armory

SURVIVE OR THRIVE

Thriving in a Software-Centric World with Continuous Delivery

1 Introduction: The 3rd Industrial Revolution, a Software-Centric World

Most people who have taken middle school history can reel off basic facts about the first and second industrial revolutions, but few are familiar with the third. Introducing new manufacturing processes, mass production, and factories, the first revolution (mid-18th century) transformed society from largely rural and agrarian into a hybrid that included urban centers. The second revolution (mid-19th century) extended the first with rapid advances in materials, transportation, and communication, enabling production and industrialization at larger scale and higher speed. Recently we've entered the third industrial revolution, one that powers—and is powered by—an assembly line of ideas, rather than a physical assembly line. The hallmark of this third, digital revolution is a software-centric world where products and experiences are software-driven and woven into almost every part of human life. Companies must adapt both technologically and culturally in order to compete in this world. Those companies that can adapt successfully and compete will thrive. Those companies that can't compete may merely scrape by and survive—and some, ultimately, will not.

This eBook will examine the drivers of a software-centric world; what it takes to survive and thrive; how to mature your software delivery lifecycle (SDLC) and software management processes to deliver quality products in a software-centric world; and how automated software delivery tools like Spinnaker accelerate software maturity. You'll learn from real-world examples of companies that adapted and thrived. And, you'll learn why approaches like continuous software integration and delivery (CI/CD) enable organizations to compete in a software-centric world by safely increasing the velocity of innovation.

2 What is a Software-Centric World?

It is not a stretch to say that the world is driven by software: It's now stitched into almost every part of your daily life. For example, 95% of American consumers have used self-checkout kiosks in retail stores, and nearly 50% use them on a weekly basis. The in-store grocery shopping experience itself is rapidly being replaced by delivery applications like Instacart and others. It's estimated that this year 20% of American adults will use grocery delivery applications, representing a 25% year-over-year growth. Unsurprisingly, the grocery delivery business is a big one: Instacart is valued at $8 billion ahead of its IPO.

As consumers interact with software throughout the day—often without realizing it—companies must focus on building innovative applications that provide a seamless, outstanding user experience. The shift to a software-centric world brings opportunities for new business models for companies, as long as they can rapidly respond to market demands and trends. This requires a paradigm shift in how companies perceive themselves. They can no longer be companies that use software to do X. They must become software companies that do X. Software must be at the center of an organization's identity, not merely a capability.

Thriving in a software-centric world requires a paradigm shift. You can no longer be a company that uses software to build widgets. You must now *be a software company* that builds widgets.

Adding to this fundamental shift is the reality that software is underpinning the economy, and in some ways is poised to take over in places where hardware previously ruled. Intelligent, nimble software design and management is enabling this. Consider the Tesla Model S. "The behavior of the car six months from now could be radically different because software can reshape the capability of the hardware continuously, exceeding the speed of customer demand." The power of Tesla's software far exceeds the power of its physical manufacturing. Another example is in comparing 4G to 5G technologies in mobile communications. In LTE (4G), specialized hardware performs the specific functions required to power mobile networks. As telecommunications companies shift to 5G, those functions have been virtualized in software, and generic hardware can be deployed across their networks. Sprint, Verizon, AT&T, and others will have to become experts at writing and deploying software at scale.

3 Which Companies Succeed in a Software-Centric World?

It's clear that the ability to provide a robust software experience is critical in order to survive and thrive as an organization. What kinds of companies can provide that experience and succeed in a software-centric world? There appear to be two kinds: companies that began life as software companies, and those that successfully shifted to become software companies.

Companies that Begin Life as Software Companies

Many of the companies that dominate our lives—and the stock market—today were founded as software companies. They are market leaders because of their unwavering focus on delivering robust applications. These companies are built upon digital infrastructure that enables users to access products or services in new ways. Some have disrupted established industries by approaching consumer needs with unique software solutions.

In this section, we'll explore a series of software companies that have several things in common. These companies:

- Replace traditional product development with agile innovation.
- Quickly bring basic products to market and continuously improve upon them with customer feedback.
- Iterate quickly and have short feedback loops.
- Place customers at the center of all development considerations.

