



PLAYBOOK

# Dojos

A Practical Guide to Creating and  
Executing Dojos at Scale



# Table Of Contents

---

1. [About This Playbook](#)
2. [Dojo Introduction](#)
3. [Dojo Types](#)
4. [Dojo Space Setup](#)
5. [Hiring & Staffing Coaches](#)
6. [Dojo Intake](#)
7. [Dojo Experience](#)
8. [Remote Dojo](#)
9. [Metrics, Measure, & Success](#)
10. [Templates](#)
11. [Appendix](#)

# About This Playbook

---

This playbook is a step by step user guide which will help you with designing, creating, executing and scaling your Dojos. The flow of the playbook assumes you are progressing from start to finish. You can also use this playbook for individual topics or areas of opportunity where you are looking to start or improve. This playbook provides Liatrio's expert opinion on specific topics, and guides you through various stages of creating, executing and scaling your Dojos.

The approaches outlined in this playbook is a general guide and Liatrio recommends a tailor-made approach that fits your organizational needs and structure to be more successful. Our point of views and expert opinions are derived from our experiences in transforming our clients and accelerating their software delivery.

This playbook is intended for:

**Leaders and Executives** who are responsible to create, own, and deliver Dojo at scale for your organization.













**Coaches & Experts** who are looking for a practical solution to design and execute a Dojo experience for product teams and leadership.

**Dojo Operations Experts** who are looking for a practical guide to setup a successful Dojo space, communication and support model for Dojo execution at scale.

**Champions** who are looking for more practical solutions to assess and determine if Dojo are the right vehicle for transformation and accelerating software delivery.

## How to Use This Playbook

If this is your first time utilizing Dojos, this table will guide you to the playbook sections that address your organization's priorities:

			
Playbook Section	Strategy/Plan	Lighthouse	Scale
<a href="#">Dojo Brochure</a> 	✓		
<a href="#">Dojo Introduction</a> 	✓		
<a href="#">Dojo Types</a> 	✓		✓
<a href="#">Dojo Space</a> 		✓	✓
<a href="#">Hiring &amp; Staffing</a> 			✓
<a href="#">Dojo Intake</a> 		✓	✓
<a href="#">Dojo Experience</a> 		✓	✓
<a href="#">Remote Dojo</a> 		✓	✓
<a href="#">Metrics &amp; Measurement</a> 		✓	✓

# Dojo Introduction

A dōjō (道場) is a Japanese word for a space focused on immersive learning. At its core, the Dojo is a cultural movement focused on helping organizations improve and accelerate their software delivery.

Teams and leaders leave the Dojo experience motivated and excited after applying modern software delivery techniques to their own product.

Liatrio has found the dojo to be the ONLY meaningful vehicle capable of team modernization where the learnings and desire to continuously improve sticks.

The Dojo provides an immersive, hands-on experience that improves a team's way of thinking and working so that organizations can deliver products faster to customers. This dedicated experience and immersive learning helps organization to improve on:



## Culture

Enables a change in mindset and behaviors for both the delivery teams and their leadership.



## Practices

Learn new technical and non-technical practices, apply and reinforce knowledge to shift teams toward modern ways of working.



## Experimentation

Immersive, distraction free environment that allows teams to apply experiments and learnings by failing early and fast.

*The best learning and growth happens only when people apply newly gained skills to their product backlog and are able to break past existing organizational constraints.*

## Benefits of Dojo

The Dojos help enterprises to

- Evolve as modern, technologically advanced learning organizations and enable teams to develop a new way of working
- Create a culture of continuous learning and improvement as teams become high performing and productive in delivering value to customers
- Create an inspiring work environment to expand existing expertise and attract new talent
- Enable engineering excellence and teams and engineers are immersed in a new way of working and rapidly upskill themselves
- Improved time to market for product and service as product teams are able to increase their Lead Time improve deployment frequency
- Realize significant cost improvement by reducing waste, removing organization silos and improve quality & efficiency through automation
- Culture of continuous learning and improving motivate teams and organizations to transformation

## Common Misconceptions of Dojo

- Dojo is a training center focused on training and providing materials and contents
- A short term investment to increase productivity & quality
- Only “product” teams are invited to the Dojo and not platform, infrastructure, security and others
- It is only for developers and engineers and not for the entire team
- Dojo is mandatory for all teams to obtain access to modern tools and platforms
- Dojos would need unicorn/super star coaches to execute and scale

## Dojo Types

---

Our typical Dojo offerings include the following:



### Product Team Dojo

Immersive, 6-week experiences where teams participate in a broad scope of learning and applying to their products/services.

---



### Leadership Dojo

Short-term experiences that help leaders learn about Dojos which enable them to support the teams.

---



### FlashBuilds

Short-term, immersive experiences focused on specific outcomes.

---



### Workshops/Labs

Learning opportunities attended by individual team members.

---



### Product Chartering/Roadmapping

Business focused experience aligning product team and delivery teams



## Product Team Dojos

Our tailor made and immersive Dojo experience for product teams results in a foundational and holistic transformation of a team's technical capabilities, practices, and ways of working. Product team Dojos are typically 6 weeks long. A full-stack team enters the Dojo with their own product/services backlog. The entire team should participate, including the product owner, engineers, scrum masters, and designers (if the product is customer facing). As part of these extended-duration activities, the team runs in hyper sprints (typically two and a half days each). All learning and applying on the team's product/backlog is done within the hyper sprint cycle and all work is tracked by tickets.

Expert coaches will be embedded with the team and guide them through a transformational journey uplifting their product through modern execution techniques, methods, and engineering practices.

### Types of Product Teams

Any and all types of teams should be encouraged to come through the Dojo, ideally you want to bring shared services/platform & infrastructure teams first before any application teams and it is very important for them to go through a Dojo experience as it would help the shared services teams to get aligned and prepared to provide support for application teams during and post Dojo.

- Application Teams
- Shared Services (Tools, Platform, etc..)
- Infrastructure and Operations
- Security and Compliance
- Site Reliability Teams

At the end of the Dojo, the team celebrates what they've accomplished during the Dojo experience and showcases their success to broader leadership and stakeholders.



## Leadership Dojo

Our leadership Dojos are designed to provide executives and leaders a holistic experience to show what a successful transformation looks like including how to uplift and support your teams. These immersive experiences typically last no more than 2 – 3 days, blending lightweight, hands-on experiences with practical learning of core concepts and practices that the leaders' teams are adopting. Instead of building products, these sessions are designed to help leaders understand what happens in a Dojo and demonstrate why they should encourage their teams to participate in a Dojo and how they can support their team's success.

A typical Leadership Dojo experience is designed to provide the following:

- Understand key failure points of low-performing software delivery teams
- Identify and create a baseline organization capabilities for modern rapid delivery
- Hands on experience on how accelerated software delivery will look like using modern execution techniques, methods, and engineering practices
- Visualization and experience target-state engineering organization with high performing teams
- Guidance and assistance for leaders to adapt to servant-leadership that drives holistic and successful organizational change
- Understand how to enable, support, and empower product teams as they go through their Dojo experience
- Discover and line up initial list of lighthouse teams or early adopters
- Discuss how to scale Dojos across your Line of Business and eventually the whole enterprise

Liatrio recommends executing a few rounds of leadership Dojos first to help leaders align and understand their teams needs and enable them as change agents within their organization, before executing product team Dojos.

## Details for planning these types of Dojos:

Leadership/Manager types of Dojos are really customized based on the needs of the organization, team, and team members attending. Several common areas are usually included such as strategy, hands-on labs, and solutioning blockers, for example.

Creating a leadership/manager Dojo is all about meeting the client/organization where they are currently at, and helping them drive through transformation changes that will elevate their delivery work, execution of that work, and the leadership support needed to clear obstacles that impede or hurt the organization transformation.

**The primary purpose of these specific Dojos** are to ensure all levels of the organization understand and know how to support Dojos. However, Dojos are often the vehicle to drive transformational level work as well. As such, these Dojos help to establish, build out, and continue the progress of transformational work at the mid to senior levels of an organization.

### Step 1

How to start and build out the custom plan for these Dojos begins with gathering information such as:

- Organization structure, roles and responsibilities, company history (especially any previous transformations/growth strategy work)
- Delivery work
- Platform
- Enablement activities
- State of learning
- State of improvement
- Culture
- Product roadmaps
- Company and product goals/milestones
- Problematic areas
- Discovery/exploration/interviews with leaders, engineering activities
- Observations of delivery teams
- Metrics/Measures

## Step 2

From these discovery areas, we begin to identify areas that may be impeding flow or effectiveness, or that are not developed/fully defined/not understood yet that will enable flow and effectiveness.

- Identify areas needing improvement
- Prioritize those areas
- Review with Liatrio Dojo team
- Review with client sponsor

## Step 3

Create the goals of what the leadership Dojo/manager Dojo is trying to achieve. These goals should be observable and measurable. Some examples of the goals may look like this:

- Alignment on core technology principles, and create these principles which will be shared out to teams
- Create strategy/roadmap for how platform will be created, used, and maintained. A road map will be created during the Dojo.
- Determine metrics and measures that teams and leaders can use to improve on delivery work and help discover areas of improvement. Metrics will be selected during the Dojo.

## Step 4

Create a high-level agenda based on step 3 (this agenda is an example and would most likely include more items)

- Welcome, intros, icebreaker
- Determine Core Technology Principles
- Strategy Roadmap on Portfolio to Product; developing strategy, timeline, areas of opportunity
- Next Steps (take home work, sharing progress, planning for next iteration of leadership Dojo)

### Step 5

Next, review with the Liatrio team for alignment and then review with the sponsor(s) for any other areas that they may wish to have included, such as HR level activities like job family changes for example. Your agenda may look like this:

Day/Time	Topic	Notes
Monday 8:30 – 9am	Welcome  Agenda  Icebreaker	Share agenda, housekeeping, flow for the day  Have team and coaches intro themselves  Conduct icebreaker to shift minds to learning and sharing
Monday 9 – 9:45am	Determine core technology principles (small breakouts)	Small teams work on creating core principles they feel is needed to establish guidelines for teams.  Small group returns to large group and shares/demo's progress  Large group aligns on core principles Principles are added to company communication channels
Monday 9:45 – 10am	BREAK	Coaches should be mingling with team during breaks/lunches to get 1:1 feedback and con't with discovery work (record notes in central location)
Monday 10am – 12pm	Strategy/Roadmap Product/Portfolio team alignment	Organizing teams with DevOps mindset, roles needed, timelines, company goals to product and team alignment
12 – 2pm (working lunch)	Roles and Responsibilities for DevOps teams	What roles are needed and why? How to organize teams with resources available. Where does infra/platform fit into teams?

