**Chapter 6. Tools for Understanding and Using Cloud Native Patterns**

OK, so what just happened during WealthGrid’s second attempt at a cloud native transformation? Why is WealthGrid having such a difficult time building a new cloud native system, even when they are putting tons of money and people into the project?

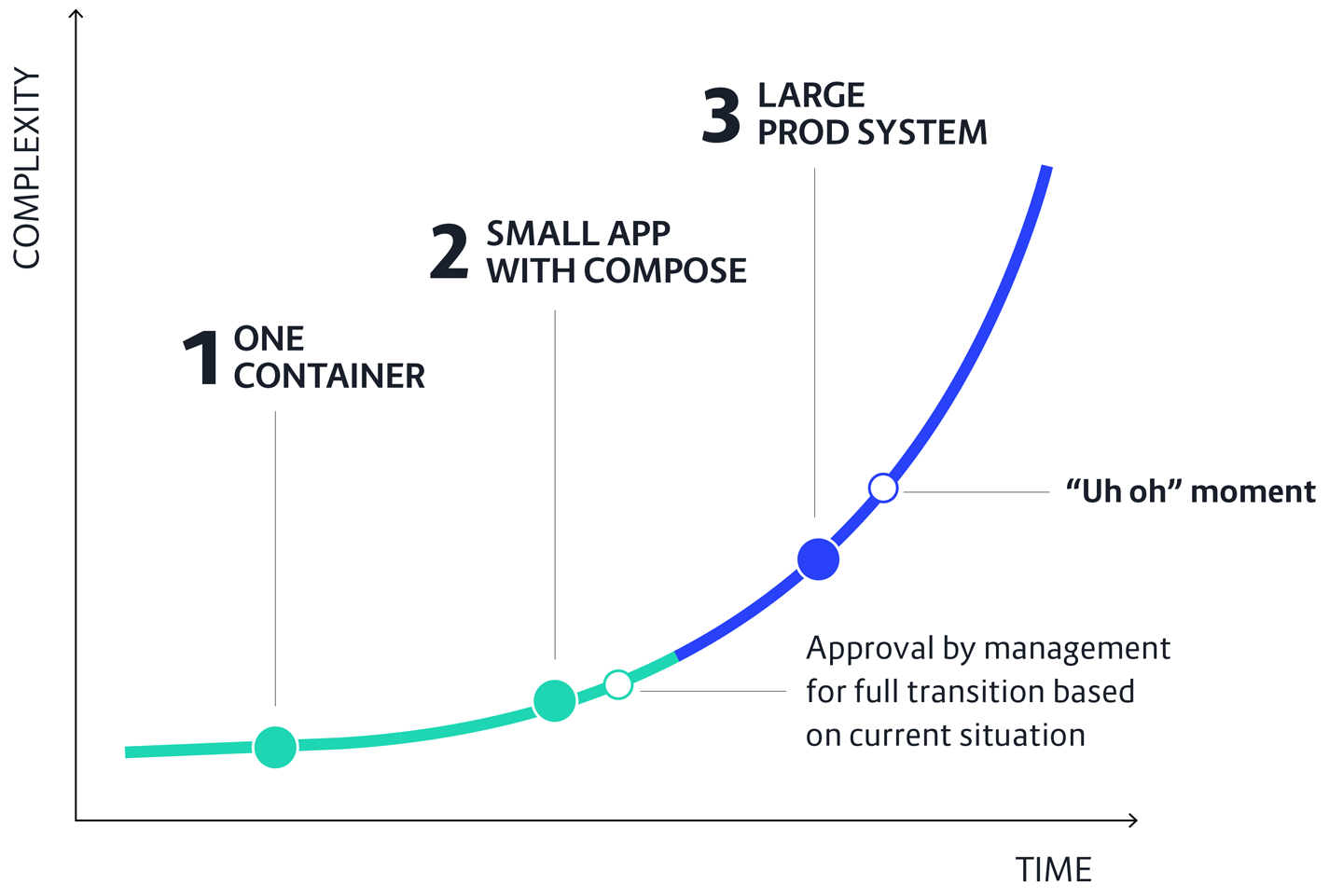
The short answer: cloud native is new, complex, and requires a very different way of thinking. Alas, this is apparent only once you have already moved a significant distance down the migration.

These eventual but inevitable difficulties take everyone by surprise because the first stages of a cloud native transformation are genuinely (and deceptively) easy. Engineers go to the conferences and they see Google Cloud Platform guru Kelsey Hightower or someone like that doing magic on the stage and producing results very quickly and efficiently. Then they go try it themselves by setting up one container, or maybe a simple application with a few containers running Java, and it goes great. It’s actually pretty simple and can be done in just a few hours and they feel like they’ve got this, no problem-o. This is the point where most engineers go to management and say, “We have to adopt this technology! We love it—it’s excellent and so easy we can do it without any big investment.” So they get the approval, and move happily forward, expecting things to continue on being just that easy and successful—while they continue working the same way they’ve always done.

This kind of bottom-up push to go cloud native, originating from the engineers, is one of the two ways most enterprises enter into a migration.

The other type of push is top-down, where an enterprise’s executives are hearing how hot cloud native tech is and decide that their company must do it too, even if they have no idea what it is all about. They are reading articles in professional publications about how amazing cloud native is and hearing from vendors how easy it will be if they just spend money on the right products. Due to this lack of understanding, the tendency in the top-down scenario is to grab onto one concept—DevOps, Kubernetes, etc.—and order the tech staff to adopt it.

In both scenarios there is no actual adoption strategy, and the decision to move ahead is made based on lack of understanding coupled with the misbelief that the process will be fairly simple and straightforward. The problem is that the complexity of this is dramatically different as you start moving to a large-scale production environment. At some point you realize, *Uh oh, this is not going at all like we thought it would.* [Figure 6-1](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch06.html#the_cloud_native_adoption_curve) shows the cloud native adoption curve and how it moves from simplicity to complexity—and where the “Uh oh” moment inevitably occurs.



**Figure 6-1. The cloud native adoption curve starts out fairly flat as transformation initiative starts out with early experiments with single containers and small test applications (using an external cloud database platform like Compose, to keep things simple). The curve gets steep fast, though, when it’s time to turn these small experiments into an actual production system**

This is exactly what happened at WealthGrid.

