Most software companies fall short of perfection, and that’s just the way it goes. It’s routine for most software firms to slap beta on the product and ship.

Most companies aren’t Apple, however. In case you missed it (and if you did, where have you been?), Apple released imperfect mapping software for the latest iPhone. The Washington Monument ended up on the wrong side of the street. Bus routes didn’t appear in many major cities. Mere glitches for most firms, but for Apple it was embarrassing. The firm has not quite learned how to do beta.

Lost in the ensuing brouhaha was a fundamental economic question. To wit: is it best to own a newly invented and complementary piece of software that works with lots of other software? Even Siri doesn’t have an obvious answer to that question. Let’s consider it.

The basics

The conventional economics of this topic begins with a simple distinction, between complements in demand and complements in supply. This distinction provides a good start, albeit not everything.

Complements in supply arise when the costs of producing two goods are lower in a single organization than two. Sometimes this is called economies of scope.

Cable television and high-speed cable Internet service can illustrate. Cable firms were already in the TV delivery business, and already had the lines. They had to change the switching to support the Internet. Using an existing line saved expense. The cost of supplying these two goods would be much more expensive if they involved a redundant set of lines.

Notice I said “high-speed” service. Economies of scope might not exist at lower speeds, because lower-speed service also can viably come from many sources. Also, notice that the cost savings would exist whether or not demand existed for the second service. In other words, demand conditions don’t play a role in whether economies of scope exist.

Don’t get me wrong. Demand is not a trivial detail. One factor that slowed cable company entry into Internet in some areas, for example, was user reluctance to adopt the Internet from cable firms with poor maintenance reputations. In other words, demand conditions don’t play a role in whether economies of scope exist.

Modern markets

It used to be conventional wisdom that realizing economies of complements in manufacturing required owning the entire production process. It used to be thought that one firm had to coordinate the whole process.

Widespread outsourcing has put that notion to rest. Many firms will sacrifice
complements in supply and willingly incur the costs of an external supplier if it yields large gains from using an inexpensive assembly process.

That said, firms do pay attention to departures from the realizing complements in supply, often to protect trade secrets. For example, US firms often insist on the use of proprietary parts in their Asian assemblers, introducing extra steps and costs into assembly. Some designers will break up a complex design among multiple assemblers to prevent any single assembler from accessing a design, again, introducing extra steps and sacrificing some of the gains from complements in supply.

Strategic issues also can shape the sequences of development of complements when those are outsourced. For example, some firms will not release any information about a design until the last possible moment. That gives their own designs a jump on imitators in the market, and it also lets the designer remind everyone else who is in control.

Lest you doubt that firms act this way, Apple is notorious for such behavior. For example, Apple recently reminded everyone of this policy (yet again) when it changed the connector design, and made everyone wait for information about it.

Now consider a special case, with platforms that coordinate multiple participants’ behavior. Examples are Android and Apple in smartphones, Facebook in social networks, Google in search engines, or Microsoft in games. In these situations, the platform owner can’t supply all the complements, and must choose between internally supplied components and negotiations between the disparate interests of carriers, application makers, users, assemblers, and advertisers.

On the one hand, a successful platform encourages others to provide applications, and the leader permits exploration in areas that no leader could have planned. After all, it’s just not possible for any single firm to come up with the entire variety of applications—witness the hundreds of thousands of apps for Android and Apple platforms.

On the other hand, the platform leader could have a vision for new functionality in a platform and may need to demonstrate it with conviction by developing applications. Microsoft did that with Halo in the Xbox, for example, and that induced sales, which motivated many other firms to make their own apps.

Free features
Modern high-tech economies make this topic particularly confusing when a pervasive component works with insiders and many outsiders. Usually a platform provider finds it in its interest to price that pervasive complement at zero.

Sometimes it’s easy to understand why this happens. For example, Apple includes a low-quality pair of ear buds with every iPod, effectively pricing them at zero. Apple doesn’t want a teen to get home, rip open the box, and not be able to use the product right away due to lack of earphones.

These ear buds are a free low-quality default, which is good enough for most teenagers. Users who want better pay for it by buying better from someone else.

A trickier tradeoff arises with free defaults that involve some arm-twisting. Years ago, for example, Microsoft “advised” other application firms on the design for their pull-down menus, but didn’t leave them with much room to negotiate. The uniformity (arguably) benefited users by developing a coordinated design, and it reduced Microsoft’s support costs. Why was that a problem? It reduced design options primarily from non-Microsoft suppliers, which restricted competitive alternatives.

Google also finds itself in this territory today. It favors results using its own maps and other information it provides, arguing that this saves users’ time. It also can (reasonably) argue that complements in supply makes this practice inexpensive to users. That is, they don’t want to have to design and support multiple designs, so it’s cheaper if they do it all. Integration also (perhaps) lets them cheaply accumulate multiple new innovations. Unsurprisingly, however, it also ends up reducing the supply of information from non-Google providers, which looks like a restriction on competition.

As a digression, I would say this: I have never met a policy maker who likes these situations. Nobody wants to deter a firm from lowering its own costs, but who will protect the marketplace for new entrants if nobody intervenes?

Anyway, let’s circle back to Apple’s issue with maps, and reassess. Users were seemingly satisfied with Google’s maps on iPhones. News reports suggest that Apple’s management didn’t like giving Google so much information about its users. Apple’s management also thought there were more possibilities for developing new functionality that Google wouldn’t allow (for example, on the voice for turns).

Those reports make it sound like users experienced complements in demand no matter who owned the map, but Apple missed opportunities to realize economics of scope from not owning both complements. So, Apple tried to realize those gains by making its own map.

Apple only got in trouble because it made its own map the default, and users complained about it and about the difficulty of finding alternatives. Conspiracy theorists have claimed that Google deliberately didn’t make something ready, but if Apple behaved as it has in the past, then it is more likely that Apple simply didn’t give Google much information in advance. Google didn’t have something ready because it could not.

Should Apple have done this? Yes, if they can generate complements in supply without sacrificing complements in demand. Otherwise, no. And it looks to me like an imperfect map suggests they may have sacrificed some complements in demand—and if this remains unfixed for long, it will go down as an error.

Frankly, I also do know one additional thing: Apple ought to find a way to slap “beta” on the label on its maps.

continued from p. 64