Leveraging Lightweight Task Notification Systems to Enhance Clarity, Continuity, and Coordination in Early-Stage Software Development Teams

Sarun Rojanasoton

Graduate students in Knowledge and Innovation Management, College of Arts, Media and Technology, Chiang Mai University.

Corresponding Author: sarun_roj@cmu.ac.th

Received: March 28, 2026. **Revised:** April 24, 2026. **Accepted:** April 27, 2026.

Abstract

Early-stage software teams relied heavily on informal communication for task coordination; however, this created a structural tension: communication was abundant, while shared visibility into task status remained insufficient. This study investigated how a lightweight structured task notification system could improve knowledge continuity and team coordination in such environments. Using Design Science Research (DSR), the researcher designed and deployed a notification artifact within a Thai software startup of approximately 25 employees, embedding four automated notification types and a single-click task completion button into the team's existing Lark platform. Three temporal phases were examined: a pre-artifact baseline with no task records, a nine-month broader deployment covering 571 tasks, and a two-month focused pilot with six participants across 20 tasks.

Pre-intervention analysis identified three recurring coordination breakdowns: mismatches between actual and recorded task start times, with 9.5% of tasks never being tracked; systematic omission of small tasks; and inaccurate completion status, with 20.3% of tasks never being marked as completed. The central finding was socio-technical: the same artifact produced inconsistent results during nine months of unstructured deployment but achieved 100% task tracking and completion rates when supported by a bounded project scope, clear usage expectations, and managerial commitment. Post-intervention evaluation using an adapted Kirkpatrick framework indicated improved task visibility, reduced verbal follow-up, and enhanced retrospective traceability within the pilot context. Five transferable design principles were proposed for practitioners implementing lightweight coordination interventions in resource-constrained teams.

Keywords: Socio-Technical Systems, Task Coordination, Lightweight Notifications, Design Science Research, Knowledge Management

Introduction

Early-stage software development teams face a distinctive coordination challenge. They need to move quickly, adapt constantly, and accomplish this with limited resources, creating pressure to minimize formal processes and documentation wherever possible (Paternoster et al., 2014). Task coordination in such teams is largely achieved through direct communication, including verbal instructions, chat messages, and brief meetings. While this supports rapid decision-making and preserves flexibility, it also means that task-related knowledge lives mainly in individual memory and temporary communication channels rather than in persistent shared records the whole team can access.

This creates a paradox: communication is abundant but coordination remains insufficient. Teams discuss tasks in chat, make decisions in meetings, and give instructions verbally, but none of this gets captured in a way that everyone can refer to later. The result is that managers spend time asking people what they are working on, team members duplicate effort because they cannot see what others are doing, and completed work goes unrecognized because nobody updated the board.

Coordination theory provides a useful foundation for understanding these challenges. Malone and Crowston (1994) defined coordination as the management of dependencies between activities, arguing that coordination problems arise not from a lack of effort but from inadequate management of these dependencies. Herbsleb (2007) extended this to software engineering, showing that coordination breakdowns stem from insufficient visibility into interdependent work. The socio-technical systems perspective (Mumford, 2006; Baxter & Sommerville, 2011; Clegg, 2000) holds that effective work system design must optimize both social and technical subsystems together. A well-designed technical artifact deployed into misaligned social conditions will produce inconsistent results. Baxter and Sommerville (2011) identified emergent system properties that arise from the interaction between people and technology, properties that cannot be predicted from either subsystem alone. That principle is central to the present study.

The organization studied is a Thai software startup with approximately 25 employees operating in a hybrid work arrangement using Lark as its primary communication and project management platform. Before February 2025, no task board existed. A structurally significant

access constraint further shaped coordination: only six individuals three managers and three team lead held permission to create or edit task board entries, creating a systemic bottleneck that persisted throughout the study period.

Research Objectives

This study addresses the following research question: How can lightweight structured task notifications improve knowledge continuity and team coordination in early-stage software development teams? Two objectives guide the research:

1) To analyze critical communication and coordination challenges that affect knowledge visibility, task ownership, and team alignment in early-stage software teams.

2) To design and evaluate a lightweight task notification system that improves clarity, knowledge continuity, and coordination without disrupting agility

Conceptual Framework

This study integrates three analytical frameworks, each serving a distinct purpose. The DIKW hierarchy (Rowley, 2007) serves as the diagnostic lens. Applied at the task level, raw communication events constitute Data; structured task records represent Information; shared team understanding of task status constitutes Knowledge; and the capacity to design better coordination interventions represents Wisdom. The primary breakdown in this study occurs at the Data-to-Information boundary, where informal communication is not reliably converted into structured task records. This task-level application is distinct from DIKW's typical use at organizational or strategic levels.