WealthGrid’s first attempt to adopt cloud native was a classic bottom-up approach. Due to lack of resources, though, the initiative didn’t get far enough for the team to reach the point where difficulties would begin to reveal themselves. Jenny’s engineers had had some success with building small pieces of the system, even if they had to start and stop a lot. Jenny’s natural assumption, and theirs too, is that this seems actually pretty easy; the only thing holding us back is lack of time to focus on it. Based on their experience they fully believe that, if they just have more help and no distractions, they could crank out a new cloud native platform in just a few months.

So when Jenny first approached the CEO and board asking for support to go all in, company-wide, on a cloud migration initiative, this is what she told them. Naturally they approved her request and allotted significant resources. After all, top-down pressure had also been building at WealthGrid. The engineers just acted first.

For the second attempt the whole company tried going all in, sidelining the current system to focus on building a brand-new one as fast as possible. That didn’t work either. In fact it went quite terribly wrong. What happened? And what else is there for them to do?!

**So Many (Deceptively Simple) Tools**

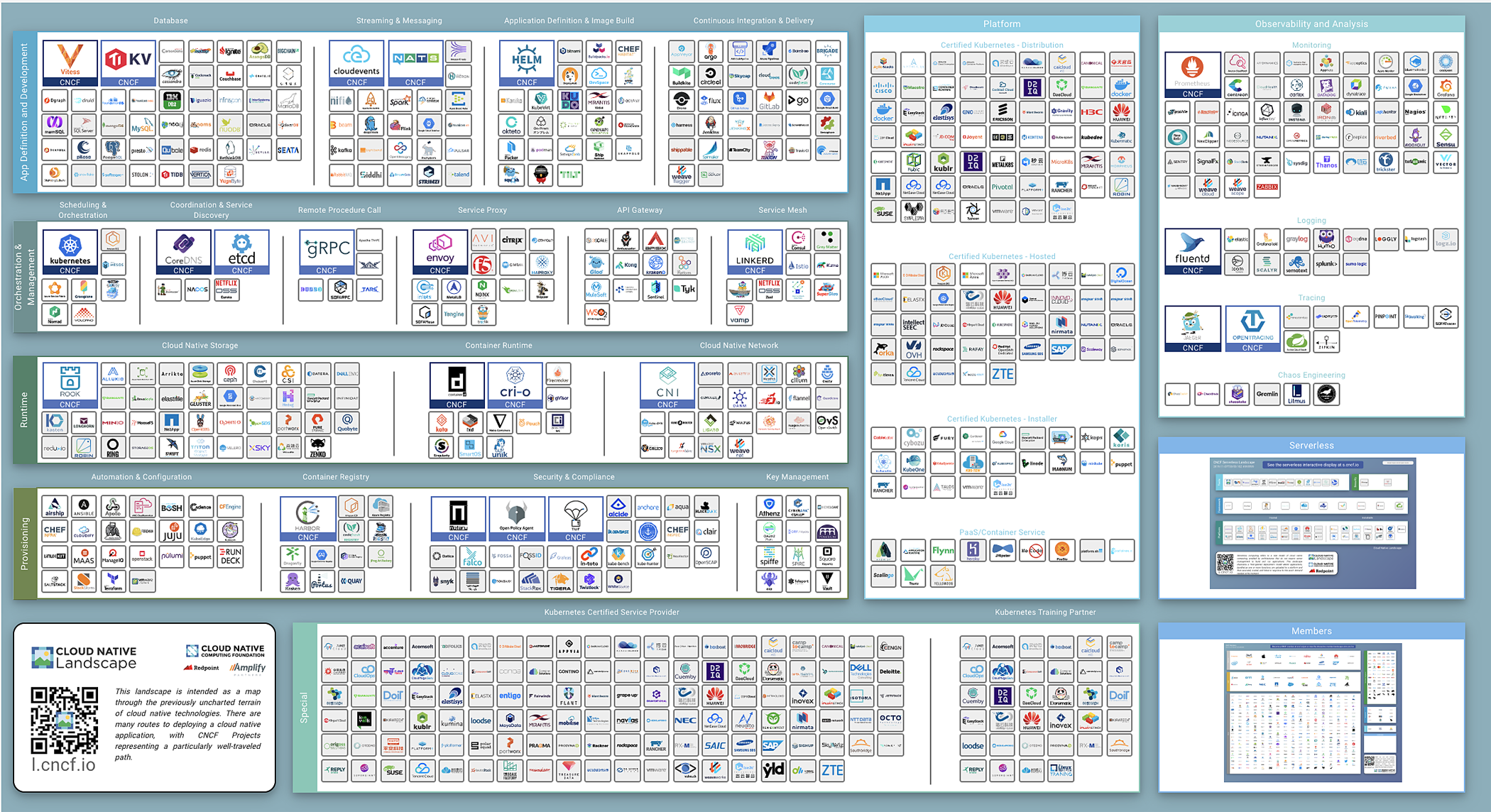
What happens? Why do things fall apart? The reason for this is best explained by the Cloud Native Computing Foundation’s landscape—there are more than 1,200 projects currently monitored by the CNCF representing the cloud native ecosystem and its landscape. And there are new projects added every week!

[Figure 6-2](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch06.html#the_cloud_native_computing) shows the landscape at the time this book is being written, with its plethora of options. You don’t have to become intimately familiar with all of them, but you do need to understand the basics of what they do, how they work, and how they fit together.

Gaining even the most basic grasp of this insanely large landscape means first understanding what the different tools do. Next, how all those tools fit together. Then, both how they integrate between themselves and how they integrate with older systems. Finally, you have to establish what kind of development and delivery processes you need to use in order to be successful using them all.

This requires a lot of new knowledge. It’s not simply installing some new tech, “lifting and shifting” to the cloud. Not just updating the tools you use, but revolutionizing *how* you go about using them.

Let’s introduce some tools. A few things that will help Jenny solve WealthGrid’s problem.



