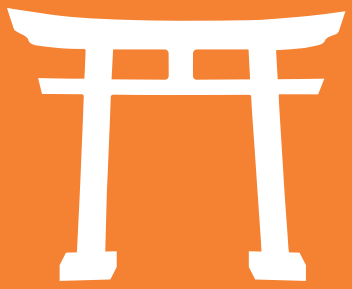
Getting Started with Dojos



A Dojo Handbook

IT REVOLUTION
DEVOPS ENTERPRISE FORUM
2019

25 NW 23rd Pl
Suite 6314
Portland, OR 97210

Getting Started with Dojos: A Dojo Handbook

Produced in the United States of America

Cover design and interior by Devon Smith

For further information about IT Revolution, these and other publications, special discounts for bulk book purchases, or for information on booking authors for an event, please visit our website at ITRevolution.com.

The 2019 DevOps Enterprise Forum was sponsored by XebiaLabs.

Preface

In May of this year, the fifth annual DevOps Enterprise Forum was held in Portland, Oregon. As always, industry leaders and experts came together to discuss the issues at the forefront of the DevOps Enterprise community and to put together guidance to help us overcome and move through those obstacles.

This year, the group took a deeper dive into issues we had just begun to unpack in previous years, providing step-by-step guidance on how to implement a move from project to product and how to make DevOps work in large-scale, cyber-physical systems, and even a more detailed look at conducting Dojos in any organization. We also approached cultural and process changes like breaking through old change-management processes and debunking the myth of the full-stack engineer. And of course, we dived into the continuing question around security in automated pipelines.

As always, this year's topics strive to address the issues, concerns, and obstacles that are the most relevant to modern IT organizations across all industries. Afterall, every organization is a digital organization.

This year's Forum papers (along with our archive of papers from years past) are an essential asset for any organization's library, fostering the continual learning that is essential to the success of a DevOps transformation and winning in the marketplace.

A special thanks goes to Jeff Gallimore, our co-host and partner and co-founder at Excella, for helping create a structure for the two days and the weeks that followed to help everyone stay focused and productive. Additional thanks goes to this year's Forum sponsor, XebiaLabs. And most importantly a huge thank you to this year's Forum participants, who contribute their valuable time and expertise and always go above and beyond to put together these resources for the entire community to share and learn from.

Please read, share, and learn, and you will help guide yourself and your organization to success!

—Gene Kim
June 2019
Portland, Oregon

Collaborators

Ross Clanton
Executive Director, Technology Modernization, Verizon

Jaclyn Damiano
Associate Director, Cloud Engineering & Enablement, Verizon

Carmen DeArdo
Senior VSM Strategist, Tasktop

John Esser
CEO, Veracity Solutions

Eric Passmore
Partner, Commerce and Consumer Office Convergence, Mircosoft

Introduction

The digital economy is here, and with it, organizations are facing an unparalleled rate of change. Enterprises are having trouble keeping up in order to sustain market relevancy. Technology is enabling new consumption patterns, and many enterprises are undertaking large-scale transformations to build the new capabilities they need in order to adapt to a changing world. Many of these transformations center around four key outcomes:

- **Operating model:** moving from projects to products
- **Culture:** enabling a highly-empowered and collaborative workforce
- **Technology practices:** growing Agile and DevOps ways of working
- **Technology architecture:** moving to modern platforms and technologies

What Is a "Dojo"?

Dojos can be a uniquely powerful vehicle for accelerating transformation across all four of these pillars. "Dojo" means "place of the way" in Japanese. The Dojo, in our context, is a space that is designed to host an immersive learning experience where full-stack teams come to learn modern engineering, product, and Agile practices.

Teams participate in Dojo experiences, typically referred to as a "challenge." In these challenges, team build real products and work from their actual backlogs with the entire process while being led by a Dojo coach. The work process within the Dojo is typically comprised of hyper sprints, a common pattern of two-and-a-half-day sprints over twelve sprint cycles (i.e., six weeks). During a team's time in the Dojo, they complete several sprints, and it's through this repetition

and increased learning cycle that the team builds an expertise of new skills. Having the cross-functional team working on a real problem while leveraging new technologies, architectures, and approaches to product development enables the team to master skills and work practices quickly.

What Is Not *a "Dojo"?*

A Dojo is not a way of accelerating delivery of existing projects. When a team enters the Dojo, their priority is to learn together and leverage new technology and new ways of working. Be wary of leaders who want to put their teams in a Dojo simply because they have aggressive delivery dates and they want to "speed their teams up." The reality is that you need to be willing to slow down in the short run in order to speed up in the long run. Teams must have the physical and mental space to learn. Organizations will benefit over the long run as teams become more productive and can deliver on increasingly complex tasks.

A Dojo is also not a purely Agile or Lean coaching practice. The focus should be on creating more effective teams, and that requires bringing together multiple practice areas. A Dojo works best when teams are hands on with both new technology and product development approaches.

Expected Outcomes of a Dojo

In order to compete in "The Age of Software,"[1] companies must have a continuous learning environment, which includes not only technology learning but, perhaps more importantly, improved ways of working, patterns, and practices. They also need to be effective at collaboration across teams and areas of expertise. This collaboration is vital as teams need to master new technologies, integrate them together, and provide business value at a continually accelerating pace.

