**Chapter 4. Beyond Patterns: Behavior, Biases, and Managing Evolution**

*We humans, sitting as we do atop the evolutionary apex, like to think we are logical and rational—not animals driven by impulse and instinct. In fact, though, we are prone to all kinds of proven biases, cognitive quirks, and mental shortcuts. These cognitive biases presumably served our hunter–gatherer ancestors well by allowing fast decision making: when facing down a predator, speed was more valuable than accuracy.*

*Many of these biases continue to serve us well. If followed by a stranger when we walk alone at night, our fight-or-flight instinct is more valuable than engaging that stranger in a discussion about the unequal distribution of wealth in our society. Other times, however, these very same biases can cause us to think and act quite irrationally, to our own detriment.*

*In cloud native, cognitive biases can lead us to try to build a cutting-edge, distributed microservice system by following traditional (comfortable) approaches we’ve used to build software for the past two decades. In this chapter we take a look at cognitive biases and discuss the ones most likely to impact a cloud transformation—and how some biases can be harnessed to work for, instead of against, success.*

Imagine a commercial construction company in your town, one that is well established and successful, and known for producing solid, high-quality apartment buildings. Now imagine that the company’s leaders decide to broaden the business mission: they start building bridges. Halfway through the very first bridge project, however, it becomes obvious that something is not right. This bridge has doors, chimneys, and even decorative bay windows*—*all things that the company has always included, to great acclaim, in previous projects. Of course, chimneys and windows have zero utility in a bridgespan. These features may even render the new construction less stable and secure. But this is how the company has always built things, so it continued doing just that in their newest project.

Sounds ridiculous, right? A mistake that no one would ever make. A thing that could never happen in the real world.

Unfortunately, it *does* happen. The cloud native equivalent occurs distressingly often when companies attempt to transition to the cloud. We have observed over and over that, ironically, these migrations can contain a great deal of internal resistance to the very changes being sought. As a result, companies rebuild their operations and infrastructure in the cloud but then start trying to build cloud native applications exactly the same way they’ve built software for the past 10 or even 20 years. It’s the cloud native equivalent of building a nice fieldstone fireplace right in the center lane of that new highway bridge.

Implementing the complex technology portion of a cloud native transformation can actually be the easiest part of the overall project. A successful migration depends on more than simply embracing, say, microservices and Kubernetes. As we talked about in [Chapter 2](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch02.html#the_human_challenge_of_cloud_native), the organization itself must also change significantly in terms of culture and psychology in order to make effective use of its shiny new cloud native tech. These human-centered changes are the areas we most often see causing the greatest problems for a company undertaking a cloud native migration. But why is this?

**Conway’s Law**

The answer is simple but also difficult: human-centered changes are hard because, well, we are human. We like doing the things we are good at, in the familiar ways we are comfortable doing them. These biases toward sticking with the easy and familiar are not inherently bad. In fact, they exist for a reason: most of the time they work pretty well to save effort and energy. We just need to be careful to catch those times when they don’t.

In the same way, pre-cloud techniques and technologies once served us well too, and for a long time. The tricky part in any of these instances is to recognize the point when change becomes required—such as when the decision is made to transform an organization into a cloud native entity.

Going cloud native requires giving up those old familiar ways no matter how successful they have been for us previously. For many organizations, this means no longer operating as a hierarchy while they construct monolithic applications. These design patterns are useful, valuable even, in their original context. In a cloud setting, however, they conflict with cloud native principles and can even cause the best-supported migration to fail.

Why do the old ways conflict with cloud native? The reason is Conway’s law.[**1**](https://learning.oreilly.com/library/view/cloud-native-transformation/9781492048893/ch04.html#ch02fn1)

Conway’s law essentially states that system architecture will come to resemble the structure of the organization that contains it. Cloud native architecture requires distributed systems, period. This means succeeding as a cloud native entity requires enterprises to transform themselves from traditional hierarchical processes*—*or even relatively modern Agile practices*—*to a decentralized organizational structure. Cloud native systems architecture literally dictates this reality; Conway’s law merely describes it.