**Figure 6-2. The Cloud Native Computing Foundation’s cloud native Interactive Landscape (November 2019)**

**Tools to Take You the Rest of the Way**

Before we can dive completely into the patterns, we first need to establish some of this new knowledge. The two concepts introduced in this chapter will provide crucial background understanding for applying the pattern design that follows. We are going to examine the important difference between proficiency and creativity and why managing them correctly is key to succeeding in any kind of digital transformation… like the one WealthGrid is attempting. Next, we work with McKinsey’s Three Horizons model for visualizing how and when to apply creativity versus proficiency and, most important of all, how to balance them properly,

**Proficiency Versus Creativity**

WealthGrid is a highly proficient company. Things are optimized for delivering its core business mission consistently and well. This is a good thing, right? Stability, reliability, and quality drive profits and thus are valuable assets in any system.

Most traditional companies prize proficiency, the ability to complete well-defined and predictable tasks with maximum efficiency. This is how you deliver maximum value in a relatively stable context with few, if any, unknowns or surprises. In this context, creativity and innovation are viewed with skepticism. Innovation introduces unknowns into a highly regimented system—and with unknowns come risk.

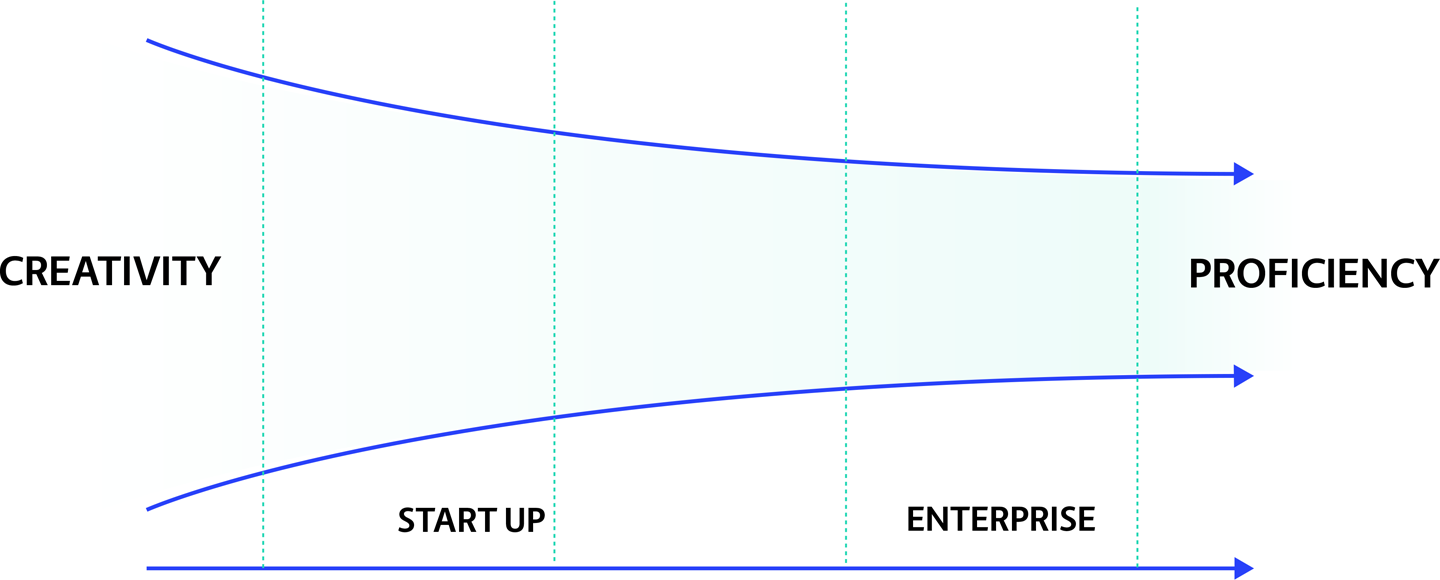
However, as we have seen, companies in nearly every imaginable sector no longer have the luxury of a stable, predictable environment. Things are changing fast in every way, from new tech to new competitors. This calls for being able to respond—being able to change—to meet these new challenges whenever they arise. It means switching out of highly efficient, algorithmic delivery (proficiency) temporarily in order to change what needs changing—your product, your way of working, your company itself—through innovation and creativity. Once you’ve made the changes, then focus returns to proficient delivery of your improved products/service…until the next time a new challenge calls for a creative response.

So, yes, proficiency is a good thing…up until the moment that it isn’t and you need to adapt to new circumstances. Most struggle to adjust, however, because they are trapped in proficiency mode. Again, many successful companies got that way through focusing on the bottom line: proficient delivery of their core product/service. They are really good at what they do, but in the process of becoming exactly that good at exactly that thing, they forgot how to work any other way.

Fortunately, once a company recognizes this reality it’s possible to take corrective moves toward creativity. But which moves, and how?

**Want Fries with That?**

A useful tool for conceptualizing the flow between creativity and proficiency, which are essentially two sides of the same coin, is called the Knowledge Funnel, shown in [Figure 6-3](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch06.html#roger_martin_apostrophe_s_knowledge_fun).



**Figure 6-3. Roger Martin’s Knowledge Funnel concept as applied to the process of moving from a highly creative startup to highly proficient enterprise delivery of products/services**

The Knowledge Funnel was introduced by Roger Martin in his book *The Design of Business: Why Design Thinking is the Next Competitive Advantage.* According to Martin, every new idea goes through three major stages as it progresses to full adoption: Mystery, Heuristics, and Algorithmic. We are going to look at these stages in terms of McDonald’s fast-food restaurants, because it is such an excellent example of pure proficiency at work.

*Mystery* comes first: you have a new idea and you think it’s good, but you have no real understanding of how (or even if) it will work or how to implement it.

Imagine the very first McDonald’s hamburger restaurant, founded in 1948 in San Bernardino, California, by brothers Richard and Maurice McDonald. It was a business serving food, but with one brilliant twist: you could walk up to the counter and get food right away instead of needing to sit down at a table and wait for service. No one knew at that point if this was a good idea or a bad idea; it was simply a new idea.

*Heuristic* follows the mystery stage. A heuristic is any approach to discovering something new or solving a problem that applies a practical method to reach the immediate next goals. A heuristic is not guaranteed to be optimal or perfect, only sufficient for getting things working well enough for the time being.