Media Consumption

Netflix is frequently the first name that comes to mind when considering a thriving media software company. Consumers have shifted media consumption from one model (renting videos from Blockbuster) to another (streaming content from many providers, including Netflix). Not everyone remembers Netflix's original business model: delivering DVDs by postal mail. Then-CEO Reed Hastings "saw that postage rates were going to keep going up and the internet was going to get twice as fast at half the price every 18 months. At some point, those lines would cross, and it would become more cost-efficient to stream a movie rather than to mail a video." Hastings saw those statistics, watched the success of streaming-music startup Napster, and did the calculations. Because Netflix is first and foremost a software company, it was able to rapidly pivot from its original business model and focus on delivering streaming content. Today, Netflix has a market capitalization of around $160 billion, and almost 150 million streaming subscribers.

Transportation

The most notable feature of the thriving software companies in this market sector is that they don't actually own any vehicles. Companies like Uber and Lyft instead created software platforms to connect riders and drivers, becoming some of the highest-valued companies in the world. As Fast Company describes it: "The real innovation that Uber and Lyft have brought to bear is in the transformation of the user experience of your ride: the ability to gauge your driver's distance from you; the presentment of the driver's name and the make and model of his or her car; the option to follow along with the route to your destination; and then the prompt to rate and review your ride at the end. These are the kinds of things that make an Uber or Lyft ride fundamentally different from stepping off the curb and waving down a taxi. They rely on technology, of course, but really, they amount to 'designing the ride,' or the application of user-centric thinking to enhance the experience of being driven across town." Both companies have created software that dramatically increased the size of the transportation market itself. It is estimated that half of the rides taken would have otherwise been taken by foot, bike, public transport, or perhaps not even at all. The software platforms designed by these companies enlarged the taxi market by an estimated 2-3x. Uber is valued at $100 billion, and Lyft at $17 billion.

Accommodation

Another frequently referenced software company that disrupted an entire industry is Airbnb. The company has a massive market valuation in the accommodation industry, despite not owning any actual hotels—just as Uber and Lyft have large valuations in the transportation industries despite not owning any fleet vehicles. Like those companies, Airbnb created a software platform that enabled customers to interact with service providers in a new and transformative way. And, because of the flexibility enabled by being a software company, Airbnb has been able to shift and add to its business model as the market changes around it. Initially offering only a mattress or spare bedrooms, Airbnb quickly evolved to offer entire homes, guest/host messaging, and reviews. It most recently added guest "experiences" in addition to accommodations. Airbnb is valued at $31 billion ahead of its impending IPO.

Companies that Successfully Shift from 'Analogue' to 'Digital'

The second group of companies that are likely to thrive in the software-centric world are those that successfully shift from an analogue to a digital mindset. These are companies that began life not as software companies, but as companies that used software to accomplish their initial business objectives. They saw the digital revolution coming and realized they would need to become software companies in order to stay relevant and compete in the marketplace. McKinsey says of these companies: "They didn't have digital in their DNA: Companies have to adapt their business model to the new opportunities, and the mindset of their employees. Companies that hail from the analogue world have a long path ahead of them—that of technology-enabled transformation." In this section we'll examine two companies that have shifted successfully to thrive in the software-centric economy: John Deere and Capital One.

John Deere

Founded in 1837—just after the first industrial revolution—John Deere is a manufacturer of agricultural, construction, and forestry machinery, and today is also a highly innovative software company. It has kept pace with waves of progress since its founding: fertilizers, pesticides, and genetics in the 20th century, and digital innovation in this century. Embracing software has led the company to what it calls "precision

agriculture." Leveraging a combination of acquisitions and internal development, John Deere has built the John Deere Operations Center – a centralized cloud where customers can upload and access all critical data from mobile devices. Analyzing the uploaded data, the company helps farmers spread seed and fertilizer on fields in an optimal manner, significantly increasing the yield per square meter. In addition, the data analysis saves fuel, reduces repair times, and makes optimal use of the vehicle fleet. The on-site data is transmitted by sensors in farmer's vehicles to the Deere data center. Its customers have seen measurable results, including a 95% reduction of chemical use, which improves yield and reduces costs to growers. John Deere continues to invest in John Deere Labs, focusing on software innovation, machine learning, and artificial intelligence to stay competitive.

