

# Executive Influence, Founder-Led Coding, and Repo-Driven PM Work

PM Daily Digest

2026-03-23

## Executive Influence, Founder-Led Coding, and Repo-Driven PM Work

*By PM Daily Digest • March 23, 2026*

This issue focuses on Jessica Fain’s executive influence framework—meeting setup, incentive alignment, and trust-building—plus two adjacent shifts: founder-led coding at v1 and repo-driven PM work that grounds specs in reality.

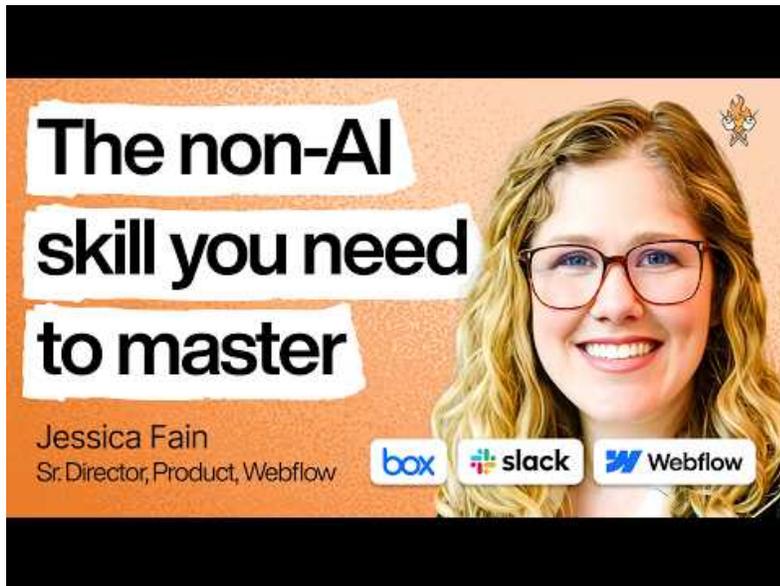
### Big Ideas

#### 1) Influence is becoming a higher-leverage PM skill

Jessica Fain’s argument is that as execution gets easier, PM leverage shifts toward deciding what work survives and bringing others along. She frames influence, stakeholder management, and learning as the “10x skill” [1]. Her core mental model is to treat executives like users, with the same curiosity and empathy you would apply in discovery [1, 2].

- **Why it matters:** Executives are usually context-switching across budget, people, legal, and other issues before they enter your review. They have not been centered on your problem the way you have [1, 2].
- **How to apply:** Build reviews around executive context: reset the room quickly, map your pitch to their incentives, and use feedback to learn rather than defend.

“People don’t realize that an executive’s calendar is like a strobe light



going off.” [2]

*The art of influence: The single most important skill left that AI can't replace / Jessica Fain (0:00)*

## 2) Founder-led coding is emerging as a zero-to-one pattern

Andrew Chen expects more “founder-led coding” as non-technical founders use AI code generation to build v1 products themselves. His analogy is founder-led sales: do the work yourself early, even if you are not yet great at it, because the point is learning and validating the product [3].

- **Why it matters:** At the earliest stage, the value is learning and validation, not polished process [3].
- **How to apply:** If you are working on a true v1, use AI-assisted building to test the idea, uncover constraints, and tighten the product before you scale delivery.

## 3) Repo access is becoming a PM reality-check layer

Community examples show PMs using GitHub repo access, often with AI, to understand architecture and business logic, baseline feature areas against current code, inspect event tags, write better tickets, and generate release notes [4, 5, 6, 7, 8].

- **Why it matters:** Multiple practitioners describe repo access as a way to start from reality rather than assumptions and to improve prioritization and roadmap conversations with engineering [9, 10].
- **How to apply:** Treat the repo as a source of product context, not just engineering context, especially before writing a PRD or debating scope.

## Tactical Playbook

### 1) Run a better executive review in 6 steps

1. Start with a 30-60 second reset: why you're here, where the last discussion ended, today's goals, and how the meeting will run [1].
2. Ask if there is anything else they hoped to cover so you can adapt before the conversation drifts [1].
3. Match the format to how they process information—doc, design, customer story, dashboard, or experiment [1].
4. Tie the proposal to how they are measured: goals, key metrics, OKRs, or current board pressure [1].
5. Go in to learn, not to convince. Use questions like “That’s so interesting. What led you to believe that?” to surface the belief behind the feedback [1].
6. Bring options, not just one answer. Fain’s “Stuart plus two more” method was to return with what the executive asked for plus two additional versions that created a real decision discussion; in one example, her team turned around a new document in two days after a weak review [1].
  - **Why it matters:** Fain’s warning is that once you go much past 60 seconds at the top, you’ve lost them [1].
  - **How to apply:** Use the opener to re-create context fast, then spend the rest of the meeting learning what the executive actually believes and needs.

### 2) Build trust the senior way

1. Kill or deprioritize ideas that are not working [1].
2. Make the decision criteria explicit and say when you’ll come back with a call [1].
3. Shrink large changes into smaller experiments or proof-of-concepts to lower perceived risk [1].
4. Follow feedback quickly; if you wait a week, the executive has often moved on [1].
  - **Why it matters:** These moves signal aligned incentives with the company outcome rather than attachment to your own roadmap [1].
  - **How to apply:** Define kill criteria, propose the smallest credible test, and turn around follow-ups while the conversation is still warm.

### 3) Use the repo before you write the spec

1. Ask AI or Copilot to explain the current implementation in plain English [11, 12].
2. Check existing code paths for edge cases and scope assumptions before you write the PRD [9, 6].

