double Diamond model was criticized for its lack of explicit iterative paths [12], a shortcoming that the Design Council later addressed by adding arrows and loops to its framework [3].

Understanding how design unfolds in detail requires moving beyond how we subjectively conceptualize it. While qualitative studies utilizing interviews, surveys, and case studies provide essential insights into designer intent [e.g., 5, 9], these methods alone struggle to track the complete, non-linear sequence of actions across large, systemic scales. Large-scale, artifact-based studies in professional contexts remain rare, aligning with Wolf et al.'s [15] call to rigorously study design's exploratory, iterative nature. While software engineering has long embraced observability to monitor and optimize complex systems [1], we define design observability as the continuous, data-driven analysis of this telemetry to empirically track, understand, and support the realities of design operations at scale. With the advent of digital design tools, such as Figma, practitioners now have access to the required repository of design telemetry, such as the version histories, comments made, component usages, and layout architectures embedded in the archived history.

Building on this data we conducted a longitudinal analysis of 102 design files from diverse products at Google and contribute in two key ways: 1) providing empirical evidence that, while the design process moves through established macro-phases, the operational micro-transitions between states are highly non-linear and self-looping, and 2) demonstrating a novel methodological approach that leverages digital design telemetry to model these states, offering a blueprint for organizations to build context-aware tools and processes that support actual design operations.

2 Methodology

Our approach combines design file telemetry data (e.g., node churn per version, component-to-instance ratios, and comment density) with manual coding. First, using the Figma API, we programmatically extracted auto-generated version checkpoints (captured whenever a designer manually saves or after 30 minutes of inactivity) to capture historical snapshots of 102 design files. To ensure we observed the complete design lifecycle, we randomly sampled these files from feature design projects across various Google products, restricting our selection to those tracing the end-to-end development of a new product, feature, or component. Second, two researchers manually coded the resulting 4,036 snapshots (average of 39 snapshots per file; inter-coder reliability = 0.86) using an adapted version of the UX design workflow outlined by [9]. To track design file activity, we focused on the core phases where designers actively manipulate the digital canvas (i.e., Ideation, Prototyping, User Testing, and Handoff) while adding an initial Setup phase. Finally, we trained a Hidden Markov Model [14] using the telemetry data as observations to predict the manually coded phase labels. This approach enabled us to 1) identify which objective signals correlate with specific design states and 2) to model the transition probabilities between them which allows us to predict the current and future stages of a design project in extant work.

3 Key Insights

Classical design theory proposes an idealized, linear process. However, extracting state-transition probabilities from our telemetry data (Fig. 1) reveals four characteristics that define a highly non-linear workflow:

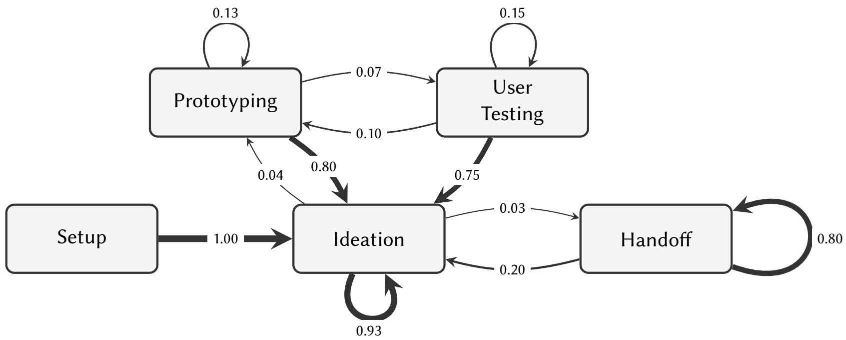

Fig. 1. State-transition pathways derived from 102 longitudinal product design lifecycles. Numbers indicate non-zero transition probabilities between design phases (adapted from [9]).

