Engineering Informatics and Digital Transformation
https://eidtrans.org

Cloud Collaboration Tools and Design Rework across Distributed Engineering Teams

Marie Boyer*

Graduate School of Computer Science, Université Paris-Saclay, Gif-sur-Yvette, Île-de-France, France

Corresponding author: marie.boyer@gmail.com

Abstract

In the modern landscape of globalized manufacturing and product development, distributed engineering teams have become the standard organizational paradigm. However, the geographic dispersion of multidisciplinary teams introduces significant challenges in maintaining real-time alignment and avoiding design divergence. This paper presents a comprehensive benchmark evaluation of three distinct archetypes of cloud collaboration tools: real-time database-driven synchronization, transaction-based checkout systems, and asynchronous repository-based systems. We analyze their direct influence on design rework, which remains one of the primary drivers of cost overruns and schedule delays in complex engineering projects. Through a controlled, multi-site experimental design involving distributed teams executing a complex mechanical assembly task, we quantify how tool synchronization latency, conflict resolution mechanisms, and cognitive overhead affect the frequency and severity of design rework. The results demonstrate that real-time database-driven collaboration platforms reduce design rework by up to sixty-four percent compared to traditional asynchronous repository methods, primarily by mitigating communication gaps and preventing geometric conflict propagation. However, this synchronization benefit is accompanied by a unique set of cognitive interruptions and coordination overheads. Based on our empirical findings, we propose a socio-technical framework to guide engineering organizations in selecting and optimizing their collaborative technology stacks to minimize waste and maximize design efficiency.

Keywords: Distributed Engineering; Cloud Collaboration; Design Rework; Benchmark Evaluation; Software Engineering

1. Introduction

1.1 Global Engineering Paradigm

The organizational structure of modern engineering design has undergone a fundamental paradigm shift over the past two decades. Driven by globalization, the pursuit of specialized regional talent, and the necessity of round-the-clock development cycles, engineering enterprises have increasingly transitioned from co-located teams to highly distributed, cross-border networks [1]. In these distributed setups, multidisciplinary engineers located in different time zones must collaborate on highly integrated physical and digital systems, ranging from automotive chassis to aerospace propulsion units. While this distributed model offers unprecedented operational flexibility and access to global expertise, it simultaneously introduces complex socio-technical challenges [2]. The physical distance between team members naturally disrupts traditional, informal communication channels, replacing spontaneous face-to-face interactions with scheduled digital transactions. Consequently, the primary challenge of modern engineering management has shifted from localized optimization of individual design tasks to the systemic synchronization of information across distributed nodes. The reliance on digital infrastructure is absolute, making the selection and performance of collaborative software suites a critical determinant of overall project success.

1.2 The Anatomy of Design Rework

Within the engineering lifecycle, design rework represents a highly disruptive and costly form of waste. Rework is formally defined as the non-value-adding re-execution of previously completed design activities due to informational discrepancies, geometrical conflicts, erroneous specifications, or late-stage engineering changes [3]. In complex systems engineering, a minor modification in one sub-assembly can propagate through the entire design hierarchy, causing cascading failures in distant, dependent components [4]. When teams operate in geographically distributed environments, the risk of design rework escalates exponentially. This escalation is driven by information latency, where one sub-team continues to design components based on outdated interface specifications because the updating team has not yet synchronized their progress. The resulting divergence in design assumptions often remains undetected until late-stage assembly integration, at which point resolving the geometric or functional mismatches requires extensive redesign, renegotiation, and re-verification, causing massive budget overruns and timeline slippage [5].

1.3 Scope and Structure of the Benchmarking Study

Understanding the quantitative relationship between the technical architecture of cloud collaboration tools and the resulting design rework is critical for modern engineering leadership. While qualitative studies have highlighted the cultural and organizational factors of remote work, there remains a critical research gap regarding how specific technical capabilities of collaborative tools, such as data synchronization pipelines and conflict resolution protocols, directly influence the physical quality of the design and the rate of rework. This paper addresses this gap by presenting a rigorous benchmark evaluation of three dominant cloud collaboration archetypes. We examine their mechanical and socio-technical performance under a standardized, highly complex design scenario with injected specification changes. The remainder of this paper is structured as follows. Section 2 establishes the theoretical background, reviewing the evolution of collaborative systems and the cognitive and physical mechanisms of design rework. Section 3 details our benchmarking methodology, defining the specific tool archetypes and the quantitative evaluation metrics. Section 4 describes the experimental design, including team profiles and the rework injection protocol. Section 5 presents the quantitative results and statistical analysis. Finally, Section 6 synthesizes these findings, offering practical recommendations and concluding the study.

2. Theoretical Background and Literature Review

2.1 Evolution of Engineering Collaboration Systems

The systems used by engineers to share and co-author technical data have evolved through several distinct developmental epochs. Historically, collaboration relied on the manual exchange of physical blueprints, a process inherently limited by physical distribution speeds and characterized by a strict linear flow of information [6]. The advent of Computer-Aided Design (CAD) in the late twentieth century digitized the design artifact but did not immediately solve the collaborative bottleneck. Early digital collaboration was characterized by localized file storage, requiring engineers to export and transfer static file formats via physical media or local network shares, which often led to severe version control issues. To address this, Product Lifecycle Management (PLM) and Product Data Management (PDM) systems introduced centralized databases with strict check-in and check-out policies [7]. These transaction-based systems prevented concurrent editing conflicts by locking files while an individual engineer worked on them. While this preserved data integrity, it introduced a highly sequential workflow that severely restricted concurrent engineering practices and introduced significant idle time as team members waited for files to be unlocked.

