CAMO
HKU Centre for AI, Management and Organization

June 2026

JIE GONG, JIN LI, AOFEI CHEN, AND KUI WANG

Tapread: From R&D Queue to Business Flow

Executive Summary

Many fast-moving companies have faced a common challenge: every request, no matter how large or small, is submitted to a single queue and processed one after the other. Although the funnel helps maintain order, it slows down the team. Tapread's overseas technology team employed AI to recast this as a portfolio problem. It categorized them by risk and gave each type its own delivery path. Product and operations personnel used AI to develop small, low-risk tools; engineers worked with AI to implement mid-sized projects across technical stacks; and core systems stayed on the full engineering process, kept off the lighter paths. Monthly output increased from 5–8 items to 20–25 without increasing manpower.

Background

Qimao is one of China's largest digital reading companies, best known for its free novel app with a large user base across China. Tapread is its overseas web literature platform, connecting readers in many countries with web novels and their authors. Operating across many markets makes the work less predictable than a settled home market: what performs well in one country can do poorly in the next, advertising channels change frequently, and reader preferences vary across regions.

Keeping pace with these markets means constant operational change, such as new creatives, channels, and landing pages, and that work generates a steady stream of technical requests. The requests range widely in size, from full product features down to a quick script that pulls and cleans operational data, a configuration tool, an internal dashboard, or an ad-testing tool.

The Bottleneck

Traditionally, every technical request followed the same fixed, serial process. A business team initiated a request, a product manager wrote it up as requirements, R&D fit it into a schedule and developed it, QA tested it, and operations put it to use. This sequence kept things orderly, but it was slow.

One reason was capacity: R&D headcount was fixed, but the number of requests increased rapidly, leading to a long queueing time for even small requests. In an advertising-driven business, the wait can be very costly: a tool delivered after the campaign window had closed is worth little.

Information was also lost during the process. As a request moved down the line it was rewritten at each step. A question such as “Can we build dozens of ad variations at once?” had to be translated into product language, then technical language, and finally into a testable logic statement. Each handoff preserved structure, but the urgency and context were sacrificed.

©2026 by HKU Centre for AI, Management and Organization

The fundamental issue was uniformity: the process treated a change in a core user-asset system the same as a throwaway internal script. Both were placed into the same queue and waited the same way, so a one-day project could take two or three weeks, mostly because of waiting. Essentially, this created a sorting question: which project requires the entire R&D process, and which just needs fast validation?

The Intervention

In October 2025, Tapread ran a pilot within its overseas business technology team consisting of nine R&D members, three product managers, six operations and advertising colleagues, and two QA members. The pilot stayed away from high-risk systems and began with low-risk, high-frequency work such as internal dashboards, tools to set up and test advertising campaigns, and small scripts for pulling and cleaning operational data.

The team applied three types of AI tools to perform different functions.

First were general coding assistants such as Cursor and Codex, which helped engineers to code, debug, and refactor code, as well as navigate parts of the system they did not normally work in. With these tools, a single engineer could work cross-stack, handling the frontend, backend, and data side of a task that previously required a specialist in each.

Second were general chat assistants such as ChatGPT and Claude, which are widely used across the team. Business, product, and engineering members used them to decompose a request into parts, discuss the approach, write interface documentation, generate test cases, and review what went wrong when an incident occurred. The chain of translation between them grew shorter because everyone could work from the same plain-language description.

Third were tools that Tapread built for itself. The team provided the knowledge base and project context to the AI tools, so they understood the company's specific systems and content, and then put them to work on tasks that only a web-literature platform has: translating stories for new markets, checking content quality at scale, sorting reader comments and complaints, and analyzing advertising material.

Together, these tools changed who could supply technical work. Before, only R&D team could, which is why every task, no matter how simple, was queued for the R&D team. Now, with the help of AI, product and operations staff can build small tools themselves, and a single engineer can span the stack on their own.

To take advantage of this extra capability, Tapread sorted requests into three routes by risk. Low-risk work, like the tools the pilot had started with, was delegated to product and operations teams. Middle-risk tasks were assigned to individual engineers. High-risk requests such as core transactions, recommendation algorithms, user assets, and payment links remained on the traditional R&D route. In this way, Tapread managed its technical work as a portfolio of three routes rather than a single queue.

Results

With this new model, small and mid-size work was delivered much faster. For example, small tools that used to take three to five days now could be done within a day. Mid-sized work that took weeks now takes days. Counting all types of requests that entered the R&D queue, monthly throughput increased from 5–8 to 20–25, yet the number of R&D staff remains nine. High-risk systems on the governed path follow the full cycle, which takes five to seven days.

Additional routes contribute to the speed gain. Moreover, within each route, a request also moved faster, as it passed through fewer hands and cleared with less waiting and bouncing back.

Measure	Before reform	After reform
Small tools and simple configuration	3–5 days	**Within a day**
Medium-complexity business needs	1–2 weeks	**1–3 days**
Monthly demand throughput	5–8 items	**20–25 items**
R&D headcount	9	**9**
Tools built by operations / product	~ 0	**4**
Production incidents	0	**0**

Advertising demonstrates the changes concretely. The work moved from making ads by hand to running them at scale. In the past, a tool to build ads in bulk would have spent so long in the R&D queue that the campaign window would close before it shipped, so the work stayed manual. After the change, operations and R&D built the tool directly and shipped a gray version (a limited first release, before a full rollout) in three days, well inside the campaign window. The team could now create ads in batches, test many combinations at once, and adjust budget and creatives while a campaign was still live.

The speed of the new system also brought some risk, and the team encountered it once. In one fast launch, a tool had been built too fast to fully map how the modules it relied on connected to one another. When it went live, an overlooked dependency took the tool down briefly. Because a gray release and rollback procedure were already in place, the team reverted within five minutes and no customer noticed the impact. Tapread turned this incident into a rule: When a task involves several modules, core dependencies, or a large portion of the system, the team now maps the module relationships, interface dependencies, rollback plan, and validation checklist before it goes live. The same logic governs upkeep: a tool can stay lightweight while it is used rarely, but once it becomes frequent or embedded in the business, R&D moves it into a governed production environment.

Takeaway

Tapread changed who can do the technical work. Before, only R&D could, so all requests, whether a change to a core system or a throwaway internal script, waited in the same queue. Now, work is sorted by risk: product and operations build small, low-risk tools with AI, one engineer does mid-size work across the stack, and high-risk systems go through the full process.

The key takeaway is that AI enables the business to distinguish between the work that needs a careful process and the work that only needs to be fast and safe. For a company that moves quickly, that distinction is more important than speed alone. Some work must be done right, even if that takes time. Other work must be done before the window of opportunity expires. What mattered was that Tapread stopped putting both through the same process.