The Kirkpatrick evaluation model (Kirkpatrick & Kirkpatrick, 2006) structures the post-intervention assessment across four levels: Reaction (participants' immediate perceptions of usefulness), Learning (conceptual understanding of the coordination logic embedded in the artifact), Behavior (observable changes in practices such as board accuracy and verbal follow-up frequency), and Results (organizational outcomes). The Kirkpatrick model was not designed for evaluating information systems interventions, and applying it here requires a degree of interpretive adaptation. Level 4, which was originally intended to capture training return on investment, is used in this study only to record early observable indicators rather than

confirmed organizational outcomes, given the two-month pilot duration. The model is retained nonetheless because its hierarchical structure enables a systematic distinction between surface-level reactions and deeper behavioral change a distinction that is analytically useful when evaluating whether a lightweight artifact shifts how people work, not merely what they think of it. The DSR process model (Peffers et al., 2007) provides the overarching research logic, guiding the study from problem identification through artifact design, demonstration, and evaluation. Together, DIKW diagnoses the knowledge transformation problem, DSR structures the design response, and Kirkpatrick evaluates the intervention's impact at multiple levels of depth.

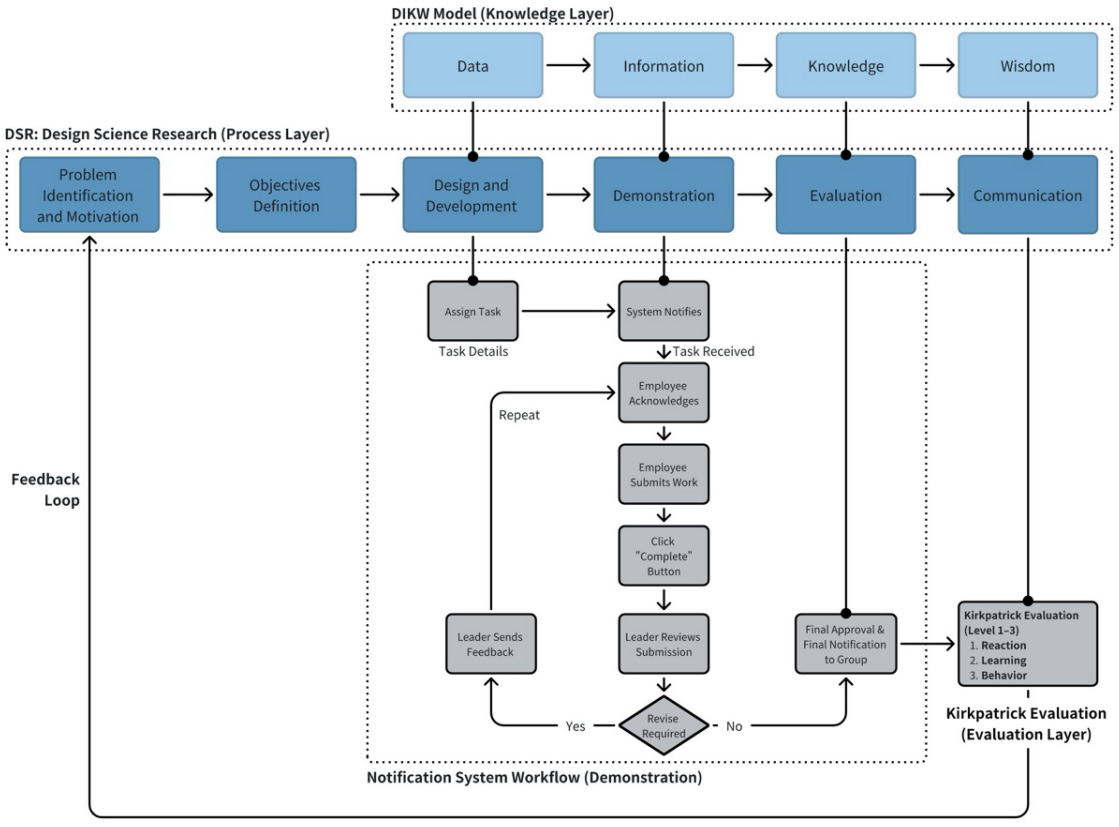


Figure 1 Integrated Research Framework

Source: Developed by the author, synthesized from Rowley (2007), Peffers et al. (2007), and Kirkpatrick & Kirkpatrick (2006)

Literature Review

Coordination in Startup Teams

Paternoster et al. (2014) found that startups commonly adopt agile practices in a loose, informal way, relying on direct communication for task coordination. While this supports rapid decision-making, task-related knowledge remains in individual memory and temporary communication rather than in persistent shared records. Stray and Moe (2020) found that high communication volume does not automatically translate into effective coordination the organization in this study displays exactly that pattern.