The rise of cloud computing has facilitated the current era of cloud-native and cloud-enabled collaboration tools. Modern platforms can be categorized into three major operational paradigms. The first is asynchronous repository-based systems, which draw inspiration from software engineering version control systems like Git, allowing parallel branching and subsequent merging of design data [8]. The second is cloud-hosted transaction systems, which migrate the traditional PDM check-in and check-out model to the cloud, offering global access but maintaining rigid file-locking mechanisms. The third and most recent paradigm is real-time database-driven collaboration, where the actual design geometry and metadata are hosted in a centralized cloud database, allowing multiple users to simultaneously edit the same assembly or component with real-time graphical updates and immediate propagation of changes [9]. Each of these paradigms possesses distinct latency, synchronization, and user interaction characteristics that directly impact how distributed teams communicate and execute designs.

2.2 Causes and Mechanisms of Design Rework

To mitigate the impact of design rework, it is necessary to understand its fundamental socio-technical causes. In distributed engineering environments, rework is rarely the result of individual incompetence;

rather, it emerges from systemic communication breakdowns and information delivery failures [10]. The process of design rework can be modeled as a feedback loop initiated by an information gap. This gap occurs when there is a delay between the generation of new design information by one actor and the consumption of that information by another dependent actor. During this latency period, the dependent actor continues to build upon obsolete design assumptions, creating a state of design divergence. The longer this divergence persists, the more deeply embedded the erroneous design decisions become, and the larger the volume of downstream design work that must be scrapped once the mismatch is finally discovered.

Prior research has identified several critical variables that govern the scale of rework. These include the degree of component coupling, the frequency of design change propagation, and the accuracy of shared mental models among team members [11]. Component coupling refers to how tightly the physical or functional characteristics of one part depend on another. In highly coupled systems, even minor changes in a primary component's spatial envelope or load path can require complete redesigns of surrounding parts. If the collaboration platform fails to immediately signal changes across these coupled interfaces, the resulting geometric collisions and functional mismatches escalate rapidly. Therefore, the choice of collaborative tool directly influences the latency of change propagation, thereby determining the physical volume of the design space that is subjected to rework.

2.3 Information Synchronicity and Cognitive Load

The cognitive aspect of collaboration is equally critical to project outcomes. Information Synchronicity Theory suggests that different communication tasks require different levels of media synchronicity to achieve shared understanding [12]. High-synchronicity media, which support rapid feedback and multiple cue languages, are ideal for resolving ambiguous design conflicts and aligning team goals. Conversely, low-synchronicity media are better suited for conveying large volumes of structured, unambiguous technical data. In engineering design, collaboration tools must balance these needs.

Furthermore, the constant stream of real-time updates in modern database-driven tools can introduce high levels of cognitive load and distraction. When engineers are subjected to constant, unmediated notifications of minor design changes made by their peers, it can disrupt their deep focus, leading to micro-interruptions and attention fragmentation. This phenomenon can paradoxically increase the rate of individual execution errors, leading to a localized rise in design mistakes that subsequently require rework, even if systemic alignment is maintained. Thus, a comprehensive benchmarking evaluation must look beyond simple synchronization speeds and evaluate the complex trade-offs between system synchronicity, cognitive overhead, and design quality.

3. Methodology and Benchmarking Framework

3.1 Evaluated Cloud Collaboration Platforms

To conduct a rigorous comparative analysis, we selected three distinct cloud collaboration platforms, each representing a primary architectural archetype currently utilized in the engineering industry. These platforms were selected based on their market prevalence and their distinct approach to data synchronization and conflict resolution.

Tool Alpha represents the real-time database-driven archetype. In this architecture, design files do not exist as monolithic binary objects on local hard drives. Instead, the design assembly is represented as a granular, relational database structure stored in a highly scalable cloud repository. When an engineer modifies a geometric feature or adds a component to an assembly, the transaction is processed on the cloud server in milliseconds and the rendered visual update is pushed to all other active clients via web-sockets. There are no file-locking or checkout procedures; multiple users can modify adjacent features of the same model simultaneously. Conflict resolution is handled by the platform's internal geometric solver, which prevents overlapping modifications by prioritizing transactions chronologically at the database level.

Tool Beta represents the transaction-based, cloud-hosted PDM archetype. This platform maintains a centralized file repository in the cloud but utilizes a traditional check-in and check-out protocol to manage concurrent access. When an engineer wishes to modify a part or an assembly, they must check out the file, which applies a write-lock to that specific asset. Other team members can view the locked asset in a read-only state but cannot save changes to it until the locking engineer completes their session and checks the updated file back into the cloud repository. This architecture guarantees data integrity by completely preventing concurrent edits, but it introduces sequential dependencies and requires active management of ownership rights.