As an example, consider developing a new mobile application where existing online services need to be extended to a larger customer base while having no downtime and be faster and more responsive. Having teams work together in an immersive space, specifically allocated to learning and improving, creates a safe and structured environment. This safe and structured environment allows teams to overcome the

cultural resistance to change. Target chief information officer (CIO) Mike McNamara states:

> **One vital part of Target's process was the creation of an accelerated learning environment our teams call "the Dojo." It's an immersive, six-week session where teams execute their normal work with Agile coaches on site to support them and provide anything they need from a DevOps point of view. The Dojo has been fantastic in getting teams engaged with Agile and DevOps, removing the natural resistance and fear of change, and then supporting the team through the changes while maintaining productivity. It's been a huge success for Target. And as we move through the journey, we continue to use the Dojo to refine, reinforce and strengthen our engineering capabilities.**[2]

Graduates of the Dojo become evangelists for modern practices and Dojo concepts, which can then help uplift other teams that they join and interact with. As other teams see examples of improved ways of working, they begin to change as well, accelerating the rate of change across the entire enterprise. Stories are powerful change agents!

In addition, the improved capabilities of these teams lead to accelerating the delivery of business value—particularly if teams are working in a product model.[3] This improves team happiness, which studies indicate is a key leading indicator of productivity.[4]

Ultimately, Dojos are instituted with the expectation that they will provide sufficient return on investment (ROI) by: accelerating or shortening transformation time, including scaling the transformation across the organization, and improving organizational learning and change, which is a key contributor to transformation efforts not meeting business expectations.

Dojos optimize for speed rather than efficiency or cost optimization, with the acknowledgement that business agility and high-value flow is ultimately worth the investment. Teams learn to not only focus on building things the right way through improved Agile and DevOps practices, they also learn to build the right things through the value orientation they master through product practices. In this scenario, speed comes from the fact that you are significantly reducing waste from your delivery process.

Most of the waste in IT delivery comes from the fact that teams spend time building things that don't add value. Huge improvements on time to value (TtV) occur when (1) you optimize effort around high-value work, (2) you learn to build and deliver in small increments, and (3) you automate your end-to-end delivery process. Generally, you can get even better outcomes on cost and efficiency as a byproduct when you choose to optimize for speed.

Maximizing the Power of a Dojo

The maximum power of the Dojo occurs when it becomes a nexus of learning and immersion for Agile principles, product development, and technology adoption. Building the right things in the right way ultimately creates the most value for an organization. If the Dojo can bring these three elements together, teams will maximize their potential to truly change and transform.

Agile

Agile is the process, method, and set of principles that forms the foundation of thinking and approach to delivery. Key Agile tenets—incremental delivery, continuous feedback and improvement, and working on high-value items—create a way of approaching software delivery that maximizes time to value.

Many teams fail to realize the full impact of a DevOps transformation because they have poor Agile implementation and practices from the outset. It may actually be a waterfall process wrapped in Agile terminology. Phrases such as "Scrum-But," "Water-Scrum," or "Scrum-Fall" are reflections of these suboptimal Agile implementations.

Ultimately, the Dojo should be an outstanding model and implementation of Agile. It should challenge and help teams make necessary adjustments to their own Agile implementation in order to maximize value delivery.

Product

There are two components that need to be addressed within the Dojo regarding the term "product"—product orientation and product management.

Product orientation refers to the need for a team coming into the Dojo to adopt, if they haven't already, a product orientation rather than a project orientation. A product orientation generally reflects a team that is funded long-term to solve, create, and build against a business need. The team creates value by continually addressing and delivering solutions to those needs.

Product management is the practice and discipline of managing and translating the business needs into that which can be developed and delivered by the product team.

Within the Dojo, both of these aspects must be reflected in the immersive experience. Teams should experience the Dojo in such a way that they emerge with a long-term perspective, strong identification, and ownership for what they deliver.

Technology

DevOps transformations often occur in concert with new technology adoption. For instance, many companies will pair DevOps adoption with their cloud-adoption efforts. The infrastructure agility of the cloud often reduces technology hurdles in DevOps tooling and system. At a more foundational level, DevOps often brings new tools and systems which need to be incorporated at the team level. Automated release management, artifact management, and application monitoring are all examples of new tools or technologies that teams must learn to use.

Again, as with Agile and product orientation, the Dojo must utilize and provide support and learning in the use of any technology key in the DevOps adoption.

How to Create a Dojo

Think of Dojos as your transformation immersion center. To build a case to invest in Dojos, focus on how you will use them to grow and accelerate your technology transformation. In doing this, Dojos are often run from a centralized organization within a company or division. Typically, this organization has insight to the strategic vision and goals of the engineering organization and remit to implement enterprise-wide change. The Dojo is intended as a shared, learning resource, so funding and support should come from this centralized group. Some companies have a charge-back model

they use to fund their Dojo. It should be noted that we're aware of at least one enterprise that is selling Dojo services to partially fund their own. That may be a model to be considered when funding Dojos.

The overwhelming sentiment of those who have set up Dojos is that the goal should be to invest in developing in-house space and staff, which will help to encourage excitement for and evangelism of the Dojo. Dojo success has been greatest when there are respected engineers at the center of the movement. Choose coaches who are seen as leaders in their peer group. This will help give the Dojo the initial strategic direction and credibility it needs to build a demand pipeline from other engineering teams.

Another option for Dojo creation—especially if the organization does not have the resources in space, people, and expertise—is to engage with an outside services or consulting firm that has experience in setting up and running Dojos. This may be a great option for getting a Dojo up and running quickly and to evaluate the ROI before the organization makes a long-term commitment.