## Step 6

With the agenda reviewed and agreed upon, begin building out content, material, immersive activities that fit the agenda and team going through the Dojo. The following sections will go deeper into SOME areas that can be brought into the Dojo, but it is not an exhaustive list.

### Examples for content ideas for Leadership Dojo:

- Icebreakers
- Alignment
- Strategy/Roadmapping
- Metrics/Measures
- Friction/Opportunity Areas
- Standards/Core Principles for teams (CI-CD, branching, toolchain, quality of code)
- Chartering all of the Dojo's for company (used to establish offerings, coaching, resources, outcomes, defaults for teams, duration, team selection, etc)
- Setting guiding principles for CI-CD/DevOps/Expectations of teams
- Light hands on labs (platform, testing, deployment, data collection, etc)

### Examples for content ideas for Manager/Eng Mgr Dojo:

- Icebreakers
- Alignment
- Strategy/Roadmapping
- Metrics/Measures
- Friction/Opportunity Areas
- Solutioning for blockers or friction areas
- Standards for teams (CI-CD, branching, toolchain, quality of code)
- Chartering Dojo's for company
- Light hands on labs (platform, testing, deployment, data collection, etc)

## Overview, Purpose, Execution of Leadership/Manager/Eng Mgr Dojos:

### Objective

To help executives/leaders/engineering managers better understand and support software engineering/product teams as they move through transformation changes.

### How

Leadership Dojos and engineering managers Dojos are custom offerings from the Dojo team that focus on how to help organizations be effective at **CREATING AND LEADING** high performing software engineering/development teams as they through transformation. They have proven effective at aligning business goals to technology goals, as well as aligning leaders to individual team members. Both of these improve team effectiveness and efficiency of software/hardware delivery work. In some cases, leadership and management Dojos have similar, lightweight activities, but scope and depth can vary.

Typical Leadership Dojos begin with sharing common goals of the Dojo, from engineering enablement to platform/toolchain choices and then can move into overall roadmap / goals alignment, strategies, metrics/measures, friction areas as one example set for a Leadership Dojo.

Management or Engineer Manager Dojos are often more direct involvement with application or software/hardware, tools, code, DevOps functions/culture, and other areas that delivery teams work with, *alongside* some of the strategies that support and enable team delivery. Some examples include having engineer managers work with an existing reference app to learn how to build, test, deploy code and to see the impact of getting code to prod within minutes.

Depending on what areas the organization prioritizes, or may need assistance with these are some common areas that may be worked on are outlined below.

### Duration

Typically leadership Dojos last 1 – 2 days, depending on transformation work needing to be done and the level of activity that needs to be worked on. Manager/Engineering Manager Dojos are approximately 1 day in length as well. Both experiences are meant to be immersive and could be split into several half day's if needed, ideally it would be done concurrently.

## Who

Who should attend Leadership and Manager/Engineering Manager Dojos? Leadership Dojos are intended for the levels above direct team leadership, these roles are typically directors, VP's, C-level officers as these roles typically 'own' strategy and decisions that span areas greater than a team or portfolio would. Manager/Engineering Manager Dojos are designed for leaders as well, but the focus is a bit lower as these leaders may be more in direct contact with teams and may even contribute directly as an individual contributor to the team and as such more hands-on time/labs may be needed.

## Delivery/Frequency

Delivery of the leadership and manager/engineering manager Dojos are best organized so that a leadership Dojo goes first, then a few weeks later an Manager/Engineering Manager Dojo would occur. This is intentional, as the feedback and lessons learned from Leadership Dojo are passed into the Manager/Engineering Manager Dojo for action/solutioning and the feedback from the Manager/Engineering Manager Dojo gets passed back into the NEXT Leadership dojo for action or strategy update work. This cycle can continue indefinitely, or as long as there are areas of work/concern/structure that needs to be addressed. As the most urgent matters are delivered on, cadence may decrease but it is highly encouraged to stay aligned through shorter duration events, collaboration channels and shared deliverables.

Leadership and manager/engineering manager Dojos are often done on a recurring basis. For example leadership dojos might be done as an 'off site' (gathering together in person provides the highest impact value and productive work in shorter times than remote only) for a full day or several half days, and then occur every 3 – 4 months to work on feedback from engineering leaders/managers.

## Leadership Dojo

### Alignment

- Goals as a company — what is driving the change, what is the value of our goals? Why do we need them? How do you consume the goals into daily work?
- Business strategy (timelines) — blockers, issues that have come up in the past
- Product Strategy — what are the products? Do product teams know their goals?
- Do we have a product/portfolio mindset? If not, how do we enable that
- Data models/domain model for data — who owns it, what does that ownership entail, how to build and create structures for that data model that teams will need to create, consume, enrich and maintain
- Reference pipelines could be an activity to help shape what the delivery teams need to align to. Sets standards in place.
- Toolchain — what tooling/functionality do we need to align on?
- Community/Collaboration — how do we communicate effectively with teams, team to team, team member to team? Where will information be, how do you stay connected to continual changes?
- Culture — what is our culture, what is it we want our teams and our company to represent, how do we model that behavior from team member to team leaders?
- Roles and responsibilities — within teams and outside teams. Expectations of team members. Clarity on roles going forward.
- Team member support — fail fast culture (learning always, failing early, building trust with leadership)

### Strategy

- Business to technology strategy — What are the business goals and how will technology help shape and support those? What gaps are there in the technology portfolio that needs to be addressed to support the business goals? What are the timelines and blockers that have caused issues? What are the friction points the business is having with technology? How do we create and empower community, team building, leadership within teams?
- Technology to team strategy — What are the technology goals that the teams need to know and adopt? How will the teams be held accountable for commitments? What blockers are the teams hitting that may cause slow downs or misunderstandings? What are the data strategies, security strategies, quality standards, tool chain purpose/reasoning?
- Team to team strategy — What dependencies do we have between teams, between data, between app ownership? What gaps are in the teams that cause friction or slowness?

- Roadmaps — product to portfolio roadmapping can be done in the Dojos or can use the high level roadmaps and build from them
- DevOps/Agile — what is our strategy for both, how do we bring this strategy into day to day work, how is it modeled?
  - 4 pillars of DevOps (Culture, Flow, Fast Feedback, Automation)
- Technology strategy — various technologies require strategies for teams to have guide rails to select, use, extend, etc. Examples may include what are the business' guidelines for open source tools, database choices, data storage, building and sharing API's at team or portfolio level, data ownership, data boundaries, shared data, etc.
- Engineering Culture — fail fast, experiment, demo's, demo days, guest speakers, continuous learning, opportunities

### Metrics/Measures

- Why do we want to measure teams, portfolios, technology areas? What do we want to measure? How would we get the data? How will we serve the data to people/teams/etc? Team effectiveness over team negatives.
- DORA4 metrics, why they matter
- Other team metrics (Jira, ticket systems, change management, custom metrics, pipeline, logging)
- Consuming metrics at team, portfolio level, organization level — actions needed, informational metrics, system health metrics, etc.

### Friction/Opportunity Areas

- Areas of opportunity that have been displayed
- Areas of opportunity that may arise/anticipated
- Technology gaps
- Business gaps
- Leadership gaps
- Roles and responsibilities

## Leadership Example Agenda

(customized based on current organizational priorities)

Time	Topic	Notes
9 – 9:45am	Business Strategy/Alignment	Business to share company goals for quarter and for year
10 – 10:45am	Technology Roadmap to meet business goals	Outline areas that technology will enable, build, support to deliver on business outcomes
11 – 11:45am	Culture	What do we want our team members to understand, how to deliver on expectations
	Metrics	What do we want to measure, how will we use it, how will we gather it?
12 – 12:45pm (working lunch)	Friction/Opportunities	Break out, small groups
...		

## Manager/Engineering Manager Dojo

In addition to topics above, more detailed work is often done with engineering managers as they may be people managers and/or direct contributors to active software/hardware development work.

### Manager/Engineering Manager Example Agenda

*(customized based on current organizational priorities)*

Time	Topic	Notes
9 – 9:45am	Technology Strategy	Outline areas that technology will enable, build, support to deliver on business outcomes
10 – 10:45am	Hands on Lab	Work with demo app, learn to build, test, deploy for example
11 – 11:45am	Metrics/Measures	What are they, how to use them, etc.
12 – 12:45pm (working lunch)	Feedback on Issues	Breakout sessions to work on solutions to common issues (prioritized)
...		

### Technical/Hands On

- Using a demo app, or existing app, add new feature to app and deploy it
  - Follow quality standards and add unit tests
  - Validate testing quality (code coverage, test passing)

### Team

- Team
  - Experiments — creating hypothesis, trying out experiments, learning to fail fast
  - Roles and responsibilities within team
    - SM
    - PO
    - Tech/Devs
  - Ways of Working
    - 3 Amigos — why does it work, tying to dev workflow
    - DoD/DoR/Working Agreements

- Ceremonies (demo's/retrospectives/sprint planning, establishing goals)
- Flow — what is the dev workflow, has it been established, creating it if needed
- Team to team dependencies — how to manage, coordinate, sharing epics/boards
- Team strategies — branching, testing,
- Personas — defining, including into story work
- Flow/orchestration of work and deliverables
- Metrics and measures and how to use them
- Retros — powerful tool to foster team member growth and improve team delivery AND address issues that can create friction within/between team members or/or teams
- Demos — creating engineering culture, demo days, business language skills
- Ticketing systems — how to use, ticket based engineering, small story sizing
- Estimation/Pointing as a team

### Current Issues/Blockers/Concerns

- Leaders polled to gather feedback on team, organization, leadership feedback
- Solutioning/Clearing Blockers
- Small group breakout sessions to problem solve and help clear team level/organization level blockers
  - Solution work defined — polling before session on issues teams are seeing, reviewing outputs from metrics, team delivery success/failure rates, gaps in tooling/automation, technical issues, etc.
  - Team to prioritize solution work, what areas can be improved, what areas are beyond the scope of the managers/engineering managers (escalate to Leadership dojo)
  - Team does small break-outs on solutioning/clearing blockers
  - Demo solution work to full team
  - Solution implementation — what can be implemented now, soon, later? Feedback reviewed and shared to Leadership dojo

***Template for Leadership/Manager/Engineering Manager Dojos is in the Appendix***



## FlashBuilds

FlashBuilds are designed for teams to learn and solve a very specific problem (technical or non-technical), and typically last anywhere from two days to two weeks. Expert coaches work alongside dedicated teams to co-create the solution, transferring knowledge throughout the process. Ideally teams going through flashbuilds should have gone through a full Dojo experience in the past so that they are used to the new working environment. Given the short duration of flashbuilds, the Dojo is focused less on holistic and cultural change rather focused on learning and adopting on specific new technologies.