Capital One

Software is everywhere, and your phone is the new bank branch, with always-on connectivity to financial services. Banks and financial institutions have had to rapidly adapt in order to thrive in the software-centric world. Banking analyst Jeff Kagan says that to thrive, financial companies have to think like tech companies—very company that wants to be a winner going forward must think this way. Capital One stands out as a winner in this imperative by transforming itself into a software company that delivers digital banking to customers.

Capital One CIO Rob Alexander explains that instead of simply becoming a more innovative bank, he transformed his organization "from an IT shop in a bank to a technology company." This enabled Capital One to turn out quick updates to its applications and services and add new features with the speed and agility required by its customers. The company achieved this by reorganizing itself as a "high-productivity software engineering organization" and committed to both a fundamental technological and cultural transformation.

Capital One adopted Agile development, invested in mobile applications, open source technologies, artificial intelligence, and machine learning. It also migrated legacy applications to the public cloud and today builds all new applications on AWS. Capital One decided to stop managing its own infrastructure, instead focusing on software innovation. The company also evolved its cultural identity to a definitively software-based organization. Less than 10 years ago, the company had 2,500 people in IT and fewer than half of them were in engineering roles. Now there are 9,000 people in IT, and 85% are in engineering.

There is a clear benefit for Capital One in the wake of its decision to shift from a financial company that used software to deliver financial services to a software company that delivers financial services. Its most recent net margin reports showed nearly 10% higher profitability than industry competitors, and it is recognized as a leader in innovation.

"Financial companies have to think like tech companies: Every company that wants to be a winner going forward must think this way"

-Jeff Kagan

4 What is Software Delivery Maturity?

It is inspiring to read about successful companies that have been inherently software-focused since their founding, or companies that have successfully migrated from analog origins to digital rebirths. But it's challenging to determine how to kickstart your organization to a state of consistent high velocity, high quality, innovative software delivery. As software becomes a core competency and a key competitive differentiator, your software release process must focus on rapid and reliable releases. This is especially true as organizations face common challenges in delivering software, which includes embracing open source technologies and migrating software delivery targets to public, private, or hybrid cloud environments.

In the face of these drivers and challenges, many companies are choosing to adopt a continuous integration/continuous delivery (CI/CD) model. This enables organizations to standardize on a set of best practices for software delivery, maintaining a high degree of innovation and a rapid release cadence while ensuring reliability and safety.

The software delivery maturity framework developed by Armory—analyzed in the next section—helps companies assess where they are in the software development lifecycle (SDLC). It also examines what tools and technologies can help propel development organizations toward greater CI/CD maturity, and ultimately toward the ability to compete and thrive in today's software-centric market.

In the following sections, we'll look more closely at what software delivery maturity means to organizations, how to evolve and mature your organization's software maturity, and how CI/CD tools like Spinnaker can assist in that process.

5 Stages of Software Delivery Maturity: From Surviving to Thriving

DATA CENTER DEPLOYMENTS	HYBRID CLOUD ADOPTION	MULTICLOUD GOLDEN PATH TO PROD	CONTINUOUS DELIVERY ADOPTION	SOFTWARE DELIVERY AUTOMATION
Bare Metal or VMs Mutable Deployments SSH into Prod Deployments = Events Manual & Error Prone Dev vs. Ops	Data Centers + Lift & Shift No Standard Path to Prod No Global Compliance Complicated Rollbacks No Service Ownership Inconsistent Deploys SLA Failures	Kubernetes in Data Centers Pipelines as Code Global Compliance Policies Dedicated DevOps Immutable Deploys Confident Rollbacks Manual Judgments Strong Integration Test Coverage	Deploy Continuously in Background Full Embrace of DevOps Culture Monolith Apps into Microservices App Teams Fully Self-Service All Teams Deploy with Same Platform Manual Canaries	Automated Canaries Automated Rollbacks Machine Learning Powered Anomaly Detection SLA Transparency on Per-App Basis Chaos Engineering Automated Dependency Analysis Feature Flagging Value Stream Map Developer Analytics
Frequent Outages 20+ Manual Steps Weeks/Months to Deploy 1-2 Deployments/Month	Some Outages 10+ Manual Steps Days/Weeks to Deploy 2-10 Deployments/Month	Few Outages 1 to 3 Manual Steps Hours to Deploy 10-20 Deployments/Month	Minimal Outages 0 Manual Steps Minutes to Deploy 100+ Deployments/Month	Rare Outages 0 Manual Steps Minutes to Deploy 1000+ Deployments/Month
Late Majority	Early Majority	Early Adopters	Early Adopters	Innovators