Create a Space for Learning and Fun

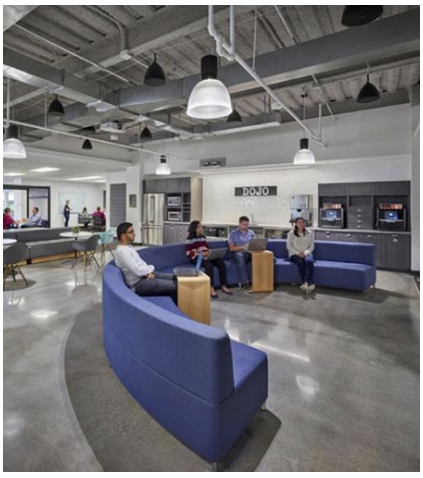


Figure 1: Verizon's New Jersey Dojo Lounge

Used with permission. Copyright Verizon

Figure 2: Target's Dojo

Used with permission. Copyright Target

Space can be a very powerful enabler for culture change and learning. When building a Dojo model, you will need to identify a space to house the Dojo. Try to put your Dojo in a high-traffic area so that you can broaden the exposure of what teams are experiencing. This helps create curiosity within the organization and in driving future demand for your Dojo services. The Dojo should be open, inspiring, and fun. See Figures 1 and 2 for examples. In order to accomplish this, one might consider creating a space with the following components:

Team Tables that Optimize Collaboration.

There is a decision to be made about how to fashion these tables. For instance, you can install individual desks grouped together that include dual monitors at each station. A company may also have a "family-style" table with no monitors, allowing for boundaryless communication. The most important factor to consider with the layout of the table is that everyone sits within close proximity to one another, hopefully allowing for in-person collaboration to become inevitable. A common practice is also to have a tech cart, or monitor, at the head of each team table for quicker sharing of screens and possible video connection to team members who are remote.

Whiteboards to Foster Collaboration and Visualization of Ideas

Whiteboards can be set up as temporary "borders" designating team spaces. Oftentimes, companies will designate a whiteboard that lists essential information about the team—their name, learning objectives, etc. This serves as a reminder for the team as to why they are there. It also helps orient other teams and visitors to the Dojo in understanding what the team is focused on.

Open Space to Enable Inter-Team Collaboration

Teams learning from others in the Dojo is a great way to get people out of their silos. Some of this learning will happen naturally as teams overhear other conversations that may be relevant to their own progress. Additionally, some of the learning will happen as people begin networking with other teams, as well as learning how they can help support one another and what further opportunities may exist.

A Communication and Collaboration Hub

These are typically referred to as demo lounges and are often the focal point for activities in the Dojo. A demo lounge creates a space for all teams in the Dojo to meet and share their work. This is another great way to get your stakeholders who aren't in the Dojo, such as management and business partners, to come in and participate in the team's learning and progress. To set up a demo lounge, you'll typically need a projector or large monitor surface, comfortable and informal seating, sound amplification, and a way to video conference with other Dojos or organizations. Demos held in the lounge are a fantastic way to gather different teams to share information and gain feedback that will help build better products. You can also leverage demo lounges to drive broader, community-engagement events. Hosting external meet-ups and fostering an internal community by running fun activities, such as movie days, can also bring people together.

Information Radiators

Transparency of the work happening and the principles of the Dojo are critical to fostering an overall cultural experience. Use information radiators, such as team-performance dashboards and signage that reflects Dojo principles (i.e., "Culture eats strategy for breakfast." —Peter Drucker), to help foster this. Though it may seem simple, it's important that the principles of the Dojo are made visible within the space. The constant reminder of how we should be working is important as people are attempting it differently, i.e., experiment, fail fast, slow down to speed up, etc.

Ensure the Space Fosters Fun and Creativity

Having activities, gadgets, and games around helps teams unplug, connect, and think. Additionally, having these in the Dojo also helps teams when they need to recharge or put some space between themselves and their work in order to reinforce productivity. Lastly, it serves as a way for people to get to know one another.

Staffing the Dojo

Dojo Coaches (Required)

Similar to a martial-arts Dojo that's overseen by a master, a transformation Dojo is headed by coaches who understand the big picture and assist the teams in "showing the way." Sometimes Dojo coaches may bring in assistance from other subject-matter experts, depending on the needs of the team and what they are trying to accomplish while in the Dojo. For example, user-experience (UX) designers and specialists may be brought into a Dojo if the team is designing a new, customer-facing front end to help the team learn more about user design. Depending on the learning goals set up for your Dojo, anywhere from one to a few coaches would likely be assigned for a "two-pizza" team, or seven to ten learners. The primary factor determining the need for multiple coaches working with a team will be the diversity or practice you are working on. Often, there's one coach who is more focused on product and Agile practices and another who's more focused on the technology practices, such as CI/CD and cloud-native development.

Coaches may be assigned to multiple teams, especially as those in the Dojo become more self-sufficient over time. One practice followed by some companies running Dojos is to have a coach assigned to two teams at a time, one team that's in the first three weeks of their challenge and another that's in the final three weeks. This works well as the demand on a coach's time is much greater at the beginning of a challenge when the teams are first learning the new ways of working.

You should staff your Dojo with coaches who have skills in more than one of these competencies. Even if they don't have the capabilities for all of these areas from when they start, you should strive to broaden their skills through pairing and cross-skilling among coaches. It's typically best to identify coaches who have a foundation as software engineers. Teams will be working through specific problems, and they need a coach who can be hands on and give the right technical direction to get them unstuck.

