



THE FIVE IDEALS: DEVELOPERS, DIGITAL DISRUPTION, AND THRIVING IN THE AGE OF DATA

Gene Kim

The Age of Software and Data is here. Fast beats slow. And fast and big will win almost every time.

But how do organizations move fast when development teams under immense pressure to deliver are paralyzed by cultures of fear and blame? And when progress is bogged down by processes and impenetrable bureaucracy, where even the smallest changes require endless committees, paperwork, and approvals?

In my previous book, *The Phoenix Project*, my goal was to explore and reveal the necessary but invisible structures required to make developers (and all engineers) productive, and reveal the devastating effects of technical debt and complexity. To do so, I outlined The Three Ways to show how IT work has more in common with traditional manufacturing plant work than most ever imagined.

The Three Ways are as follows.

The First Way emphasizes the performance of the entire system, as opposed to the performance of a specific silo of work or department—this can be as large as a division (e.g., Development or IT Operations) or as small as an individual contributor (e.g., a developer, system administrator).

The focus is on all business value streams that are enabled by IT. In other words, it begins when requirements are identified (e.g., by the business or IT), are built in Development, and then transitioned into IT Operations, where the value is then delivered to the customer as a form of a service.

The outcomes of putting the First Way into practice include never passing a known defect to downstream work centers, never allowing local optimization to create global degradation, always seeking to increase flow, and always seeking to achieve profound understanding of the system (as per Deming, who we'll hear from again below).

How do organizations move fast when development teams under immense pressure to deliver are paralyzed by cultures of fear and blame?

The Second Way is about creating the right to left feedback loops. The goal of almost any process improvement initiative is to shorten and amplify feedback loops so necessary corrections can be continually made.

The outcomes of the Second Way include understanding and responding to all customers, internal and external, shortening and amplifying all feedback loops, and embedding knowledge where we need it.

The Third Way is about creating a culture that fosters two things: continual experimentation, taking risks and learning from failure; and understanding that repetition and practice is the prerequisite to mastery.

We need both of these equally. Experimentation and taking risks are what ensures that we keep pushing to improve, even if it means going deeper into the danger zone than we've ever gone. And we need mastery of the skills that can help us retreat out of the danger zone when we've gone too far.

The outcomes of the Third Way include allocating time for the improvement of daily work, creating rituals that reward the team for taking risks, and introducing faults into the system to increase resilience.

To quickly recap the Three Ways, they are:

THE FIRST WAY: Systems Thinking

THE SECOND WAY: Amplify Feedback Loops

THE THIRD WAY: Culture of Continual Experimentation and Learning

Rather than a handbook (I've written some of those, as well), I decided to write *The Phoenix Project* as a novel to introduce the core human element in organizational transformation, allowing readers to recognize and understand the internal and inter-personal challenges (and triumphs) all workers experience in the process.

My new book, *The Unicorn Project*, retells the story of *The Phoenix Project* from the perspective of development and architecture. It explores the invisible structures that are required to make developers (and all engineers) productive, which is absolutely required for the initiatives they work on to achieve their goals. It is about technical debt and the incredible damage it creates, as well as how we create the opposite conditions—an architecture that enables developers to quickly, independently, and safely develop, test, and deliver value to customers, and where we are continually focused on elevating developer productivity.

What makes this problem so insidious is that architecture and the systems that developers rely on for their daily work are invisible. Everyone can see features, or the application. Therefore, they are easy to fund. However, far less visible are the APIs and architectures those features rely on, as well as the processes and infrastructure that developers need to get their daily work done.

It is also about rebellion and coalition-building, and about how small teams are able to achieve all the amazing outcomes described in *The Phoenix Project*, despite seemingly endless resistance from powerful, entrenched functional silos. These are stories about transformational leadership and courage, which are very much inspired by the heroic journeys I've had the privilege of observing and studying for the last seven years within the DevOps enterprise community.

It also examines the problems posed by data, and what is required to actually use it to solve our most pressing business problems. Digital disruption requires us to understand our customers, and is greatly aided by data we already have, which we need to manipulate, transform, and safely allow teams to use.

So much of the genesis of the DevOps movement was to address the huge problems of getting code into production. These days, there is an orthogonal problem of getting data from where it resides (often trapped in data warehouses) to where it can be used by teams to help their organizations win in the marketplace. I'm hoping that this book conclusively demonstrates to all that data is ultimately a software effort, and can be massively helped by DevOps principles and patterns.

