

Version-Controlled PM Work, Validation Before Scale, and Fintech Proof-of-Work

PM Daily Digest

2026-06-28

Version-Controlled PM Work, Validation Before Scale, and Fintech Proof-of-Work

By PM Daily Digest • June 28, 2026

This brief centers on one operating shift: storing AI-era PM artifacts in version-controlled repos instead of scattered folders. It also pulls tactical lessons on validation, release documentation, fintech job hunting, and reusable PM skill libraries.

Big Ideas

- **Treat GitHub as PM infrastructure, not just a code tool.** Aakash Gupta's framework is to version-control the PM artifacts that now shape AI-assisted work: personal `CLAUDE.md`, skills, autoresearch configs, eval criteria, `PLANNING.md`, and code [1]. The structure is three repos: a private workspace, a shared-tools repo with stripped context, and one repo per initiative [1]. **Why it matters:** this makes drift reversible, changes reviewable, and project history durable instead of scattered across folders and local files [1, 2]. **How to apply:** separate personal context, team assets, and project records rather than keeping everything in one workspace.

Tactical Playbook

1. **Adopt the seven-step rhythm.** Pull, branch, edit, commit, push, PR, merge—using natural-language commands in Claude Code rather than memorizing git syntax [1, 2]. **Why it matters:** it standardizes how PM artifacts change and adds review points when work is shared [1]. **How to apply:** skip branches and PRs only for solo work in a private repo; use them for shared tools and team workflows [1].
2. **Use version control for four PM-specific jobs.**

- Roll back a degraded skill or reviewer by checking out the previous hash [2]
- Prune `CLAUDE.md` regularly; the source warns that files over 200 lines cause Claude to deprioritize instructions [2]
- Treat `git log` as your autoresearch experiment record [2]
- Version evals so a score drop—like 0.82 to 0.71—can be traced to model regression versus changed criteria [2]

Why it matters: this turns AI workflow quality into something diagnosable rather than anecdotal. **How to apply:** commit after each meaningful change, and make commit messages specific enough to explain what changed and why [1].

3. **Do security work before the first push.** Use `.gitignore`, run a secret scan, and exclude API keys, `.env` files, customer data, internal docs, HR feedback, raw Slack exports, and sensitive screenshots [1].

“A private repo is not a privacy strategy.” [1]

Case Studies & Lessons

- **Validation still comes before scale.** One PM described two Australian startups with functional MVPs and \$0 revenue: an employee cognition platform with 13 staff averaging about 110k AUD, and a consumer cash-back app with about 15 full-time employees [3]. The reported common issue was weak product validation, combined with founders removed from day-to-day operations [3]. **Why it matters:** funding and team size can make unvalidated bets more expensive, not safer [3]. **How to apply:** validate continuously before expanding team or burn.
- **Documentation needs an audience, not total coverage.** In a release-doc discussion, commenters warned that documents meant for “everyone” and “everything” usually fail because the purpose and reader are not defined [4, 5].

“A document like this that tries to serve everyone ends up serving no one.” [6]

Why it matters: unclear audience is often a sign the team has not defined what the document should actually do [5]. **How to apply:** write launch docs against one job—decision, alignment, readiness, or reference—then scope content to that job.

Career Corner

- **Break into fintech PM with one end-to-end proof piece.** One practical recommendation: build a real fintech-ish side project and use that story in every interview [7]. **Why it matters:** it gives you concrete evidence of product thinking, execution, and domain interest when your

resume title does not yet do that. **How to apply:** be ready to explain the problem, target user, product choices, trade-offs, and what you learned.

- **Local compensation benchmarking can be noisy.** One Chandigarh Tricity PM reported earning 25 LPA with 8 years of experience and said some small or mid-sized B2B SaaS firms are offering below online benchmarks of 35+ LPA [8]. **Why it matters:** broad market ranges may not match local offers. **How to apply:** gather location- and role-specific datapoints before negotiating.

Tools & Resources

- Examples of shareable PM skill repos mentioned in the source: gstack, with 23 specialist skills; pm-skills, with 100+ skills across 8 plugins; and claude-code-pm-course, an interactive Claude Code course [1]. The same post also points to a PM `.gitignore` template for keeping secrets out of GitHub [1]. **Why it matters:** these are concrete examples of what a shared-tools repo can look like. **How to apply:** borrow structure and reuse patterns, but strip personal and company-specific context from anything you publish [1].

Sources

1. GitHub for PMs: Version Control for Everything You Build With AI
2. substack
3. r/ProductMgmt post by u/Impossible_Banana649
4. r/ProductManagement comment by u/SmartName_
5. r/ProductManagement comment by u/Particular-Fennel-67
6. r/ProductManagement comment by u/steakinapan
7. r/prodmgmt comment by u/Ok-Wedding8049
8. r/ProductManagementJobs post by u/Old_Set8146