**Interlude II. First Attempt at Transformation–Cloud Native “Side Project”**

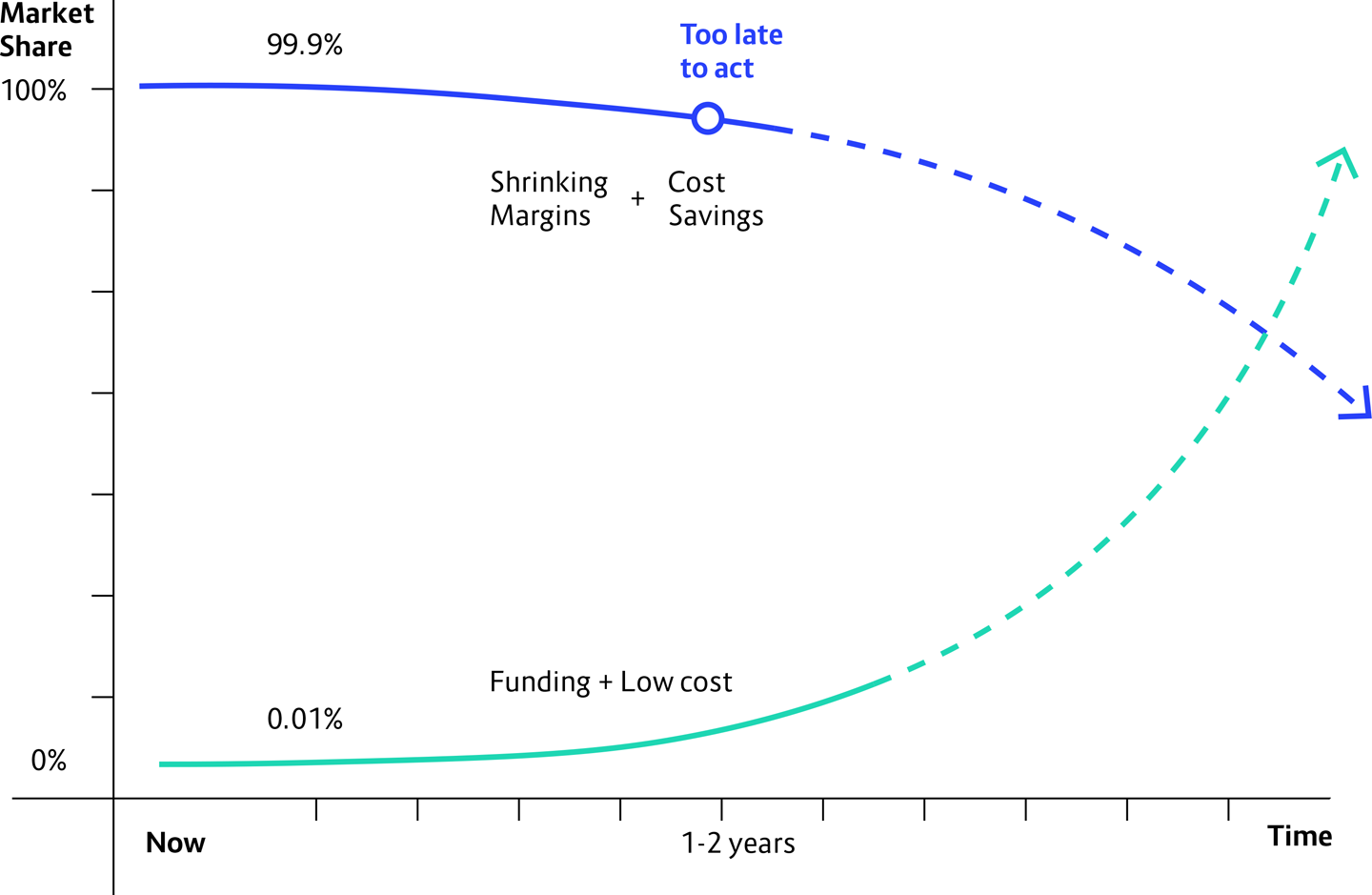
*This interlude includes a description of Jenny’s first attempt to lead WealthGrid through a cloud native transformation. One of the most frequent mistakes we see is a company trying to treat a cloud native transformation as just another simple technology shift. Instead of allocating the resources necessary to implement a major organization-wide evolution of both tech and culture, the initiative gets treated as just another minor upgrade. It’s assigned as a side project, or as just another system upgrade added to the backlog of standard tasks. Since getting started with cloud native can be deceptively easy, this may even seem to work well at first. But the reasons why this is a poor way to approach a transformation begin to reveal themselves as the project progresses. Things rapidly grow more complex, however, and more and more problems arise; meanwhile, the transformation team still has to keep working on existing system, too. Eventually the initiative will grind to a halt, though no one is quite sure why or what happened.*

When last we saw Jenny, our technical program manager, she had come to the realization that now is the time for WealthGrid to make the move to the cloud.

She had been experiencing pressure for just such a move from both above and below. The engineering team she oversees really wants to try out new tools and technology. Some of WealthGrid’s executive leadership, meanwhile, has been reading about how all modern businesses must go cloud native. Recently, upper management also realized that having cutting-edge cloud tech would help them attract and retain engineering talent.

