CHAPTER

1

Behind Every Great Company

In this book, I want to share and highlight the differences between how *the best* companies create technology-powered products and how *most* companies create products.

The differences are both fundamental and striking.

The differences certainly include what many people think of as "product culture," but strong product companies often have very different cultures from one another, so it clearly goes beyond that.

For example, consider Amazon, Google, Apple, and Netflix. I would argue all four are very strong product companies, having consistently innovated for many years, yet they each have very different cultures.

I still believe culture is extremely important, but there is something about great product companies that is more fundamental.

It comes down to the views they have on the role of technology, the purpose of the people who work on the technology, and how they expect these people to work together to solve problems.

Moreover, I don't think it's an accident that, despite their different cultures, these four companies have the most important elements in common.

What I will try to do in this book is untangle the parts of the cultures of these companies that are more a reflection of their founders' personalities from those that are essential to consistent innovation.

I want to share the important lessons I've learned regarding what separates the best from the rest.

One surprising common thread among many of the best product companies is the legendary coach, Bill Campbell. During their formative years, Bill literally provided executive coaching to the founders of Apple, Amazon, and Google, as well as several others.

To get a sense of Bill's views and values, here is one of my favorite quotes about the role of leadership in a strong product company:

> *Leadership is about recognizing that there's a greatness in everyone, and your job is to create an environment where that greatness can emerge.*

This book is all about identifying what makes such an environment, and I want to encourage you to consider adopting these important practices and behaviors.

Please note that I am not arguing that these strong product companies are models of virtue. All of them have been justifiably criticized about some of their policies and practices.[1]

But when it comes to the ability to consistently innovate, all four of these companies have demonstrated their skills, and I believe there is much to be learned from them.

At the core, I see three critically important differences between the strongest product companies and the rest:

The first is how the company views the role of technology.

The second is the role their product leaders play.

The third is how the company views the purpose of the product teams—the product managers, product designers, and engineers.

Let's take a closer look at each of these.

---

[1]For an unflinching critique of these companies when it comes to their policies, see the writings of Professor Scott Galloway (www.profgalloway.com).

The Role of Technology

There is a fundamental difference between how strong companies view the role and purpose of technology as compared to most other companies.

At its most basic level, the vast majority of companies view technology as a necessary expense. They know it's important, but they think of it more as a cost of doing business. If they can outsource the labor, even better. Fundamentally, they don't really consider themselves in the technology business. Instead, they think of themselves as in the insurance business, or the banking business, or the transportation business, or whatever. Certainly, they need some technology to operate, but it's viewed as a subservient role to "the business."

Because of that, in most companies, technology teams exist *to serve the business*. That is very often the exact phrase you will hear. But even if they aren't explicit about it, the different parts of "the business" end up driving what is actually built by the product teams.

In contrast, in strong product companies, technology is not an expense, it *is* the business. Technology enables and powers the products and services we provide to our customers. Technology allows us to solve problems for our customers in ways that are just now possible.

Whether the product or service is an insurance policy, a bank account, or an overnight parcel delivery, that product now has enabling technology at its core.

As such, in *strong* product companies, the purpose of the product team is *to serve customers by creating products customers love*, *yet work for the business*.

That is a profound difference, which impacts nearly everything about the company and how it works, and results in much higher motivation and morale. And most important, it results in a much higher level of innovation and value for customers and the business.

Strong Product Leadership

In *most* product companies, the role of true product leadership is largely missing in action.

Instead, they are mainly there as facilitators, responsible for staffing the in-house (or even worse, outsourced) feature factory, and keeping the trains running on time.

In *most* companies, there is no product strategy. Notice I didn't say a bad product strategy—I mean literally *no* product strategy. The feature teams are simply there "to serve the business."

The business certainly has reasons for what they request or put on the roadmaps, but they very rarely have a product strategy, or even the skills or data required to create one.

The stakeholders end up providing product teams with a prioritized list of features and projects that they need completed this quarter or this year. So, the "product strategy," if you could even call it that, is really about trying to please as much of the business as possible.

When technology product companies moved to Agile methods over the past 10–20 years, many managers and leaders questioned whether they were still necessary, since team members would be expected to take a much more active role in how they work.

I realize this is counterintuitive to many people, but while moving to truly empowered teams does require moving away from the old command-and-control model of management, it does *not* mean you need fewer leaders and managers. It means you need *better* leaders and managers.

It's actually easier for a manager to manage (often micromanage) in the old command-and-control style. It's not hard to assign a team a list of activities, or a list of features to build, and just tell them to do the work as fast as they can.