This is a transformational change due to the distributed nature of cloud native, which is based on an architecture of small, loosely coupled modular components, and it leads directly to more decentralized organizations. In order to successfully transform themselves into cloud native entities, organizations must evolve a collaborative, experimental culture to truly take advantage of innovations offered by the cloud. This is harder than it sounds! Traditionally, organizations needed to be massively risk averse, to minimize uncertainty at all costs. Wholesale risk-aversion becomes embedded in the company psychology to the point where the idea of experimentation, of exploring and testing many different answers, can feel genuinely terrifying.

**Cognitive Biases**

We can explain all day long to these clients how and why cloud native’s experimental culture and other organizational changes are the path to success, but we can’t carry them to the destination. They have to walk there for themselves. No matter how many cloud native transformation patterns we map for an enterprise, they simply will not be effective until its people are ready to apply them to their work.

Understanding the underlying causes*—*and that they exist in the first place*—*is the way to move beyond these self-inflicted limitations. We are finding that an effective form for both understanding and explaining this comes in the form of cognitive biases.

A cognitive bias is a kind of systematic error in thinking that affects the decisions and judgments that people make. They are basically hardwired into the human brain. We all have such biases in one way or another, even if we think that we’re personally immune to them (there is even a specific bias named for this: “bias blind spot”). This is understandable*—*when making judgments and decisions about the world around us, we of course like to think of ourselves as objective, logical, and capable of evaluating all relevant available information. Cognitive biases, though, are an unavoidable part of being human. They are baked into our thought processes and decision making.

Such biases are not always bad, though. In fact, they evolved for a reason.

**Why Biases Are Useful**

Biases can sometimes trip us up, leading to poor decisions and bad judgments. But they also serve us.

The human brain is a powerful processor that is constantly taking in and making decisions/acting on innumerable inputs. But, just like any computer processor, it also has limits. If we had to think about every possible option when making even the simplest of decisions, it would take a ridiculously long time. Due to the sheer complexity of our surrounding environment and the staggering amount of information it contains, our brains developed adaptive methods for simplifying the information processing load. Cognitive biases, in effect, are mental shortcuts*—*known as *heuristics—*designed to help us to selectively attend to input, make decisions, and then act on them quickly.

Such shortcuts served us well as early humans, back when our greatest existential threat involved becoming dinner for a saber-toothed tiger. Cognitive biases serve us in dangerous or threatening situations.

For example, imagine walking alone on a city street late at night and spotting sudden movement in the nearby shadows. Cognitive bias has us wired to assume this likely to be some sort of predator*—*muggers being the modern equivalent of saber-toothed tigers*—*and that we need to get to safety as fast as possible. Even though the perceived threat could actually be an alley cat or stray plastic grocery sack blowing in the breeze, we are unlikely to stick around to find out. Our “mysterious movement in nearby darkness” mental shortcut leads directly to worst-case scenario (*Predator*! *Run*!) to get us out of the way of potential danger.

That is an example of attribution bias*—*attributing a cause to an event without knowing what was actually happening and then responding to that limited interpretation regardless of reality. This and similar automatic responses are rooted in human evolution and are pretty easy to recognize. Though they can be surprisingly accurate, they still represent an error in thinking.

Things get even more complicated, though, when emotions, personal motivation, and social or cultural pressures inevitably get mixed in. When we use these pre-programmed shortcuts to assess situations and then make decisions, it all happens unconsciously–and beneath an overlay of those emotional factors. This is how subtle biases can creep in without us ever noticing to influence and distort the way we see, understand, and respond to the world.

Fortunately, simply being aware that this happens at all, that biases exist and influence everyone, is a powerful first step toward combating such distortion and influence. The second step is recognizing what biases might be at play in our own processes, whether personal or organizational.

**Biases, Patterns, and Behavior**

The idea that cognitive biases exist and influence us has become widely accepted over the past few decades and has even inspired various fields of academic research and study.

The concept of cognitive bias was first established in the 1970s by Amos Tversky and Daniel Kahneman. Both were Israeli social scientists who eventually relocated to the United States, Tversky teaching at Stanford and Kahneman at Princeton. They continued working closely, however, and together they pretty much invented the field of behavioral economics while racking up both a MacArthur Foundation “Genius Grant” and the 2002 Nobel Prize in Economic Sciences. Kahneman summarized several decades of their joint discoveries in his 2011 best seller, *Thinking, Fast and Slow*.