In addition, it's common for teams to leverage their Dojo experience to move them toward new platforms, tools, and technology stacks. In this scenario, they need a coach who can explain how to transition existing technology to new architectures. Having solid, technical work already established, many of these coaches can learn the necessary Agile and product practices to further broaden their skills. Even within the broad spectrum of technology practices, a single coach typically isn't going to have

expertise across all of them, so you will still require multiple coaches to build your full-stack coaching team.

Some of the common skills these coaches have are:

Agile Coach

An Agile coach should be skilled in multiple Agile practices and frameworks including scrum, kanban, and possibly scaling framework. Coaches should not be dogmatic about an Agile framework but should leverage mindsets and practices to help teams define the process that works best for their needs.

Product Coach

A product coach should be skilled in various product-discovery practices and techniques, as well as understanding how to validate ideas with customers to determine value. Product management skills are also critical for coaches as they will be guiding these teams on how to negotiate, prioritize, be a stakeholder, and influence the direction of the products being built. This includes understanding and valuing the different types of work that goes into building and running a product, such as features, defects, and tech debt.

Technology Coach

Across your Dojo team, it's important to have a broad set of technology skills including front-end development, back-end development, technology integration, and infrastructure and platforms. The technology coach should have knowledge of a broad set of development languages, especially those that are commonly used within your enterprise. Beyond these skills, there are common technology practices that you'll want to teach in your Dojos such as:

- **DevOps:** Leveraging CI/CD, configuration as code, test automation, and performance telemetry methods to operate and instrument product and service delivery.
- **Application Security (SecDevOps):** The practice of building secure code, automating validation of this code, and enabling accountability for security within product and service teams.

- **Cloud-Native Development:** Building loosely coupled, dynamic, and resilient applications; leveraging microservices; and learning and applying 12-factor principles.
- **Public Cloud:** Leveraging public cloud services to configure, build, and run application environments.
- **Integration:** Building APIs, consumable enterprise services, and integrations, as well as defining and guiding integration patterns, techniques, and technologies.

Subject-Matter Expert Coach (As Needed)

These may be experts in specific toolsets or platforms. Common examples would be architects, continuous delivery platform–engineers, kubernetes platform engineers, site reliability engineers, or similar.

While the subject-matter experts (SMEs) don't necessarily need to be constantly present in the Dojo, it is important for them to be accessible—either through close, physical proximity to the Dojo or through the appropriate ChatOps channels. Coaches typically only pull these individuals in when there are deep issues teams are grappling with that are beyond the coaching team's knowledge for that technology. There are benefits for these SMEs to participate in the Dojo as well, as it provides them direct customer exposure to the developers who are using their platforms and services, which helps them build more customer empathy and provide better solutions.

Dojo Operations Manager (As Needed)

There are a lot of moving parts to running a Dojo, and it's important to have someone who has a clear focus on all of the operational needs. Some of the common responsibilities of a Dojo Operations Manager are:

- Space planning and maintenance, as well as arranging for any fixes or reconfigurations that may be needed.
- Coach scheduling and ensuring the appropriate staffing is secured to handle Dojo demand.
- Arranging consults with teams interested in entering the Dojo.

- Managing the logistics for teams coming into the Dojo, including their onboarding, placement within the Dojo, and their exit.
- Managing the logistics of other stakeholders visiting the Dojo, such as tours to prospective customers or to external companies interested in learning about the Dojo.
- All communication-related activities around the Dojo including Dojo signage and potentially marketing.

Lead Coach (As Needed)

The primary responsibility for this position is to ensure the coaching staff is constantly teaching in order to meet the skill demands of the engineering teams. These coaches are generally quite broad in terms of experience and have significant knowledge in training teams. In addition to coaching the more complex teams, lead coaches will continuously evaluate the other coaches, as well as provide feedback and actionable ideas or plans to mitigate any skill gaps. This role is more important when starting up your Dojo and growing your coaching staff, and it is common to initially source this position from an outside consultant more skilled in Dojos.

Dojo Manager (As Needed)

This manager position is responsible for making a case for the Dojo, aligning it with the broader enterprise strategy and agenda of the technology organization, building and growing the Dojo team, and working with stakeholders to reach continuity with the value teams for their Dojo experience. Through heavy collaboration with stakeholders and the Dojo coaches, this position is generally responsible for establishing the best services, offerings, and practices to focus on in the Dojos based on the needs of the business.

Selecting Teams and Building a Demand Pipeline

Based on the goals for your Dojo, you should develop criteria around the selection of teams who will make good Dojo candidates.

Teams who generally benefit the most from the Dojo experience have these characteristics:

- Full-stack, multidisciplinary team; "two-pizza" team
- In-person (additional options for when this isn't feasible are discussed later in the paper)
- Are able to commit at least six hours a day, investment for four to eight consecutive weeks
- Committed to learning
- Combine Agile, product, and technology

When starting a Dojo program, it's important to identify and work with a few early-adopter teams who are highly motivated to improve their practices. Focus your energy on working with these teams to demonstrate early success that can be showcased across the organization. Additionally, take every opportunity to grow and develop powerful change agents who are passionate about the ways of the Dojo. These individuals will further promote the Dojo concept across the organization, even after they finish their Dojo experience, which helps drive future demand for Dojo services.

Enterprises have started Dojo programs using both "top-down" and "bottom-up" strategies to build a Dojo-demand pipeline. Organizations may also look to key metrics focused on delivery performance to indicate that a team is ready for a Dojo experience: baseline cycle time, team velocity, deployment frequency, and mean time to repair (MTTR). These can be good measures to start with baselining teams on their performance before demonstrating how they are able to continuously improve these measures as they progress through their Dojo experience—and after they leave the Dojo. As teams become more advanced, it's good to incorporate more value-oriented measures for the products that they are building as well, so they can ensure improved delivery performance is being channeled toward the best business and customer outcomes.