This new idea was apparently one whose time had come, because the first McDonald’s restaurant was instantly successful. The McDonald brothers naturally thought to expand, opening new locations until they had four total. Things were still going very very well—Americans were eagerly embracing the concept of fast food—and they wanted to keep expanding. Starting with restaurant number five, however, they ran into a problem. The quality of their hamburgers was not scaling along with the growth; it was going down, actually, because they couldn’t keep things consistent. The McDonald brothers realized they had to keep production fast and quality high, but they didn’t know how to scale.

The *Algorithmic* stage is the next stop on the Knowledge Funnel, representing fully optimized and streamlined proficiency.

Ray Kroc was a restaurant-supply salesman who was intrigued by the fact that the first McDonald’s restaurant had ordered eight milkshake mixers. Most restaurants had one, at most two, and Kroc was curious to see what this new place was doing with so many more than the usual hamburger joint required. When he visited the place, he was instantly intrigued by their innovative approach. He was so impressed, in fact, that he persuaded the founders to let him try turning their concept into a nationwide chain of restaurants all based on a single, easily replicable model.

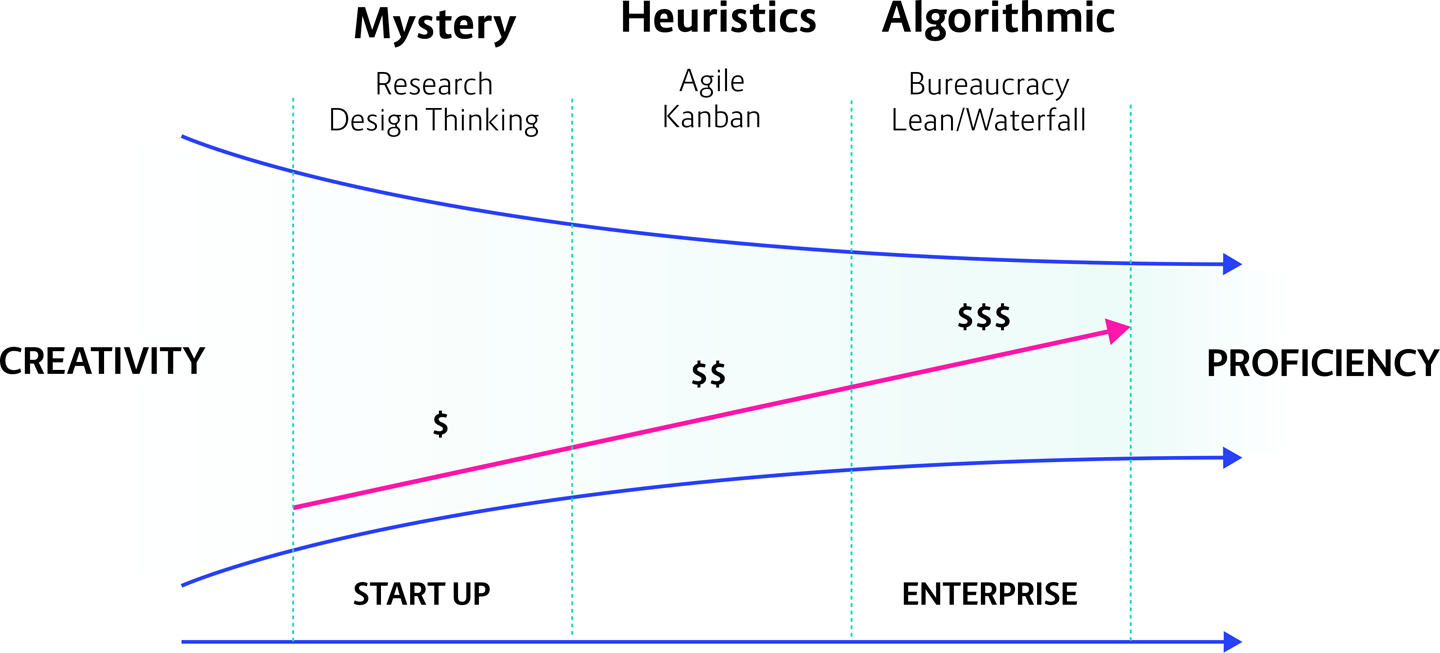
Kroc eventually bought the brand from the McDonald brothers and made McDonald’s what it is today: essentially, a fully algorithmic operation. When a new franchise opens, it is done literally by the book: each franchisee is given a guide in which every step in running a McDonald’s restaurant is well defined, extremely specific, and very clear. Every decision has already been made in order to optimize quality, consistency (a Big Mac in Birmingham, Alabama, tastes exactly like a Big Mac in Buenos Aires, Barcelona, or Beijing), and, above all, speed.

In short, the McDonald brothers came up with a truly innovative idea, but Kroc was the one who recognized what it could become—if proficiency were applied.

**Creativity, Proficiency, and Enterprise**

Unfortunately, when businesses evolve from a scrappy new startup to become a proficient and as-algorithmic-as-possible operation, they often forget how to be a startup—that is, how to re-enter the mystery state over and over in order to research and introduce new ideas to their business. This would be like McDonald’s sticking to their original menu of regular burgers, fries, and shakes, never to invent their famous triple-patty Big Mac, much less Chicken McNuggets. McDonald’s again is an excellent metaphor for our exploration of proficiency versus creativity, because the company has historically continued to introduce innovative new menu items (produced in, of course, a fully proficient fashion).

How do these concepts apply beyond fast food to the world of businesses and software, though? [Figure 6-4](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch06.html#the_flow_between_creativity) helps us visualize this.



**Figure 6-4. The flow between creativity and proficiency through the different stages of a company’s journey from startup to enterprise, and from mystery to algorithmic delivery**

In [Figure 6-4](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch06.html#the_flow_between_creativity) we still see the Mystery, Heuristics, and Algorithmic stages. Let’s look at how they apply to typical enterprises.

**Mystery:** Startups by nature inhabit the creative phase—when building a new thing, by definition you don’t know if it is going to succeed or not. The beginning phase, for a startup, is almost purely creative: the time when you have an idea and need to develop it. There are many, many ideas in the world, which is why there are so many startups. Most of these ideas are probably not very good, which is why so many startups fail.

**Heuristics:** Once your brilliant new idea has found success in your market, it’s time to move to this next phase. In order to begin building up the company, to grow and find success in your market, you must become more proficient. This means turning focus from figuring out how to get your idea working at all (an inherently creative process) to how to get it working efficiently.