Software Delivery Maturity (Stage 1 and 2)

Companies at these early stages in the SDLC still use software to do X. They are not yet inherently software companies. These companies likely still house applications in data centers, although some may be beginning to shift monolithic apps to the cloud. They are more often than not relying on mutable infrastructure. Development and operations organizations are still largely bifurcated internally: applications developers write software and pass it off to operations team members, who are tasked with deploying the software. The path to deployment is manual, with many steps and long deployment cycles. Rollbacks are complicated.

For these companies, the path(s) to production are complicated and messy. Compliance and security policies, business processes, and infrastructure are tangled together with staging, testing, and production environments. With so many manual steps, repeatable deployments are difficult. Because the process is complex and there is little transparency into the many moving pieces, it is difficult to accelerate deployment and innovation in this environment.

For companies at this early stage of software maturity, the manual deployment process and the long deployment cycle make innovation extremely hard to achieve and deliver.

Read more about the benefits of immutable infrastructure here.

Handling outages in the early stages of software maturity.

Updates cause 70% of outages in production. Because companies in Stages 1 and 2 don't have automated rollbacks, deployment must be halted. In response, team members introduce many more steps to ensure safety in future deployments, including integration and unit testing, manual quality assurance, and manual judgments (including some requiring manager approval). This further slows the deployment process, in turn reducing the total number of deployments. This is an extremely common scenario and one that causes many companies to miss service level agreements (SLAs) and deployment goals. It also impedes innovation.

STAGE 3

Software Delivery Maturity Stage 3

Companies at this stage of the software maturity model are starting to adopt continuous integration. They are likely breaking monolithic applications into microservices, deploying to multicloud targets, adopting Kubernetes, and embracing immutable infrastructure. As they begin to adopt CI tools like Spinnaker, the number of manual steps required to deploy are reduced from around 20 to 3, and the time to deployment is reduced from days or weeks to hours. Companies can more than double deployments, from 2–10 times per month to 10–20 times per month. And, they can roll back when

needed. These companies are also likely to have begun the shift to a DevOps culture: The same engineers who write software are deploying that code into production, ensuring ownership along the delivery lifecycle and removing some of the traditional silos that slow innovation and delivery in earlier stage, less mature companies.

> This is a critical tipping point in software maturity. Companies can choose to tread water at this stage and remain static with CI. They may not experience growth or rapid innovation, but they will benefit from an increased number of deployments and the ability to roll back when needed. However, if organizations coalesce around CI and choose to continue on the path toward fully automated continuous delivery, the pace of innovation will increase, and they can focus on truly becoming software companies. This is the pivot point in software maturity between merely surviving, and truly thriving.

STAGE 4

Software Delivery Maturity Stage 4

Companies that have made it to this stage have committed to continuous delivery. Applications have been broken into microservices, and platforms have been standardized. Software is starting to be deployed continuously, like running water—there are no manual steps, deployments take minutes, and it's easy to deploy ~100 times/month or more. This makes it easy to iterate quickly, respond to customer requirements, and rapidly deliver new features or products to the marketplace. These companies are starting to thrive.

STAGE 5

Software Delivery Maturity Stage 5

Companies at this stage are innovators. They have completely automated their software delivery. There are no manual steps, deployment is continuous and ongoing, and they safely and reliably have thousands of releases each month. Netflix releases code more than 4,000 times daily! That release cadence doubled after Netflix built and implemented Spinnaker. Companies with automated CD also have completely integrated DevOps cultures, with engineers owning applications end-to-end. Companies

at this stage identify wholly as software companies, building complex software with the goal of creating software-first customer experiences. It should come as no surprise that companies known for software maturity and practicing continuous delivery are thriving: Netflix, Google, and Amazon among them.

Do more microservices mean more complexity?

As you move through the software maturity model, you begin to ship smaller diffs and break your applications into more microservices. At first glance, it may seem like you are adding complexity. In reality, you are adding more safety and efficiency. Why?

- You can add a feature to one component without affecting another.
- The SDLC cost-per-feature is reduced.
- It's much more scalable, with easier rollbacks.

Don't be afraid of microservices!