In his book, Kahneman identified two “systems” of thinking in the human brain: one conscious and deliberate, the other impulsive and automatic. System 1, our instinctive “fight or flight” wiring, resides in our lower brain and is a legacy of our saber-toothed tiger days. System 2 represents our rational and, above all, aware mental processes required to apply logic to decisions, exert self-control, and deliberately focus attention on non-life-threatening things like office work.

At only a few thousand years old, System 2 is a relatively new feature in the human brain. It evolved to help us function in a more complex world as our primary functioning became less about hunting for dinner (while avoiding becoming anyone else’s dinner) and more about engaging in more abstract survival activities like earning money and posting on social media.

Unfortunately, these two systems don’t play nicely together, or even take turns. Instead, they often fight over which system gets to be in charge in any given situation you face. Whichever one wins determines how you respond.

The way things are supposed to work when we encounter a problem is for System 1 to activate first. If the problem turns out to be too complex or difficult for System 1 to solve quickly, it hands off to analytical and slow-thinking System 2 to figure things out. The reason is because, again, we are wired to cut corners and save energy whenever possible: the law of least effort states that, given any task, the human brain will apply the minimum amount of energy it can get away with. This is why System 1, our impulsive snap-decision brain, gets first crack at most situations.

Trouble arises when the brain perceives problems to be simpler than they actually are. System 1 thinks, “I can handle this!”—even though it actually can’t—and we end up making a bad decision or reaching an erroneous conclusion. To illustrate this in action, Kahneman uses a simple math logic challenge called the “bat and ball problem”:

*A baseball bat and a ball have a total price of $1.10. The bat costs $1 more than the ball. How much does the ball cost?*

Take your time. Think it over. Got it?

If your instant answer is $0.10, we regret to inform you that System 1 just led you astray. If the ball costs 10 cents and the bat costs $1 more than the ball, you ended up with $0.10 plus $1.10, which equals $1.20. Try working the problem again.

After actively pondering things for a minute or two—i.e., activating System 2—you’ll see that the ball must cost $0.05. At $1 more than the ball, that means the bat costs $1.05. Combine the two and you reach the correct $1.10 total.

What just happened here? Well, if the brain gauges that System 1 can just handle things, it won’t bother to activate System 2. We make a snap judgment and then happily move forward in the wrong direction. The process is so automatic and deeply rooted as to be completely unnoticeable when it’s happening. At least, when it gets pointed out to us that our solution to the bat and ball problem is wrong, our System 2 brain can get called into action to overrule System 1’s unmediated impulse answer. But that’s not so easy in the real world, when we’re dealing with unpredictable people and unexpected situations rather than a straightforward arithmetic problem.

Failing to realize that System 1 does not suffice and therefore it’s time to activate System 2 is a universal human problem, according to Kahneman. “Unfortunately, this sensible procedure is least likely to be applied when it is needed most,” he wrote in *Thinking, Fast and Slow*. “We would all like to have a warning bell that rings loudly whenever we are about to make a serious error, but no such bell is available, and cognitive illusions are generally more difficult to recognize than perceptual illusions. The voice of reason may be much fainter than the loud and clear voice of an erroneous intuition, and questioning your intuitions is unpleasant when you face the stress of a big decision.”

This is the process that happens when we apply cognitive biases in a decision-making situation: we too often take the mental shortcut. Unfortunately, according to Kahneman and others who have studied bias, it is basically impossible to alter the leap-before-you-look System 1 part of our brain. We simply can’t control the human hardwiring that served us so well for so long, though we can strive to be aware that it does exist and is always trying to jump into the driver’s seat.

The most effective check against letting System 1 auto-drive our responses ultimately lies not within ourselves but in the people around us: others can perceive our biases and errors more readily than we ourselves are able to. Ultimately, the good news is that we can use external observation to create strategies and policies to help us monitor our individual decisions and predictions. With this approach we can weed out cognitive biases and faulty heuristics in order to make decisions more slowly but correctly, instead of fast but wrong.

This is where patterns come in. (This is, after all, a book about patterns.) Patterns, pattern languages, and transformation designs are all highly intentional and carefully crafted. As such, they function as System 2 tools that can be applied in complex scenarios of many kinds to help keep us on a rational and reasonable path. Patterns themselves can also function as a kind of external observer: later, when we get to our actual cloud native patterns, you will find that they each include a list of the biases that are likely to emerge in that situation.

