

# Leading Agile Transformation from a Beach

---

*Two Agile Coaches' Journeys from Mandatory Failure to Masterless Mastery*

Michael Carey and Todd Kromann

**Contents**

Mandatory Failure ..... 1

Giant Baby Steps ..... 2

Benevolent Despotism ..... 2

Un/Learning ..... 3

Humble Competence ..... 4

Masterless Mastery..... 4

Todd and Mike have both experienced the growing pains that come from becoming Agile and helping others with the same. They had similar experiences and learned similar lessons, but their journeys have been surprisingly different. One took 15 years, the other took 5, and both took longer than they had to.

## Mandatory Failure

*“In governing, don't try to control.*

*In work, do what you enjoy.*

*In family life, be completely present.” – Lao Tsu*

**Todd's** journey began as a consultant, working for an auto manufacturer where the CIO demanded that the company move to an entirely new methodology. The CIO was an imposing figure. She pounded her fists on the podium and said, “You will be onboard with this process or you will not be onboard at this company!” She then pointed to Todd's area and said, “And here is the guy who is going to transform you.” Todd had a sickening feeling in his stomach as the hostile glares from 400 managers were directed at him.

Through sheer force of will, the company began to move to the new process in an effort to appease their leader. The direction was very top-down, command-and-control. The CIO had inadvertently just asked everyone in the company to jump off a cliff. Todd successfully helped them do just that. His initial engagement with the company lasted a year. The expected improvements did not come, despite his best efforts.

Roughly four years after leaving, Todd revisited the company and they were still failing. The people were still trying to make the CIO's way work, and therein was the problem. This process did not belong to the people. They did not take ownership of it. They weren't doing this because they wanted to; they were doing it because they didn't want to lose their jobs.

**Mike** entered his company as they were just getting into the swing of Waterfall. Their PM career path had just been established and their ISDLC was fresh off the presses. As a Junior Software Developer, he struggled with the constant churn and politicking that came as by-products to the new Waterfall methodology. He didn't know that there was another way – a better way – to do things. He was fresh out of college and had been warned that the “real world” would be different than academia.

In an attempt to make things run more smoothly, they became a matrixed organization; unfortunately, this had the side-effect of finger pointing and stifled collaboration. Mike did his best to make his projects as successful as they could be, but all of them left a bad taste in his mouth. He knew they could do better, but struggled to improve with subsequent projects. He noticed a few small things. For example, smaller projects worked better, as did those where you worked with people you'd already worked with. But every project seemed too big and no two projects' teams were identical.

Mike worked for about two years in this methodology before he decided he was sick of it and was going to try something called “Agile”.

## Giant Baby Steps

*“The journey of a thousand miles starts from beneath your feet” – Lao Tsu*

**Todd** left the auto manufacturer for a time and consulted with an insurance company. He was to assist with their Agile transformation. They took a much more human-centered approach and did a fantastic job of getting buy-in. Yet their process seemed to take much longer to emerge and mature than anticipated. They were willing to experiment, but experiments were drawn out and took too long to learn from.

While Todd appreciated their patience with people, he often found himself wondering how to speed up the rate of change. His engagement ended before the company arrived at their originally intended destination. It took over eight years to achieve the desired level of company-wide maturity.

**Mike** became the tech lead for a new project that he decided would be Agile. Unfortunately, he had no idea what that meant. He started reading up more on what Agile meant as the project entered the Initiation and Requirements phases. He began doing design work and planning out construction Sprints while trying to wrap his head around the concept of Story Points.

Then Mike managed to get Agile training for his team, shortly after the Construction phase of the project started. They started to see the error of their ways and began to retro-fit practices to help them run the rest of the project in a more Agile way. They had already planned on multiple releases, so the requirements that had been gathered and designed were only a portion of the full scope of the project. The team experimented a lot over the course of a year and learned a lot about how not to do Agile.

The project ended up losing funding after the first release, about a year after it started. Despite the many stumbles and fumbles of the team, the project highlighted the success of Agile in Mike’s area – they had a project cut in less than half and were able to deliver more than just documentation.

## Benevolent Despotism

*“When you are content to be simply yourself and don't compare or compete, everybody will respect you.” – Lao Tsu*

**Todd** moved on to another Insurance company with plenty of experience under his belt. He knew what was best for them. All he had to do was show the way; they would surely see the wisdom and follow. The company liked what they heard and saw that Todd clearly knew what he was talking about. Unfortunately, Todd put the focus on him and the practices he brought instead of getting the teams to own their transformation.

To be fair his practices were impressive. Todd was able to share with them many great ways of doing, and they liked what they saw. In fact, Todd became the best practitioner they had. Of course, he wasn’t brought in to become their star practitioner; he was brought in to develop star practitioners.

By the time he left, about two years later, many were still struggling with the change instead of feeling empowered to continuously improve themselves.