6 Why Software Delivery Maturity Matters

As the market valuation of software-born companies like Uber or Netflix show, delivering software innovation pays off. The successes of John Deere and Capital One show that embracing a digital transformation can revitalize companies moving into new markets or product lines. Moving up the software maturity framework enables companies to experiment faster and deliver more innovation by iterating on products. A recent McKinsey report underscores that accelerating cycle time is the key to competing in a digital world.

Like John Deere or Capital One, moving from manual software delivery to automated continuous delivery enables inherently analogue companies to become software companies. This shift drives engagement and customer feedback. And delivering more code, more efficiently, pays off. Automation and software maturity decrease the error surface and increase innovation rates. High-performing companies:

- Deploy code 200+ more frequently
- Move 100+ times faster from commit to deploy
- Recover from incidents 2600+ times faster
- Have a 7x lower change failure rate

For the ability to innovate, deliver to customers, and iterate based on feedback, software competency and maturity is the distinction between companies that will simply survive and those that will thrive.

"Accelerating cycle time is the key to competing in a digital world."
-McKinsey

7 How to Move Along the Software Delivery Maturity Model

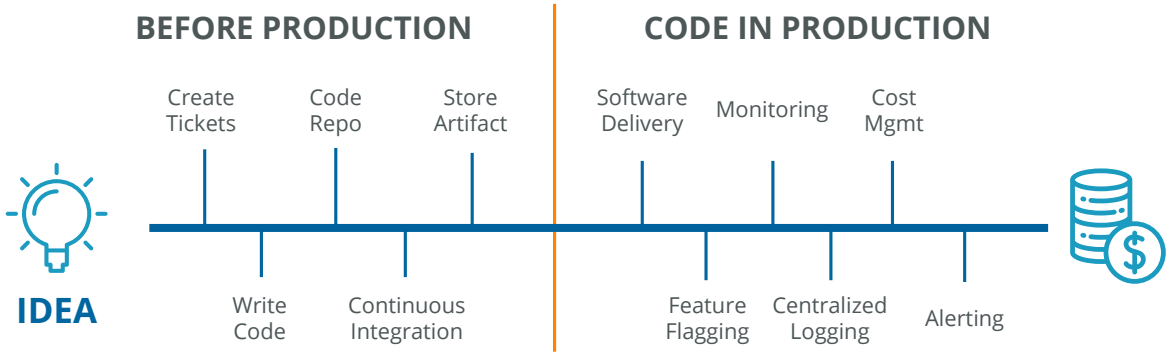
If you want to move your organization along the software maturity framework and compete in your market as dominantly as the "thrivers" in Stage 5, it is critical to understand some fundamental truths about the SDLC and the current state of software delivery.

How to Move Along the Maturity Model

TRUTH #1

You Must Measure and Manage Risk in Your SDLC

Imagine being responsible for a physical manufacturing plant with completely opaque windows. You can't see the actual assembly line or anyone working on it. You're asked to reduce the number of steps to increase efficiency, or to make the assembly line safer. It's an impossible task: You can't measure and improve what you can't see.



Your SDLC is exactly the same. Taking an idea to production is a complex digital assembly line. Until you can clearly understand and measure it, you can't make it more efficient or reduce the risks, increasing your efficiency and delivery velocity in the process. There's a great deal of complexity in the SDLC: many steps, vendors, data pools, risks, and many individuals and teams making contributions. It's vital you take the time to truly instrument your SDLC.

What can you measure in the SDLC? Here's just a small sample:

- Service-level agreements (SLAs)
- Service-level objectives (SLOs)
- Service-level indicators (SLIs)
- Time to deploy (TTD)
- Number of outages
- Number of steps to deployment in production
- Number of deployments (per day/week/month)
- Number of rollbacks
- Time to onboard new engineers
- Time to roll out new services
- Engineering time to patch

The more you measure, the more you know. If you spend the time to quantify your SDLC, you are in a better position to understand what can be automated and where it lives. You can measure delivery velocity and reliability before and after adopting the automated processes of CI/CD, to document the associated efficiencies and ROI. With 70% of outages caused by updates, automating software delivery reduces your error surface substantially.

Companies at Stages 4 and 5 in the maturity framework can easily quantify both their infrastructure and the steps in their SDLC. This enables them to point to the benefits of moving to continuous delivery, including the time efficiencies in deployment, the reduced number of steps and errors, and the ROI and cost savings.