- **Setup is the only strictly linear phase.** The process begins with absolute predictability. Setup phases transition directly into Ideation 100% of the time in our dataset. The sole instance in our model guarantees forward progression without backtracking, divergence, or self-looping.
- **Ideation acts as a gravitational hub.** Rather than a transient stepping stone, Ideation is highly self-sustaining with a 93% self-loop probability. It serves as the universal baseline and rarely advances directly to testing or handoff, acting instead as the central anchor that all subsequent phases eventually collapse back into.
- **Evaluation triggers macro-structural resets.** Extant research suggests prototyping and user testing are deeply interconnected [2], but our data shows transitions between the two are rare. Instead, both stages overwhelmingly trigger disruptive resets back to abstract Ideation (80% from Prototyping; 75% from User Testing). Evaluation may force teams to deconstruct their conceptual architectures and restart.
- **Handoff is a stable but porous conclusion.** While Handoff serves as a relatively stable terminal state (80% self-loop probability), it does not guarantee finality. Even after achieving production readiness, files regress directly back to Ideation 20% of the time potentially due to addressing technical constraints or late-stage developer feedback.

4 Discussion

The core contribution of this work to the CSCW and industry communities is demonstrating how we can quantify the design process at scale, and its numerous loops and collaborative processes over time, such as peer feedback, user testing, and handoff with a developer. We find that while the idealized design process exists from afar, we need to attend to the messy details to build better tooling and enable healthy collaboration. To this end, we suggest three distinct recommendations.

First, design can adopt the same rigor that the broader field of software engineering has embraced in DevOps ([1]; e.g., tracking deployment velocities, pull request cycles, and code churn). By establishing telemetry pipelines over design files, cross-functional leadership can gain an empirical understanding of product development. Instead of relying on a designer's subjective update during a sync meeting, such systems can automatically flag that a file has experienced a sudden surge in component churn or a re-emergence of placeholder text, programmatically signaling that the project has undergone an unexpected reset. Collaborative partners such as developers or project managers could be notified that there may be larger shifts to the timeline because of these edits.

Second, current collaborative design platforms are fundamentally architected to support linear construction. Tools such as Figma treat design history as a chronological timeline of edits. This mismatch may cause friction in collaboration as it fails to reflect the iterative leaps and structural resets inherent to the design process. Tools should ideally be sensitive to the stage a design file is in and where it will go next to maximize helpfulness (e.g., when a file enters the prototyping stage, the tool could anticipate the likely structural reset by automatically provisioning an adjacent ideation canvas to capture the resulting pivot). Systems can capture each of the "dead ends" in design which might be useful in a review to show paths not taken and explain the reasoning or explicitly highlight the current "best" version of a design to draw comments and attention to the very latest.

Third, understanding the objective reality of the design process can help us to design better Generative UX tools. While AI agents can generate hundreds of changes to a design file in seconds, understanding if and where these capabilities help or hinder the design process is imperative. Design agents can be prompted to follow similar exploratory paths, sending in-progress designs for evaluation or user-testing and looping back to ideation for bigger resets. Design Agents should also realize that handoff and implementation is a loop as well, where often the design must change because of what can or cannot be built or alignment with brand or design system guidelines.

While our telemetry reveals a highly non-linear workflow, it is crucial to contextualize these findings within the boundaries of our instrumentation and sampling strategy. First, Figma is predominantly an environment optimized for design iteration and exploration, while evaluative testing and developer execution often occur in entirely separate platforms. As a result, the tool naturally captures upfront exploration far better than it captures final implementation. Second, the state transition probabilities in our model are highly dependent on the organizational characteristics of the design practice we observed. In reality, design practices vary widely and teams might distribute workflows across multiple files, or use a single file for an isolated feature update rather than a full product trajectory. Therefore, our findings do not refute the existence of a broader, linear macro-process, nor do they claim to comprehensively define every variation of design practice. Rather, they provide empirical evidence of the chaotic, cyclical micro-processes required to turn abstract requirements into handoff-ready artifacts within a single workplace, yet one with largely different software development practices across teams.