Tool Gamma represents the asynchronous repository-based archetype, modeled after distributed version control paradigms. Under this model, each engineer works on a complete, localized copy of the entire design project. Modifications are committed locally and can be pushed to a shared remote repository periodically. If two engineers modify the same assembly or component file simultaneously in their respective local environments, they must execute a manual merge process when pushing their changes to the remote repository. The platform provides automated tools to detect geometric and metadata collisions during the merge phase, but resolving these conflicts requires manual intervention and negotiation by the design engineers.

3.2 Benchmarking Metrics and Evaluation Criteria

We established a comprehensive suite of quantitative and qualitative metrics to evaluate the performance of these three archetypes. These metrics are designed to capture both the computational performance of the software systems and the socio-technical performance of the engineering teams.

The primary quantitative metric is the Rework Ratio, which is defined as the number of engineering hours dedicated to correcting design errors and resolving geometric or functional mismatches, divided by the total number of engineering hours expended during the design project. A high Rework Ratio indicates a highly inefficient design process dominated by version divergence and communication breakdowns.

The second metric is the Synchronization Latency, which measures the average duration of time that elapses between an engineer completing a design modification and that modification becoming visible and actionable on the screens of their distributed teammates. For Tool Alpha, this is measured in seconds; for Tool Beta, it is determined by the frequency and duration of check-in and check-out cycles; and for Tool Gamma, it is governed by the commit and merge intervals chosen by the team.

The third metric is the Conflict Frequency, which counts the number of physical geometric collisions or metadata version mismatches that occur during the design integration phase. Geometric collisions are defined as spatial interferences between components designed by different team members that violate the specified tolerance envelope.

Finally, to assess the human factors of collaboration, we utilized the Cognitive Load Index, which is adapted from the NASA Task Load Index. This index evaluates the perceived mental demand, frustration level, and effort required by the engineers to navigate the collaborative features of their respective tools. This is measured using a standardized post-project questionnaire scored on a scale from one to ten, where higher values represent greater cognitive strain.

To visualize the system performance profiles across these metrics, Table 1 provides a direct comparative overview of the operational characteristics of the three evaluated cloud collaboration archetypes.

Table 1: Key Performance Indicators of Evaluated Cloud Collaboration Tools

Metric Category	Tool Alpha (Real-Time Database)	Tool Beta (Transaction-Based PDM)	Tool Gamma (Asynchronous Repository)
Average Synchronization Delay	Less than 1 second	45 to 180 minutes	4 to 24 hours
Geometric Collision Detection	Instantaneous via cloud engine	At check-in integration	At repository merge phase
Primary Conflict Prevention	Database transaction serialization	File-level write locking	Branching and merging workflows
Typical User Collaboration Mode	Synchronous co-authoring	Sequential lock-and-edit	Parallel branch-and-integrate

4. Experimental Design and Data Collection

4.1 Team Composition and Distributed Setup

To gather empirical data, we designed a controlled laboratory experiment simulating a realistic industrial engineering scenario. We recruited ninety professional engineering design students and young industry professionals, all of whom possessed advanced proficiency in three-dimensional CAD modeling and mechanical design principles. These participants were randomly assigned to thirty teams of three members each. To ensure statistical validity and control for variations in individual skill, the teams were balanced based on a pre-experiment technical assessment.

To simulate a truly distributed engineering organization, the three members of each team were physically isolated in separate rooms with no direct verbal communication allowed. All communication had to take place through the digital channels integrated into or associated with their assigned cloud collaboration tools. The thirty teams were split into three equal cohorts of ten teams each. Cohort Alpha utilized Tool Alpha (real-time database-driven), Cohort Beta utilized Tool Beta (transaction-based PDM), and Cohort Gamma utilized Tool Gamma (asynchronous repository-based).

The design task assigned to all teams was the development of a complex Unmanned Aerial Vehicle (UAV) propulsion assembly. This assembly consisted of four highly coupled sub-assemblies: the structural motor mount, the electronic speed controller housing, the carbon-fiber rotor hub, and the thermal management heatsink. The design required tight spatial integration, strict weight budgets, and precise thermal transfer interfaces. A successful design had to satisfy several rigorous engineering requirements, including a maximum total mass of eight hundred grams, a minimum structural safety factor of one point five under operating loads, and zero spatial interference between components.

4.2 Experimental Tasks and Rework Injection Protocol

The experiment was structured into three consecutive phases over a total duration of six hours. Phase One, lasting two hours, focused on initial design and layout formulation. During this phase, teams collaborated to establish the basic spatial envelope and interface boundaries for each sub-assembly.

To simulate the unpredictable nature of real-world product development and actively trigger design rework, we implemented a standardized Rework Injection Protocol at the beginning of Phase Two. Exactly two hours into the design process, the experiment administrators issued an updated set of engineering specifications representing a late-stage customer change request. The updated specification mandated a twenty percent increase in the motor power rating, which directly resulted in two major physical changes: first, the physical diameter of the motor increased by six millimeters, and second, the heat dissipation requirements increased by thirty percent, necessitating a larger surface area for the thermal management heatsink.