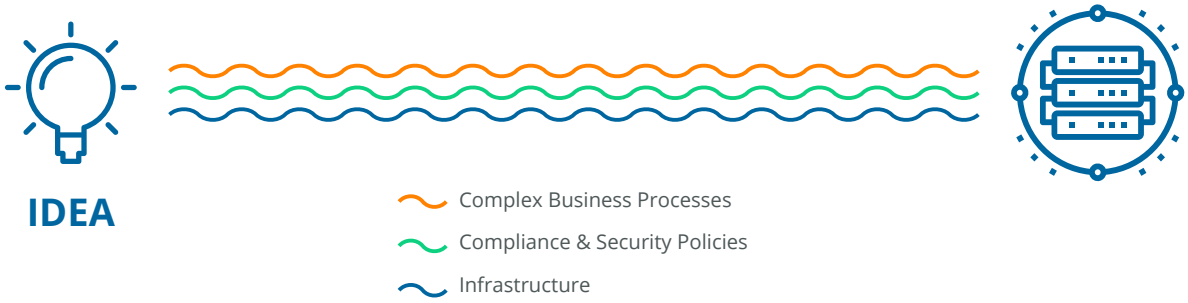
TRUTH #2

You Must Understand the Complexities of Delivering to Cloud Infrastructure

The line from idea to deployment isn't straight. It's a tangled web of complex business processes, compliance and security policies, and infrastructure, including environments, configuration, and management. This web continues to grow messier for a variety of reasons:

- There are multiple paths to multiple deployment targets.
- Monolithic apps are being broken into multiple microservices.

- Infrastructure is continuing to fragment.
- You have to rebuild for each stage (development/staging/production, quality assurance, manual judgment), and for each delivery environment.



The complexity will only increase as innovations and expansions in public cloud infrastructure grow. Delivering to AWS isn't as simple as "delivering to AWS." It can include delivering to EC2, ECS, EKS, Lambda, and Fargate. In an environment as fragmented as this, companies need standardized deployment paths that enable them to confidently, safely, and rapidly deliver software without sacrificing innovation.

By moving to continuous delivery, companies in Stages 4 and 5 have automated paths to production and have codified their deployment pipelines and infrastructure management. Deployments are easy, repeatable, and safe. This makes it easier for teams across the organization to roll out new features and innovate. And, these organizations can continue to innovate in tandem with the public cloud providers without fear of increasing complexity in deployment targets.

TRUTH #3

You Must Approach DevOps as a Cultural – Not a Tactical – Imperative

Achieving software maturity to compete in a software-centric world requires a combination of technological and cultural adjustments. In Stages 4 and 5 of the software maturity framework, a fully integrated DevOps culture is key to the success of automated continuous delivery. Armory's CEO defines DevOps as "getting really good at moving through the SDLC with more safety, velocity, and innovation, where everyone rallies around a joint understanding of the importance of software not just to survive, but to thrive." To fully embrace the third revolution's "assembly line of ideas," you need an idea-centric organization that supports the technologies required to ship innovative software rapidly.

How to deploy the best ideas, at scale:

- Conceptualize an idea and ship it as software with safety and velocity.
- Iterate on the idea and resulting product with fast feedback loops.
- Mechanize and automate the delivery process for efficiency and safety.

The fundamental misalignment between development and operations is well known. Development teams want to move fast, break things, and make as many changes as possible to get the best software product. Operations teams want stability, safety, and a static environment in production to avoid utages. There is an inherent cultural tension that must be overcome in order for software to unlock enterprise value. It can't become a sticking point.

There are other factors that contribute to the challenge also. As noted earlier, there are many components and risk factors in the SDLC. Multiple internal groups own their systems of record, and don't cherish giving up control. There is frequently a lack of data transparency or process transparency between these silos. To solve the inherent tensions between the goals of development and operations, organizations that want to become software-centric need to fundamentally rethink how DevOps is approached.

While DevOps in software delivery can be approached tactically—engineers owning their code end-to-end—the successful and mature software companies that approach DevOps strategically and truly embrace it culturally, are the ones that thrive. DevOps has to be seen as a cultural imperative and not simply as a way to "do" software delivery. Companies in Stages 4 and 5 of the software maturity framework embody this cultural imperative. They deploy innovative ideas at scale with no silos between development and operations.