**Algorithmic:** You improve the idea until it becomes a viable business, and you can deliver reliably and steadily. Congrats, you have now reached full enterprise status!

Just as each of the three stages of business development has specific characteristics, how an organization functions internally while moving through each of them is also very different. There are whole books dedicated to exploring the mysteries of group behavior at different points along the path from crazy idea to fully established and delivered business value. Two of these in particular helped us crystallize our thinking around creativity, proficiency, and how to address and balance both through cloud native transformation patterns.

*Loonshots: How to Nurture the Crazy Ideas That Win Wars, Cure Diseases, and Transform Industries,* by Sahfi Bahcall, is a compelling look at how companies, individual teams, or really any group having a common mission will suddenly change from a culture of embracing wild new ideas (while in Mystery mode) to rigidly rejecting them (once Algorithmic stage is reached), just as flowing water can transform into inflexible ice. Bahcall draws upon the scientific field of phase transition to illustrate how group behavior is influenced by the structure of its containing organization.

Likewise, Daniel Coyle’s *The Culture Code: The Secrets of Highly Successful Groups* examines how diverse groups end up internalizing the same homogenous mindset and how this can lead to the stifling of innovation within an organization.

What do these look like over the life cycle of a typical enterprise? In the beginning you want to use research, design thinking, all kinds of creative ways of doing things, because this time is all about exploration. As a successful direction emerges, you begin to define your product or service and your process for delivering it. However, you are still able to pivot quickly and change direction because there is still some flexibility and creativity in your processes. If things turn out not to be working, it’s still possible to change course.

At some point, though, when you have honed things down to delivering a core value proposition, you reach the algorithmic stage. The inevitable outcome of this is building in bureaucracy to keep things running smoothly and consistently. This happens because it is only by reaching the algorithmic stage and full proficiency that you can generate significant revenue. Simply by being successful, you undergo a process of organizational evolution. Early on, you are a handful of crazy people trying all kinds of wild things, and the point is not to make money (well, not yet, anyway) but grow your idea. Once you prove your idea works, though, you want to grow the business.

In such cases, companies tend to become Waterfall organizations with a strong hierarchy, or perhaps a Lean one like the Toyota Production System. The Lean production model focuses only on that which adds value, mainly through reducing everything else (because it is not adding value). Unfortunately, this almost always means jettisoning creativity, which has no functionality in a proficient system because you no longer have an idea you need to explore. Now you have a successful product and the sole need becomes only to keep delivering that product as cheaply as possible. Perhaps that product improves over time, but the focus is now all on proficiency and no longer on evolution.

This does not mean proficiency is inherently bad or creativity is automatically good. An enterprise needs both, in proper balance—a balance that can shift according to need. There always needs to be at least a small portion of effort going to improving and trying new things. Because if you are only proficient and some kind of disruption or other significant change happens unexpectedly—one that requires you to shift to a new or different way of doing business—you have no way to respond.

When an existing business gets challenged by the stranger in their town, they swing from proficiency to overvaluing creativity, because they suddenly realize they don’t have enough of it. Startups, meanwhile, tend to overvalue proficiency, which is what *they* don’t have since they are often seat-of-the-pants organizations with few resources. They aspire to the stability that proficiency represents, but this leads to a different problem. Many startups jump too soon in trying to introduce quality and so try to establish an algorithmic delivery model too early, locking themselves in and diverting resources when they still need to be focused on fully developing their idea. Thus, they are trying to leave creativity prematurely in favor of proficient delivery of a product that is not yet fully baked.

So, yes, you need both proficiency *and* creativity. One is not better than the other, or more important than the other—it’s the balance that is important. You need both, but not at the same time. This is because they both need to be managed differently. You cannot have teams that are working both proficiently and creatively at the same time, since these are conflicting mandates:

* Proficient teams require high repetition to deliver the same thing, over and over, very efficiently and reliably, and at the highest quality possible. High repetition, high feedback, small set of very specific rules. The emphasis is on skills and repetition.
* Creative teams, on the other hand, have no specific list of tasks. Their work requires open-ended thinking that is more like puzzle solving. This doesn’t mean that creativity equals chaos: there is still a guiding purpose behind it, and tools to use. To effectively nurture innovation there must be a goal and the strong support and safety of a space that allows open-ended experimentation. Autonomy is crucial: once the goal is established, let the team find solutions in whatever way they can discover.
* Both types of teams are just teams, composed however your organization’s team structure works. It’s their jobs that are different: the proficient teams are your bottom-line delivery workers, the creative teams are focused on research and next steps. Typically you’re going to have more teams tasked with delivery than innovation/creativity.

Trying to have one team work using both proficient and creative approaches at the same time is simply futile. They are two different mandates, and they conflict with each other. You are asking a single team to focus on optimizing repetitive ongoing delivery processes while simultaneously innovating on them, which means neither area is going to get full attention or best effort, and both are going to suffer.

Thus an organization needs both kinds of teams, and they need to be distinct and separate.

They also, of course, need to work together. The proficient teams focused on bottom-line delivery of product need to tell the creative teams—who are in charge of keeping the company looking forward and engaged with innovation—what the actual problems are in the market that the company needs to solve. The creative teams, meanwhile, need to return something from their development pipelines that the proficient teams can use in real life that is useful to customers.

Ideally this is a dynamic relationship—a well-adapted organization can move as needed between times of greater proficiency versus times of increased creativity.