Readers should note, however, that using the Dojos as a "stick" to correct poor performance will not yield positive results. Dojos should be seen as an inspiring place to learn modern product and engineering practices, not as a source for teams to accelerate a near-term delivery goal. Dojos do not equal war rooms.

While Dojos typically have strong support from executives, sometimes there are challenges in obtaining alignment from middle management. Middle managers may have a difficult time prioritizing learning over executing, depending on the corpo-

rate culture. If one reviews what is driving this behavior, they are likely to realize the incentive structure for a manager is based on delivery—the Dojo asks them to slow down for the short term in order to learn, so that they can ultimately build better products faster. Specific investment should be made to help leaders understand the changes teams are going through as they learn new ways of working. A best practice to support this is to provide leadership-focused services in the Dojo as well, helping leaders learn through an immersive experience.

Dojo Life Cycle and Formats

Dojo Life Cycle

The overall life cycle for a team's engagement in the Dojo is:

- **Consult:** A short meeting set with interested team to describe what a Dojo is and how it could help. It's meant to give prospective teams enough information to decide if the Dojo is a good fit for them, and for the coaches to assess whether they are a good fit for the Dojo.
- **Charter:** Typically a half-day meeting set with the team entering the Dojo, as well as any critical stakeholders. The team aligns on their outcomes for the Dojo through articulating their elevator pitch, goals, key metrics, working agreements, and completing a team skills assessment.
- **Dojo experience:** The most engaged step in the life cycle as it is the teams' dedicated experience in the Dojo directly working with the coaches and other stakeholders for an extended duration. They will be leveraging one of the formats and offerings later discussed.
- **Release back into the wild:** Team is surveyed about their Dojo experience. They are expected to share what they have learned with others outside of the Dojo. Coaches follow up in four to six weeks to check in on the "stickiness" of what was learned and identify further opportunities on Dojo engagements for developing gaps in practices.

Dojo Formats

Dojos can offer many formats from which teams can learn. These formats are optimized for different types of learning outcomes and team constraints. Some common formats are:

- **Challenge:** Full-stack teams enter the Dojo with their own backlog. These Dojos should include the entire team, including the product owner, engineers, scrum masters, and designers if the product is customer facing. The focus of the challenge is a holistic transformation of a team's practices and ways of working. As a result, a challenge is an extended-duration activity (typically six weeks) where the team runs in hyper sprints (typically two and a half days each). At the end of the challenge, there is generally a celebration with the team's broader leadership and stakeholder group, signifying their accomplishments throughout their Dojo experience.
- **FlashBuild:** This format is meant for teams who are trying to solve a very specific technical practice problem, such as establishing a CI/CD pipeline, applying a development or testing framework, configuring cloud or container based services, etc. It's a smaller format—typically less than one week. Teams are still dedicated, but the expert coaches work alongside them to cocreate the solution, transferring knowledge throughout the process. Given the short duration, there is less focus on learning the ongoing practices for how teams should work with this new technology.[5]
- **Workshops:** One- to several-hour events designed to offer education and exposure to self-selecting individual engineers. Examples include events focused on running in the cloud efficiently, leveraging kubernetes, fortifying your code, introduction to test-driven development, etc.
- **Leader workshops:** Provide an immersive workshop that is typically no more than a couple days, blending lightweight, hands-on experiences with a lecture on the core concepts and practices that their teams are adopting. The goal of these sessions is not for the leaders to build full-stack products but for them to get their hands on enough information to understand what their teams are going through. Specifically, focus should be placed on teaching

leadership behaviors necessary to enable and empower teams to be successful in this model.

Core Offerings

The Dojo should have some enumerated core offerings, which are customized based on the team's needs. Some of these offerings could be based on topics including security, CI/CD, DevOps, cloud, Agile, and product. There's also a concept of naming offerings across these topics to orient around the outcomes teams strive to achieve as a result of their Dojo engagement. Examples may include "breaking the monolith," "responding to security incidents," etc.

An effective Dojo is able to meet teams where they are at. The coaches are able to assess the team's strength and weakness, relative to their learning goals and aspirations. Based on this, the Dojo experience is continually customized and adapted to ensure the teams are getting the best experience for their needs. Do not orient your Dojos around a set curriculum as it is too rigid. Remember you are trying to change how people work, not certify them on a set of skills and ceremonies that they are unable to effectively leverage in practice.

Variations from the Standard Dojo

Research shows that communication works best when it's in person.[6] It's not a far stretch to say that learning also works best in person. There's a lot of benefit to the physical Dojo space in terms of being able to provide a safe, controlled environment to learn in—building community and connections, creating energy and buzz, etc. Given the global nature of business, it's not often feasible for team members to be colocated for a Dojo experience. We've outlined some variations that we believe should be experimented with to measure learning efficacy in these formats.

Remote Team Members/Non-Colocated

This is the same as a standard Dojo challenge with one major exception: there is no physical space to colocate the team. The entire team is virtual.

The largest impediment to success in this format is keeping participants engaged for six or more hours a day over the phone or via teleconference. It's difficult to achieve high-context communication between members and to keep the group focused when it is easier for them to multitask when remote. It also makes it far more difficult for the coach to capture the attention of the group.

To be successful, you'll need to invest in tooling that helps facilitate high levels of remote collaboration and communication in a technical context. Some tools that may be beneficial are Slack, Screenhero, Google Jamboard, and Sococo. Additionally, some of the common source-control platforms, such as GitHub and GitLab, have fairly advanced social-coding capabilities within them.