### Examples of Flashbuild experiences:

- Establishing a CI/CD pipeline
- Applying a development or testing framework
- Improving code quality
- Increase code coverage
- Expand automated functional testing
- Configuring cloud or container-based service
- Behavior Driven Development
- Feature mapping and breakdown
- Roles and responsibilities reset

### Outcomes of Flashbuild experiences:

- Bring visibility to the challenges that are present to complete the work with the normal processes in place
- Teams have the ability to collaborate and work together with various departments to remove impediments / improve flow
- Concentrated efforts towards a shared goal that has tremendous impact

### Example Agenda — Security Focused Flashbuild

#### Day 1

- Ice Breaker and Intro to Dojo
  - Dojo Portal (15 mins)
- Discuss / Define Flashbuild measurements
- Pipeline design / Path to Prod (20 Mins)
- Tech Assessment (10 mins)
- Sensible Defaults Collector — Coaches

- Local cluster config (Docker Desktop, Kubectl, K9s) (45 mins)
- IDE Review/Setup (10 mins)
- Implement Helm from temp repo — StarterKit (20 mins)
- Configure Local Helm Deployment for productinfo-step-canada-ui-experience (30 mins)
- Deploy UI to local K8s (1.5 hrs w/ troubleshooting)

## Day 2

- Sensible Defaults review (20 mins)
- Install and demonstrate SonarLint for each team members IDEs (30 mins IntelliJ & VS Code install and troubleshooting)
- Deep review on CircleCI pipeline (config.yml) including SonarQube job integration (1 hour)
- Integrate SonarQube in UI Pipeline
- Baseline code coverage and identify target improvement ( % )
- Setup pre-commit hook with .husky
- Implement jest for test coverage and optionally add it to git-hook

## Day 3

- Expert to discuss JS to TS uplift from current state — 9am CT
- Intro to Snyk
- Integrate Snyk into IDE
- Integrate Snyk into UI repo
- Remediate 100% of Medium, High, and Critical vulnerabilities for UI repo
- Enable Auto-PR's from Snyk
- Enable Gating in UI
- Submit Veracode Offboarding request for UI repo

## Day 4

- Front End Testing Strategy — 9am CT
- Snyk vulnerabilities — Cont.
- Update README w/onboarding instructions
- Review metrics & measurements
- Next Steps



## Workshops/Labs

Educational workshops/labs are designed to offer education and exposure to select engineers on specific practices and tools which typically last a few hours to 2 days. Workshops/labs are generally instructor-led or a series of self-guided learning activities that are not usually performed with their product/backlog.

These workshops/labs are sometimes designed as prerequisites for more foundational learning of tools and methodologies.

Examples includes

- Scrum/Kanban
- Test Driven Development
- Docker
- Kubernetes
- Ansible
- Introduction to cloud & cloud native



## Product Chartering/Roadmapping

Product Chartering/Roadmapping Dojos are customized to support the **business** needs of an organization. Some of the business reasons why a team should consider a product chartering Dojo are to help align technology work to the business goals, using discovery activities to assist the business with creating product roadmaps and to then help break down product or portfolio level business goals into actionable backlog items that a development team can begin to execute on. Product Dojos are also great to help the business side of an organization build out roadmaps, milestones and learn how software delivery iterates to deliver features the business needs for customers.

Think of product chartering/roadmapping as what happens to the **left** of Jira (or ticketing systems) *before* stories are brought into the team. This left side is more the product side, the symptoms of these types of product issues are poorly defined requirements/features, unclear clears, unclear prioritization of features, milestones that may not be achievable (either too aggressive or too weak due to multiple factors). Additionally, feedback from the product side and the development team usually uncovers more of what each side is struggling with. At the end of the day, all these types of issues result in delivery decreases, commitments not being met, features not matching business needs, even antagonistic relationships between product team and delivery team.

Additional purposes for product chartering roadmapping may be to level set teams, multiple teams, and/or leaders on broad strategies and how to break those strategies into smaller actionable work, or roadmaps, to further refining sub-strategies that may be needed to align and coordinate teams.

So far we have talked about *why* an organization/team can benefit from Product Chartering/Roadmapping dojo's, next we will go into the *how* of creating them and *what* should be covered.

## How to Prepare, Create, Deliver Product Chartering/Roadmapping

From a high level the flow for creating product chartering/roadmapping:

- Exploratory meetings with product owner and their hierarchy
- Exploratory meetings with delivery team key roles
- Identify friction areas, pain points, mis-alignments areas (don't try solve for these in exploratory meetings, more information and work needs to be done)
- Create high level agenda/bullets of areas to be focused on for product chartering/roadmapping
- Align with Liatrio and client sponsors on high level agenda/bullet list of areas to focus on
- Create mid to low level flow for 1 – 2 experience addressing the areas to be focused on

As with creating any Dojo experience, preparation and exploration beforehand are key to understanding the organization, team, and areas of opportunity that play a role in creating an experience for the team/organization which we will deliver. Having 2 – 3 days planned out in detail is a great starting point but flexibility and facilitation are keys to driving success for the experience!

For example, you may plan for 30 min of architecture or roadmapping alignment time and find a team struggling with that area. You could push it to the parking lot to keep movement going, or you can elect to con't to deep dive to get that alignment and understanding across the full product/delivery team. The coach has to be agile in practice as well.

Often, with chartering teams for Product Dojos, many things remain unknown at the beginning of the charter experience and are discovered during the charter itself. As we prepare to bring teams into a product type Dojo we do consults first with the team leads to learn more to ensure the team is ready to go into the dojo before chartering the entire team and learning there are blockers that prevent them from proceeding. Similarly, with product chartering/roadmapping Dojos, we as coaches need to learn more before we go into the experience so we can prepare and plan more for the *how* and the *what* parts of the experience.

Preparation/Exploration is a broad topic, but in this case it can be simplified by using the 'business' lens to help determine who to engage with, what areas to explore with them, and then building out their dojo plan based on those findings. For example, if a product team is struggling with not understanding the business goals, or who the

customer is, or what they truly think is valuable then exploration/interviews should start at product owner level and move upwards to include the team, direct reports, business leaders tied to the product hierarchy. The goal of these sessions are to clarify what friction areas the team is having around business/product, and exploring with the business side to clarify, or even to help shape the company to product to team strategies (alignment comes from organization-product/business-team). Often these exploratory conversations will reveal mis-alignments between product and delivery teams and those mis-alignments can be opened up more and examined for action that the Dojo can help drive.

The other end of exploratory sessions should be within the delivery/product team. Meeting with the engineering manager (and their director as well), tech leads, test leads, scrum master (if the team has one), business/systems analysts, architects and other roles that support the team. These sessions often reveal more if conducted in small groups or 1:1's to get real feedback vs group feedback (which tends to be more generic).

Between explorations with the product owner and their hierarchy, and the delivery team and its key roles, the areas of friction or opportunity will be more apparent. From the areas of friction you can begin to shape the Dojo experience, similar to leadership Dojo flow.

Taking the information from these exploratory interviews and friction items will help shape the experience but there are also common areas and activities that need to be worked on throughout the Dojo to bring the full team together and work on the alignment, and this is part of lessening the friction areas.

The following 2-day product chartering/roadmapping example is targeting issues around alignment as well as helping the product team shape the features into actionable stories for the development team.

Day 1	Flow/Agenda	Activity/Supplies Needed/Facilitator
9 – 9:30am	<ul style="list-style-type: none"> <li>Welcome/Orientation to Dojo/Outcomes</li> <li>Intro's of team/coaches</li> <li>Icebreaker</li> </ul>	Icebreaker activity defined
9:30am – 1:30pm (working lunch or extend by 1 hour)	<ul style="list-style-type: none"> <li>Chartering</li> <li>Product</li> <li>Teams Goals/personal goals (inputs for roadmapping)</li> <li>Arch/Business Flow</li> <li>Product Capabilities</li> <li>Working Norms/Team Agreement/WoW</li> <li>Roles/Responsibilities</li> <li>Challenges</li> </ul>	<p>Use mural/miro board or physical boards if all in person (best results and time efficiency if done in person)</p> <p>Prep before hand with these items on the boards (markers, erasers, multiple boards)</p>
1:30 – 1:45pm	BREAK	
1:45 – 5:00pm	<p>Roadmapping — Capabilities to technical Delivery</p> <ul style="list-style-type: none"> <li>Types of roadmaps, strategies, vision, Org to product goals</li> <li>Epics/Goals for current year, and current quarter to start with</li> </ul>	<p>Have product team prepared to share company goals, business goals and product goals</p> <p>Review types of roadmaps, purposes for roadmaps, how to get full team providing input to roadmaps</p>

Day 2	Flow/Agenda	Activity/Supplies Needed/Facilitator
9 – 9:30am	<ul style="list-style-type: none"> <li>● Recap, agenda for day 2</li> <li>● Icebreaker</li> </ul>	Icebreaker activity defined
9:30am – 5pm	<ul style="list-style-type: none"> <li>● Epics/breakdown/priorities for current quarter to end of year</li> <li>● Epics to Story Mapping</li> <li>● Story Breakdown exercise               <ul style="list-style-type: none"> <li>○ Optional (estimation, pointing stories consistently)</li> </ul> </li> <li>● Vertical slicing</li> <li>● Parking Lot/QA</li> <li>● Next Steps (business rules for example, setting a mini-chartering session quarterly with product/delivery team reviewing features, roadmap alignment and epics, etc)</li> <li>● Team event, Happy Hour/Apps</li> </ul>	<p>Full team: using whiteboard to track epics, prioritize epics based on product needs</p> <p>Small team breakouts, each taking one of the prioritized epics and breaking it down to small stories.</p> <p>Share as large group, then talk through vertical slicing, and have team return to stories to see if the MVP of vertical slicing is correct. Con't to break down stories and then re-share as large group with vertical slicing, team to offer comments and adjust stories</p> <p>If estimating is needed, suggest using the Fruit Game to illustrate relative to absolute estimation and help get the team aligned on estimation. Also introduce agile poker as a way for full team to contribute pointing and build estimation skills doing it together</p>

As the example shows, day 1 is pretty well defined and serves as the alignment day. Often teams have never gone through this level of detail on their product and the chartering day brings insight that will help provide alignment and context. Day 2 can dive deeper into areas that are identified as problematic or needing more guidance on.