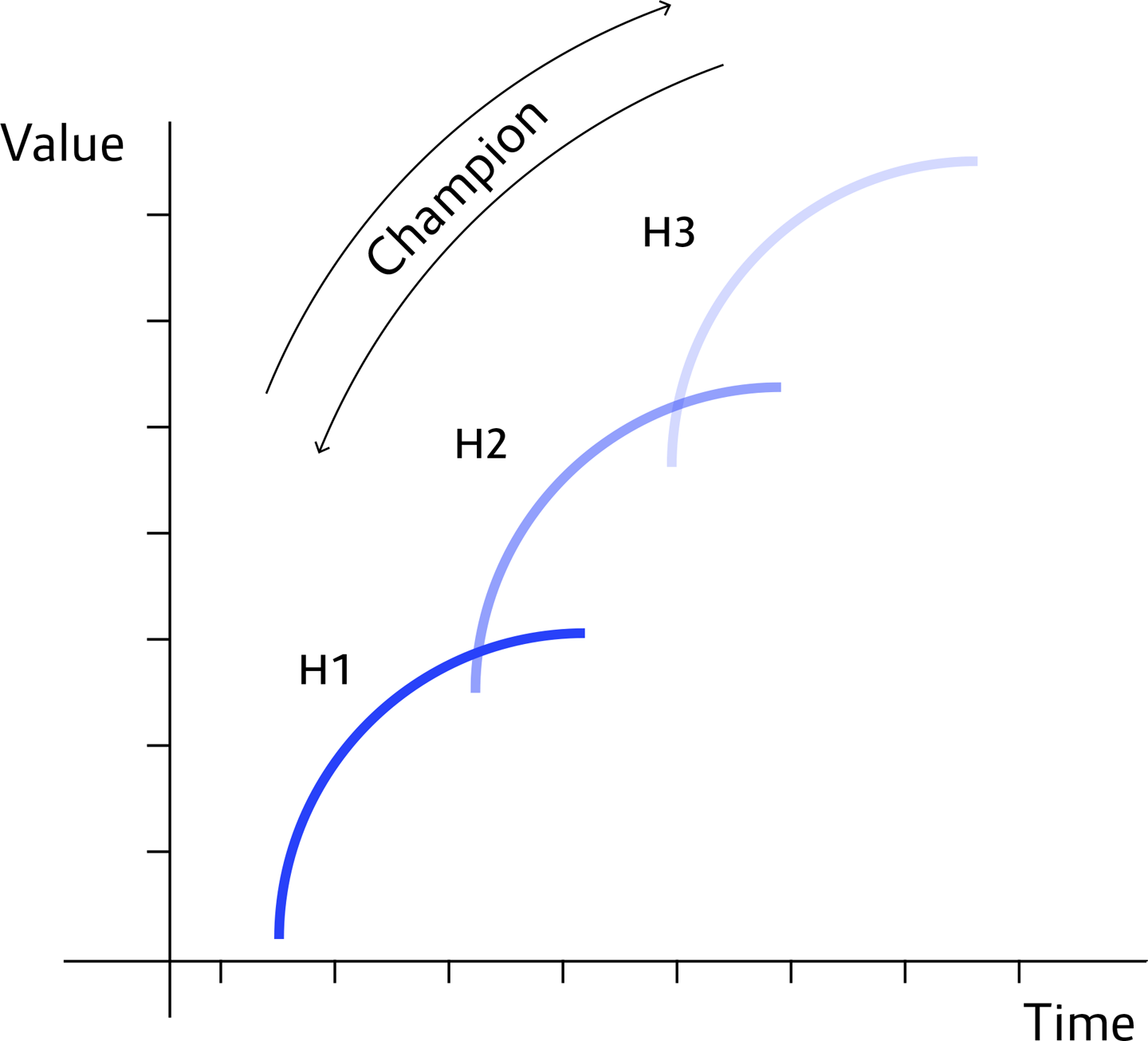
It is difficult to achieve this, but possible. Striking the balance requires maintaining the separation of proficient and creative teams while closely coordinating between them. Different styles of management are required for each, and a designated champion (or two, or even more) is needed to act as a kind of translator to manage whenever there are handoffs between their respective efforts.

The way to integrate proficiency and creativity, organizationally speaking, uses our next tool: McKinsey’s Three Horizons model.

**Three Horizons**

Once you understand the differences between creativity and proficiency, and the relationship between them, we use the Three Horizons model to understand how to blend and balance investment in developing new products within a company, while still delivering efficiently and reliably (and, of course, profitably).

The Three Horizons is an incredibly useful tool for identifying your business’s current state and then strategizing adaptive actions—either to take now or to hold for when they might be needed in the future. In order to use it most accurately, we break creativity into two different categories: innovation and research (more about those below). Let’s jump right in: [Figure 6-5](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch06.html#mckinsey_apostrophe_s_three_horizons) shows the three stages.



**Figure 6-5. McKinsey’s Three Horizons Model showing the relationships between delivery, innovation, and research**

* **H1:** Horizon 1 represents your current core business—presumably, the things that provide your cash flow and main profits. This also includes logical next-step development of/iteration upon any product or service you are making right now.
* **H2:** Horizon 2 is investment in innovation: taking ideas that have been shown to work in the H3 incubator and productizing them for real customer use. This means both introducing new technologies and new ideas into existing products or changing your existing products themselves to something entirely new. At this point you are working with a promising idea that looks like it is going to grow to become a successful part of your bottom-line business. (Interestingly, cloud native itself is now still mostly at H2: it’s working at many organizations but is far from easy or even completely stable and requires special knowledge to implement.)
* **H3:** Horizon 3 is research. Pure exploration of new ideas, research projects, pilot programs. Nothing that you can really use right now—here is where you are looking at the long term, things you may need to know or use a few years down the line. H3 is about awareness: staying abreast of what is coming next so you at least have some understanding when it does get here. Some of what you discover in H3 will get moved to H2 for further investment and development. Not everything you investigate in H3 will pan out or end up being useful, but that’s OK. Sometimes experiments just don’t work.
* **Time:** The x-axis does not represent a linear progression of time; it is not telling you when to apply each stage—now, later, or far in the future. *Companies must work within all three horizons at the same time.* In this model, “time” demonstrates how an enterprise moves, throughout the course of a normal venture’s life cycle, between the three interrelated cycles.
* **Value:** The y-axis represents the growth in value that organically occurs when an organization addresses all three horizons simultaneously and, as we shall see, in proper balance.

Companies start with some crazy big idea and founders who create a company around it. If you find success, you have to hire more people, which means starting to create specific heuristics so everyone can work together. If you are really successful, at some inevitable point you form a hierarchy, with set positions, routines, processes—it’s a one-way ratchet. Nobody wants to go back to the zero-process chaos of an early startup, no matter how creative a time it was.

What this means, though, is that most enterprises in the world use economy of scale to optimize both operations and profits; naturally, then, it becomes where all their efforts focus. Only a very few companies keep creativity on the side and keep it feeding in.