Non-Dedicated Space

This is the same as a standard Dojo challenge with one major exception: the team enters a space that is not persistent.

This is actually a common pattern when companies are just getting started with their Dojos. You can get mobilized relatively quick with a few coaches, but it can often take a bit longer to secure and build a permanent space. The largest impediment caused by this circumstance is the missed opportunity to directly optimize your space in order to facilitate your cultural aspiration. As an example, participants are not surrounded by Dojo principles as they are with the physical Dojo.

Dojo-in-a-Box

This is the same as a standard Dojo challenge, but the Dojo coaches are sent to the teams in their normal working environment. It isn't too different from traditional Agile coaches. The exception being that the Dojo will bring specific materials, practices, and approaches. While this may work when you have some remote locations that you can't easily service from your Dojo, leaving teams in their normal working environment makes it more difficult to drive a transformative experience in how they're working as you aren't pulling them out of their comfort zone.

2 x 4

For teams who are not colocated but want some of the benefits of an in-person Dojo experience, the 2 x 4 may be the best option. This format is the same as a standard Dojo challenge, but the amount of time that the team is colocated is limited to the first two weeks. After the first two weeks, all the team members return to their "home" location for the remaining four weeks of the challenge. In fact, this option can be combined with many of the other alternative formats previously described.

The companies that have experimented with this approach have seen a lot of benefits from getting the team together even for the first two weeks. It helps to quickly move through the forming, storming, and norming process on the model. Additionally, it allows for everyone to get on the same page of their Dojo outcomes quicker and enables them to get better foundational relationships in place, which will help when they move back to a remote context.

Measuring the Success of the Dojo

Dojos are established to help teams tackle new technologies and different processes through a direct experience. As the teams work through problems, they acquire new skills and demonstrate an ability to apply them to their own projects and products. After interviewing several *Fortune* 500 companies, we found they're using a variety of methods to evaluate success.

Delta Air Lines

As their primary evaluation tool, Delta Air Lines uses a set of thirty questions with "yes" or "no" answers. The questionnaire covers capabilities needed by a modern software organization to ship the software quickly and with low failure rates. The questionnaire is available online, and a facilitator is there to help those unfamiliar with the terminology. By comparing the answers before and after the Dojo, the organization is able to assess progress.

In addition to the questionnaire, Delta Air Lines customizes Dojo objectives to each team. Before teams enter a Dojo, they complete a "Discovery Phase" where they identify what they want to accomplish in the Dojo. At the end of this phase, teams

write down their own set of goals and objectives. Teams evaluate their own progress against their Discovery Phase objectives using common scrum definitions of Ready (DoR) and Done (Dod).

Delta Air Lines also has a new set of cloud-native software tools, services, and architecture. Before teams are allowed to onboard to the new technology, they must complete a Dojo and get hands-on exposure. Looking at the teams who have completed the onboarding requirements and have implemented features on the new technology stack provides another measure of success.

Target Corporation

At Target, Dojos are an optional learning experience, and teams must sign up to participate. Teams define their own goals and measures of success in a charter. The goals are intended to be completed in a single Dojo. Charters have four sections:

- **Overview:** Name for the Dojo, explicit duration, brief explanation of why this is important, and a visual explaining the people and teams contributing to the success of the Dojo.
- **Goals and measures:** The desired outcomes and measures of success.
- **Skill matrix:** Explicit skills individuals attain during their participation.
- **Working agreements:** Norms of behavior during the Dojo.

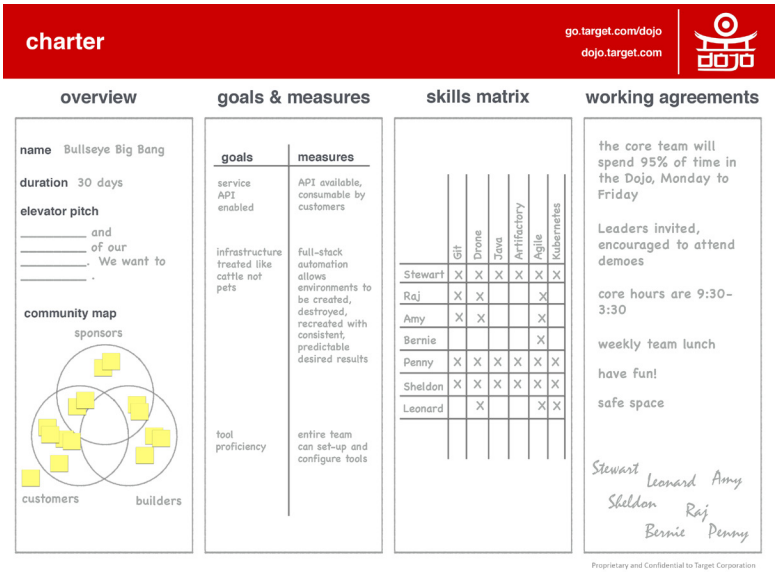


Figure 3: Dojo Charter Example

The "Goals and Measures" section of a charter must have at least one learning goal and one real-work deliverable. Teams select their own learning goals, and the best are those that are outcome focused. From the sample (see Figure 3) the learning goal is tool proficiency. Having real-work deliverables means teams are making changes to production services and applications. Completing real work makes the Dojo productive; it aligns learning in the Dojo with daily work, and it encourages meaningful changes by impacting things.