Linking each pattern to potential biases is our effort to help with your situational awareness—to show the reason that you may be doing (or not doing) certain things is because of a particular bias or to overcome another specific bias. Let’s start by taking a look at the common biases that we have seen influence people and organizations when they undertake a cloud native transformation.

**Nudges**

Over the decades, the idea of cognitive biases has become widely accepted as one of those things that simply make us humans. Research has made it clear, furthermore, that they are hardwired and inalterable.

The only possible thing we can change is our behavior—the actions we take while under the influence of a bias. The best way to change bias-driven behavior takes the form of incentives or “nudges.” A nudge is a subtle cue or context change that prods you toward making a certain decision, while leaving all options open. Nudges don’t force us toward any one action; our actions are entirely voluntary. However, nudges do subtly guide us in a particular direction.

The notion was introduced by Richard Thaler, a University of Chicago behavioral economist, and Cass Sunstein, a Harvard legal scholar, in their book *Nudge: Improving Decisions about Health, Wealth, and Happiness*, first published in 2008. They draw directly on Kahneman’s work, particularly the concept that there are fast and slow systems of human thinking. (Thaler and Sunstein call System 1 “the automatic system” and System 2 “the reflective system” but the upshot is the same.) The book demonstrates quite powerfully that we all are being nudged all day, every day—sometimes toward good decisions, sometimes toward bad ones.

For example, if your company’s cafeteria places fresh fruit by the cash register instead of pastries, you are more likely to take the nudge to choose a healthier snack or dessert. Conversely, when a server asks “Would you like fries with that?” you are being nudged toward a choice that benefits the restaurant’s profit margin (though not your own healthy best interest). The good news is that we can design systems and environments to counter*—*or even harness*—*biases so that we will be nudged toward making a beneficial choice.

The most powerful form of nudge is known as the default. Default nudges are choices set up so that if you do nothing, you will be choosing the desired thing by simply going with the option presented. This has been used to raise the number of people who become organ donors in the United States. Many states have tried to boost organ donor rates by implementing a default nudge. They have shifted from an explicit opt-in question when people apply for a driver’s license or renewal (“Would you like to be an organ donor?”) to making all applicants organ donors by default, with the chance to explicitly opt out. The default doesn’t force anything—applicants may still freely choose whether they’d like to be a donor. But the change leads to many more organ donors, and more lives saved, because social and behavioral sciences research shows most people accept whatever is listed as the default option.

So, ultimately nudges are a way to manage bias*—*maybe even the only way.

Take, for example, the cognitive bias known as *present bias—*the tendency to favor immediate rewards at the expense of our long-term goals. This is well demonstrated by how a large majority of people will say that putting money into a retirement savings account is important and how few people will actually follow through and do so. They mean to take some money out of their next paycheck and start an account eventually…just, not right now.

In this case, present bias has contributed to a severe shortfall in retirement savings in the United States, where millions of Americans face the very real likelihood of getting too old to continue working*—*but without the means to stop. According to a 2018 study by Northwestern Mutual, a financial services company, 21% of Americans have no retirement savings at all, while an additional 10% have less than $5,000 in savings. A third of the population currently at, or about to reach, retirement age has less than $25,000 set aside for financial support for their golden years. How did this happen? Historically, participation in retirement savings programs has been voluntary. Present bias led most people to keep that money in their current paychecks, rather than opting in to set money aside for their golden years.

While present bias has so far proved intractable, employers have been able to nudge employees into contributing to retirement plans by making saving the default option: now you have to take active steps in order to not participate. Yes, laziness or inertia can be even more powerful than bias!

This is a classic example of “choice architecture”—the way in which our decisions are influenced by how the choices are presented. People can be nudged by arranging their choice architecture in a certain way, like by placing healthy foods in a school cafeteria direct at eye level while putting less healthy choices like chips and cookies in harder-to-reach places. Individuals are not actually prevented from eating whatever they want, but arranging the food choices that way causes people to eat less junk food and more healthy food.

“How does this apply in cloud native software development?” you may well ask. Well, biases permeate all human undertakings, and cloud native’s decentralized hierarchy means that teams, as well as individuals, are no longer tightly overseen and directly managed—and so sometimes biases climb into the driver’s seat. Even more importantly, however, is that transforming into a cloud native organization requires getting comfortable with uncertainty and change (more about this in later chapters).