An adaptive business constantly evaluates and recalibrates the relationship between its three horizons, pursuing the optimal balance between proficiency and creativity. To manage this across an organization, it’s important to have people called “champions,” who understand those different horizons and move the technology across those three horizons.

**We Are the Champions?**

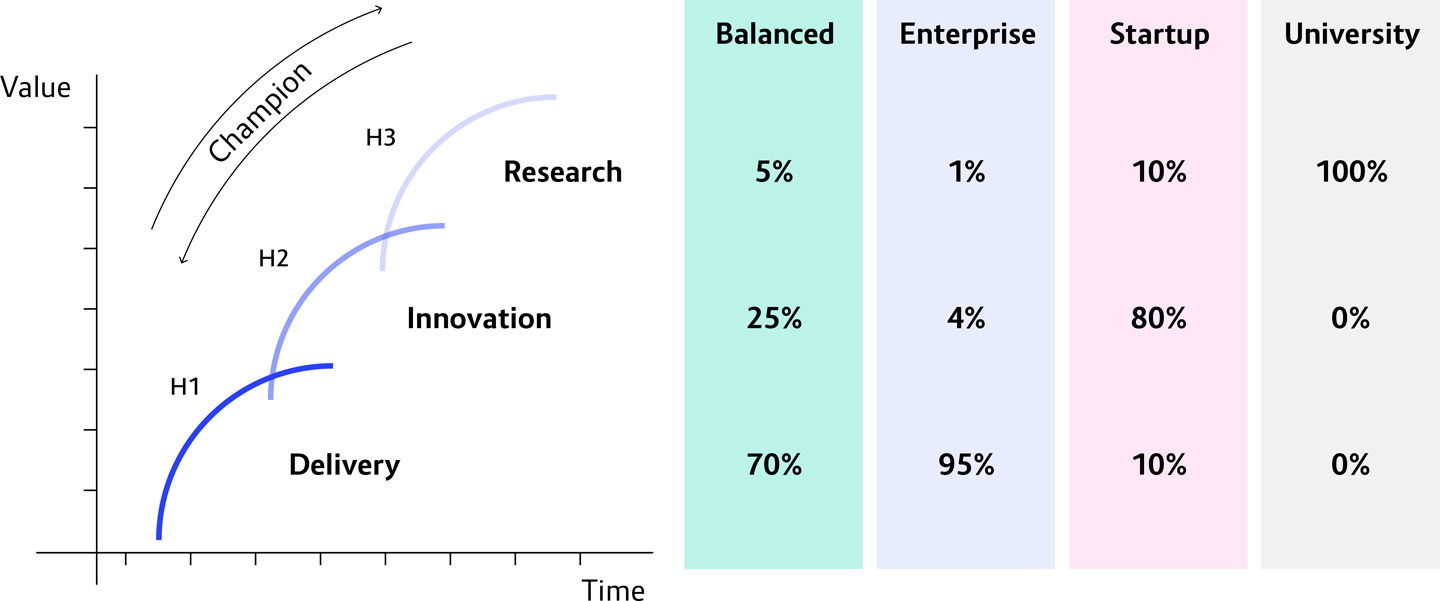
The champion is the person who keeps a firm fix on the bottom line while also pushing the likely next step—and keeping an eye on whatever crazy future thing could be coming next. It’s a lot to track, and so it’s good to have someone officially in charge of doing just that.

Champions are well aware that for a good company in a stable market, a good general ratio between the three horizons means putting most of the company’s efforts into delivering the core business. A small, but still significant, portion gets directed into practical innovation: basically, building the logical, market-demanded next step in terms of feature or functionality. Finally, a little bit gets left open for research.

Of course, this ratio represents a “business as usual” situation. It should change whenever the business environment shifts, in direct response to the type of change—like when a disruptive stranger suddenly shows up in your market.

Companies who lack a champion (or, more to the point, aren’t even aware of what a champion does and why they should have one) often skew the balance. This happens unintentionally, because they are fully focused on the business at hand and forgetting to build in some innovation to keep their creative juices flowing.

[Figure 6-6](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch06.html#recommended_ratios_of_the_three) puts some numbers behind these varying scenarios and the proper ratio for each of the horizons, in different circumstances. Note that these are approximate numbers to demonstrate the relationship between proficiency, innovation, and research, not hard-and-fast definitions that you must hit exactly (or else). They are based on our own experience with what we have observed works well in companies that achieved not just a successful cloud native transformation but also a new ability to respond to changing circumstances as they arrive.



**Figure 6-6. Recommended ratios of the Three Horizons in different business environments**

* **Balanced:** Normally you would want to invest 70%–80% of effort into delivery, invest 15%–25% into innovation, and maintain a small budget for research (5%).
* **Enterprise/Full Proficiency Focus:** Many enterprises, though—the ones we mentioned earlier, who forget how to be creative in the pursuit of pure proficiency—look more like 95/4/1. To be honest, this is even more realistically 95/5/0 at most traditional Waterfall companies. Too many focus solely on delivery, caring only about the products they are selling right now.
* **Startups/Mostly Innovation Focus:** By nature, startups are an innovative type of company. They are in the process of building a proper product, so they mostly live in the middle. They may dabble a bit in H3 (10%), and of course they are working their way as fast as they can to inhabit H1, so there is some groundwork already laid there (10%). But H2 is their home turf (80%).
* **Research:** Universities are pretty much the only entities you will find spending 100% of their time and effort on research. These are the only places that can do pure exploration without needing to worry about monetization in the near future.

Dedicating 5% of a company’s resources to research may not sound like a lot, but it’s crucial. That 5% is where you are gaining knowledge and retaining your ability to be creative when necessary. Where this goes wrong, where companies end up in trouble, is when they are not thinking—there is no strategy—and they try to move straight from research to delivery.

For example, in a cloud native transition, a company’s engineers might try to adopt a microservices architecture when they know nothing about it, have no background, and so end up approaching it all wrong. They researched enough about microservices to recognize that this is the right thing to do, but they are rushing to get things working, which means they skip the middle stage and try to squeeze it directly into delivery. Skipping H2, which is the pragmatic development phase for creating heuristics around delivering new ideas, might sound like a way to speed things up. But instead it’s a recipe for failure, since they haven’t taken time to understand how it works and fits together. An innovation champion’s job is to prevent just this sort of short-sighted corner cutting.