Socio-Technical Systems Theory

The socio-technical systems tradition holds that effective work system design requires joint optimization of both social and technical subsystems (Mumford, 2006; Clegg, 2000). Baxter and Sommerville (2011) identified emergent properties that arise from the interaction between people and technology properties that cannot be predicted from either subsystem alone. This principle is central to the present study: the same notification artifact produced markedly different outcomes under different social conditions, confirming that technical deployment alone is insufficient.

Knowledge Management and the DIKW Hierarchy

Centobelli et al. (2017) found that knowledge management in startups is typically individual-centric, creating fragility as teams grow. More recently, Shujahat et al. (2019) demonstrated that knowledge management practices in small, agile firms succeed primarily when embedded within daily operational workflows rather than implemented as standalone systems, a finding that directly supports the embedded-workflow design approach taken in this study. Bolisani and Scarso (2016) showed that knowledge management tools succeed only when perceived as immediately useful and culturally aligned. The DIKW hierarchy (Rowley, 2007), applied at the task level in this study, reveals that the primary bottleneck occurs at the Data-to-Information boundary, where informal communication events fail to convert into structured, retrievable records.

Notifications as Coordination Mechanisms

Iqbal and Horvitz (2010) found that notification effectiveness depends critically on relevance, timing, and actionability, but few studies have examined notifications as team-level

coordination artifacts. Riedl et al. (2021) extended this line of inquiry by examining how notification design shapes collective rather than individual attention in distributed teams, finding that notifications bundled around task boundaries are significantly less disruptive than those triggered by individual events a distinction that informs the workflow-integrated design used in the present study. Mark et al. (2008) cautioned that frequent digital interruptions increase stress and reduce performance, which informed the design emphasis on minimizing notification types. Thaler and Sunstein (2008) demonstrated that embedding action buttons within existing interfaces makes the preferred behavior the path of least resistance a nudge principle directly applied in the artifact's single-click Complete button. Existing tools such as Jira, Trello, and Asana offer task tracking capabilities, but small teams frequently underutilize them due to overhead that exceeds their coordination needs. The Lark platform used here integrates communication and task management, creating the technical precondition for embedding the completion action directly within the chat notification.

Taken together, the literature reveals a consistent structural problem in early-stage software teams: coordination fails not because teams communicate too little, but because the communication they produce does not get converted into shared, retrievable records (Malone & Crowston, 1994; Paternoster et al., 2014). Tools exist that could address this, yet small teams persistently underutilize them, and heavier platforms tend to make adoption worse rather than better (Stray & Moe, 2020; Smite et al., 2023). What remains underexplored is whether a lighter intervention, one that does not demand a new tool or a new workflow but rather embeds structured coordination signals within the environment the team already uses, can reliably move communication data across the boundary into structured information. The literature on notification design offers partial guidance (Iqbal & Horvitz, 2010; Mark et al., 2008), and nudge theory provides a design logic (Thaler & Sunstein, 2008), but empirical studies that apply these ideas specifically to task-level coordination in small, access-constrained software teams are scarce. This study addresses that intersection through a Design Science Research approach, using a single Thai startup as a bounded case in which the coordination problem, the technical constraints, and the intervention can be examined together.

Research Methodology

This study follows Design Science Research (Peffers et al., 2007) within an interpretive, constructivist paradigm. The interpretive stance means the study prioritizes understanding how participants experience coordination practices rather than measuring outcomes through statistical inference. Analytical generalization, not statistical generalization, is the appropriate standard: findings generalize to theory and to similar contexts, not to populations (Yin, 2018). The qualitative single-case approach is consistent with DSR's emphasis on evaluating artifact utility in depth within a specific organizational context.

The researcher holds a managerial role within the case study organization, which provided deep contextual access but introduced social desirability risk in participant accounts. A reflexive field journal was maintained throughout. Two instances illustrate this concretely. First, delayed task-board updates were initially framed as a discipline problem; a journal entry flagged this, prompting recoding toward structural factors such as access limitations and verbal task initiation, which shifted interpretation from individual blame to a systemic bottleneck. Second, uneven use of the Complete button was initially read as artifact resistance; focusing more on participant explanations gave a more balanced picture of how the system was used. Triangulation across three data sources provided additional mitigation.