Both of these cloud native realities, thus, make it important to understand that biases are operating in the first place and must be taken into consideration and countered—often, interestingly, with more bias. For example, ambiguity provokes anxiety, which in turn leads us down a well-worn path to many different biases. Knowing this, in the cloud native context we can counteract with an abundance of information and small, manageable experiments to build certainty and familiarity while reducing anxiety. You fight phobias with exposure to the thing that is feared, right? Well, the very common bias known as “status quo effect” is essentially change phobia…but when change becomes a way of life, you aren’t afraid anymore. When experimentation becomes routine, the new is no longer to be feared. Innovation as a routine process is, after all, a core cloud native principle.

**Common Biases and Nudges**

Cognitive bias has become a popular mainstream topic, and many, many examples have been defined. Wikipedia’s “List of Cognitive Biases” contains, as of this writing, 193 entries. The list of the many different flavors of flawed thinking, all apparently hardwired into the human brain, literally ranges from A (the ambiguity effect, or “The tendency to avoid options for which the probability of a favorable outcome is unknown”) to Z (the Zeigarnik effect, where “uncompleted or interrupted tasks are remembered better than completed ones”).

The list is broken down into three types of bias:

* *Decision-making, belief, and behavioral biases* that affect belief formation, business and economic decisions, and human behavior in general
* *Social biases*, a form of *attribution bias* that describes the faulty assumptions that affect our thinking when we try to explain the cause of our own behavior or that of other people (“Rather than operating as objective perceivers, people are prone to perceptual errors that lead to biased interpretations of their social world,” Wikipedia helpfully explains.)
* *Memory biases* that either enhance or impair the recall of a memory or that alter or shift its details

All of these are things that can hold a project back, slow it down, or even cause it to fail.

We have identified the 24 cognitive biases that, in our experience, most commonly show up during cloud migration projects. Most of these biases fall into the category of decision-making, belief, and behavioral biases, with a couple of social biases thrown in. We have also included any related nudges that can flip a bias from being a problem to being a force for positive change.

These commonly occurring biases are presented in the following list, starting with each one’s Wikipedia**5** definition followed by a description of how it tends to present itself in the course of a cloud native transformation.

*Ambiguity effect*

The tendency to avoid options for which missing information makes the probability of the outcome seem “unknown.” An example of ambiguity effect is that most people would choose a regular paycheck over the unknown payoff of a business venture.

*Cloud native relationship*: This bias is the main reason to run experiments early in a transformation, to understand the project’s scope and fill in the missing information gaps. Otherwise, people tend to do what they have always done, because they know exactly how that worked, even when it no longer applies to the new context.

*Authority bias*

The tendency to attribute greater accuracy to the opinion of an authority figure (unrelated to its content) and be more influenced by that opinion.

*Cloud native relationship*: In traditional hierarchical (Waterfall) organizations, authority figures have to know more than others below them in the hierarchy. In cloud native, this bias is even more dangerous, as the managers have less understanding of what’s going on in a highly complex distributed architecture based on new technologies. They need to be careful to avoid giving orders or even providing opinions as they will be automatically received as “correct.”

*Availability heuristic*

The tendency to overestimate the likelihood of events with greater “availability” in memory, which can be influenced by how recent the memories are or how unusual or emotionally charged they may be.

*Cloud native relationship*: “Everyone is talking about Kubernetes so suddenly it must be the right thing to do!”

*Bandwagon effect*

The tendency to do (or believe) things because many other people do (or believe) the same thing. Related to groupthink and herd behavior.

*Cloud native relationship*: When Gartner puts certain tech on their chart, everyone decides to adopt it even without understanding how it relates to their use case.

*Bystander effect*

The tendency to think that others will act in an emergency situation.

*Cloud native relationship*: This is very relevant for overlapping responsibilities. In a Waterfall organization with its many specialized teams, for example, when a task doesn’t officially belong to anyone, no one will volunteer to pick it up. In cloud native, where teams are responsible for independent execution, there needs to be clear communication so all necessary tasks get covered.

*Confirmation bias*

The tendency to search for, interpret, focus on, and remember information in a way that confirms one’s preconceptions.

*Cloud native relationship*: Ignore all those inconvenient facts and embrace the information that supports your opinion. There is always plenty of data to choose from, so it’s easy to cherry-pick. If you have a system administrator with 20 years of traditional IT experience, he’ll be very creative in finding lots of very important reasons not to move to the cloud. If an engineer is dead set on using some cool tool he heard about at a conference, he will ignore all information showing that another tool might actually be a better fit.