Now let’s look at the flip side. Sometimes, proficiency is, for the most part, what a company needs, with side initiatives into innovation and pure research. And these can even go to zero, temporarily, if needed. An example would be when quality is suffering in an enterprise’s main offering, and focus needs to be fully on delivery for a while. But this is a temporary pause, and in a healthy company (especially one with a champion on the job) once the crisis passes, the ratio gets rebalanced.

There is another scenario, however, where many companies do exactly this—pull all resources out of innovation and research in order to focus on the bottom line—and it is absolutely the wrong move. This would be when a company is under serious pressure from a new competitor, or a previous competitor that has introduced something new that is disrupting the whole market. Suddenly the established company is rapidly losing market share, and its leaders panic. The most likely path for a traditional company in this case would be to start cutting costs, especially in development. In times of distress, the R&D department is the first to suffer. But by doing this, the company is basically guaranteeing that it will not ever be able to adapt to the new market conditions and emerge as an equally strong contender.

**So What the Heck Happened at WealthGrid?**

*How do we use these tools to understand what went wrong*—*and make it go right?*

When we first met WealthGrid, it was a classic Waterfall-process company focused on the bottom line: delivering its core business value as efficiently and profitably as possible. If WealthGrid had an innovation champion, which they most assuredly did not, this person would have analyzed the Three Horizon ratio as 95% of efforts invested in proficiency, 5% in innovation, and 0% in research.

It was in this 95/5/0 context that Jenny and her team first attempted to move WealthGrid onto the cloud. As we have seen, this did not work. At the time, they chalked it up to not having enough time to focus on the migration because the existing system always took priority. We now have new tools to gain a deeper understanding into the broader forces that caused this to happen.

Essentially, WealthGrid’s first attempt failed because they tried to implement cloud native—which requires a creative and innovative approach—from a proficiency mindset.

Jenny and the engineers put cloud native technologies and tools into the Scrum backlog along with all their other tasks for running the existing proficient system. Then they tried to execute it using the same type of processes they’ve always used, using the same sort of management by deadlines, sprints, and stress. Many companies use sprints as a development framework; Scrum usually sets a goal with two weeks to deliver it. The problem is that when you are running as fast as you can to deliver, it’s hard to look around you with a mind open to innovation: it is impossible to be predictively creative.

So this initial attempt did not work. The dirty secret here is that, even had Jenny’s team been able to work 100% only on building the new system, they still would have failed. And for the same reasons, though perhaps from a different direction, that attempt number two also then proceeded to fail.

WealthGrid’s second attempt to go cloud native was basically the full reverse: *Now we will go into complete creativity!* The company moved a majority of resources—in this case, people—to the transformation project. That is, they redeployed most of their engineers from working on the original proficient system to building the new one. Dedicating resources to a transformation is a great first step. Unfortunately WealthGrid then took multiple wrong steps, all of which added up to the same thing: still attempting to deliver this brand-new technology and way of working by using proficiency-centered processes and culture.

There was no adoption of new ways of thinking or doing things, no application of a design thinking process to identify new ways to do creative problem solving, no purpose setting. The second attempt to build a cloud native system still used a Scrum and/or Waterfall approach to building. Just about every engineer in the company may have been playing around with different tools and cloud providers, but they were all using the same old knowledge and the same old way they have been doing successfully (and proficiently) for so long. This is why things went wrong for WealthGrid—*both* times.

Cloud native is a new way of doing things. It is not predictable, at least not to people who lack a good understanding of how it works. Most of the people at WealthGrid did not have this knowledge. Probably no one, not even Jenny, truly understood the full intricacies of cloud native architecture; certainly, no one at WealthGrid had any experience actually building a cloud native system. Lacking this understanding and experience, they of course used what they knew, the tools and techniques at hand. They didn’t know what they didn’t know.

The initiative may have moved forward pretty well at first, but inevitably the complexities multiplied as the system grew in size. Things slowed down more and more until the entire project simply could no longer progress at all. Meanwhile, the old system—the one WealthGrid is still running on!—had been languishing, with no new features or functionalities being delivered to the customer for over a year.

This was the third crisis point. WealthGrid was still committed to becoming a fully cloud native company. But it also needed a way to continue delivering value to customers while it worked to find the right path—the middle path between proficiency and creativity—to finally deliver the long-delayed new platform.

**Summary**

Digital transformation, ultimately, requires a balance between innovation and pragmatism, between creativity and proficiency. Some companies attempt to innovate but do so by trying to deliver creativity using proficient processes—that is, long-held practices and beliefs that worked well for them historically but don’t work with cloud native architecture. This leads to failure, or at best a low-functioning, improperly implemented attempt. There may be a few microservices running on a few Kubernetes clusters, but no real value is being delivered—especially in comparison with how much time and money has just been invested.

Others go all in on innovation, attempt to abandon the old system completely to build a new one from scratch, and still get lost. Many times these companies are trying to be like Google, one of the most creative (not to mention fully cloud native) organizations around. The common misbelief is that being like Google means being all in on creativity—let’s say something like 98% creative. Google’s real focus, however, is very much on proficiently delivering their existing products and services while investing very intentionally in small but targeted and highly impactful creative initiatives. Putting in the terms we have been using in this chapter, Google’s balance would actually be closer to 2% creativity and 98% delivery. The point is, they do have a balance, and it is what works for them. The real problem for most established and successful companies is that they have no idea how to be creative at all, in an effective way, at any number.

Proficiency is important. Creativity is important. Neither is better, and both are necessary. Proficient teams need to be managed in a way that supports their focused, stable and efficient delivery of bottom-line core business value for the company. Creative teams are managed for open-ended exploration of next steps, so the company stays innovative and ready to take responsive and adaptive next steps whenever needed.

It’s not surprising at all that WealthGrid attempted to transform itself into a creative cloud native company using proficiency-focused Waterfall/Agile processes. This is an extremely common thing, and we have seen it many times, in enterprises of all sizes and from all sectors. Certainly, if they knew better, they would do things a different way. The right way.

Now let’s see what the right way looks like.