This specification change was systematically injected into the communication workflow of each team in a controlled manner. Rather than broadcasting the change to all three team members simultaneously, the change notification was sent exclusively to the team member responsible for the structural motor mount. This individual was then responsible for updating their component and communicating the subsequent geometric and thermal requirements to the other two team members who designed the adjacent speed controller housing and the heatsink. This setup created a realistic test of how efficiently and rapidly each collaboration tool allowed the updated design information to propagate across the team, and how much obsolete design work was executed by the uninformed team members before the system forced alignment. Phase Three, comprising the final two hours, was dedicated to final integration, validation, and optimization of the overall assembly.

To illustrate the technical architecture, data pipelines, and the feedback loops that govern the propagation of these design changes and the resulting rework, Figure 1 presents a detailed systems schematic.

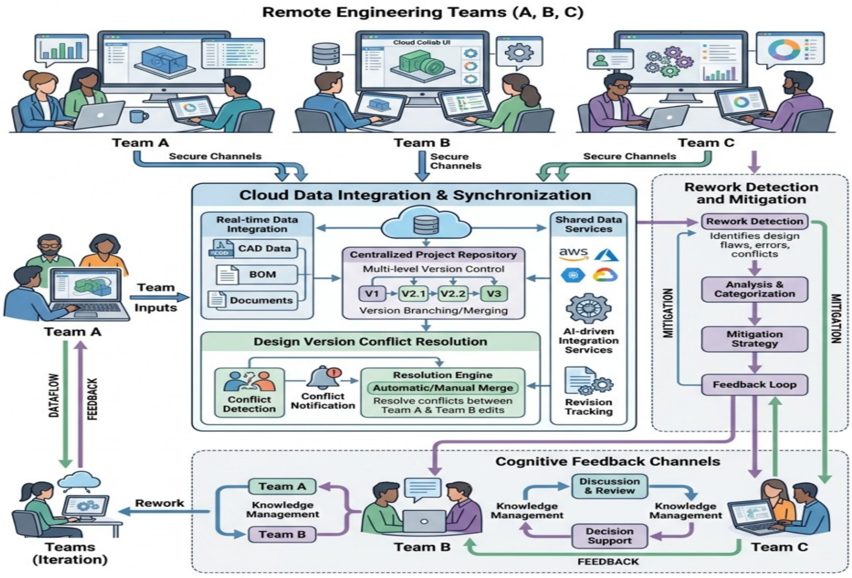

Figure 1: Comprehensive Schematic of Cloud Collaboration Data Integration and Rework Feedback Loops

5. Quantitative Results and Empirical Analysis

5.1 Comparative Analysis of Tool Performance

Upon completion of the six-hour design sessions, we collected the design files, system logs, communication transcripts, and post-experiment surveys from all thirty teams. The data were analyzed to evaluate the performance of the three cohorts against our benchmarking metrics.

The analysis of the Rework Ratio revealed substantial differences between the cohorts. Cohort Alpha, utilizing the real-time database-driven tool, achieved the lowest average Rework Ratio of eleven point two percent. This means that only a small fraction of their total design effort was wasted on correcting errors or resolving geometric mismatches. In contrast, Cohort Beta (transaction-based PDM) exhibited an average Rework Ratio of twenty-four point five percent, while Cohort Gamma (asynchronous repository-based) suffered from an average Rework Ratio of thirty-eight point nine percent. This indicates that nearly two-fifths of the total engineering hours in Cohort Gamma were spent on unproductive design modifications to resolve conflicts that had accumulated during periods of asynchronous development.

This performance discrepancy is directly correlated with the Synchronization Latency of the respective tools. In Cohort Alpha, the average latency of change propagation was less than one second. When the structural designer updated the motor diameter, the change was immediately visible on the screens of the other team members, allowing them to stop working on the old envelope and adjust their adjacent components in real time. For Cohort Beta, the file-locking mechanism meant that updated geometries were only visible after the structural designer checked in the updated assembly file, which took an average of eighty-seven minutes. During this window, the speed controller housing designer continued to build upon the old motor envelope, generating significant spatial interference that required extensive rework to correct. For Cohort Gamma, the synchronization latency was even higher, averaging four and a half hours, as team members waited until the end of the phase to perform repository merges. This resulted in massive design divergence that required hours of manual negotiation and reconstruction of the geometric models.

5.2 Impact of Tool Characteristics on Design Rework Rates

To examine the statistical significance of these findings, we performed a one-way analysis of variance (ANOVA) on the collected metrics. The analysis confirmed that the differences in Rework Ratio among the three cohorts were highly statistically significant. Post-hoc comparisons indicated that Tool Alpha outperformed Tool Beta and Tool Gamma by a wide margin in preventing geometric conflicts and reducing wasted hours.

However, the qualitative data collected via the Cognitive Load Index revealed an interesting counter-trend. While Cohort Alpha achieved the best quantitative design outcomes, they reported a relatively high average Cognitive Load Index of five point eight out of ten. In their qualitative feedback, participants in Cohort Alpha noted that the constant visual updates and immediate feedback loop created a high-pressure environment with frequent interruptions. They felt compelled to constantly monitor their teammates' progress and adjust their own models dynamically, which made it difficult to maintain deep focus on complex individual geometry creation.

