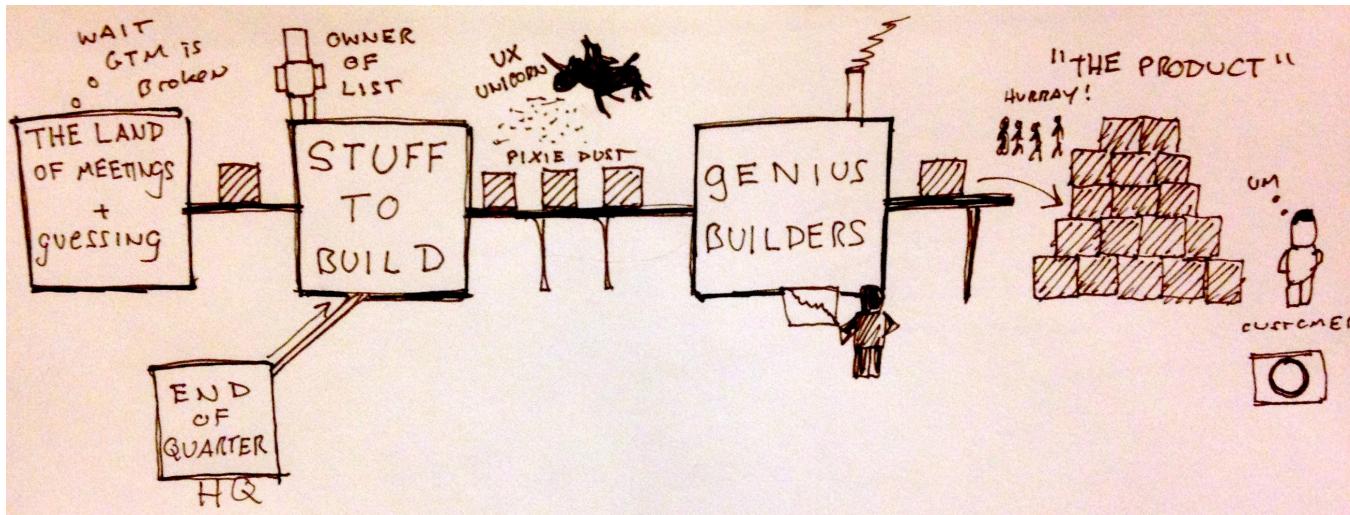


## 12 Signs You're Working in a Feature Factory



John Cutler [Follow](#)  
Nov 16, 2016 · 4 min read



- I've used the term *Feature Factory* at a couple conference talks over the past two years. I started using the term when a software developer friend complained that he was "just sitting in the factory, cranking out features, and sending them down the line."

How do you know if you're working in a feature factory?

1. **No measurement.** Teams do not measure the impact of their work. Or, if measurement happens, it is done in isolation by the product management team and selectively shared. You have no idea if your work worked
2. **Rapid shuffling of teams and projects** (aka Team Tetris). Instead of compelling missions or initiatives, teams deal in feature and project assignments. Chronic multitasking and over-utilization
3. **Success theater** around "shipping" with little discussion about impact. You can tell a great deal about an organization by what it celebrates
4. **Infrequent (acknowledged) failures** and scrapped work. No removed features. Primary measure of success is delivered features, not delivered outcomes. Work is rarely discarded in light of data and learning. Often the team lacks the prerequisite safety to admit misfires
5. **No connection to core metrics.** Infrequent discussions about desired customer and business outcomes. Team cannot connect work to key business and customer satisfaction metrics. Impossible to connect iterations to "what matters most"
6. **No PM retrospectives.** Product managers do not conduct regular retrospectives on the quality of their product decisions and compare expected benefits to actual benefits. Developers have "passing tests", but product managers do not. Product managers view velocity and output as their key performance

indicator

**Obsessing about prioritization.** Mismatch between prioritization rigor (deciding what gets worked on) and validation rigor (deciding if it was, in fact, the right thing to work on). Prioritization rigor is designed exclusively to temper internal agendas so that people “feel confident”. Lots of work goes into determining which ideas to work on, leaving little leeway for adjustments and improvisation based on data.

**Roadmaps show a list of features**, not areas of focus and/or outcomes

**No tweaking.** Once work is “done”, the team moves immediately on to the next “project”, leaving no time to iterate based on qualitative and quantitative data

**Culture of hand-offs.** Front-loaded process in place to “get ahead of the work” so that items are “ready for engineering”. Team is not directly involved in research, problem exploration, or experimentation and validation. Once work is shipped, team has little contact with support, customer success, and sales

**Large batches.** Without the mandate to experiment, features are delivered in single large batches instead of delivering incrementally. You might still work in sprints (yay, we’re “Agile”), but nothing new is reaching customers at the conclusion of each sprint

**Chasing upfront revenue.** Features are implemented to close new deals. While not inherently wrong, the economic justifications are often flimsy (at best), and fail to account for the non-linear increase in product complexity (you make the quarter, but you pay for it many times over later). Again, this reinforces the idea that features are the unit of value measurement. Product decisions lack a sound economic grounding

**Shiny objects.** Low visibility for refactoring work and debt work-down. Low visibility for overall value delivery capabilities. As mentioned, primary measure of success is new feature output. Little appreciation for the health of the whole product as opposed to shiny new objects. Little awareness around impact of new features on usability (and maintainability and extensibility) of existing product

For translations of this post see: Korean

I've written about addressing this problem. Check out these posts:

[A 12-Step Plan for Recovering Product Managers](#)

[Success Theater ...](#)

[Your product management approach needs to evolve](#)

[Dear Product Manager...](#)

[Quit Planning Ahead and Keeping People Busy](#)

[Beat the Feature Factory—With Biz Chops](#)

[Beat the Feature Factory: Run Pre-cap Design Studios](#)

[SaaS and the Impostor Clairvoyant PM](#)

[Agile: Don't Exchange Waterfalls for Whirlpools](#)

[Visualizing Debt, Rework, Cut Corners, and Frustration](#)

[The Overlap](#)

[5 Simple Questions to Drive Validated Learning](#)

[Stop Setting Up Product Roadmaps To Fail](#)

How Much Does A New Feature Cost?

User Stories and Data

... actually it is a theme in most of my posts



*Hacker Noon is how hackers start their afternoons. We're a part of the @AMI family. We are now accepting submissions and happy to discuss advertising & sponsorship opportunities.*

*If you enjoyed this story, we recommend reading our latest tech stories and trending tech stories. Until next time, don't take the realities of the world for granted!*



