My first job out of Stanford's Computer Science program was as a Product Manager at Google. It was an amazing first job. I shared offices with engineers who had written the textbooks I used in college. I helped create Google Maps, which is still my proudest achievement as a product designer and engineer. And I learned how to be effective on large scale software projects. By the time I left Google to start my first company, FriendFeed, I had worked on so many projects at scale, I felt extremely confident I could start one myself.

However, being a product manager at a large company is different than founding a startup. For one, you are judged differently. While, in theory, product managers are judged on the success of the products they work on, in practice, large companies also judge product managers on their ability to manage all the people and departments that have a stake in a product's outcome. Did you engage the PR team enough time before launch? Did you integrate your product with the CEO's pet project? Did you convince the competing executive of your product direction before the big executive review? At software companies that aren't as enlightened as Google, product managers are judged more on these political issues than they are on any aspect of the products they work on.

That's why so many engineers and product managers that come out of larger companies have trouble with the concept of *leverage* that Edmond Lau talks about in *The Effective Engineer*. They are effectively trained to care about low leverage activities because the bureaucracy that trained them values and rewards them. The most successful engineers I've worked with in my career were the few that were able to see past these bureaucratic idiosyncrasies and recognize the one or two things that would really impact their product's success. The engineer that taught me the most about leverage is, without question, Paul Buchheit.

Paul was one of the co-founders of FriendFeed. He had previously created Gmail, and while we hadn't worked together much at Google, we had enough mutual respect to join forces with Jim Norris and Sanjeev Singh to start a company together in mid-2007. More than any person I've ever met, Paul was willing to challenge conventional thinking, and he completely changed my perspective on engineering and product management.

Whenever we would encounter a challenging technical problem, I would ask "How should we do this?" Paul would respond, often obnoxiously, "Why do we need to do it at all?" Instead of trying to solve impossible problems, he would more often challenge the assumptions so we could simply work around them. At times, it almost seemed like Paul was lazy—given any sufficiently hard project, Paul would question the purpose of the project. But he was almost always right. Unless the project was destined to make or break our nascent company, why spend our precious engineering resources on it?