Depending on exploratory/interview/observations with the delivery team, and product team it may be necessary to prepare for a day 3 option. Other reasons to extend into a 3rd day could be to take the high level alignment, roadmapping, and epics that are prioritized and begin to break those epics into stories. If the team is ready for more advanced project and team management, Jira roadmapping can be introduced as way for the program/portfolio to track epics, dependencies between epics/stories, and even begin building out cross team epics and dependencies.

Finally, during the 2 or 3 day's of product chartering/roadmapping it is important to recognize that high-level and mid-level planning takes time and many inputs to be successful and blockers will come up; from missing knowledge, uncertain architecture, features that have not been done before, and other uncertainties. Use the parking lot feature as you would with all Dojos or team-level strategy/planning work to hold these blockers and challenges to the side and as with Jira feature development aim for MVP for each of the areas you are exploring with the team over trying to be exhaustive in thinking and planning for every scenario, potential risk, or unknown methods. These can derail the overall success of getting the product chartering/roadmapping to a state where work can move forward successfully.

After moving to the point where you feel like the outcomes set forth are completed, and if there is time remaining in the 2 – 3 days, look at the parking lot and challenges with the team and prioritize with them the areas that are the next level obstacles or unknowns that need to be addressed for mid-term or long-term success of the product. It may be worth creating strategies with the team to outline initial steps that can be done to further break down those challenges/obstacles.

For example, you complete all the core outcomes of the product chartering/roadmap Dojo. You have several items on your challenge list and in the parking lot. With the team it is decided that the architecture design seems 'good enough' to work for MVP but maybe it lacks some clarity on an integration or future state area that will be needed for the next feature. Use the time remaining to outline the strategy to complete more of the design, add that in as research story to the backlog and include acceptance criteria and enough description that the story can be picked up. Suggest ways that the story can be worked on, as a full team, or a POC/experiment/fun Friday activity that the team and SME's can plan for time to deep dive on solving for it.

## Next Steps/Wrap Up

As you wrap up the product chartering/roadmap Dojo, always clearly record the summary of what was decided, artifacts created during the session and where they will be stored, ensure Jira is updated if at all possible (otherwise things tend to get lost or forgotten), and then outline the next steps for the people attending. If that includes scheduling another product chartering/roadmap Dojo to continue into the next features, or to stay aligned, or extend the product roadmap to delivery stories, add it to the team's schedule so it can be planned for.

## Dojo Space Setup

---

Dojos are immersive, dedicated, open-space environments where development teams can learn and apply modern engineering practices, product based thinking and reinforce key principles of trust, collaboration, and transparency. Dojos enable teams to build a solid foundation in engineering practices and technologies, cultivating the cultural change needed to build new skills and capabilities. The goal is to help enterprises become modern, technologically advanced, continuously improving and learning organizations and enable teams to develop a new way of working.

A Dojo environment should be dynamic, open, fun and collaborative with some key elements to give the best experience for the teams at enterprise organizations. Designing a Dojo environment with intention enables organizations to reinforce the working norms, workflows, and methodologies that strengthen team collaboration, foster open communications, and reinforce healthy development practices that individuals and teams can continue to improve once they return to their working space.

### Purpose of a Dojo Space

Designing the right Dojo space matters a lot, as Dojos have the power to drive the culture change needed in your organization. The space you create should specifically foster openness, collaboration, and sharing. You break down the walls and create a fun and engaging environment that inspires creativity. Create teaming spaces that encourage different groups to come together, demo, and share their learning with each other. In more than one company, I've had Dojo participants come up to me and tell me that they feel like they are working at a different company.

So, a Dojo should be built in an open environment that allows as much interaction between various folks as possible. This kind of environment also helps teams focus and solve problems using shared knowledge, build good pairing practices, and facilitate both movement within the space and face-to-face, collaborative interaction. In this open-flow space, each team works on their own projects, coaches help teams focus on their backlogs and provide constant support, and individuals and teams interact in both structured and unstructured ways in order to make continuous progress on their work.

When the Dojo model is up and running, teams will constantly filter through the Dojo space. Individual teams may work in a Dojo for only a couple days or for up to 2 – 6 weeks.

## What an Open, Collaborative Dojo Space Looks Like

The Dojo layout should be an open environment without walls, enabling teams to remain in constant communication. We have found that a successful Dojo space contains the following:

**Coaches' Corner:** A coaches' corner is a set location where all Dojo coaches work. Because the coaches are physically nearby, teams and individuals can reach out to coaches when they have questions.

**Demo Lounge:** The demo lounge should seat anywhere from 20 – 100 people depending on the space. In this lounge area, teams will showcase the work that they have accomplished within the Dojo.

**Executive Bullpen:** Executives onsite have a dedicated space where they can work while also observing efforts that take place in the Dojo.

**Platform Support Space:** Platform support teams are permanent residents of the Dojo. They have their own dedicated space located on the outskirts of the Dojo environment. If a Dojo team needs help, they can get the support they need in real time.

**Huddle Spaces:** The Dojo encourages constant collaboration and meetings. Each Dojo needs one to three huddle spaces located on the outskirts where teams can gather and meet for sprint cleaning or backlog grooming.

**Information Radiators:** One to two information radiators (walls with screens) display Dojo team metrics, team challenges, and solutions that various Dojo teams are working on.

## Dojo Space Setup and Layout

We recommend separating each dedicated Dojo space by 20 – 30 feet to minimize noise and interruptions while teams work. If possible, the Dojo space setup should have no permanent fixtures, enabling spaces to evolve as needed.

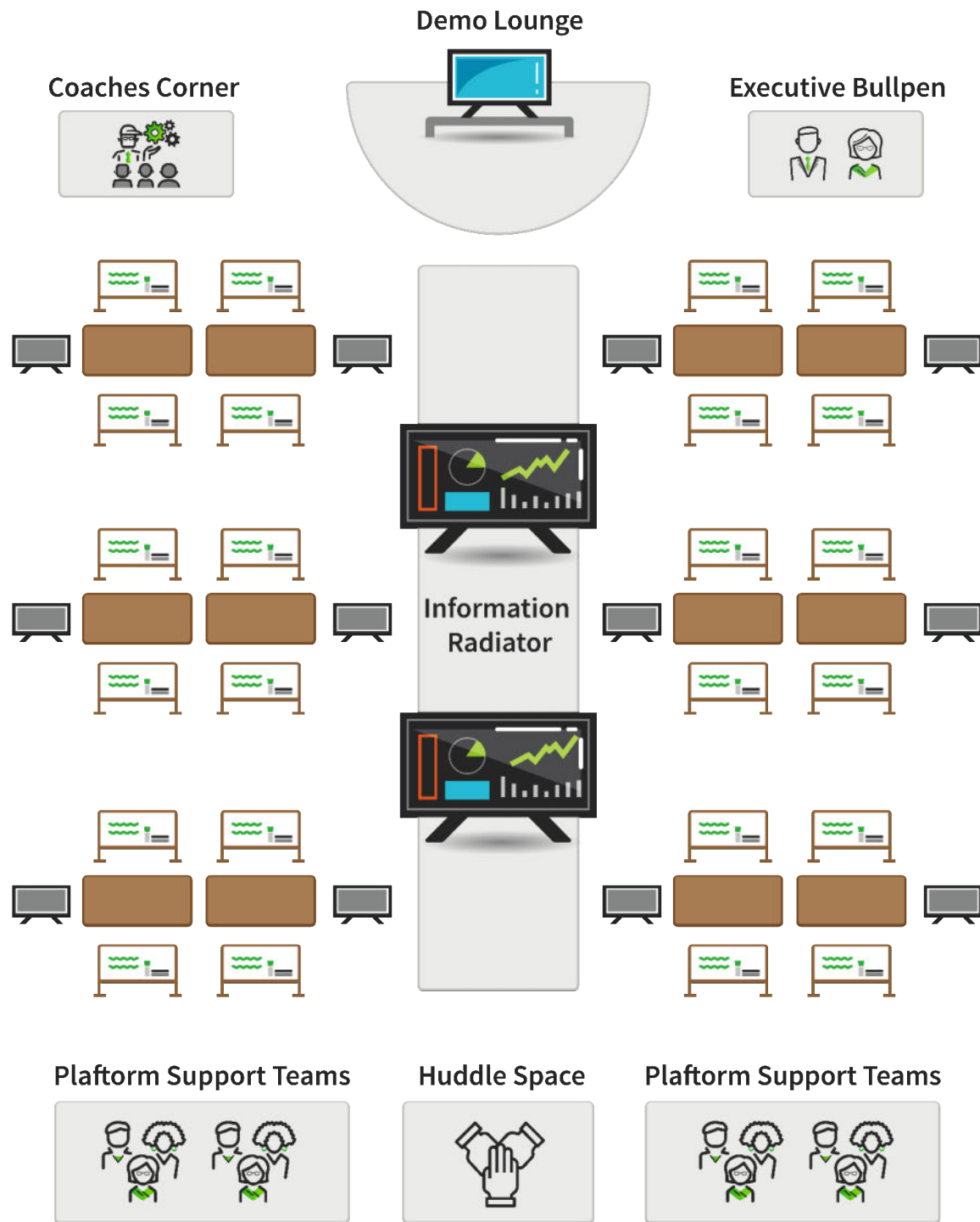
Each Dojo space should have the following:

- ✓ 2 – 4 whiteboards
- ✓ 2 – 3 tables
- ✓ 2 tvs

As teams work on their backlogs, they can use whiteboards to learn content and solve problems. Whiteboards can also be used to help separate the Dojo space for different teams.

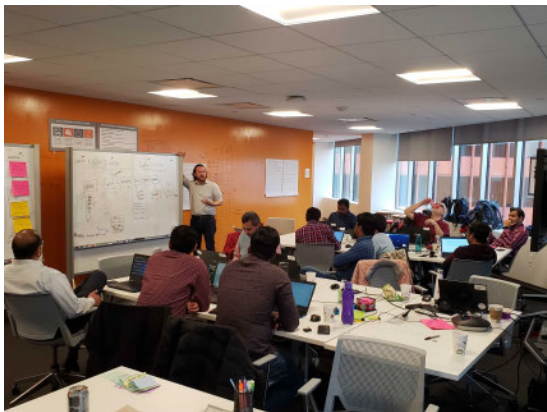
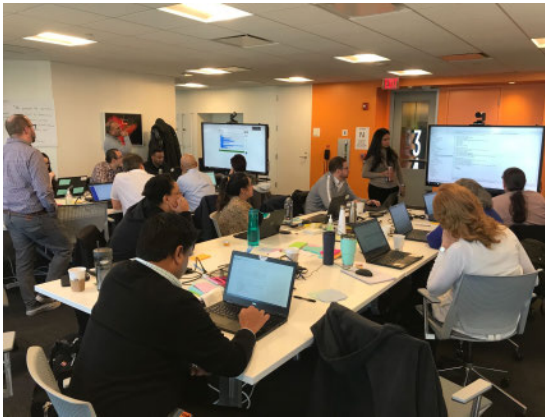
Tables can be easily moved to allow teams to work in the manner that best suits them. TVs located at both ends of each table allow teams to display content when they are building solutions. Situating the information radiator in the middle of the room divides the Dojo spaces and enables easily accessible display of up-to-date information.