3. Search comments and tags when you need to understand instrumentation, events, or legacy behavior [7].
  4. Use what you learn to write clearer tickets and summarize changes or PRs into release notes [8, 6].
  5. If the team is already sharing AI workflows, version prompts, skills, or agents in GitHub so others can reuse them [4, 5].
- **Why it matters:** Several commenters describe this as a direct way to save wasted spec writing and improve engineering conversations [9, 10].
  - **How to apply:** Do a repo pass before major scoping work, then bring the resulting constraints and edge cases into discovery and prioritization.

## Case Studies & Lessons

### 1) Jessica Fain used PM skills to get inside the exec decision-making loop

In 2017, Fain approached Slack CPO April Underwood with problems she saw in how work was getting done, paired them with proposed solutions, and tried to understand Underwood and other senior leaders. Underwood says that approach “used PM skills on me” and helped land Fain the Chief of Staff role; she also credits Fain with helping Slack through IPO prep, competitive pressure, and a leadership transition [13]. Fain later said she took the role because her product ideas kept dying and she wanted to see how executive decisions were actually made from the inside [2].

“She used her PM skills on me, the CPO - it was classic needs-finding and the best kind of selling” [13]

- **Lesson:** Influence with executives starts before the meeting: identify the organizational problem, bring possible solutions, and show that you understand leadership constraints [13, 2].
- **How to apply:** When a cross-functional problem keeps repeating, package it the way you would package a user problem: pain, evidence, candidate solutions, and the executive’s likely incentives.

### 2) Slack’s Customer Love Sprint turned quality work into a strategic signal

At Slack, the team paused normal engineering work for two weeks and let engineers pick fixes that would be good for users. PM, design, and support supplied ideas, but the only rule was that the work had to ship something good for users [1]. The sprint produced 65 improvements and included a judging process that involved executives [1]. Fain says it helped bring back a part of the culture the company had lost and aligned with what leaders believed differentiated Slack in the market [1].

- **Lesson:** A short execution sprint can be a product strategy tool, not just

a morale exercise, if it ties visible user improvements to company-level differentiation [1].

- **How to apply:** If quality or craft is drifting, run a time-boxed improvement sprint with clear shipping criteria, broad idea input, and executive visibility.

### 3) Repo read access prevented wasted PRD work

One startup PM said repo read access let them inspect existing code paths before writing a PRD and discover that some edge cases were already handled, while some changes they expected to be a large lift were actually easy under the current architecture [9]. Another PM said repository analysis with Claude Code gave their product team better information for prioritization and for discussions with IT on deadlines and roadmap changes [10].

- **Lesson:** Technical context can materially change the product plan before requirements are written [9, 10].
- **How to apply:** Insert a codebase review step between discovery and spec drafting when the product area is complex or legacy-heavy.

## Career Corner

### 1) In the AI era, influence looks like career leverage

Fain argues that as execution complexity drops, PM leverage shifts away from being the best note taker or experiment runner and toward deciding what work survives and getting people to buy into it [1].

- **Why it matters:** This is the part of PM work she frames as the enduring leverage point [1].
- **How to apply:** Deliberately practice stakeholder mapping, review prep, and post-meeting follow-up alongside core execution skills.

### 2) Show seniority by acting like a CPO

Fain’s advice for becoming more senior is to be the deepest domain expert in the room and “act like a CPO” [1]. She pairs that with expanding your viewpoint from local optimization to global company thinking [1].

- **Why it matters:** Seniority here is framed as scope of judgment, not just scope of ownership.
- **How to apply:** In reviews, speak to company outcomes, not just team outputs, and bring solution-oriented thinking rather than only surfacing problems [1].

### 3) Use AI as a rehearsal partner for tough reviews

Fain describes a colleague who trained a model on past product review transcripts and expects PMs to run PRDs or pitches through it to identify likely

pushback. She also suggests using AI against your own known weaknesses, such as thin data or thin UX thinking [1].

- **Why it matters:** This turns past organizational feedback into reusable coaching [1].
- **How to apply:** Before a major review, test whether your doc would survive the objections your executives typically raise [1].

## Tools & Resources

- The art of influence: The single most important skill — the full conversation with Jessica Fain on executive influence and stakeholder management [14]
- The art of influence: The single most important skill left that AI can't replace | Jessica Fain — video version with the meeting opener, trust-building tactics, and the Slack sprint example [1]
- How do PMs use access to Github repos for their work — a practical thread on repo access for PRD baselining, release notes, metrics debugging, and prompt sharing [6, 9, 7, 4]
- **GitHub Copilot** — worth exploring if you need help turning legacy repos into plain-English docs, understanding recent changes, or drafting technical spikes [12, 11]
- **Claude Code** — PM teams describe using it to analyze repositories, understand architecture and business logic, and improve prioritization conversations with engineering and IT [10, 5]
- **GitHub as a prompt/versioning layer** — some PMs are sharing prompts, skills, agents, and queries there with engineers and other PMs [4, 5]

---

## Sources

1. The art of influence: The single most important skill left that AI can't replace | Jessica Fain
2. X post by @lennysan
3. X post by @andrewchen
4. r/ProductManagement comment by u/userreddit
5. r/ProductManagement comment by u/userreddit
6. r/ProductManagement post by u/Wmonk47\_2071
7. r/ProductManagement comment by u/ryan\_with\_a\_why
8. r/ProductManagement comment by u/512damon
9. r/ProductManagement comment by u/84tiramisu
10. r/ProductManagement comment by u/pidgeonsarehumanstoo
11. r/ProductManagement comment by u/onethreeone
12. r/ProductManagement comment by u/Sufficient-Rough-647
13. X post by @aunder

14. X post by @lennysan