8 How software-centric organizations thrive with Spinnaker

As companies move toward continuous integration and delivery, many are choosing to adopt Spinnaker as the continuous delivery platform of record. Spinnaker is an open source, multicloud continuous delivery platform for releasing software changes with high velocity and confidence. Spinnaker was originally built at Netflix and was released to the open source community in 2015. A robust and engaged community has since built up around it.

Spinnaker enables customers to:

- Automate software delivery.
- Remove manual, error-prone, and time-consuming deployment processes.
- Enable speed, iteration, safety, repeatability, and a focus on innovation and software quality, instead of just on software delivery.
- Introduce economic efficiencies and better regulatory compliance.

Spinnaker delivers these advantages by providing:

- An internal platform for release, delivery, and deployment standardization.
- Easy management across hybrid cloud platforms, with deployment to all providers in the same "golden" pipeline.
- A common delivery mechanism.
- Simple, repeatable deployments.
- Automated cloud scaling.
- Single-pane-of-glass view.
- An extensible architecture that hooks into external tools and platforms (e.g., events, notification, pipeline orchestrators, authentication, and authorization).
- Zero application downtime during deployments.
- Safe and easy rollbacks.
- Microservices deployed to optimal targets without brittle, scripted paths to production.
- Automated testing and canary analysis, enabling you to move fast without breaking things.
- Guardrails: A way to empower developers to move quickly and innovate, while simultaneously ensuring that the controls and restrictions that protect organizational best practices are followed.

Migrating to Spinnaker enables organizations to address each of the "Maturity Model Truths" from the previous section. This accelerates and simplifies the SDLC, moving companies toward greater delivery maturity.

How does Spinnaker help companies address?

TRUTH #1

Measure and Manage Risk in Your SDLC

It's vital to quantify your existing software delivery environment to understand the efficiency, reliability, and velocity benefits Spinnaker can deliver. A company recently worked with Armory to migrate its software delivery to Spinnaker and took the time to understand several variables in its SDLC. These metrics included the number of manual steps to deploy a single service to production, the average engineering TTD, onboarding time, time to patch, and other metrics. By continuing to measure the same things during the maturation from CI to CD with Spinnaker, the company had a compelling ROI to move all of its applications to automated delivery.

Metric	Explanation	Before Spinnaker (avg)	After Spinnaker (avg)
Number of steps to deploy service into productions	The number or manual steps, including manual validation, in all staging environments through production. (Each manual step introduces the potential for human error, which was responsible for many deployment failures and service outages.)	25 steps	1 step
Engineering TTD	The amount of time for an engineer to deploy from staging through to production	60 minutes	<1 minutes
Automation TTD	The average amount of time to takes to deploy regardless of engineer or service	60 minutes	31 minutes
Onboarding time	The time it takes for new engineers to be comfortable deploying code to production	3+ days	30 minutes
Engineering time to patch	Time to patch	5+ days	0 (automated patching)

Clear visibility into your SDLC quantifies what areas need improvement and provides continuous justification for the benefits of automation as you mature software delivery in the organization.

TRUTH #2

Understand the Complexities of Delivering to Cloud Infrastructure

Spinnaker directly addresses these complexities. Without Spinnaker, the complex path to multiple deployment targets can be visualized like this:

With Spinnaker, that path is radically simplified:

Spinnaker disentangles the messy path to production posed by compliance and security policies, business processes, and staging/testing/production environments and infrastructure, enabling companies to accelerate toward delivery maturity. It provides a mapping layer between your DevOps team and deployment targets. And, by abstracting business and technical processes, compliance, and

security policies from infrastructure choices, it gives companies access to the underlying innovation in cloud deployment targets.

Spinnaker enables customers to deploy workloads to optimal targets, such as GDPR compliant targets, the best AI-compute targets, or the most economic targets. The best engineers from the top cloud providers are building paths into their clouds: AWS engineers are building the Lambda integration, Pivotal engineers are building the PCF integration, and Google engineers are building the k8s and GCP drivers, for example. The involvement of engineers from cloud providers enables users to leverage innovation happening in the cloud and simplify deployments despite the increasing fragmentation.