Take a look below at our snapshot of the ideal Dojo space layout and design:



We've found that the Dojo layout described above fosters transparency, collaboration, and effective teamwork.

## Examples of Dojo Spaces created by Liatrio



## Conclusion

A good Dojo space fosters a culture of learning and enables teams to collaborate, learn and accomplish capabilities together. A physical space gives teams a sense of disconnect from the existing way of working and inspires them to come to the Dojo to learn new ways of working while working on their real world product/services. Enterprises will benefit long term with this culture of continuous learning and improvement as teams become high performing and productive in delivering value to customers.

We recommend not to go big bang with your Dojo space for the first few teams coming through the Dojo. Starting off with a small space or even a large conference room should suffice, learn from the first few iterations and apply them when building out your own dojo space. Within the small space, try to put up team's artifacts on the walls/windows (ex: are architecture diagrams, team norms, skill matrices, etc.) Most of the Dojos we set up for the clients have started off in a conference room. You want to build a brand for your Dojos and slowly make the dojo space more open, creative and fun, in order to attract and inspire teams to come and learn together.

The Dojo will evolve over time, the key is to get started, learn and adapt along the way.

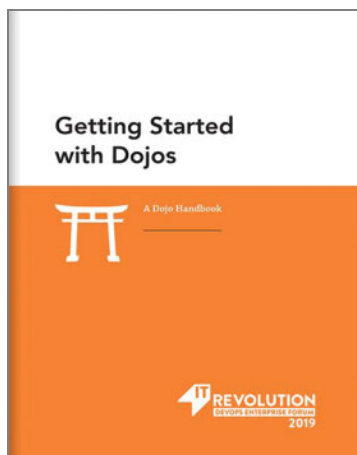
## References

IT revolution's recent book on [Getting Started with Dojos](#) and Joel Tosi & Dion Stewart's recent book [Creating your Dojo](#) talk extensively about Dojo space and the value for it. Some of our favorite quotes:



*“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”*

*“Space can be a very powerful enabler for culture change and learning. Try to put your Dojo in a high-traffic area so that you can broaden the exposure of what teams are experiencing.”*



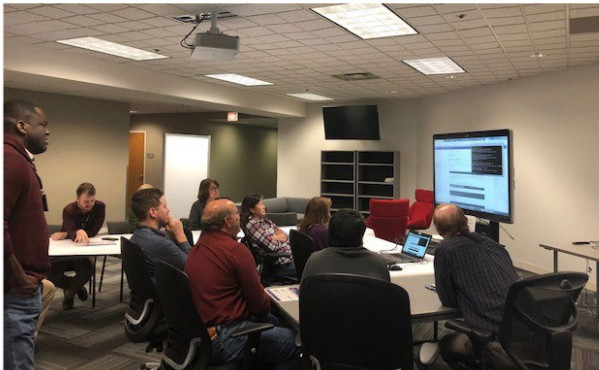
*“A dojo space is supposed to feel different than a standard working environment. Having a separate physical space demarcates the dojo as something different from the normal day-to-day work.”*

*“First and foremost, a dojo space is designed to stimulate and support learning and is well served by the same visual display tools that you find in a studio.”*

[Dojo Consortium](#) is a community of Dojo practitioners from 30+ organizations and are connected together via slack, annual gathering, bi-weekly video conference, regional meetups, etc. to share and learn from each other's challenges and accomplishments. Special thanks to the consortium members for sharing pictures of their Dojo for our blog.

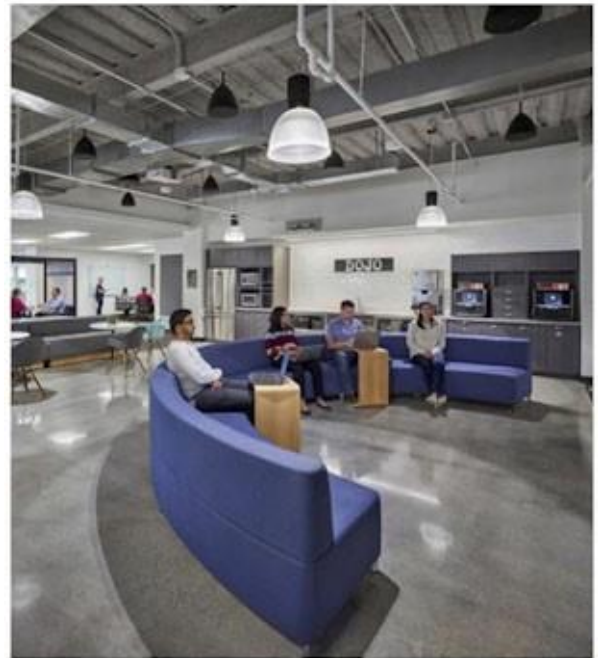
## **DELTA TEAM MOBING ON AN ISSUE**

USED WITH PERMISSION. COPYRIGHT DELTA



## **TARGET'S DOJO**

USED WITH PERMISSION. COPYRIGHT TARGET



## **VERIZON'S NEW JERSEY DOJO LOUNGE**

USED WITH PERMISSION. COPYRIGHT VERIZON

# Hiring & Staffing Coaches

---

Coaches will make or break your Dojo experience, so hiring and staffing a Dojo with skilled coaches is very critical for the success of the Dojo. On a high level, a coach should:

- Have the confidence and expertise on product, lean and engineering practices.
- Be able to inspire engineers within your company to listen to them.
- Understand the big picture & assist in showing teams the way.
- Possess a blend of good engineering and coaching skills.

Dojo coaching squads should be formed and aligned based on findings from the Dojo Discovery<sup>1</sup> phase that is done for a value stream or business unit level.

We recommend pairing Dojo coaches for each Dojo experience<sup>1</sup> and be assigned to a team based on the team's outcomes and goals identified during Charter<sup>1</sup> as part of Dojo intake phase<sup>1</sup>.

Coaches should have opportunities to continuously learn and improve based on their Dojo experiences, creating and building Dojo as a product and rotate from regular coaching assignments to embed as hands-on engineers with platform teams so they can learn and grow by doing.

## Focus Area For Dojo Coaches

A coach cannot have breadth AND depth in all areas but rather he/she should be an expert in 2 – 3 focus areas with a depth of knowledge & experience. These focus areas include Agile/Lean, product design, product development, architecture, engineering & development practices, quality engineering, cloud native engineering, etc.

### Engineering

- End-to-end software delivery lifecycle skills & perspectives to troubleshoot team's challenges by working hand-in-hand
- Technical framework knowledge & ability to speak to POs, BSAs, SMs, Devs, Quality Engineers/Analysts, etc.
- Expertise in cloud & cloud native, pipeline, quality engineering, app security, infrastructure automation practices and tools

---

<sup>1</sup> discussed in detail in upcoming sections of this playbook

## Product and Lean

- Expertise in Product breakdown, feature mapping, design thinking to drive customer first way to thinking for the teams
- Process expert and drive immersive learning to uplift new teams and reinforce/re-learn core iterative concepts
- Pragmatic approach to the application of Lean, Agile & DevOps practices to guide teams to the solution

## Knowledge and Skill Competency

Expanding from the focus area for Dojo coaches, in order to successfully enable Product (Software Delivery) teams, Dojo coaches need to be prepared across four general categories:

1. Behavioral / leadership skills
2. Technical knowledge / skills
3. Soft skills
4. Program knowledge

Each category represents a potential capability that either a coach has currently or will need to uplift or upskill.

Category	Evaluation Dimension	Description
Behavioral /leadership skills	Able to teach how to find solutions vs. giving answers	Coaches are change agents so for long-term change, coaches should not only give answers but lead Dojo participants to the answer. There's a give-and-take in this process, but Coaches need to be able to judge what level of detail they should or shouldn't go
	Critical thinking/ problem capacity	This capability relates to the coaches ability to anchor on data around them (e.,g. concerns from dojo participants, pipeline failures, etc.) to identify what actual root causes could be (e.g. lack of any good practices) and implement sound logical solutions that create a mixture of short-term and long-term consequences.
	Thought Leadership	Coaches will not only be evangelists but will also need to, at times, drive strategy and approaches

Category	Evaluation Dimension	Description
		for Product teams that have gaps (e.g. if Product team is missing a QE and only have QA, what should the team do)
	Ability to help others	This is a core mindset of a coach — the focus on helping and enabling Product Teams is what all coaches need to anchor on.
Technical knowledge /skills	Technical aptitude focus on pipeline / software delivery (skill, knowledge)	Coaches will need to have an end-to-end Software Delivery perspective, as well, skills to supplement. Coaches will need to be able to demonstrate skill but also to enable stronger Product delivery teams, needs to be able to know and convey key engineering practices Additionally, pipeline skills are one the core competencies of a Dojo Coach will need - the ability to navigate the run of a standard pipeline for a sandbox application Dojo Coaches will need to be able to understand concepts across software delivery at fundamental level and enough to triage and solve.
	Understanding and ability to run DevOps pipeline	Coaches need to understand: <ul style="list-style-type: none"> <li>● How a pipeline is architected</li> <li>● What capabilities it has/solves for and what it lacks</li> <li>● Why it was architected</li> <li>● What are potential upcoming features / pipelines</li> <li>● What limitations does the pipeline have? (if possible, also how do we solve it?)</li> </ul>
	Iterative methodology	Knowing the different capabilities to help uplift new teams (e.g. transitioned from Waterfall) or reinforce/relearn core iterative concepts (e.g. hypothesis-driven, Scrum basics, etc.)
	Product structuring	Coach will need the ability to understand at a high-level what the product is and if the product at a high-level has/is missing structure.