Research Phases

Three temporal phases are distinguished. Phase 1 (before February 2025) is the pre-artifact period during which no task board existed. Phase 2 (February to November 2025) is the broader deployment period during which the artifact was technically functional across all projects but adoption was inconsistent and unenforced, generating 571 task records. Phase 3 (December 2025 to January 2026) is the focused pilot in which the artifact was evaluated within a single bounded client project under deliberate social conditions: explicit usage expectations, a clearly defined project scope, and visible managerial commitment.

Phase 2 is not a control period in an experimental sense; it is a naturalistic deployment that generated evidence about which coordination problems persisted even when a technical system was available. The contrast between Phase 2 and Phase 3 same artifact, different social conditions provide the primary analytical lever for the socio-technical finding.

Participants and Data Collection

Six participants constituted the complete population of artifact users during the focused pilot: P1 (Developer, 4 months, on-site), P2 (Developer, 4 months, on-site), P3 (QA/Tester, 10 months, on-site), P4 (UX/UI Designer, 12 months, on-site), P5 (Developer/IT Consultant, 14 months, fully remote), and P6 (Project Manager, 8 months, authorized to create tickets). Selection was purposive based on active project involvement (Patton, 2015). Three data sources enabled triangulation: semi-structured interviews in Thai (30–60 minutes each, conducted February 2026), non-participant observation of team coordination activities from February 2025 to January 2026, and document analysis of task board records, notification logs, and system timestamps. Interview transcripts were analyzed using reflexive thematic analysis (Braun & Clarke, 2006). Research trustworthiness was assessed following Tracy's (2010) criteria of credibility, sincerity, and rich rigor. All participants provided informed consent and were assured of confidentiality and the right to withdraw at any time.

Table 1 Participant Summary

Code	Role	Tenure	Work Mode	Interview
P1	Developer	4 months	On-site	45 min
P2	Developer	4 months	On-site	35 min
P3	QA/Tester	10 months	On-site	40 min
P4	UX/UI Designer	12 months	On-site	30 min
P5	Developer / IT Consultant	14 months	Remote	45 min
P6	Project Manager	8 months	On-site	60 min

Artifact Design

The notification artifact was implemented using Lark Base's built-in automation feature; no external coding or third-party tools were required, ensuring maintainability by non-technical staff. Four automation rules were created: Task Start (notification to team channel and assigned employee when a task is created), Reminder (deadline approach), Complete

(employee clicks button in chat notification; system records timestamp and notifies manager), and Manager Approval (manager accepts or requests revision). The core design decision was embedding the Complete button within the Lark chat notification itself, making completion performable in a single click within the team's existing workflow, consistent with nudge theory (Thaler & Sunstein, 2008). The four-type limit was informed by notification fatigue research (Mark et al., 2008).