While this command-and-control style may be easier for the manager, it creates teams of mercenaries with no empowerment in any meaningful sense.

In contrast, in *strong* product companies, the product leaders are among the most impactful leaders in the company.

They are responsible for staffing and coaching the product teams; they are responsible for the product strategy and converting the strategy into action; and they're responsible for managing to results.

Empowered product teams depend on skilled product managers, product designers, and engineers, and it is the leaders and managers who are responsible for recruiting, hiring, and coaching these people.

Further, a focused and compelling product strategy—based on quantitative and qualitative insights—is among the most important contributions of product leadership.

Empowered Product Teams

In *most* companies, the technology teams are not *empowered* product teams, they are what I call here *feature teams*.

Feature teams look superficially like a product team. They are cross-functional, with a product manager, a product designer, and some number of engineers. The difference is that they are all about implementing features and projects (*output*), and as such are not empowered or held accountable to results.

The feature teams get to work first designing the features on the roadmap, maybe doing a little usability testing, and then proceeding to building, QA testing, and deploying the features (known as *delivery*).

These feature teams sometimes claim they're doing some product discovery, but they rarely are. They've already been told what the solution should be; they're not empowered to go figure out the solution themselves. They're just there to design and then code.

In these feature teams, there is usually a person with the product manager title, but they are mainly doing *project* management. They are there to ensure the features get designed and delivered. Necessary perhaps, but this is not *product* management.

Because the teams are provided, or are pressed to provide, roadmaps of features and projects, the focus of the team is *delivery*—delivery of these features. And features are output. Even if someone were to complain of lack of business results, who would you hold accountable?

In contrast, in strong product companies, teams are instead given *problems to solve*, rather than features to build, and most important, they are *empowered to solve those problems in the best way they see fit*. And they are then held accountable to the results.

In the empowered product team model, the product manager has a clear responsibility, which is to ensure that the solutions are *valuable* (our customers will buy the product and/or choose to use it), and *viable* (it will meet the needs of the business). Together with a product designer who is responsible for ensuring the solution is *usable*, and a tech lead who is responsible for ensuring the solution is *feasible*, the team is able to collaborate to address this full range of risks (value, viability, usability, and feasibility). Together, they own the problem and are responsible and accountable for the results.[2]

So, to summarize feature teams vs. empowered product teams:

Feature teams are cross-functional (a product manager doing mainly project management, a product designer, plus some engineers), and assigned features and projects to build rather than problems to solve, and as such they are all about output and not business results.

Empowered product teams are also cross-functional (a product manager, a product designer, and engineers), but in contrast to feature teams, they are assigned *problems to solve*, and are then empowered to come up with solutions that work—measured by outcome—and held accountable to results.[3]

Product Discovery

If you have not yet read *INSPIRED*, then you might be wondering: *What is so wrong with the business owners and stakeholders deciding what goes on the roadmap*, and therefore what the engineers should build?

This is considered the first and most important principle of product discovery: *our customers, and our stakeholders, aren't able to tell us what to build*.

It's not because our customers or stakeholders aren't smart or knowledgeable.

[2] To be clear, the designer and tech lead contribute much more than simply ensuring usability and feasibility respectively; what I'm referring to here is who we hold responsible and accountable for each risk.

[3] There is actually a third type of technology team, which is referred to as a *delivery team* (or "Scrum team" or "dev team"). A delivery team doesn't even pretend to be a true product team. They are not cross-functional, and they are not empowered. There is a product *owner* (responsible for administering the product backlog) and some number of engineers. They are purely about output (code and ship). If you're running a process like SAFe, then this is unfortunately you, and truthfully, I have no idea why you would want to read this book, since what I describe here is the polar opposite both philosophically and practically.

There are two fundamental reasons why our customers and stakeholders aren't able to tell us what to build:

First, the customers and stakeholders don't know what is just now possible—they are not experts in the enabling technologies, so they can't be expected to know the best way to solve the problems we're focused on, or even if the problem is possible to solve. It's often the case that innovations solve problems in ways that customers and stakeholders had no idea was possible.

Second, with technology products, it's very hard to predict in advance what solutions will work. There are many reasons why product ideas don't deliver the results we hoped. All too often we are excited about some idea, but our customers are not, so they don't buy what we thought they would. Or, we discover the idea has major privacy or security issues. Or we find out the idea will take much longer to build than anyone expected.

Empowered product teams understand these inherent issues, and product discovery is about *discovering a solution that our customers love, yet works for our business*.

We refer to this as product *discovery* to acknowledge that we understand what we can't know in advance, and to emphasize that our task is to *discover* a solution that is *valuable*, *usable*, *feasible*, and *viable*.