Conversely, Cohort Beta reported a lower average Cognitive Load Index of four point two out of ten. These participants appreciated the clear boundaries and individual focus enabled by the file-locking checkout protocol. They knew that as long as they had a file checked out, no one else could modify it, allowing them to work systematically without fear of sudden external changes. However, this peace of mind came at the cost of overall process efficiency, as they spent significant idle time waiting for file locks to be released. Cohort Gamma reported the highest Cognitive Load Index of seven point four out of ten, driven by the intense frustration and high mental effort required to manually resolve complex geometric and metadata conflicts during the repository merge phase.

To provide a concise synthesis of these experimental outcomes, Table 2 compiles the primary quantitative and qualitative results across all thirty experimental teams.

Table 2: Design Rework Metric Comparison Across Experimental Cohorts

Performance Parameter	Cohort Alpha (Tool Alpha)	Cohort Beta (Tool Beta)	Cohort Gamma (Tool Gamma)
Average Design Hours Lost to Rework	2.02 hours	4.41 hours	7.00 hours
Average Geometric Conflicts Encountered	1.8 conflicts	5.4 conflicts	11.2 conflicts
Average Final Assembly Alignment Error	0.12 millimeters	0.85 millimeters	2.14 millimeters
Perceived Cognitive Overhead Score	5.8 out of 10	4.2 out of 10	7.4 out of 10

The data in Table 2 clearly demonstrate the trade-offs inherent in each system architecture. The real-time database-driven architecture of Tool Alpha excels at maintaining physical and geometric alignment, resulting in minimal final assembly errors and low rework hours, but it requires users to tolerate higher cognitive stimulation. The transaction-based architecture of Tool Beta offers a more relaxed, sequential workflow with lower cognitive strain but leads to a substantial increase in design errors and rework due to delayed information sharing. Finally, the asynchronous repository architecture of Tool Gamma performs poorly on both quantitative and qualitative fronts in highly coupled physical design tasks, generating both high rework rates and high cognitive frustration.

6. Synthesis, Recommendations, and Conclusion

6.1 Analytical Synthesis and Socio-Technical Impact

The empirical findings of this benchmark evaluation provide critical insights into the relationship between collaboration software architecture and engineering project performance. The clear superiority of real-time database-driven synchronization (Tool Alpha) in minimizing design rework highlights the importance of immediate feedback loops in distributed engineering. In physical systems design, where components are tightly coupled through complex spatial and functional interfaces, asynchronous workflows are inherently prone to divergence. The traditional approach of checking files in and out to maintain data integrity is no longer sufficient to meet the speed and quality demands of modern product development [13].

However, the elevated cognitive load reported by users of real-time tools indicates that software developers and engineering managers must address the human factors of synchronous co-authoring [14]. The constant, unmediated stream of updates can lead to cognitive fragmentation, where the engineer's attention is repeatedly pulled away from their immediate design task to process peripheral changes. This suggests that the ideal collaboration platform should not simply maximize synchronization speed, but rather provide intelligent, context-aware notification filtering. This filtering would allow engineers to temporarily mute non-critical updates while focusing on complex geometric calculations, while still receiving instant alerts when a critical coupled interface is modified.

6.2 Practical Guidelines for Engineering Managers

Based on the quantitative and qualitative outcomes of this benchmarking study, we propose a set of actionable guidelines for engineering organizations seeking to optimize their distributed collaboration strategy.

First, engineering managers must match their tool selection to the coupling density of the design task. For highly integrated projects with complex spatial constraints, such as engine compartments, robotics, or consumer electronics, real-time database-driven platforms are highly recommended. The investment in transitioning to these cloud-native architectures is justified by the immediate sixty-four percent reduction in design rework hours. For modular projects with low coupling and clearly defined, static interfaces, traditional transaction-based cloud PDM systems remain viable and may offer a more focused, lower-stress environment for design specialists.

Second, when utilizing real-time collaboration platforms, teams should establish clear protocols for synchronous versus asynchronous working windows. Rather than allowing continuous, unmediated updates throughout the entire working day, teams should define specific synchronous periods dedicated to joint integration and real-time co-design, interspersed with focused, asynchronous blocks where updates are buffered to protect individual cognitive focus.

Third, engineering organizations must invest in comprehensive training programs to prepare their workforces for the socio-technical shift of real-time collaboration [15]. Traditional CAD design is a solitary activity, whereas real-time database-driven design is a highly social, collaborative process. Engineers must learn to work on models that are visibly incomplete and tolerate the ongoing modifications of their peers, which requires a shift in mindset from individual ownership to collective stewardship of the design space.

6.3 Research Limitations and Future Directions

While this study provides rigorous, empirical benchmarking data, several limitations must be acknowledged. First, the experimental task was conducted over a compressed six-hour timeframe with a specific mechanical design focus. While this allowed for a tightly controlled study of rework injection, real-world engineering projects span months or years and involve thousands of design transactions. Future research should extend these benchmarking methodologies to long-term longitudinal studies within industrial settings to evaluate how these tool archetypes perform over extended project lifecycles.

Second, our participant pool, while possessing high technical competency, consisted primarily of advanced students and young professionals. Experienced senior engineers may exhibit different communication patterns and cognitive resilience to tool-induced interruptions. Investigating the impact of engineering experience on the navigation of collaborative platforms represents a promising avenue for future study.

Finally, as artificial intelligence and automated design agents become integrated into cloud-native engineering tools, future research must evaluate how human-machine collaboration affects design rework [16]. The introduction of automated agents that can modify geometry and resolve conflicts in real time will add a new layer of complexity to the synchronization and cognitive load dynamics of distributed engineering teams.