To compliment the Three Ways, I reveal in this new story The Five Ideals. I use the story to point out the role of bottlenecks and red tape in the DevOps practices within a century-old auto parts retailer and manufacturer that is on the brink of failure, and demonstrate that companies operating by the Five Ideals will flourish.

Experimentation and taking risks are what ensures that we keep pushing to improve, even if it means going deeper into the danger zone than we've ever gone.

THE FIVE IDEALS

The First Ideal is **Locality and Simplicity**. We need to design things so that we have locality in our systems and the organizations that build them. And we need simplicity in everything we do. The last place we want complexity is internally, whether it's in our code, in our organization, or in our processes. The external world is complex enough, so it would be intolerable if we allow it in things we can actually control! We must make it easy to do our work.

The Second Ideal is **Focus, Flow, and Joy**. It's all about how our daily work feels. Is our work marked by boredom and waiting for other people to get things done on our behalf? Do we blindly work on small pieces of the whole, only seeing the outcomes of our work during a deployment when everything blows up, leading to firefighting, punishment, and burnout? Or do we work in small batches, ideally single-piece flow, getting fast and continual feedback on our work? These are the conditions that allow for focus and flow, challenge, learning, discovery, mastering our domain, and even joy.

In its briefest form: The Third Ideal is **Improvement of Daily Work**. The Fourth Ideal is **Psychological Safety**, where we make it safe to talk about problems, because solving problems requires prevention, which requires honesty, and honesty requires the absence of fear. In manufacturing, psychological safety is just as important as physical safety. And finally, the Fifth Ideal is **Customer Focus**, where we ruthlessly question whether something actually matters to our customers, as in, are they willing to pay us for it or is it only of value to our functional silo?

To unpack those last three a little, consider how a century ago, when mass production revolutionized industry, the role of the leader was to design and decompose the work and to verify that it was performed correctly by armies of interchangeable workers, who were paid to use their hands, not their heads. Work was atomized, standardized, and optimized. And workers had little ability to improve the system they worked within.

Which is strange, isn't it? Innovation and learning occur at the edges, not the core. Problems must be solved on the front-lines, where daily work is performed by the world's foremost experts who confront those problems most often.

That's why the Third Ideal is Improvement of Daily Work. It is the dynamic that allows us to change and improve how we work, informed by learning. As Dr. Steven Spear said, "It is ignorance that is the mother of all problems, and the only thing that can overcome it is learning."

The most studied example of a learning organization is Toyota. The famous Andon cord is just one of their many tools that enable learning. When anyone encounters a problem, everyone is expected to ask for help at any time, even if it means stopping the entire assembly line. And they are thanked for doing so, because it is an opportunity to improve daily work. And thus problems are quickly seen, swarmed, and solved, and then those learnings are spread far and wide, so all may benefit. This is what enables innovation, excellence, and outlearning the competition.

The opposite of the Third Ideal is someone who values process compliance and TWWADI, "The Way We've Always Done It." It's the huge library of rules and regulations, processes and procedures, approvals and stage gates, with new rules being added all the time to prevent the latest disaster from happening again.

You may recognize them as rigid project plans, inflexible procurement processes, powerful architecture review boards, infrequent release schedules, lengthy approval processes, strict separation of duties... the list goes on and on. Each adds to the coordination cost for everything we do, and drives up our cost of delay. And because the distance from where decisions are made and where work is performed keeps growing, the quality of our outcomes diminish. As W. Edwards Deming once observed, "a bad system will beat a good person every time."

You may have to change old rules that no longer apply, or change how you organize your people and architect your systems. For the leader, it no longer means directing and controlling, but guiding, enabling, and removing obstacles. General Stanley McChrystal massively decentralized decision-making authority in the Joint Special Operations Task Force to finally defeat Al Qaeda in Iraq, their much smaller but nimbler adversary. There the cost of delay was not measured in money, but in human lives and the safety of the citizens they were tasked to protect.

Innovation and learning occur at the edges, not the core. Problems must be solved on the front-lines, where daily work is performed by the world's foremost experts who confront those problems most often.

That's not the celebrated servant leadership we read so much about, it's transformational leadership. It requires understanding the vision of the organization, the intellectual stimulation to question the basic assumptions of how work is performed, inspirational communication, personal recognition, and supportive leadership.