5 Open Questions

We conclude our perspective by posing three open questions to the CSCW and collaborative tool-design communities:

(1) **How do we design interfaces that support non-linear workflows without introducing cognitive overload?** If returning to earlier states of abstraction is a healthy property of professional work, how can tools help teams visualize, track, and navigate across multiple historical design trajectories simultaneously?
(2) **How can tracking telemetry be used to support creativity?** As we move toward design observability, how can we ensure that tracking metrics like node churn or iteration velocity is used to improve collaborative tools and team resourcing, rather than serving as a micromanagement tool that penalizes creative exploration?
(3) **How should companies prepare for the exponential churn of Generative AI?** If ideation already acts as a gravitational hub with a 93% self-loop probability, how will collaborative design tools prevent AI from permanently trapping teams in infinite generative loops or in pushing initial ideas to production without proper reflection and user feedback? How can observability frameworks distinguish between intentional human resets and automated AI canvas noise?

References

[1] Titus Barik, Robert DeLine, Steven Drucker, and Danyel Fisher. 2016. The Bones of the System: A Case Study of Logging and Telemetry at Microsoft. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*. 92–101.

[2] Jonathan Chen and Dongwook Yoon. 2024. Exploring the Diminishing Allure of Paper and Low-Fidelity Prototyping Among Designers in the Software Industry: Impacts of Hybrid Work, Digital Tools, and Corporate Culture. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 1024, 14 pages. doi:10.1145/3613904.3642774

[3] Design Council. 2019. Framework for Innovation. https://www.designcouncil.org.uk/resources/framework-for-innovation/. Accessed: Jun 20, 2025.

[4] Steven Dow, Julie Fortuna, Dan Schwartz, Beth Altringer, Daniel Schwartz, and Scott Klemmer. 2011. Prototyping Dynamics: Sharing Multiple Designs Improves Exploration, Group Rapport, and Results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2807–2816. doi:10.1145/1978942.1979359

[5] K. J. Kevin Feng, Tony W. Li, and Amy X. Zhang. 2023. Understanding Collaborative Practices and Tools of Professional UX Practitioners in Software Organizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1–20. doi:10.1145/3544548.3581273

[6] Jesse James Garrett. 2010. *The Elements of User Experience: User-Centered Design for the Web and Beyond* (2nd ed.). New Riders Publishing, Berkeley, CA.

[7] Sarah Gibbons. 2016. Design Thinking 101. https://www.nngroup.com/articles/design-thinking/.

[8] Colin M. Gray. 2016. "It's More of a Mindset Than a Method": UX Practitioners' Conception of Design Methods. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 4044–4055. doi:10.1145/2858036.2808444

[9] Jie Li, Hancheng Cao, Laura Lin, Youyang Hou, Ruihao Zhu, and Abdallah El Ali. 2024. User Experience Design Professionals' Perceptions of Generative Artificial Intelligence. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 381, 18 pages. doi:10.1145/3613904.3642114

[10] Youn-Kyung Lim, Erik Stolterman, and Josh Tenenberg. 2008. The Anatomy of Prototypes: Prototypes as Filters, Prototypes as Manifestations of Design Ideas. *ACM Transactions on Computer-Human Interaction (TOCHI)* 15, 2 (2008), 1–27. doi:10.1145/1375761.1375762

[11] Alexandros Nikolaos Lotsos, Yizhu Wang, and Michael Stephen Horn. 2025. Conceptions of Design Practice From Academy to Industry: Implications for HCI and Design Education. In *Proceedings of the 7th Annual Symposium on HCI Education (EduCHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 3, 12 pages. doi:10.1145/3742901.3742905

[12] Dan Ramsden. 2023. The Limitations of the Double Diamond. https://danramsden.com/2023/07/26/the-limitations-of-the-double-diamond/. Accessed Sep 9, 2025.

[13] David J. Roedl and Erik Stolterman. 2013. Design Research at CHI and its Applicability to Design Practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1951–1960. doi:10.1145/2470654.2466251

[14] Harold Thimbleby, Paul Cairns, and Matt Jones. 2001. Usability Analysis with Markov Models. *ACM Transactions on Computer-Human Interaction (TOCHI)* 8, 2 (2001), 99–132. doi:10.1145/376929.376941

[15] Tracee Vetting Wolf, Jennifer A. Rode, Jeremy Sussman, and Wendy A. Kellogg. 2006. Dispelling "Design" as the Black Art of CHI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 521–530. doi:10.1145/1124772.1124853

[16] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. Research Through Design as a Method for Interaction Design Research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 493–502. doi:10.1145/1240624.1240704