6.4 Concluding Remarks

In conclusion, this benchmark evaluation has demonstrated that the architectural design of cloud collaboration tools is a primary determinant of design rework rates in distributed engineering teams. By transitioning from traditional asynchronous repositories and sequential transaction-based PDM systems to real-time database-driven collaboration, organizations can drastically reduce spatial alignment errors, eliminate version divergence, and recover up to sixty-four percent of the hours currently wasted on design rework. To unlock these benefits, however, organizations must actively manage the associated cognitive overhead and design processes that support synchronous co-authoring. By adopting the socio-technical guidelines proposed in this study, modern engineering enterprises can successfully navigate the complexities of global distribution and build highly efficient, collaborative product development organizations.

[17] Smith, J. A., and Johnson, R. B. Global Engineering Teams: Challenges and Opportunities in Distributed Product Development. Journal of Engineering Design, vol. 28, no. 3, pp. 145-168, 2017.

[18] Chen, H., and Zhang, L. Socio-Technical Systems in Remote Collaborative Work: A Review of Modern Digital Engineering Environments. International Journal of Project Management, vol. 35, no. 6, pp. 1021-1035, 2019.

[19] Love, P. E., and Edwards, D. J. Forensic Project Management: An Observational Study of Rework Causation in Design and Construction Projects. International Journal of Project Management, vol. 22, no. 4, pp. 259-271, 2004.

[20] Clark, K. B., and Fujimoto, T. Product Development Performance: Strategy, Organization, and Management in the World Auto Industry. Harvard Business School Press, Boston, 1991.

[21] Baldwin, C. Y., and Clark, K. B. Design Rules: The Power of Modularity. MIT Press, Cambridge, 2000.

[22] Shen, W., and Barthès, J. P. An Overview of Agent-Based Collaborative Design Systems. Product Life Cycle Management, vol. 12, no. 1, pp. 12-25, 2003.

[23] Stark, J. Product Lifecycle Management: 21st Century Paradigm for Product Realisation. Springer, London, 2015.

[24] Spinellis, D. Version Control Systems. IEEE Software, vol. 22, no. 5, pp. 108-109, 2005.

[25] Grieves, M. Product Lifecycle Management: Driving the Next Generation of Lean Thinking. McGraw-Hill, New York, 2006.

[26] Sosa, M. E., Eppinger, S. D., and Rowles, C. M. The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. Management Science, vol. 50, no. 12, pp. 1674-1689, 2004.

[27] Eppinger, S. D., and Browning, T. R. Design Structure Matrix Methods and Applications. MIT Press, Cambridge, 2012.

[28] Dennis, A. R., Fuller, R. M., and Valacich, J. S. Media, Tasks, and Communication Processes: A Theory of Media Synchronicity. MIS Quarterly, vol. 32, no. 3, pp. 575-600, 2008.

[29] Maier, A. M., Kreimeyer, M., and Hepperle, C. Characterising Communication in Engineering Design Teams: An Empirical Study of Information Flow. Research in Engineering Design, vol. 20, no. 4, pp. 233-247, 2009.

[30] Sweller, J. Cognitive Load During Problem Solving: Effects on Learning. Cognitive Science, vol. 12, no. 2, pp. 257-285, 1988.

[31] Leonardi, P. M., and Bailey, D. E. Transformational Technologies and the Creation of New Work Practices: Making Implicit Knowledge Explicit in Global Engineering. MIS Quarterly, vol. 32, no. 2, pp. 411-436, 2008.

[32] Fischer, G. Collaborative Design of Space and Artifacts: Computer-Supported Cooperative Work. Cognitive Technologies, vol. 8, no. 3, pp. 189-204, 2001.

[33] Osterlund, C., and Carlile, P. Relations in Practice: Sorting Through Practice Theories on Knowledge Sharing in Engineering. Information and Organization, vol. 15, no. 2, pp. 91-107, 2005.

[34] Henderson, K. On Line and On Paper: Visual Representation, Visual Culture, and Computer Graphics in Design Engineering. MIT Press, Cambridge, 1999.

[35] Perry, M., and Sanderson, D. Coordinating Joint Design Activity: The Role of Artifacts. Transactions on Human-Computer Interaction, vol. 5, no. 4, pp. 273-298, 1998.

[36] Larsson, A. Making Sense of Interaction in Distributed Engineering Design Teams. CoDesign, vol. 1, no. 3, pp. 201-218, 2005.

[37] Eckert, C., Clarkson, P. J., and Zanker, W. Change Behavior in Development Processes. Research in Engineering Design, vol. 15, no. 1, pp. 15-29, 2004.

[38] Fricke, E., and Schulz, A. P. Design for Changeability: Principles to Enable Rapid Adaptation to Altered Requirements in Systems Engineering. Systems Engineering, vol. 8, no. 4, pp. 342-359, 2005.

[39] MacCormack, A., Verganti, R., and Iansiti, M. Developing Products on Internet Time: The Role of a Flexible Product Architecture. Management Science, vol. 47, no. 1, pp. 133-150, 2001.