Dojo participants may span different functional roles from software development, product management, or quality assurance. Each functional role may have different skills they want to acquire. The skills map explicitly indicates what each participant would attain. In addition, Dojo participants come from across the organization, and they may be from different teams. Having an explicit set of rules will establish norms of behavior and pull the participants together into a mini-community. The Dojo participants should come up with their own ground rules, and as they progress in the Dojo, they are free to change and update them.

Teams need half a day to create their charter, and typically, charters are completed a few weeks in advance of the Dojo. An important element of success is the culture of coaching. Coaches feel responsible to help teams complete the goals they have defined in their charter. Privately coaches aim for teams to successfully complete 70% of their goals. If a team is not on track to complete their goals, a coach will intervene and open a discussion. A good coach should be able to help the team take action to overcome obstacles and reset expectations.

John Deere

At John Deere, Dojos are part of a continued effort to improve IT. During a Dojo, teams work on real backlog work items under the supervision of product, Agile, and technology coaches for periods ranging from sixty to ninety days. The IT organization identifies teams to participate, and they work to bring them into the Dojo. Through this process, teams should have improved in two areas:

- **Stability:** Increased stability through better risk management. An example would be continuous integration leading to improve uptime.

- **Frequent delivery:** Deliver value to customers more often by identifying valuable features and prioritizing the work needed for those features. An example would be leveraging Scrum to ship customer-facing features and improvements into production on a weekly cadence.

As part of measuring their continued IT improvement, John Deere collects a broad range of measures. John Deere uses the metrics that are appropriate for the type of work the team is doing. They feel it is important to emphasize each team's individual journey of improvement and reinforce the positive, cultural process and technical changes that led to these advancements. Therefore. looking at a large number of measures enables John Deere to examine improvements across a wide variety of teams. The DevOps Enterprise Forum paper *Measure Efficiency, Effectiveness, and Culture to Optimize DevOps Transformation* discusses more about measurements and metrics for your technology organization.[7]

Scenarios

Below are a couple scenarios to help you better visualize how a Dojo might work.

Scenario: Moving to Git

As a scenario, Jennifer has been tasked by the CIO to move her engineering teams to Git. Git is a distributed technology for managing source code, configuration, and documentation. Moving to a new version of source control would help the teams integrate with an improved software-development toolset, and, most importantly, a new joint venture required the use of Git for collaborative development.

Jennifer initially treated this as a code-migration project and tasked an engineer on her team to investigate what it would take to move code to Git. After the initial investigation, the engineer recommended that in order to get the best use of Git, it would require a change in workflow and that all code updates should be checked into the main code line. As Jennifer discusses this with her technical team leaders, she encounters push back as they feel checking into the main code branch will cause disaster and prevent the team from releasing because they will need to fix all bugs before shipping. Jennifer knows other teams in her company have moved to Git and

are generally happy. She asks her boss for advice, and he recommends Jennifer sign one of her teams up for a Dojo and build a project with Git.

The mandate provided by the CIO to move to Git provides Jennifer with a relatively straightforward assessment. Can her teams move their code from a project to Git? Her teams will need to learn how to use Git, and they will need to solve the problems introduced by the new workflows. Jennifer feels her teams are pushing back and do not want to move to Git. Jennifer will be relieved if her teams come out of the Dojo happy and encouraging each other to move to Git.

After the Dojo, her teams learn about Git and how it integrates with Jira and Jenkins. They are excited by the new self-service tools, and they learn how to configure their pull requests to run automated tests and code reviews. One of the developers even figures out how to prompt the developer to enter the Jira ticket(s) associated with each pull request and automates the creation of a manifest file for deployments. The teams are now excited to move to Git, and they encourage others to follow by giving demos of the new pull-request process. (See Table 1.)

Evaluation Criteria	Before Dojo	After Dojo
Team learns specific skills	No understanding of Git	Can perform add, check-in, check-out, clone, branch, and pull-request operations
Team improves the way they work	Team does manual inspection, code review, and testing before each release	Team shifts left and does code review and unit tests at each check-in
Team innovates	Jr. engineer creates release package, hand builds a manifest, and releases notes	Building the package along with the manifest and release notes is automated
Gets real work done	Team has no projects on Git	Team moves an existing software product to Git and works through several implementation issues

Scenario: Product-Based Org

NewCo is a medical company that stores and manages electronic health records (EHRs). The company feels it is falling behind its competitors, and they need to build a new product that works well on both desktop and mobile devices while supporting thousands of diagnostic and billing codes. They believe this new product will enable them to open another market by selling to hospital groups.

The executive leadership team of the company feels they need to work quickly to bring a product to the market. They decide to form a dedicated team, solely focused on building this product. For the first step, they decide to build a prototype of their mobile-ready product—deploying to the cloud, using new open-source software, and moving to a different software architecture.

Using this new technology and moving to another architecture is a big change for the team. The CIO decides to start with a small group and run a six-week Dojo. The team will use the new technology and architecture to build out a small service. After the Dojo, the small team will move onto building out the prototype. The CIO expects that the new product development will start with a team of six and eventually grow to a team of forty. To help train everyone, the CIO rotates new groups of six through the Dojo. After the teams complete the Dojo, they will be assigned onto the new product team.

The first team starts the Dojo, and they learn about the particulars of setting up and configuring the new software. They start a wiki that adds more detailed configuration instructions. The team also finds the need to generate some test data, and they add it as JavaScript Object Notation (JSON) blobs to the Git repository. When the next team comes, they leverage the wiki with the improved instructions and continue to add more to help those who are next. The process of improving the knowledge base continues as each team works through the Dojo. (See Table 2.)