*Congruence bias*

The tendency to test hypotheses exclusively through direct single testing, instead of testing multiple hypotheses for possible alternatives.

*Cloud native relationship*: When you run an experiment to prove your point rather than to find new information. So, you would run only one proof of concept (PoC), and if it works, you’ll automatically dismiss all other alternatives even without evaluating them. This is why we have a PoC pattern and a separate Exploratory Experiments pattern.

*Curse of knowledge*

When better informed people find it extremely difficult to think about problems from the perspective of less well-informed people.

*Cloud native relationship*: Our clients’ engineers frequently struggle to sell cloud native ideas to their managers due to this bias: they see so clearly why this is the right thing to do that they forget the managers have no background knowledge that enables them to understand it with equal clarity. Even our own engineers and consultants, so immersed in cloud native, need to stay mindful of seeing things from the perspective of clients who are new to cloud native.

*Default effect*

When given a choice between several options, the tendency is to favor the default one.

*Cloud native relationship*: Solutions to any problem have to consider the correct defaults, as those defaults will be adopted more frequently compared to any customized option. In other words, if you set up a cloud native platform, most people are probably going to use it exactly as you gave it to them. This is true both for the tools built into cloud platforms like Amazon Web Services or Azure as well as internal tools provided to employees. It’s why we have the Starter Pack pattern.

*Dunning-Kruger effect*

The tendency for unskilled individuals to overestimate their own knowledge/ability, and for experts to underestimate their own knowledge/ability.

*Cloud native relationship*: This bias leads to overestimating your competency at things you’re not intimately familiar with. We see this with managers who try to dictate which tools or methods will be used as part of a transformation. They have no actual cloud native experience or knowledge, but they are accustomed to calling all the shots.

*Hostile attribution bias*

The “hostile attribution bias” is the tendency to interpret others’ behaviors as having hostile intent, even when the behavior is ambiguous or benign.

*Cloud native relationship*: We need to consider this bias whenever we go into a new client’s organization and when working with internal transformation leaders who can run into this same form of resistance. In both scenarios we meet people who think that we’re there to destroy their work and wreck their company. We shouldn’t think that this is their real opinion, as it is a normal human bias arising from the fact that change frequently creates anxiety in those poised to undergo it.

*IKEA effect*

The tendency for people to place a disproportionately high value on objects that they partially assembled themselves, such as furniture from IKEA, regardless of the quality of the end result.

*Cloud native relationship*: This one can be used positively, as a nudge, and explains why we have to involve everyone in the client’s organization in every stage of planning and executing a transformation. Whoever is involved is biased to like and support the solution.

*Illusion of control*

The tendency to overestimate one’s degree of influence over other external events.

*Cloud native relationship*: This is especially common in the uncertain circumstances of a cloud native transformation. Engineers think that they know how to build microservices, and managers think that they know what it takes to do DevOps. But in reality, it is only an illusion of control. Many complex and emergent processes are very difficult to even steer, much less control. Sometimes we need to embrace some uncertainty to ultimately get results.

*Information bias*

The tendency to seek information even when it cannot affect action.

*Cloud native relationship*: Analysis paralysis. Very common in a Waterfall organization’s efforts to try to find more and more answers for more and more questions, regardless of the fact that there are only two or three possible actions to take and the benefits of one of them are very clear.

*Irrational escalation (also known as sunk-cost fallacy)*

The phenomenon where people justify increased investment in a decision, based on the cumulative prior investment, despite new evidence suggesting that the decision was probably wrong.

*Cloud native relationship*: If you’ve spent six months working on setting up an OpenShift cluster and bought all the licenses, it is very unlikely that you’re going to switch to another tool even if it’s proven to be superior. People routinely push forward with projects that are obviously not going to bring any value.

*Law of the instrument*

An over-reliance on a familiar tool or methods, ignoring or under-valuing alternative approaches. “If all you have is a hammer, everything looks like a nail.”

*Cloud native relationship*: Moving to cloud native while using old techniques/processes. Examples of this include using Scrum for innovation; deploying many microservices coupled tightly together; telling people what to do in a highly distributed team building microservices; or acting in other ways that avoid change in favor of what is already familiar.

*Ostrich effect*

Ignoring an obvious (negative) situation.