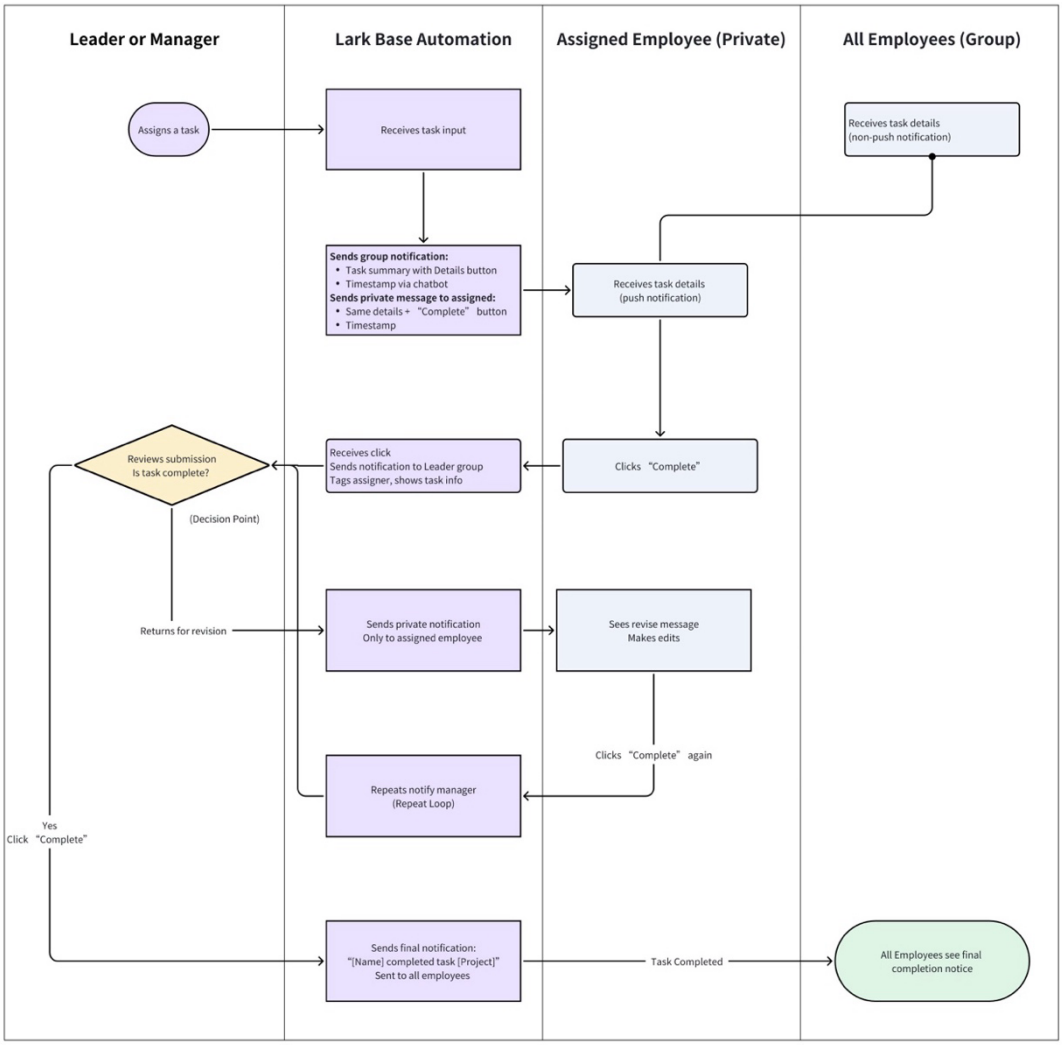


Figure 2 Post-Intervention Task Notification Workflow

Source: Developed by the author

Research Results

Pre-Intervention Findings (Phase 2: Broader Deployment)

Analysis of 571 tasks combined with interview data from all six participants revealed three recurring coordination problems. Across all three themes, the same underlying dynamic appeared: abundant task-relevant communication was not being converted reliably into structured records a consistent breakdown at the Data-to-Information boundary in DIKW terms.

Theme 1: Mismatch Between Actual and Recorded Task Start Times

The most consistently reported problem was a temporal gap between when work began and when it was formally recorded. Fifty-four of 571 tasks (9.5%) were never tracked on the board at all. Tasks routinely began from verbal or chat instructions before any ticket was created. P1 (Developer) described starting work immediately from chat messages because regular employees could not create tickets themselves, with the ticket appearing later making it look like the task had just started. The access constraint only six individuals authorized to create entries structurally amplified this gap. In DIKW terms, the task assignment event (Data) occurred in chat but was not converted into a structured record (Information) until significantly later, if at all.

Theme 2: Inconsistent Recording of Small Tasks

Tasks categorized as small QA checks, design adjustments, minor bug fixes were routinely not recorded. P3 (QA) described receiving work through private chat and completing it without any board entry. P4 (Designer) noted that iterative design revisions requested verbally never appeared in the system. At least 54 tasks were entirely missing, and an unknown additional number of small tasks were never recorded. This invisible workload affected QA and design roles disproportionately, and made workload distribution appear more balanced than it was. In DIKW terms, a significant portion of daily work never entered the information layer at all.

Theme 3: Inaccurate Task Completion Status

Of 571 tasks, 116 (20.3%) were completed but never updated to Done on the board. P6 (Project Manager) described constant verbal follow-up as necessary to determine actual project status, creating a self-reinforcing cycle: when the board was inaccurate, managers reverted to asking directly, which further reduced motivation to keep the board updated.

Notification coverage was 99.3% (567/571), confirming that automation worked reliably whenever a task was entered. The coordination gap was upstream: tasks not created at all, and completions not signaled.

Post-Intervention Evaluation (Phase 3: Focused Pilot)

The focused pilot ran from December 2025 to January 2026 within a single client project. Twenty tasks were created and tracked throughout the period.