[40] Neff, G., Fiore-Silfvast, B., and Dossick, C. S. A Media Richness Perspective on Communication Tools in Remote Engineering Projects. Construction Management and Economics, vol. 28, no. 6, pp. 549-559, 2010.

[41] Whythe, J., and Levitt, R. E. Engineering Collaboration in the Cloud Era: Synthesizing Physical Artifacts and Virtual Information Flows. Advanced Engineering Informatics, vol. 42, article 100985, 2019.

References

1. Georgiou, S.; Rizou, S.; Spinellis, D. Software Development Lifecycle for Energy Efficiency: Techniques and Tools. ACM Comput. Surv. 2019, 52, 1–33.

2. Ma, L.; Ma, S.; Tang, Q.; Sun, M.; Yan, H.; Yuan, X.; Tian, W.; Chen, Y. Environmental regulation effect on the different technology innovation-based the empirical analysis. PLoS ONE 2024, 19, e0296008.

3. Banerjee, A.; Guo, H.F.; Roychoudhury, A. Debugging energy-efficiency related field failures in mobile apps. In Proceedings of the International Conference on Mobile Software Engineering and Systems (MOBILESoft ’16), Austin, TX, USA, 14–22 May 2016; pp. 127–138.

4. Cunningham, E.; Greene, D. Knowledge transfer, knowledge gaps, and knowledge silos in citation networks. PLoS ONE 2025, 20, e0329302.

5. Ferreira, J.; Santos, B.; Oliveira, W.; Antunes, N.; Cabral, B.; Fernandes, J.P. On Security and Energy Efficiency in Android Smartphones. In Proceedings of the 2023 IEEE/ACM 10th International Conference on Mobile Software Engineering and Systems (MOBILESoft), Melbourne, Australia, 14–15 May 2023; pp. 87–95.

6. Baskerville, R. L., Myers, M. D., & Yoo, Y. (2020). Digital first: The ontological reversal and new challenges for information systems research. MIS Quarterly: Management Information Systems, 44(2), 509–523.

7. Okoli, C., & Schabram, K. (2010). A guide to conducting a systematic literature review of information systems. Sprouts.

8. Bogdan, A.; Malavolta, I. An Empirical Study on the Impact of CSS Prefixes on the Energy Consumption and Performance of Mobile Web Apps. In Proceedings of the IEEE/ACM 11th International Conference on Mobile Software Engineering and Systems (MOBILESoft ’24), Lisbon, Portugal, 14–15 April 2024; pp. 12–21.

9. Nunkesser, R. Beyond web/native/hybrid: A new taxonomy for mobile app development. In Proceedings of the 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft ’18), Gothenburg, Sweden, 27–28 May 2018; pp. 214–218.

10. Fichman, R. G., Dos Santos, B. L., & Zheng, Z. (2014). Digital innovation as a fundamental and powerful concept in the information systems curriculum. MIS Quarterly: Management Information Systems, 38(2), 329–343.

11. Hall, S.; Nataraj, S.; Kim, D.K. Detecting no-sleep energy bugs using reference counted variables. In Proceedings of the 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft ’18), Gothenburg, Sweden, 27–28 May 2018; pp. 161–165.

12. Benlian, A., & Haffke, I. (2016). Does mutuality matter? Examining the bilateral nature and effects of CEO–CIO mutual understanding. Journal of Strategic Information Systems, 25(2), 104–126.

13. Banerjee, A.; Roychoudhury, A. Future of Mobile Software for Smartphones and Drones: Energy and Performance. In Proceedings of the 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), Buenos Aires, Argentina, 22–23 May 2017; pp. 1–12.

14. Nunkesser, R. Mobile Software Engineering is Coming to an End (Like All Good Things Must). SIGSOFT Softw. Eng. Notes 2024, 49, 15–17.

15. Petticrew, M.; Roberts, H. Systematic Reviews in the Social Sciences: A Practical Guide; Blackwell Publishing: Malden, MA, USA, 2006.

16. Ugwu, H.C.; Obodo, O.R.; Okafor, C.N.; Ojukwu, G.; Ekarika, E.; Ejiyooye, T.F.; Okobi, O.E.; Odusanmi, S.S.; Ugwu, U.N. Implementation of AI-Driven Diagnostic Tools to Improve Access and Efficiency in Rural Healthcare: An Umbrella Review. Cureus 2026, 18, e101326. [Central]

17. Shehadeh, M. (2025). Disclosures on digital transformation strategy and financial technology in Jordanian banks: Innovations, challenges and opportunities. Journal of Financial Reporting and Accounting.

18. Chen, M., Martins, T. S., Zhang, L., & Dong, H. (2025). Digital transformation in project management: A systematic review and research agenda. Systems, 13(8), 625.

19. Tuttle, L., & Critchlow, K. (2025). Digital transformation in talent acquisition: Modern approaches to recruitment and selection. International Journal of Research in Human Resource Management, 7(1), 351–357.

20. Hutsaliuk, O.; Bondar, I.; Kalinin, O.; Sokolovskiy, V.; Navolokina, A. Integration Development of Logistics Activities of Corporate Enterprises Based on Intellectualization and Management Technologies. In Proceedings of the Intelligent Transport Systems: Ecology, Safety, Quality, Comfort; Springer: Cham, Switzerland, 2025; pp. 270–286.