Evaluation Criteria	Before Dojo	After Dojo
Team learns specific skills	No understanding of AWS, MongoDB, Jenkins, Git, or GitHub	Can set up services and create a CI/CD pipeline using open-source software and deploying to the public cloud
Team improves the way they work	The team was siloed by functional area. Software engineering, service engineering, product, and operations were all separate groups that handed off work through tickets and change requests.	The functional areas start to comingle and interacting more frequently. They start to sit next to each other, and they use a single, work-item backlog across their new product.
Team innovates	Previously, the team separated distinct services supporting mobile and desktop.	Team begins building a single set of services as public APIs in the public cloud. Internal and partner teams can leverage the same APIs to solve customer problems.
Gets real work done	Not able to support large number of diagnostic codes	New prototype proves out the ability to support large number of diagnostic codes.

Maintaining/Sustaining the Dojo

Many improvements are initially successful but ultimately fail due to insufficient investment and structure being put into place to scale and sustain the results. Dojo champions need to make the time to explain the value Dojos create and how it can help the overall organization reach their goals. Many of the organizations start small, with two or three Dojos from teams in the vanguard and sponsorship from a single executive. These first opportunities should be leveraged to generate case studies and short testimonials of success. In this way, funding and resources are pulled together by scrappy teams of volunteers. The challenge comes in expanding the Dojos as an offering to the larger organization.

One of the biggest challenges is funding. Some organizations have experimented with a charge-back model. A charge-back model is compelling because teams pay for Dojo services and resources as they use them. The challenge is teams often forget to include immersive learning as part of the budgeting process and often leave out specific line items for Dojo funding. Therefore, in a charge-back funding model, teams often relocate their existing budgets to pay for Dojos.

An alternative to the charge-back model is dedicated funding for Dojos as a central, shared resource. Most companies are using dedicated funding for Dojos or are moving toward that direction. They find dedicated funding is much easier to manage, and it eliminates an obstacle that prevents teams from the training and learning that will lead to success.

Staffing may also be a challenge. The Dojo roles should be rotational, so folks can move back into delivery and also move those who have experienced this into coaching roles. This is not atypical for how coaching roles are handled in an Agile or DevOps journey. A model of training for the trainer, such as local leads who perform the coaching, provides additional bandwidth.

Conclusion

Just as a student of the martial arts would progress under the direction of a master from Kata to Kata, from a white belt to red, and on to the pinnacle black belt, DevOps and other Dojos provide a powerful, immersive framework and experience for cultural

transformation. In addition, Dojos offer learning skills and techniques critical for a successful DevOps transformation.

Find a master, a space, and a passion, and get to work!

References

Ainomugisha, Gerald. "5 Great Reasons Why Happiness Increases Productivity." *6Q Blog*, https://inside.6q.io/5-great-reasons-why-happiness-increases-productivity/.

Brown, Damon. "Target CIO Explains How DevOps Took Root inside the Retail Giant." *The Enterprisers Project* (January 2017): https://enterprisersproject.com/article/2017/1/target-cio-explains-how-devops-took-root-inside-retail-giant.

"Communication on Agile Software Teams." Agile Modeling website (2018): http://agilemodeling.com/essays/communication.htm.

Hering, Micro, Dominica DeGrandis, and Nicole Forsgren. *Measure Efficiency, Effectiveness, and Culture to Optimize DevOps Transformation: Metrics for DevOps Initiatives*. Portland: IT Revolution, 2015.

Kersten, Mik. *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework*. Portland: IT Revolution, 2018.

Perez, Carlota. *Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages*. United Kingdom: Edward Elgar Pub, 2003.

Walker, Jason. "Target FlashBuilds." *Target Tech Blog* (November 2014): https://tech.target.com/2014/11/10/target-flashbuilds.html.

Notes

1. Carlota Perez, *Technological Revolutions and Financial Capital: The Dynamics of Bubbles and Golden Ages* (United Kingdom: Edward Elgar Pub, 2003).

2. Damon Brown, "Target CIO Explains How DevOps Took Root inside the Retail Giant," The Enterprisers Project (January 2017): https://enterprisersproject.com/article/2017/1/target-cio-explains-how-devops-took-root-inside-retail-giant

3. Mik Kersten, *Project to Product* (Portland: IT Revolution, 2018).

4. Gerald Ainomugisha, "5 Great Reasons Why Happiness Increases Productivity," *6Q Blog*, https://inside.6q.io/5-great-reasons-why-happiness-increases-productivity/.

5. Jason Walker, "Target FlashBuilds," tech blog (November 2014): https://tech.target.com/2014/11/10/target-flashbuilds.html

6. "Communication on Agile Software Teams," Agile Modeling (2018): http://agilemodeling.com/essays/communication.htm

7. Micro Hering, Dominica DeGrandis, and Nicole Forsgren, *Measure Efficiency, Effectiveness, and Culture to Optimize DevOps Transformation: Metrics for DevOps Initiatives* (Portland: IT Revolution, 2015).

A Special Thank You to Our Sponsor

Our mission for the Forum is to bring together technology leaders across many industries and facilitate a dialogue that solves problems and overcomes obstacles in the DevOps movement. For three days at this private event, we gather 50 of the best thinkers and doers in the DevOps space to tackle the community's toughest challenges. We ask these thought leaders to collaborate and generate a piece of guidance with their best solutions to the challenges.

We would like to thank all of our attendees and our friends at XebiaLabs for helping to make this year's Forum a huge success.