Level 1 Reaction: Participants generally found the system useful, particularly the Complete button, which P1 described as fast and convenient because tasks could be closed without the lead needing to follow up. P5 (remote) considered notifications essential for staying connected to team activity. All six participants, however, criticized notification volume as too high, with P1 admitting to reading only personally relevant notifications and ignoring others.

Level 2 Learning: Participants demonstrated understanding of the coordination logic embedded in the artifact. P2 articulated that inconsistent usage reduces data reliability across the team. A gap persisted, however, between conceptual understanding and consistent action under workload pressure, a well-documented challenge in knowledge management research.

Level 3 Behavior: Board accuracy improved from 90.5% (517/571) during broader deployment to 100% (20/20) during the focused pilot. All 20 tasks were tracked from creation through completion, and verbal follow-up for status checks decreased. Of the 20 tasks, 11 (55%) were completed on time and 9 (45%) were late; 10 (50%) required rework. These figures differ from those in broader deployment (68.1% on time, 5.3% rework), but they should not be read as evidence that the artifact worsened coordination. During broader deployment, tasks already late or with quality issues went largely unrecorded. The focused pilot surfaced delays and rework that had previously gone undetected. The descriptive data below supports pattern recognition within this study; no statistical inference across the two deployment periods is claimed.

Table 2 Coordination Metrics: Broader Deployment vs. Focused Pilot

Metric	Phase 2: Broader Deployment (Feb–Nov 2025)	Phase 3: Focused Pilot (Dec 2025–Jan 2026)	Change
Tasks tracked	90.5% (517/571)	100% (20/20)	+9.5 pp
Tasks reaching Done	79.7% (455/571)	100% (20/20)	+20.3 pp
Tasks never tracked	9.5% (54/571)	0% (0/20)	Resolved
Notification coverage	99.3% (567/571)	100% (20/20)	+0.7 pp
On-time completion	68.1% (389/571)	55% (11/20)	-13.1 pp*
Rework required	5.3% (30/571)	50% (10/20)	+44.7 pp*
Manager follow-up	Frequent	Reduced	Improved

* Reflects improved detection of pre-existing delays and quality issues, not worse coordination performance.

Discussion

The central finding of this study is socio-technical. The same notification artifact produced inconsistent coordination outcomes over nine months of unstructured deployment but achieved complete task tracking and completion coverage during a two-month focused pilot with deliberate social conditions. The technical design alone was insufficient. This is underscored by the notification coverage data: Phase 2 already achieved 99.3% coverage for tasks that were entered into the system. The automation worked. What did not work was organizational consistency in creating tasks and updating completion status. The artifact was a necessary but not sufficient condition for coordination improvement.

This finding directly supports the socio-technical systems tradition. Mumford (2006) argued that effective work system design must jointly optimize social and technical subsystems. Clegg (2000) proposed that technology should complement human capabilities within its organizational context. The present study provides empirical evidence for this principle at the level of a lightweight information systems intervention: even a simple, well-designed notification artifact cannot produce coordination improvements without aligned social conditions. The artifact is a necessary but insufficient condition for coordination improvement. The three social conditions that differentiated Phase 3 from Phase 2 bounded

project scope, explicit usage expectations, and visible managerial commitment constitute a minimal social infrastructure that practitioners should establish before deploying coordination tools.

Applying the DIKW hierarchy at the task level reveals that all three pre-intervention themes share a common structural bottleneck at the Data-to-Information boundary. Communication data was abundant: chat messages, verbal instructions, and meeting discussions generated a continuous stream of task-related content. However, this data was not reliably converted into structured, retrievable task records. Delayed ticket creation, omission of small tasks, and failure to update completion status each represent a different manifestation of the same conversion failure. The notification artifact addressed this bottleneck by automating the transformation of task events into structured records, most directly through the embedded Complete button. This task-level DIKW application extends the framework's diagnostic utility below its typical organizational and strategic scope.

The Kirkpatrick analysis reveals a meaningful gap between Levels 2 and 3: participants understood the coordination logic of the artifact but did not consistently act on that understanding under workload pressure. This gap between learning and behavior suggests that future iterations should invest in structural incentives alongside technical design. The Level 4 observations reduced verbal follow-up and an emergent performance review benefit are treated as early indicators rather than confirmed organizational improvements, given the short pilot duration.

The findings add a structural dimension to startup coordination research. Paternoster et al. (2014) documented that startups minimize documentation as a deliberate strategic choice. This study shows that even when teams intend to document their work, structural access restrictions can create systemic bottlenecks independently of individual motivation. The finding that only six of 25 employees could create task entries represents a coordination constraint not prominently discussed in the existing startup coordination literature.