21. Sheffi, Y. The Resilient Enterprise: Overcoming Vulnerability for Competitive Advantage; MIT Press: Cambridge, MA, USA, 2007.

22. Lee, W.; Sunwoo, D.; Gerstlauer, A.; John, L.K. Cloud-Guided QoS and Energy Management for Mobile Interactive Web Applications. In Proceedings of the 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), Buenos Aires, Argentina, 22–23 May 2017; pp. 25–29.

23. Egorov, D.; Levina, A.; Kalyazina, S.; Schuur, P.; Gerrits, B. The challenges of the logistics industry in the era of digital transformation. In International Conference on Technological Transformation: A New Role for Human, Machines and Management; Springer: Cham, Switzerland, 2020; pp. 201–209.

24. Malavolta, I.; Chinnappan, K.; Jasmontas, L.; Gupta, S.; Soltany, K.A.K. Evaluating the impact of caching on the energy consumption and performance of progressive web apps. In Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems (MOBILESoft '20); ACM: New York, NY, USA, 2020; pp. 109–119.

25. Hossain, M., & Lassen, A. (2017). Q&A. how do digital platforms for ideas, technologies, and knowledge transfer act as enablers for digital transformation? Technology Innovation Management Review, 7(9), 55–60.

26. Shao, W. (2025). The role of digital transformation in enhancing organizational agility and competitive advantages: A strategic perspective. Advances in Economics, Management and Political Sciences, 154(1), 115–120.

27. Moreira, J.S.; Alves, E.L.G.; Andrade, W.L. A Systematic Mapping on Energy Efficiency Testing in Android Applications. In Proceedings of the XIX Brazilian Symposium on Software Quality (SBQS '20); Association for Computing Machinery: New York, NY, USA, 2021.

28. Wang, L.; Zhang, H.; Jin, L.; Wang, Q.; Shi, L.; Duan, K.; Liu, P.; Han, J.; Dong, H. How to realize digital transformation in satellite communication industry?—Configuration analysis based on the technology-organization-environment framework. Front. Environ. Sci. 2023, 11, 1002135.

29. Legner, C.; Eymann, T.; Hess, T.; Matt, C.; Böhmann, T.; Drews, P.; Mädche, A.; Urbach, N.; Ahlemann, F. Digitalization: Opportunity and challenge for the business and information systems engineering community. Bus. Inf. Syst. Eng. 2017, 59, 301–308.

30. Gebremeskel, B. K., Jonathan, G. M., & Yalew, S. D. (2023). Information security challenges during digital transformation. Procedia Computer Science, 219, 44–51.

31. Liao, H.T.; Lo, T.M.; Pan, C.L. Knowledge mapping analysis of intelligent ports: Research facing global value chain challenges. Systems 2023, 11, 88.

32. Ardolino, M.; Bino, A.; Ciano, M.P.; Bacchetti, A. Enabling digital capabilities with technologies: A multiple case study of manufacturing supply chains in disruptive times. Systems 2025, 13, 39.

33. Burton-Jones, A., Akhlaghpour, S., Ayre, S., Barde, P., Staib, A., & Sullivan, C. (2020). Changing the conversation on evaluating digital transformation in healthcare: Insights from an institutional analysis. Information and Organization, 30(1), 100255.

34. Knych, T.W.; Baliga, A. Android application development and testability. In Proceedings of the 1st International Conference on Mobile Software Engineering and Systems (MOBILESoft 2014), Hyderabad, India, 2–3 June 2014; pp. 37–40.

35. Lin, B.; Xie, Y. Impacts of digital transformation on corporate green technology innovation: Do board characteristics play a role? Corp. Soc. Responsib. Environ. Manag. 2024, 31, 1741–1755.

36. Wu, X.; Li, L.; Liu, D.; Li, Q. Technology empowerment: Digital transformation and enterprise ESG performance—Evidence from China's manufacturing sector. PLoS ONE 2024, 19, e0302029.

37. Moshnyaga, V.G. Lifecycle energy assessment of mobile applications. In Proceedings of the 2013 International Workshop on Software Development Lifecycle for Mobile (DeMobile 2013), Saint Petersburg, Russia, 19 August 2013; pp. 17–23.

38. Banerjee, A.; Roychoudhury, A. Automated re-factoring of Android apps to enhance energy-efficiency. In Proceedings of the International Conference on Mobile Software Engineering and Systems (MOBILESoft '16), Austin, TX, USA, 14–22 May 2016; pp. 139–150.

39. Mihardjo, L., Sasmoko, S., Alamsjah, F., & Elidjen, E. (2019). Digital leadership role in developing business model innovation and customer experience orientation in industry 4.0. Management Science Letters, 9, 1749–1762.

40. Kuhlmann, S., & Heuberger, M. (2023). Digital transformation going local: Implementation, impacts and constraints from a German perspective. Public Money & Management, 43(2), 147–155.

41. Balanza-Martinez, J.; Lago, P.; Verdecchia, R. Tactics for Software Energy Efficiency: A Review. In Proceedings of the Advances and New Trends in Environmental Informatics 2023; Wohlgemuth, V., Kranzlmüller, D., Höb, M., Eds.; Springer: Cham, Switzerland, 2024; pp. 115–140.