Jenny agrees with the engineers *and* the C-suite. Furthermore, she had identified a third clear driving force that unites both sides: keeping the company’s bottom line strong and healthy. WealthGrid needs to move to the cloud, for sure, and doing this now is key to keeping the company competitive.

She is astute enough to recognize that not only is the imminent arrival of a disruptive new cloud-powered competitor a very real possibility, but that if you wait until they show up in your market and begin showing significant strength, it could very well be too late for you to catch up. Remember our exponential versus traditional growth graph? Take a look at [Figure II-1](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/afterword02.html#when_a_disruptive_new_competitor) as a reminder.



**Figure II-1. When a disruptive new competitor comes to your market, they may seem small and nonthreatening at first—but their progress will be fast and relentless**

Even if at first the stranger is just a small startup taking only a tiny bite of your market, pretty soon they are going to be eating your whole lunch! WealthGrid must take action before they even appear on the scene. And Jenny thinks she knows what to do.

Jenny has been to conferences, listened to talks, and read a lot of tech-industry articles. She understands that technology is the main factor in winning this competition. And that, furthermore, all of these technologies—public cloud, microservices, Kubernetes—are widely available to everyone. Many are even open source technologies, so they are available for free!

Really, this will just be a technical shift, she thinks. No need to put together a formal proposal and take it up the chain of command, because this falls into the category of pragmatic operations details that the C-suite avoids getting involved with. She can easily spare money out of her budget—have you seen how cheap it is to get started on the public cloud providers?!—and ask the engineers to work on implementing it alongside their usual tasks on the existing system.

This is how, without even going to her managers for approval, Jenny—together with the technical team she oversees—decides that they must go ahead and implement those things. Then they can show the entire company, once they have it ready. This should take a few months, no problem.

**How Hard Can This Be?**

The engineers are thrilled. Finally, a chance to roll up their sleeves and play with something really cool. They are eager to get started right away with their plan to put together a new platform for WealthGrid’s software development system using Kubernetes, Docker containers, and microservices. Although they are accustomed to working with older technologies, the engineering team is confident they can figure things out as they go along. Jenny is confident, too—she knows her team is smart and capable. How hard can this be?

Their enthusiasm is only slightly dimmed when they realize that of course their first priority is the existing system—keeping it up and running well while still delivering new functionality. This, after all, is WealthGrid’s core business, and where the company’s profits come from. There are projects underway for building out new features, and those also need to keep moving forward. Even so, building a new cloud system shouldn’t take too long, everyone agrees. They can work on it as a side project, when there is time, and still turn it around in a few months.

At first, things go pretty well. The team kicks off its cloud native transformation by setting up an account on a public cloud provider like Amazon Web Services or Google Cloud—all you need is a credit card, and you can be up and working almost instantly. If you want to run a single container, you can go online, click, and in 10 seconds you have a fully provisioned container running. It’s practically magic!

Or perhaps they decide to go with a private cloud on their own infrastructure, à la VMware, and set up a Kubernetes cluster on a couple of spare servers … or get a free version of one of the commercial cloud native integration platforms like RedHat’s OpenShift or Mesosphere DC/OS.

After that, setting up a small project is also pretty easy. The engineers play around with a single container, begin learning the ropes, and it goes really well. So then they try adding a small experimental app they whip together, or perhaps place an existing Java app exactly as is into a container and get that running on the cloud. Either way, at the beginning this initial trial project seems very easy—but quickly gets harder. A *lot* harder.

Dividing everything into related, containerized microservices makes intuitive sense, but no one at WealthGrid has ever worked with a distributed system before. The complexities inherent in even a single small and isolated project present a major learning curve, and there seem to be more and more things going wrong that they just don’t understand. Still, they are confident they can figure it all out, if they just find some time to focus.

Unfortunately, this dedicated time to focus on the new cloud platform simply never happens. Every time the engineering team starts to work on building out the new system, the work gets interrupted by a higher-priority feature request that comes from the market—from the customers out there using the existing WealthGrid products. The problem is, there is always pressure from customers. New or improved features are always winning the battle for assigning time to projects, and the new system always loses.

**A Year Later, with Little to Show for It**

So time passes, and after 12 months only a tiny portion of the new technologies is implemented. Even these are still strictly experimental. The engineers maybe have a few microservices running nicely on some Kubernetes clusters off on the side, but there is no significant impact on the current production environment. The results are intriguing—and also completely stalled. The team is constantly pulled away from any work on constructing the new platform by the need to continue rolling out features and support for the existing one.

This is not true failure, but there is no real success either. A year ago Jenny decided, along with her team, to go get for WealthGrid the same kind of responsive development and speed that their cloud native competitors enjoy. Their expectation was this would take maybe three to four months. Suddenly it’s a year later, and there is little to show.

Jenny’s first wakeup call was that WealthGrid needed to evolve into a cloud native company, and she responded using the resources at hand to start the transformation. Now she has a second wakeup call: what they’re doing isn’t working. It’s not leading them fast enough to the goal of successfully competing with the disruptive newcomers crowding into WealthGrid’s market.

Jenny needs to try a different approach. But what should she do?