Taken together, the findings address both research objectives. For the first objective, the pre-intervention analysis documented three recurring coordination breakdowns and traced each to a common structural failure at the Data-to-Information boundary a failure amplified by the access constraint that concentrated ticket creation in six individuals. For the second

objective, the focused pilot demonstrated that the notification artifact improved coordination accuracy and reduced verbal overhead when deployed under aligned social conditions, and the five design principles derived from the evaluation provide practical guidance for practitioners facing similar constraints.

Two threats to validity deserve acknowledgement. First, the Hawthorne effect is a recognized confound (Landsberger, 1958; McCambridge et al., 2012): P6 explicitly noted that the team was instructed to use the system consistently during the pilot. Second, the researcher's dual role introduces social desirability risk. Participants were evaluating a system designed by their manager, and the possibility that more fundamental criticisms were softened cannot be ruled out. The reflexive strategies described in the methodology section, particularly triangulation against system-generated records, provide mitigation, but positive findings should be read as indicative rather than confirmatory.

Recommendations

Based on the empirical findings, five transferable design principles are proposed. These principles are grounded in a single case and should not be treated as universal prescriptions. Before applying them, practitioners should assess alignment with at least three conditions. First, team size: the dynamics observed in this study emerged in a team of approximately 25 people with a flat reporting structure; larger teams with more complex hierarchies may require a different configuration of notification types and permission models. Second, platform integration: the artifact's effectiveness depended critically on the notification being delivered within the same platform the team already used for daily communication; if an organization's task management system is separate from its messaging environment, the friction cost of the completion action increases substantially, and the nudge logic weakens. Third, managerial commitment: the focused pilot demonstrated that the same artifact produced qualitatively different outcomes under different levels of management engagement; practitioners should treat managerial buy-in not as a background condition but as a prerequisite that must be actively established before deployment.

Table 3 Five Transferable Design Principles for Lightweight Task Coordination Interventions

Principle	Description
1. Prioritize completion signaling	Task completion is the most frequently abandoned data point. Design a one-click mechanism embedded in the team's existing communication platform.
2. Embed actions in existing workflows	Interventions requiring context switching or new interface learning face significantly higher abandonment rates.
3. Limit notification types	Start with the minimum set (start, remind, complete). Each additional type increases the risk that users stop using the system.
4. Design for selective attention	Make personally relevant notifications visually distinct from team-level broadcasts.
5. Address structural constraints alongside behavioral ones	Behavioral nudges cannot overcome access bottlenecks. Audit permission structures before attributing coordination problems to individual behavior.

This study makes three contributions. Conceptually, it extends the DIKW hierarchy to task-level coordination, identifying the Data-to-Information boundary as a primary operational failure point. As a design contribution, it proposes a lightweight notification model with five empirically grounded design principles. Practically, it shows that meaningful coordination improvements do not require comprehensive project management systems; targeting specific high-friction points with minimal structure can yield significant benefits at low cost, provided appropriate social conditions are established alongside technical deployment.

Limitations and Future Research

The study examines a single startup, a single pilot project, and a two-month evaluation period with six participants. Findings offer analytical transferability, not statistical generalization (Yin, 2018). The researcher's dual role as investigator and manager is the most significant methodological limitation: the entire analytical framework rests on a single interpreter who also designed the artifact and managed the participants. Future research should extend the deployment period, conduct multi-case comparisons across different startup contexts, and

introduce quantitative coordination metrics. The divergence in experience between remote and on-site workers also warrants dedicated investigation.

A cultural caveat also deserves acknowledgement. This study was conducted in a Thai organizational context, and certain dynamics observed during the research may reflect features of that setting rather than the intervention itself. Thai workplace culture tends toward high power distance (Hofstede, 2001), which may have influenced how participants engaged with both the notification system and the researcher in his role as their manager. Employees may have been less inclined to voice frustrations with the artifact directly, and more likely to use the system consistently during the pilot out of respect for managerial expectations rather than personal conviction. Face-saving norms may similarly have moderated the critical feedback shared in interviews, meaning that the Level 1 reaction data could reflect a degree of social accommodation that would be less pronounced in flatter organizational settings. These cultural dynamics are not unique to this organization, but they are worth naming explicitly given the researcher's dual role. Future research should examine whether the same intervention produces similar adoption patterns in teams operating under different cultural or organizational norms.