Category	Evaluation Dimension	Description
	Software Delivery Principles & Frameworks	<p>In the Software Delivery space, Coaches will need to have technical frameworks to always rely on as Coaches need to have the ability to speak to POs, BSAs, SMs, Devs, Quality Engineers/Analysts, etc. Core frameworks Dojo coaches need to know are:</p> <ul style="list-style-type: none"> <li>● Software delivery lifecycle</li> <li>● Modern architecture / engineering practices (ex: DDD, containerized, cloud-native)</li> <li>● Core Infrastructure framework</li> <li>● Product strategy / planning</li> <li>● Iterative approach(s)</li> <li>● Test automation &amp; approaches (ex: BDD, TDD)</li> </ul>
Soft skills	Empathetic	With the massive level of change affecting Product teams, Coaches will need to learn to be patient before solution-ing to really tease out root concerns (e.g. are they just venting or is there something else that's underneath the concern/question).
	Communication /Presentation	Coaches need to demonstrate ability to clearly communicate and convey both technical and non-technical concepts. As well, Coaches need to demonstrate comfort in common public speaking as presenting and facilitating will be common norms to help resolve issues and/or brainstorming.
	Facilitation	Ability to control the room and ensure that topic/goal is met for a given period (e.g. if topics don't make sense for the team to adjust approach or content, etc.)
Program knowledge	Understanding of DevOps ecosystem	Coaches should have situational awareness and understanding of the Dojo and DevOps program and key stakeholder. In addition to that, have a working relationship with the platform team engineers and leads and be aware and in sync with the latest and greatest of the capabilities being offered.

## Interview Approach for Coaches

Purpose of this screen is to understand and assess if the coach candidate can speak technically about certain topics. We should also have them explain a technical topic in layman's terms. Have a general conversation with the candidate. It keeps us engaged, gives them the ability to showcase their talents and strengths and covers what a coach needs. Here's are some examples:

1. Intro and pleasantries
2. Understand where they excel/lack
3. See how they present technical topics they are familiar with
4. How can they learn the stuff they aren't familiar with
5. How they engage technically with others
6. Take a topic (i.e. testing) then ask them to: (4 or 5x)
  - a. What is it for
  - b. What are some tools available
  - c. explain one tool technically
  - d. explain the tool's purpose and use in layman's terms.
7. Ask them about how they learn and apply new topics. What's an example?
8. How have they coached technical topics before? What's an example?

The questions below are recommended to be part of the technical screen. Have some open-ended conversations to allow the candidate to show us their best traits and for it to feel more like a conversation than a grilling.

## Getting Started

Standard meet and greet section.

- Remind the interviewee we are recording
- Give the interviewee a brief background of your history with the company
- Tell the candidate who you are and your background a little bit.
- Let the candidate tell you who they are and what they have done.
- Ask the interviewee for a brief background on their history and let them know if you've gone through the previous interviews so they don't go too in depth.

## Area of Focus

This section should allow the candidate to get more comfortable. It should also give you the ability to get some good insights about their favorite area of expertise.

Ask about what area of tech/IT/tools that they are most familiar with. Probe about:

- Use cases
- What & How they learned it
- Have they trained anyone with it
- What do they like/dislike about it

## Technical Questions

Remove the duplicate topic if the candidate already touched on one of the topics during the previous question.

During this section I will be asking you about a broad topic — testing, for example. What I would like to hear from you is a 1 minute answer that provides your understanding of the topic, some tools or techniques you've used in this topic area (how/why), and pick one tool and explain its value and use to me if I was a non-technical user.

- Give an overview of your understanding of the topic (i.e. what's it for, when is it used, etc).
- What are some tools and techniques used in the topic area?
- Explain one of the tools you listed above and go in depth about its use.
- Explain the same tool as if I was typically non-technical.

### Topics (1min each)

- Automation/CICD
- Engineering practices for a team (i.e. what does a team being cross functional mean, what skills would you coach them to be one?)
- Test Automation
- Version Control Tools
- Build Tools
- Containerization
- Local Dev Tooling
- Infrastructure as Code

### Identifying Learning & Coaching Experience

The next questions are about the candidates and how they learn/teach/coach.

- How do you learn and apply new topics? Give some examples.
- When was the last time you taught or coached another person on a technical topic?
  - What was it?
  - How did you teach it?
  - What was the outcome?
- You need to coach a team on a technical topic that is new to you. How do you approach this?
- Tell me about a time when you overcame resistance to change around a new tool or process.
- You hit a roadblock while trying to learn a new technology. How do you go about resolving those roadblocks?

### Guidelines for Growth & Expansion

Here are some guidelines and standards to evaluate, and establish continuous learning for coaches and drive success:

- The biggest learning and uplift for coaches is to ensure they rotate as engineers to actual platform and product teams on a regular cadence.
- Ongoing learning is very important and to enable that Coaches should treat Dojo as a Product and own them to build, learn and add more capabilities. When coaches are not active in Dojo, they should be building/improving their Dojo product.
- Coaches should be pairing with other coaches in the Dojo to learn from and grow each other's capabilities.

- Feedback and NPS survey from the participant would help assess the Dojo experiences and help the coaches to learn and adjust for future teams as needed.
- Periodic coaches alignment and retrospective discussion per Dojo and overall for the Dojo program will help the coaches to learn and improve from others.
- Recommend to pair the coaches to execute Dojo to provide the best coverage and experience, ideally you would want to pair a technical and a non-technical (Agile/Product) coach.
- Set standards for the Dojo coaches, these standards will help normalize the workload among the coaches and help form a cross-functional Dojo coaches group. Some examples to consider:
  - Execute X no of Product Dojos in a year
  - Execute X no of Flash Dojos in a year
  - Every coach should run dojo consults and charterings
  - Execute a leadership dojo to solve specific leadership challenges and outcomes
  - Expand the Dojo product with at least 1 new capability/offering (technical/non-technical)
  - Represent Dojo outside of the organization (like participating and contributing to local meetup and conferences)

## Conclusion

Hiring the right coach is challenging as not all good/great engineers can be coaches and vice versa. There are certain factors that make a coach special and fitting. Identifying them is very personal for every organization as the shared values are unique and different from each other. The above is just a set of guides and recommendations that would need to be interpreted, adjusted and tailor-made for an organization to hire the best coach.

# Dojo Intake

---

A Dojo intake process or journey should be designed specifically for your organization before you start bringing teams through the Dojo. All product teams going through a full Dojo experience and some flashbuild experiences should follow a similar intake process. Certain aspects of the Dojo intake (example: logistics, artifact creation, participants etc.) should be customized to your organization.

## Consult

Dojo consultation is the first step in the Dojo intake process. The purpose of the consult is interview Dojo participants and key stakeholders, which serves a dual purpose:

1. Coaches and Dojo operations<sup>1</sup> experts give a brief introduction about the Dojo, share any past experiences and explain the Dojo journey to the team members, leadership and key stakeholders.
2. Gain a high level understanding of their product, obtain enough technical and team information to determine if the team is a good candidate for the dojo experience and is ready to proceed to next steps in the process.

Attendees for a consult or initial interview include: product owner, scrum master, senior developers, and other roles on the team should be represented by at least one person.

Below is a comprehensive checklist we have developed based on a number of consultations and initial interviews our experts have done so far. Use the checklist as a guideline to have a conversation with the participants and stakeholders to help the team and the coaches to get started with their Dojo journey.

Typical items that should be touched on during a consultation:

- Must**
- What is the primary outcome the team is looking for in the Dojo?
  - Any secondary outcome the team is looking for in the Dojo?
  - Is the team full stack and can take the product from concept to release?
  - Has the product been defined? (looking for a clear understanding of what the team is responsible for, and they understand the product boundaries).
  - Is the entire team able to co-locate or participate in the Dojo?
  - How long are they able to commit to the Dojo experience?
  - Is the team composition complete? Will there be any changes to the team over the coming months (looking for a complete team that will be in the Dojo, and no changes to team like dis-banding are in the immediate forecast).

- Should**
- Are there any release, delivery, prod support, deadlines that fall within the time the team is in range for Dojo? (looking to understand if these disruptors to the dojo experience will have an impact and if so alert devops operation team to shift schedule to avoid.
  - Is the team ready to improve their current way of working? (incremental delivery, cross functional skills, learning culture)

- Could**
- Discuss with the team the app, tech stack in use (does the team have the skills to support each item in the tech stack?) pipeline (if any), challenges the team is working with.
  - Is the team using team collaboration tools today? Ask them to begin those requests and get access to them.
  - Understand their current delivery methodology being used?

From the initial consultation, make sure to establish a single point of contact for the team as well as for the Dojo coaching team to enable constant communication to the team and other interested parties throughout their Dojo journey.

## Discover

Discovery is the first step in the lifecycle as you are designing and preparing to execute Dojos for your organization, Line of Business or a value stream within your organization. This session is designed to help discover the scope, organizational challenges and identify gaps and opportunities to restructure product teams and support teams to have a successful dojo experience. A discovery session is generally tailored to a specific line of business, value stream and domain within an organization.

Our discovery sessions typically achieve the following:

- Provide the key players and stakeholders a solid understanding of Dojo and capabilities offered
- Using techniques value stream mapping and lean process mapping to identify opportunities, challenges and wastes
- Common understanding and alignment on organizational structure and technical gaps
- For tightly coupled value streams, identify opportunities to decouple applications and teams
- Create a high level matrix of product, applications, teams technical stack and
- Understand current composition of product teams, roles and responsibilities
- Identify shared service and/or functional areas to provide support to product teams
- Identify lighthouse teams and champions for Dojo and build a roadmap for the Dojo

## Key Players & Stakeholders

- Architects & tech leads
- Product managers/owners
- Development & test leads
- Platform, infrastructure release engineering leads and experts
- Functional area leads and experts (security, risk & compliance)
- Process owners
- Dojo coaches and leadership

## Charter

Chartering is the preliminary step in the Dojo life cycle which helps the team to articulate their current challenges and identify their goals and outcomes of the Dojo. This session is generally done before inviting the team to the Dojo and the entire team including key stakeholders, partners and sometimes the leadership should be involved. This session should be led by the coach(es) who is going to take the team through their Dojo experience. This is the first step and opportunity for the coaches to build relationships and trust with the entire team.