TRUTH #3

Approach DevOps as a Cultural – Not a Tactical – Imperative

Spinnaker lays the groundwork for companies to migrate from siloed development and operations functions to a truly integrated DevOps functionality that automates software delivery. Companies that have successfully embraced DevOps with Spinnaker have removed manual and error-prone deployment processes in favor of a standardized platform for release, delivery, and deployment to all targets in the same golden pipeline. Guard rails enable teams to move quickly while ensuring that best practices are followed.

Deploying Spinnaker enables companies to rapidly migrate from Stages 1, 2, and 3 of the maturity framework toward truly continuous automated delivery maturity. Ideas can be translated into software and delivered to the best targets with both velocity and confidence. Companies looking to compete in a software-centric world can confidently innovate at scale.

9 Why Armory?

Armory has more experience installing Spinnaker in complex enterprise environments than any other company. We become the expert team member who does the Spinnaker infrastructure work for you. We abstract away the complexities to get you moving as quickly as possible, letting you use Spinnaker without having to be an expert on installation, upgrades, monitoring, and maintenance. We are active members of the OSS community and are always on the lookout for (and contributing to) best practices for Spinnaker use. We also empower the rest of your team with 24/7 support, services, and training, as well as access to our SAs.

In addition to our services and support, our product is enterprise-grade, scalable, and packed with features that help you use Spinnaker at scale, including:

Managed Spinnaker

- A fully-managed instance of Spinnaker, installed in your cluster
- Armory manages Spinnaker so you don't have to, with 24/7 expert support and a 99.95% uptime SLA

Armory Terraform Integration

- Seamless integration of Spinnaker with Hashicorp Terraform
- Run Terraform plan / apply / destroy / output stages as part of your Spinnaker pipeline to fully-align your infrastructure with your deployments

Armory Pipeline-as-Code

- Dynamically create and store Spinnaker pipelines (in HCL or YAML) in a Github repository
- Enables a) version control, templates, and modularization in your pipelines, b) repeatable, scaleable pipelines by avoiding the need to use the Spinnaker UI, and c) sophisticated features such as conditional logic, multiple repo / multiple branch support, Slack notifications, and more

Armory Policy Engine

- Set up and enforce pipeline compliance rules across your entire enterprise
- Key DevSecOps feature, enabling fine-grained access control and ensuring proper compliance and configuration throughout the deployment pipeline

Armory Account Management API

- Quickly and efficiently add new accounts to Spinnaker and make them available to your users as soon as they're provisioned
- Replace manual account configuration with a simple API call that gets picked up by Clouddriver automatically

Armory DevOps Insights

- Log and display your Spinnaker data exhaust in dashboards for at-a-glance insight into your SDLC
- Gain actionable insights into your SDLC value stream and problems affecting lead time (ie. SLO/SLI, slow stages, open PRs, manual judgements, pipeline failures etc.)

Armory Halyard: Secrets

- Encrypt secrets used within your Spinnaker configuration files
- Hashicorp Vault and encrypted S3-compatible buckets are supported as secrets engines

Armory Halyard: Installation & Release Management

- Install, configure, deploy and upgrade your Spinnaker instance with ease, speed, and confidence
- Guarantee a stable, reliable release process with a suite of hundreds of continuously-running integration and quality tests

If you are looking to deploy Spinnaker or move through the maturity model, Armory has many resources, including a comprehensive Blueprint for Spinnaker Adoption.

Contact Armory today for a complimentary assessment of your software delivery practices and learn more about how your organization can benefit from fast software delivery.

CONTACT US TO LEARN MORE

Armory Is Spinnaker at Enterprise Scale

Armory is the leader in enterprise-scale continuous software delivery. Our platform is built on Spinnaker, the de-facto standard in cloud-native software delivery developed and open-sourced by Netflix and Google, and battle-tested in production by the world's biggest brands. Armory brings the power of Spinnaker to your business, coupled with enterprise-grade availability, powerful feature extensions, and 24x7 expert support and services. Ship better software, faster, with Armory. Armory is headquartered in San Mateo, CA, and is backed by leading investors including Crosslink Capital, Bain Capital Ventures, YCombinator, and others.

armory

Armory Inc. 100 S. Ellsworth Ave., 9th floor, San Mateo, CA 94401
1-888-222-3370 | info@armory.io | www.armory.io