Acknowledgements

The author would like to express sincere gratitude to Assoc. Prof. Teeraporn Saeheaw for her guidance and supervision throughout this research. The author also thanks the six participants who generously shared their time and experiences during the interviews, and the organization that provided access to the research site.

References

Baxter, G., & Sommerville, I. (2011). Socio-technical systems: From design methods to systems engineering. *Interacting with Computers, 23*(1), 4–17. https://doi.org/10.1016/j. intcom.2010.07.003

Bolisani, E., & Scarso, E. (2016). Factors affecting the use of wiki to manage knowledge in a small company. *Journal of Knowledge Management, 20*(3), 423–443. https://doi.org /10.1108/JKM-05-2015-0167

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

Centobelli, P., Cerchione, R., & Esposito, E. (2017). Knowledge management in startups: Systematic literature review and future research agenda. *Sustainability, 9*(3), 361. https://doi.org/10.3390/su9030361

Clegg, C. (2000). Sociotechnical principles for system design. *Applied Ergonomics, 31*(5), 463–477. https://doi.org/10.1016/S0003-6870(00)00009-0

Herbsleb, J. D. (2007). Global software engineering: The future of socio-technical coordination. In *Proceedings of the Future of Software Engineering (FOSE '07)* (pp. 188–198). IEEE. https://doi.org/10.1109/FOSE.2007.11

Hofstede, G. (2001). *Culture's consequences: Comparing values, behaviors, institutions, and organizations across nations* (2nd ed.). SAGE.

Iqbal, S. T., & Horvitz, E. (2010). Notifications and awareness: A field study of alert usage and preferences. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW '10)* (pp. 27–30). ACM. https://doi.org/10.1145/1718918. 1718926

Kirkpatrick, D. L., & Kirkpatrick, J. D. (2006). *Evaluating training programs: The four levels* (3rd ed.). Berrett-Koehler.

Landsberger, H. A. (1958). *Hawthorne revisited: Management and the worker, its critics, and developments in human relations in industry*. Cornell University Press.

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys, 26*(1), 87–119. https://doi.org/10.1145/174666.174668

Mark, G., Gudith, D., & Klocke, U. (2008). The cost of interrupted work: More speed and stress. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)* (pp. 107–110). ACM. https://doi.org/10.1145/1357054.1357072

McCambridge, J., de Bruin, M., & Witton, J. (2012). The effects of demand characteristics on research participant behaviours in non-laboratory settings. *PLoS ONE, 7*(6), e39116. https://doi.org/10.1371/journal.pone.0039116

Mumford, E. (2006). The story of socio-technical design: Reflections on its successes, failures and potential. *Information Systems Journal, 16*(4), 317–342. https://doi.org/10.1111/ j.1365-2575.2006.00221.x

Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. (2014). Software development in startup companies: A systematic mapping study. *Information and Software Technology, 56*(10), 1200–1218. https://doi.org/10.1016/ j.infsof.2014.04.014

Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice* (4th ed.). SAGE.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Riedl, C., Kim, Y., Gupta, P., Malone, T. W., & Woolley, A. W. (2021). Quantifying collective intelligence in human groups. *Proceedings of the National Academy of Sciences, 118*(21), e2005737118. https://doi.org/10.1073/pnas.2005737118

Rowley, J. (2007). The wisdom hierarchy: Representations of the DIKW hierarchy. *Journal of Information Science, 33*(2), 163–180. https://doi.org/10.1177/0165551506070706

Shujahat, M., Sousa, M. J., Hussain, S., Nawaz, F., Wang, M., & Umer, M. (2019). Translating knowledge management into innovation. *Journal of Business Research, 94*, 442–450. https://doi.org/10.1016/j.jbusres.2017.11.001

Smite, D., Moe, N. B., Klotins, E., & Gonzalez-Huerta, J. (2023). Remote work and post-pandemic transitions. *Journal of Systems and Software, 195*, 111509. https://doi.org/10.1016/ j.jss.2022.111509

Stray, V., & Moe, N. B. (2020). Coordination in global software engineering. *Journal of Systems and Software, 170*, 110717. https://doi.org/10.1016/j.jss.2020.110717

Thaler, R. H., & Sunstein, C. R. (2008). *Nudge: Improving decisions about health, wealth, and happiness*. Yale University Press.

Tracy, S. J. (2010). Qualitative quality: Eight big-tent criteria. *Qualitative Inquiry, 16*(10), 837–851. https://doi.org/10.1177/1077800410383121

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). SAGE