*Cloud native relationship*: Hardly requires explaining. We know that when we move to cloud native we need to make significant cultural shifts, not just change the tech. However, many companies choose to either ignore this altogether or make only small cosmetic changes meant to signal they’ve transformed their culture—and, either way, try to work in the new paradigm using old processes that no longer apply.

*Parkinson’s law of triviality (“bikeshedding”)*

The tendency to give disproportionate weight to trivial issues. Also known as bikeshedding, this bias explains why an organization may avoid specialized or complex subjects, such as the design of a nuclear reactor, and instead focus on something easy to grasp or rewarding to the average participant, such as the design of a bike shed next to the reactor.

*Cloud native relationship*: When people get together for three days of discussion and planning their cloud native migration—and then talk about tiny trivial things like which machine will Kubernetes run on or how to schedule a specific microservice. All this while avoiding large challenges, like changing organizational culture or overall architecture.

*Planning fallacy*

The tendency to underestimate task completion times. Closely related to the well-traveled road effect, or underestimation of the duration taken to traverse oft-traveled routes and overestimation of the duration taken to traverse less familiar routes.

*Cloud native relationship*: Especially operative in uncertain situations like moving to cloud native for the first time. We are eager to estimate the time and resources required but have no idea what it actually takes to move to cloud native. So, some people estimate it as a few weeks of work, maybe a couple months at most. Reality: often a year or longer. Basically if you don’t have baselines from previous experience, any estimation is totally worthless.

*Pro innovation bias*

The tendency to have an excessive optimism toward an invention or innovation’s usefulness throughout society, while failing to recognize its limitations and weaknesses.

*Cloud native relationship*: “Let’s do Kubernetes, regardless if it’s a good fit or even necessary. Because it is new and cool.”

*Pseudocertainty effect*

The tendency to make risk-averse choices if the expected outcome is positive, but make risk-seeking choices to avoid negative outcomes.

*Cloud native relationship*: Successful teams will avoid investing in improvements while everything is going OK. But once a crisis erupts they will jump on any crazy new tool or process to save themselves. This is also a challenge for us: how do we help successful teams and companies overcome this and invest in continual improvement? The best motivator is recognizing the existential risk of standing still.

*Shared information bias*

The tendency for group members to spend more time and energy discussing information that all members are already familiar with (i.e., shared information), and less time and energy discussing information that only some members are aware of.

*Cloud native relationship*: The whole team went to a Docker training, so they spend a lot of time talking about Docker—and no time at all about Kubernetes, which is equally necessary but they don’t know much about it. To reduce this one, especially in new and complex environments, teams have to learn new things all the time.

*Status quo bias*

The tendency to like things to stay relatively the same.

*Cloud native relationship*: Entire companies, and/or people within the organization, will resist moving to cloud native due to this bias: everyone wants to remain comfortably right where they are right now, which is known and understood.

*Zero-risk bias*

Preference for reducing a small risk to zero over a greater reduction in a larger risk.

*Cloud native relationship*: This is the opposite of low-hanging fruit. For example, companies want to reach 99.9999% availability, which is very difficult, yet they have no CI/CD to deliver the changes in the first place.

**What to Watch For, and How to Overcome**

A few of these biases are particularly hazardous to cloud native transformations. In particular we see the status quo bias operating in many client migration initiatives, especially in long-established companies. In this scenario, people have been doing the same set things in the same set way for 20 years and they simply don’t want to change. They may not actively protest the move to cloud native, but neither do they actively help make it happen. Such inertia, repeated in person after person across an entire organization, can seriously hinder a transformation initiative.

There are several others. Law of the instrument bias—where people tend to rely on old skills and methods for doing new things, often unconsciously—leads to more individualized problems. One example happens when a project leader within a digitally transformed company still reflexively insists on approving each small change or experiment their team may want to try. This was the policy in the previous Waterfall hierarchy, but applying it now effectively blocks their team’s autonomy—when it is just this independence and ability to iterate and experiment that drives the velocity and responsiveness of a cloud native system. Planning fallacy is another extremely common hazard; most companies enter into a transformation thinking it will be relatively quick and simple. They budget time and resources for only a few weeks or months, only to find that a proper transformation can require a year or more to successfully complete. And, finally, the ostrich effect explains the common scenario that arises when a company tries to transform its tech while ignoring the need to also transform its organizational culture and processes.