Some think it's about leaders being nice. Nonsense. It's about excellence, the ruthless pursuit of perfection, the urgency to achieve the mission, a constant dissatisfaction with the status quo, and a zeal for helping those the organization serves.

Which brings us to the Fourth Ideal of Psychological Safety. No one will take risks, experiment, or innovate in a culture of fear, where people are afraid to tell the boss bad news. In those organizations, novelty is discouraged, and when problems occur, they ask "Who caused the problem?" They name, blame, and shame that person. They create new rules, more approvals, more training, and, if necessary, rid themselves of the 'bad apple,' fooling themselves that they've solved the problem itself.

The Fourth Ideal asserts that we need psychological safety, where it is safe for anyone to talk about problems. Researchers at Google spent years on Project Oxygen and found that psychological safety was one of the most important factors of great teams: where there was confidence that the team would not embarrass, reject, or punish someone for speaking up.

When something goes wrong, ask "what caused the problem," not "who." Commit to doing what it takes to make tomorrow better than today. As John Allspaw says, "every incident is a learning opportunity, an unplanned investment that was made without our consent."

Picture this scenario: You are in an organization where everyone is making decisions, solving important problems every day, and teaching others what they've learned. Your adversary is an organization where only the top leaders make decisions. Who will win?

It's so easy for leaders to talk about the platitudes of creating psychological safety, empowering and giving a voice to the front-line worker, but repeating platitudes isn't enough. The leader must constantly model and coach and positively reinforce these desired behaviors every day. Psychological safety slips away so easily when the leader micromanages, can't say "I don't know," or acts like a know-it-all, pompous jackass. And it's not just leaders, it's also how one's peers behave. It either creates a culture of psychological safety, or it doesn't.

The Fifth Ideal is about a ruthless Customer Focus, where you are truly striving for what is best for them, instead of the more parochial goals that they don't care about, whether it's your internal plans of record or how your functional silos are measured. Ask whether your daily actions truly improve the lives of your customer, create value for them, and whether they'd pay for it. And if they don't, maybe you shouldn't be doing it.

**When something goes wrong, ask
“what caused the problem,” not “who.”**

Instead, think of the Fifth Ideal, of being truly customer-centric instead of being silo-centric. When it comes to thinking of the applications and services that you manage, ask: Which of them are customers willing to pay us for? Which ones truly enhance our competitive advantage? And which can we rely on vendors for?

A hundred years ago, most large factories had a CPO—a chief power officer—who ran the electricity generation processes. It was one of the most important roles in manufacturing, because no electricity, no production. It was a Core process. But that role has disappeared entirely. Electricity has become infrastructure that you buy from a utility company. It is interchangeable, and you choose suppliers primarily on price. There is rarely a competitive advantage to generating your own power. It is now merely Context, no longer Core. You don't want to be the organization that has a large staff providing internal power generation.

As Clay Christiansen once stated, one keeps what is “not good enough” and outsources what is “more than good enough.” I challenge you to think deeply about the Fifth Ideal and identify areas of Context that you can unload, freeing yourself from decades of technical debt, things that have been shackling you for years or maybe even decades. Imagine what you can get done without all those things dragging you down. Even though it may be more painful in the short term, you will find some unexpected and critical dividends long term.

This is never easy. You need someone who truly understands the business, someone hard-nosed who can drive standardization across the company, who truly has the best interests of the entire organization at heart, and who knows what technology can and can't do.

But imagine a world where you can make decades of technical debt disappear, where you rid yourself of bad automation built on top of bad business processes. Imagine what it could feel like to deliberately and carefully choose what to leave behind and where you could spend your time and energy instead.

The notion of simplifying the business and technical landscape of the company is breathtaking. Most leaders actually live working on complex business problems, but it would be so much better and easier if they weren't obstructed by the decades of senseless complexity and accumulated neglect.

Before I come to my final point, I'd like to briefly summarize the Five Ideals:

THE FIRST IDEAL: Locality and Simplicity. There needs to be simplicity in everything that is done. The last place an organization wants complexity is internally, whether it's in their code, their organization, or their processes. Simplicity enables locality. Locality in code is what keeps systems loosely coupled, enabling delivery of features faster. Locality in organizations allows teams to make decisions without having to communicate and coordinate with people outside the team.