**Mike** was loaned to another area to help with a struggling project. It was a in different domain, and for a different country with a different language and culture. On top of all that, almost all of the developers were contractors. He could see the flaws in how they did things and couldn't wait to show them the error of their ways.

He got involved mostly through emails and conference calls, though he was sent on a 3 week visit at the end of his engagement. While there, Mike evangelized XP practices and Scrum techniques that he knew would help get the project out of the proverbial ditch. He had been using these practices during his engagement to that point, and he showed them the results in person.

At the end of his half-year involvement, the developers had not changed at all. They were more concerned about fighting fires than fixing the root problem, and Mike's approach did not help them "see the light". Perhaps the only change Mike brought about was the resentment with which the dev team viewed him.

## Un/Learning

*"The ancient Masters*

*didn't try to educate the people,*

*but kindly taught them to not-know."* – Lao Tsu

**Todd** went back to the auto maker of his youth, this time through a different consulting company, and realized how much damage he had helped implement over five years ago. The company was struggling with a growingly over-architected process and had moved further and further from their roots. In other words, they were still trying to out-live jumping off a cliff.

He helped them to see how much better their roots were than what they had become. He worked with them to simplify their processes. They unlearned what they had learned over the past five years so that they could start to learn the right things. Over the course of 3 years, they leaned down their processes and began running much more happily and efficiently. Simplicity was a process they could really own.

**Mike** came back to his home area, no longer on loan, and became his domain's first full-time Scrum Master (a.k.a. not a Tech Lead / Scrum Master). He worked with a Scrum team that had just split from one team into two. They looked to him for answers that he tried not to give them. They would argue and try to defer to his judgment. What he found was that the root cause was usually the team's culture and the mindset that resulted from it.

He helped them agree on problems and experiment with solutions. He used his knowledge of common practices to help guide them, but tried to let them "have it his way". They almost always went in the direction that he wanted, but he helped them arrive there as if it were their decision all along. They understood both what they were doing and why.

By unlearning command-and-control and learning to experiment, they were able to fail fast consistently. After eight or nine months, Mike transitioned the teams, which had now become three, to new Scrum Masters.

## Humble Competence

*“If you want to govern the people,  
you must place yourself below them.  
If you want to lead the people,  
you must learn how to follow them.” – Lao Tsu*

**Todd** left the consultant world and joined Mike’s company. He became part of the organization within the company responsible for defining and enforcing its SDLC. He worked hard to increase the knowledge and adoption of Agile, despite finding little traction for it. He did his best to lead Agile by example, showing others on his team what “Being Agile” looked like. They added Agile to the SDLC and several teams began to implement it, with varying levels of success. He continued to act as change agent for three years before becoming a full-time Agile Coach.

**Mike’s** domain decided to go “all Agile”. All new work was to be done using Agile, and all in-flight work was to be transitioned to Agile. They went from three teams to seven, and he led them from an Agile perspective for nine months (during which time they grew to eleven teams). He did his best to learn from his past, letting the teams own their transformations and experiment to find improvements. He developed strong Scrum Masters who understood what it meant to be a Servant Leader. He was able to leave the domain in the hands of a capable successor because he brought her along instead of asking her to just do as told.

## Masterless Mastery

*“If you don't trust the people,  
you make them untrustworthy.” – Lao Tsu*

**Todd** and **Mike** met when Mike founded the Agile User Group at their company. They eventually began to work together as company-wide Agile Coaches, helping teams and individuals learn Agility so that they didn’t need help anymore. They presented at user group meetings and local PMI chapter meetings. They provided ad-hoc training for teams, including some who had no full-time Agile Coach and weren’t developing software.

Their goal is not to help teams learn how to do Agile. Best practices for doing Agile are always changing and are defined by those who truly are Agile. Todd and Mike protect people from executive mandates and help them unlearn what they have learned. They show them the benefit of short iterations, small chunks, and continuous improvement. They provide an environment conducive to learning so that people can teach themselves competence. And, as necessary, they provide teams helpful tools and practices as examples of “how” to the “why” of Agile.

Todd and Mike have learned that there are different classes of workers. Each type of worker has a different way to respond when their management tells them to jump off a cliff. First, there is the person who jumps. This is a bad employee. This employee costs time and damage to themselves and their management. Second, there is the person who refuses to jump. Subversively or in open mutiny, they will refuse to damage themselves and the company. However, they also won't accomplish anything. This employee, while good, is not very useful. The third type of employee shows management that they are asking the team to jump off a cliff. They help management understand the cliff and build a bridge across it so that the needs of the company are met without damage. This is the best employee. This is the employee you get when you don't try to control people.

The third type of employee is the type that Todd and Mike now try to emerge out of the people they work with. And these are the people who will own and drive organizational transformation so that you can take a little time to relax on the beach.