THE SECOND IDEAL: Focus, Flow, and Joy. How does your daily work feel? Is your work marked by boredom and waiting for other people to get things done on your behalf? Do you work on small pieces of the whole without seeing the bigger picture? Daily work should be completed in small batches, ideally single-piece flow, getting fast and continual feedback on your work. These conditions allow for focus and flow, challenge, learning, discovery, mastering our domain and even joy.

THE THIRD IDEAL: Improvement of Daily Work. Problems must be solved on the front lines, where daily work is performed by the world's foremost experts who confront those problems most often. This dynamic allows employees to change and improve how they work, informed by learning. When anyone encounters a problem, everyone is expected to ask for help at any time, even if it means stopping the entire assembly line. As a result, problems are quickly seen, swarmed, and solved and then learning spreads far and wide, to benefit all, enabling innovation, excellence, and outlearning the competition.

THE FOURTH IDEAL: Psychological Safety. An innovative culture should be free of fear. Employees should be able to speak about problems with honesty; honesty requires the absence of fear. No one will take risks, experiment, or innovate in a culture of fear, where people are afraid to tell the boss bad news. When something goes wrong, organizations should ask "what" caused the problem not "who" and commit to doing what it takes to make tomorrow better than today.

Imagine a world where you can make decades of technical debt disappear ... what it could feel like to deliberately and carefully choose what to leave behind and where you could spend your time and energy instead.

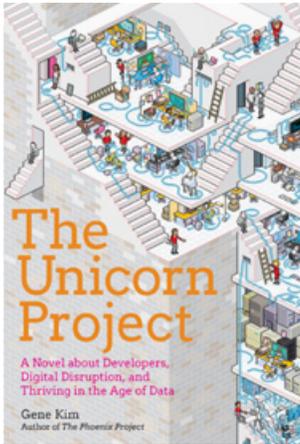
THE FIFTH IDEAL: Focus on the Customer. Ruthlessly question whether something actually matters to your customers: Are they willing to pay us for it or is it only of value to our functional silo? Ask whether a daily action truly improves the lives of customers, creates value for them, and whether they would pay for it. Are you truly striving for what is best for them or is it about more parochial goals that they don't care about, such as internal plans of record or how your functional silos are measured.

Lastly, when restructuring, many companies look to eliminate people and their salaries rather than technical debt and outdated processes. But think carefully about how each and every position you eliminate might disrupt flow, especially when you don't have locality in decision-making, as embodied by the First Ideal. For instance, what happens when you get rid of managers? Those middle managers are often your interface between strategy and execution. They are your prioritizers and your traffic cops. We all have this ideal of small teams working independently, but who manages the teams of teams? It's your middle managers. Some call them derisively the 'frozen middle,' but you'll find that properly developing this layer of people is critical to execute strategy.

When it comes to deciding on whether you should keep people or a complex technical process to manage an organization, always bet on people. 📌



Info



Ready to dig deeper into this idea?
Buy a copy of
[The Unicorn Project.](#)

Want copies for your organization or for an event? We can help:
customerservice@porchlightbooks.com 800-236-7323

ABOUT THE AUTHOR

Gene Kim is a multi-award winning CTO, researcher, and author. He was the founder of Tripwire and served as CTO for thirteen years. His books include the bestselling *The Phoenix Project* (2013) and *The DevOps Handbook* (2016), the *The Unicorn Project*, as well as *Beyond the Phoenix Project*, Shingo Publication Award-winning *Accelerate*, *The Visible Ops Handbook* (2004), and *Visible Ops Security* (2006). In 2007, *Computerworld* added Gene to the "40 Innovative IT People to Watch Under the Age of 40" list, and he was named a Computer Science Outstanding Alumnus by Purdue University for achievement and leadership in the profession.

Follow Gene at realgenekim.me or on Twitter [@RealGeneKim](https://twitter.com/RealGeneKim).



Porchlight

Curated and edited by the people of Porchlight, ChangeThis is a vehicle for big ideas to spread.

Keep up with the latest book releases and ideas at porchlightbooks.com.

This document was created on December 4, 2019 and is based on the best information available at that time.

The copyright of this work belongs to the author, who is solely responsible for the content. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License. To view a copy of this license, visit Creative Commons. Cover image from Adobe Stock.

SHARE THIS

Pass along a copy of this manifesto to others.



SUBSCRIBE

Sign up for e-news to learn when our latest manifestos are available.