## Objectives

- Obtain a solid introduction to Dojo and align on expectations
- Align on the product vision (elevator pitch)
- Understand the application architecture & flow including internal and external partners
- Technical assessment and gaps
- Map out team's path to production using Lean Process Map and identify opportunities
- Identify current challenges with team including the product, process and technology
- Identify success criteria and carve out outcome/goals for the Dojo

## Sections of Charter

Section	Description	Artifacts
Introduction to Chartering & Dojo	<ul style="list-style-type: none"><li>• Layout the schedule for the day, logistics, hand gestures, etc.</li><li>• Provide a good overview of the Dojo (What and Why) align with the team on success factors for the Dojo. Get the team excited and get them know more about the Dojo.</li></ul>	None
Product Elevator Pitch	<ul style="list-style-type: none"><li>• What your elevator pitch for your team' product (one or two sentences)</li><li>• Come up with your elevator pitch for your product as a team</li></ul>	Elevator pitch / product vision
Process Flow	<ul style="list-style-type: none"><li>• What does a customer journey or experience look like?</li><li>• What is the Business Impact?</li><li>• Who are the end customers?</li></ul>	Wiki page with answers to

Section	Description	Artifacts
	<ul style="list-style-type: none"> <li>● Functionality you're trying to deliver?</li> <li>● Expected timeline from the end customer?</li> </ul> <p>Has this been established?</p>	all the questions
Architecture /Service Interactions	<ul style="list-style-type: none"> <li>● Get an in-depth understanding of their current application architecture, components, dependencies and service interactions. If a detailed architecture diagram exists then have the lead/architect give a walk through.</li> <li>● Get a horizontal diagram and facilitate discussion of the Applications and Mnemonics that the teams interact with consistently, and end up having dependencies on.</li> </ul>	Updated and clear architecture diagram If whiteboard, Pictures uploaded to confluence
Technical/ Team Assessment	<ul style="list-style-type: none"> <li>● Review the technical stack of the application and identify any team gaps in delivering technical stack.</li> <li>● Team structure, roles and composition, identify any challenges.</li> <li>● Understand teams current experience with delivery and ownership</li> <li>● Current ways of working.</li> </ul>	Wiki page list of technical stack with any gaps
Path to Production (Lean Process Map)	<ul style="list-style-type: none"> <li>● What does the current path to production look like?</li> <li>● Identify the wait times between and lead time.</li> <li>● Breakdown the lead time and wait time by Small, Medium and Large tickets or work items.</li> </ul>	Lean Process Map (castle wall diagram) Waste
Consolidate Challenges	<p>What is a Challenge: Any disruptors to team both internal and external to their product delivery lifecycle.</p> <ul style="list-style-type: none"> <li>● Call out and capture throughout the chartering.</li> <li>● Add and additional challenges and consolidate.</li> </ul>	Challenges listed and ranked

Section	Description	Artifacts
Formalize Outcomes	<ul style="list-style-type: none"> <li>Identify which challenges can be turned into outcomes.</li> <li>Add any additional outcomes the team wants to accomplish in the dojo.</li> <li>Team will dot vote and provide a prioritized list of their outcome/goals from the Dojo.</li> </ul>	Goals/ Outcome wiki page

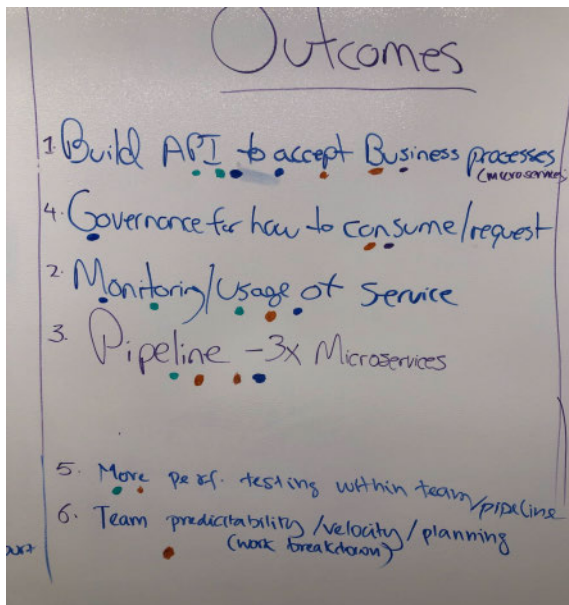
Additional	Description	Artifact
Community Map (optional)	<p>A product community map will layout key players who collaborate on key decisions. We need perspective and understanding from others that vary to give guidance on the product. Key stakeholders can often be affected by an organization's actions, objectives and/or policies. Some of the key dojo stakeholders are</p> <ul style="list-style-type: none"> <li>Sponsors: Essential stakeholders that determine the potential of a project or product.</li> <li>Customers: Anyone interacting with or influencing the product all the way up to the end user</li> <li>Builders: Developers and engineers and determine if the product is feasible and come up with an implementation plan, cost and impact.</li> </ul>	Visual vent diagram showing 3 key roles and players within
Application Flow Collaboration & Consolidation (optional)	<ul style="list-style-type: none"> <li>Separate out by Role and spend 15 minutes talking about. "How is work taken in/handed off?" "What challenges do you face in your role?" and "What human/process dependencies do you have?"</li> <li>Use the hour time to bring everyone back, and have one person from each role walk through each page, discussing what's there and facilitate questions, handing off to the next role after each.</li> <li>Roles challenges and flow hanging up in the dojo.</li> </ul>	Input > Action > Output artifact for each role Dependencies and flow of action between them

## Logistics

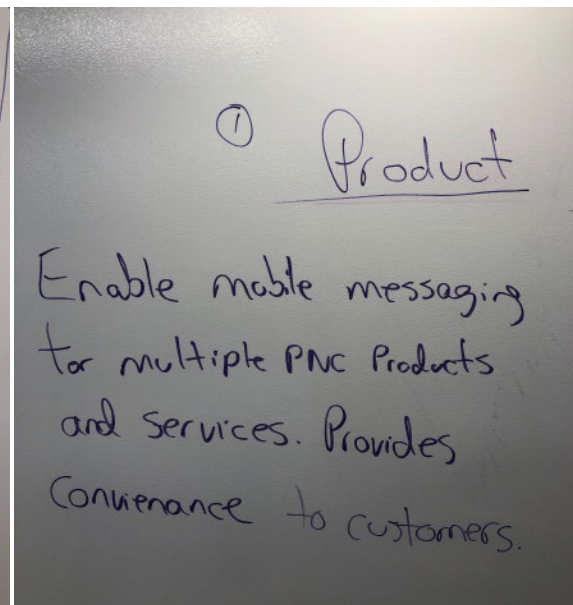
- 4 – 5 hours session
- Encourage every team member to be present in person
- Large conference room or Dojo space with video call facilities for remote participants
- Supplies
  - Whiteboard wall (large) or at least 3 whiteboards
  - Large Flipchart
  - Colored post-it notes
  - Sharpies
  - Name tags
- Snacks

## Examples of Chartering Artifacts

### Dojo Outcomes



### Product Elevator Pitch



## Challenges

Challenges

Replacing the vendor w/ in-house  
technical debt

Source of truth for data  
issues

API Security, currently open  
access

Mapping customer ids across  
multiple systems

Consumers request on-demand, support  
challenges

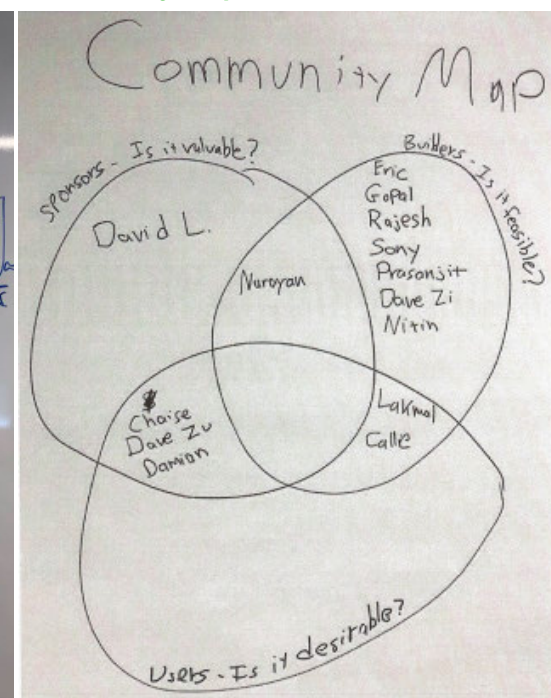
## Technical Assessment

- + JPA
- Mongo
- + Oracle + Micro-Services
- + Java / JST
- + Java Springboot
- + Docker
- + OpenShift
- + Swagger
- + Maven
- JScript
- + Karate (w/ cucumber)
- + Selenium
- + JUnit
- + Mockito
- + Spring Test
- Sonar

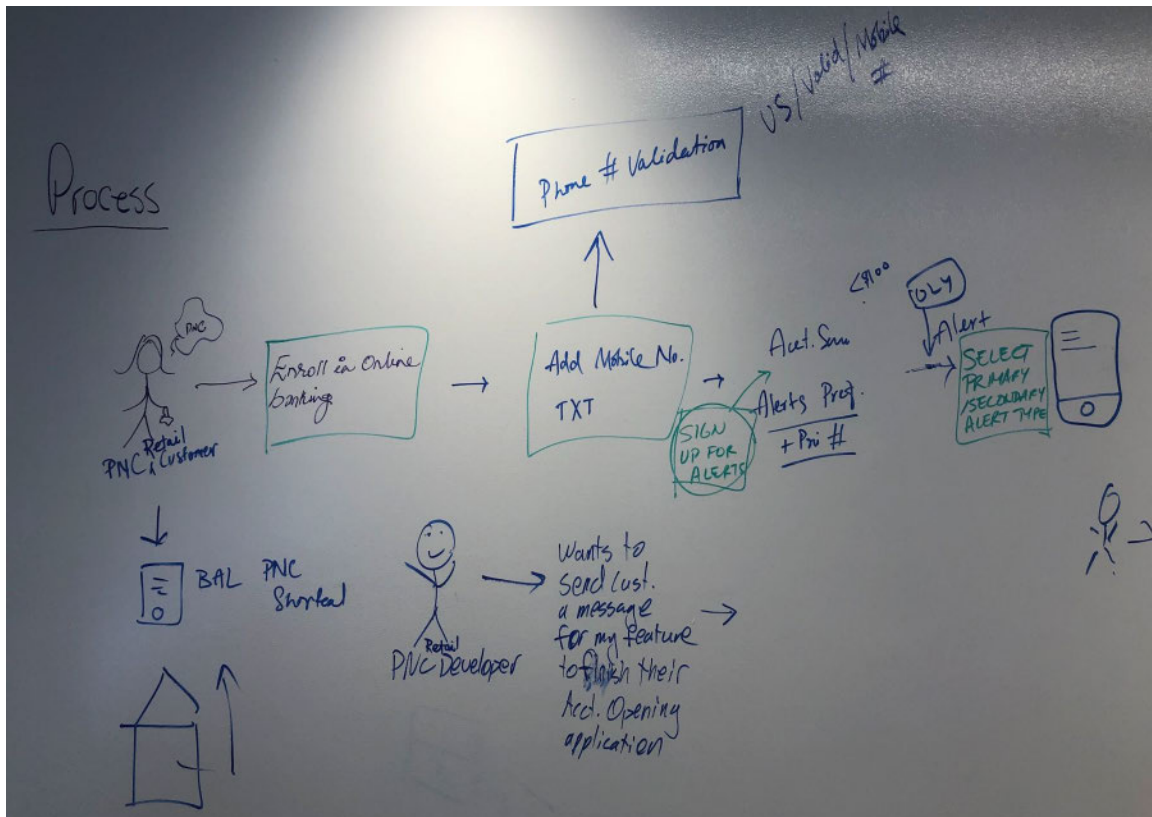
## Architecture



## Community Map



## Business Flow



# Dojo Experience

---

A Dojo experience could follow a common template on an operational level but you will realize the most success when Dojo experiences are custom made for a team based on their needs and outcomes discovered in dojo intake. In order to create a best dojo experience, we recommend to focus on the following key areas:

- Team Preparation
- Coaches Preparation
- Day 1 Experience
- Hyper-sprints (two and a half day sprints)
- Immersive Environment
- Embedded Coaching
- Outcome drive
- Demo and Celebration

## Team Preparation

Between the Dojo charter and day 1 of the Dojo, all team members would need to prepare for their Dojo experiences. Some areas to prepare are the following:

- Team need to address any action items related to their product (vision, roadmap, research), personnel (assignment, commitment, travel) and others identified from the Dojo charter
- Address all critical gaps based on the technical assessment from chartering by bringing the required personnel to the team and/or completing any foundational training/workshop.
- More matured Dojo practices can have some elearning or self learning modules as prerequisites depending on the outcomes they want to achieve in the Dojo
- Tools access and local machine setup to perform tasks in the Dojo.
- Close out any current work and active interaction in their backlog to start new hyper sprints in the Dojo
- Refine and get their backlog ready for the Dojo with coaches' help.
- Visit the Dojo a few times to get familiar with the space and setup and have casual interactive with coaches and active Dojo teams.
  - Optionally the team explores and gets familiar with past teams' experiences by seeing recorded demos, short videos, pictures of past dojos including any artifacts created by the teams in the Dojo.

## Coaches Preparation

- Alignment with the platform to ensure all the capabilities that the Dojo team would need can be supported and clarify any security and risk concerns.
- Line up support and external expertise needed for the team based on their charter
- Prepare and set (or reset) the Dojo space for the team to come in including arrangement of tables, chairs, monitors, whiteboards and supplies needed like post-its, flipcharts, markers, pens, etc.
- Socialize Dojo outcomes and any pre-table challenges to the leadership team and line up support for the team coming into the Dojo
- Consider any special needs for the participants to make the Dojo experience and environment inclusive and accommodating for all members of the team
- Create an high level checklist for the coaching squad to be sync
- Setup daily coaches for the coaching squad, operations and key platform support engineers to sync for alignment, retrospect and make any course adjustment
- Consolidate all physical and online artifacts created from the charter and bring it to the Dojo. For example:
  - Outcomes/Challenges
  - Product elevator pitch
  - Tech assessment
  - Architecture diagrams
  - Business process flow

## What Does Day 1 Look like?

Day 1 of the dojo sets the tone for the entire Dojo, the participants and the team members are sometimes skeptical and need reinforcement from the coaches on what to expect in the Dojo and how the team and coaches will be working together.

The coaches need to set some expectations on the dojo environment, here are the principles we follow in our Dojos.

- Dojo is a place of learning and improving as a team.
- Come with an open mind.
- Change is hard and happens one day at a time.
- All ideas are welcome.
- Trust the process.
- Practice! Practice! Practice!

As part of day there are some preliminary elements that would need to be establish before the team and coaches start working on the outcomes

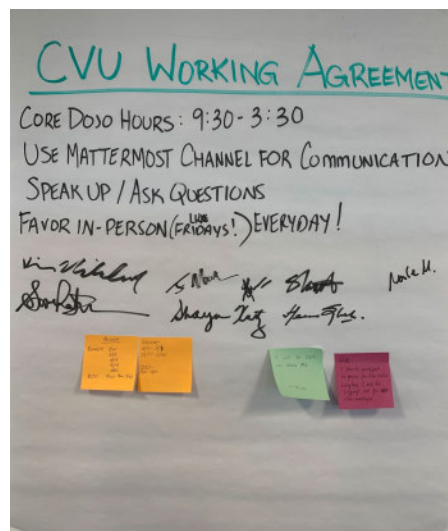
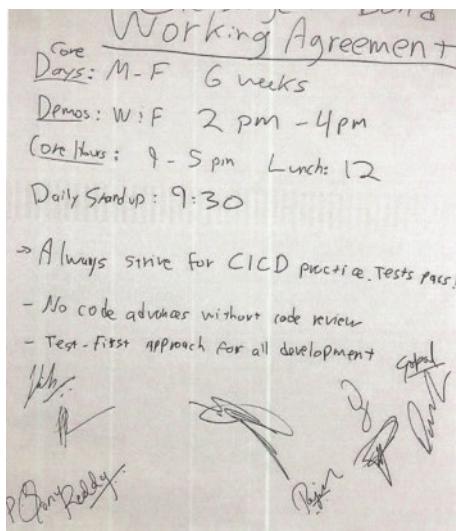
## 1. Working Agreement

The working agreement is an informal agreement between the participants and coaches to work together to accomplish the outcomes identified in the charter session. This working agreement is a living document and can be adjusted/modified as the Dojo progresses

The working agreement frames how the team will initially work together. There are certain items that need to be established on day 1 like core hours, ceremonies, pairing, etc.

Example working agreement:

- Core hours are M – F, 9:30 – 4:30pm
- Weekly team lunch
- Monthly team event (fun)
- Demo & retro will be on Wednesdays and Fridays
- Team will swarm with master build is broken
- We will always work in pairs
- Every PR should be reviewed by at least 2 engineers
- Clean up branches end of every sprint
- Expectations and commitments on social aspects like meetings, communication, how they interact with one another



## 2. Skills Matrix

The skills matrix is a visual representation of individual skills and competencies. This skills matrix includes the participants in the Dojo what they know currently and what they want to learn. Skills that you need to learn can also be outside of the Dojo. This needs to be done day 1 and final day, without exception, sometimes can be visited during the Dojo.

The skills matrix helps with the following:

- Identify skills the team needs to be successful and how will they acquire the new skills
- Assist in looking for learning and pairing opportunities as well as creating opportunities for teams to help each other
- Show progress on learning and improvement

## 3. Feedback

Coaches need to establish a mechanism to obtain feedback on a regular basis, one of the techniques we use is to use post-its and provide feedback at the end of the day or throughout. This will help teams to provide honest and instantaneous feedback directly to the coaches and coaches can adjust the dojo experience and environment based on the needs.

Note: This is in addition to retros as part of hyper-sprint, the feedback provided here is more from the aspect of the Dojo environment, experience, etc. whereas retros are from the team's perspective around their current work and things they have delivered.



## Hyper Sprints

After all the preliminary setup, now the actual work begins. The team, while working on the outcomes, learning, and applying new skills in the Dojo, follow a hyper sprint cadence — two and a half day sprints. A week is broken down into 2 sprints, Monday to Wednesday morning and Wednesday afternoon to Friday. Every sprint will have a

- Brief planning session
- Daily stand ups
- Demo
- Retros
- Additionally backlog refinement session for future sprints

### Why are Hyper-Sprints effective?

#### 1. Repetition Helps with Learning

This hyper sprint cadence helps with repetition and the repetition helps with learning and mastery. Coaches often introduce and teach new techniques and practices in a sprint and the team can practice across multiple sprints helping them to learn, apply their work and attain mastery. These shorter iterations help with the learning to stick and teams get better when they practice frequently especially when learning new skills.

#### 2. Fail Fast & Fast Feedback

With hyper sprint cadence, teams can experiment new things, obtain fast feedback. This enables a mindset of experimentation and provides a positive environment to fail fast. This helps during the phases of product research, technical spike

#### 3. Small Batch Mindset

These shorter iterations force the team to break down work to the smallest possible unit to complete them in two and a half day sprints. When teams practice a small batch way of working, their delivery pipeline is exercised multiple times helping it to be reliable and dependable for every code change they make. Team transforms their thinking from “How big or how small” to “it is too big” when sizing their stories/tasks.

#### 4. Inspect and Adapt Frequently

In two and a half day sprint iterations there is a cadence of planning, daily stand ups, demo, and retro. The retrospective that happens at least twice a week helps the teams and coaches to reflect on how the team is progressing to accomplish their outcomes, address new challenges, and adapt the course for the upcoming weeks. This sometimes can help the team to adjust the outcomes if needed based on lessons learned and experience thus far in the Dojo.

#### 5. Experiment with Safety Nets

The shorter iteration prevents members of the team from going further without feedback from the team or collaborating with others. On the other hand it helps with the experimentation mindset of trying something new with the help and guidance of coaches and other expert team members. The experimentation applies to both product and technical side, where the team can experiment product features and also new technical practices, frameworks, or tools.

### Example of a Week in the Dojo with Hyper Sprints

#### Week 1

Mon.	Tue.	Wed.	Thu.	Fri.
<b>Sprint 1</b>				
Kickoff Sprint Dojo Planning	Standup Standup Refine Refine	Demo Retro		
			<b>Sprint 2</b>	
			Sprint Planning	Standup Standup Demo Refine Demo Retro

### Immersive Environment

In an immersive Dojo environment, solitary work is discouraged, team collaboration increases, and distractions are limited. Learning retention and the overall success of a Dojo increase when the experience incorporates work on relevant, real-world products.

The best learning and growth happen only when people apply new skills to their product backlog and are able to break past existing organizational constraints.

For starters, “immersive” is an experiential style of engaging the team at as many levels as possible to facilitate DevOps learning. These levels include the following:

- Visual
- Verbal
- Physical (Kinesthetic)
- Logical
- Social

Any Dojo or activity can be done in an immersive style Dojo which implies both that the coach(s) are embedded and engaged throughout the Dojo (Dojo length doesn't matter) AND coach(s) plans out day/week/Dojo experiences to deliver the outcomes the team creates during chartering in engaging, team focused methods.

Refer to the Dojo setup section of this playbook for more guidance on setting up a successful Dojo space for immersive learning experience.

## Embedded Coaching

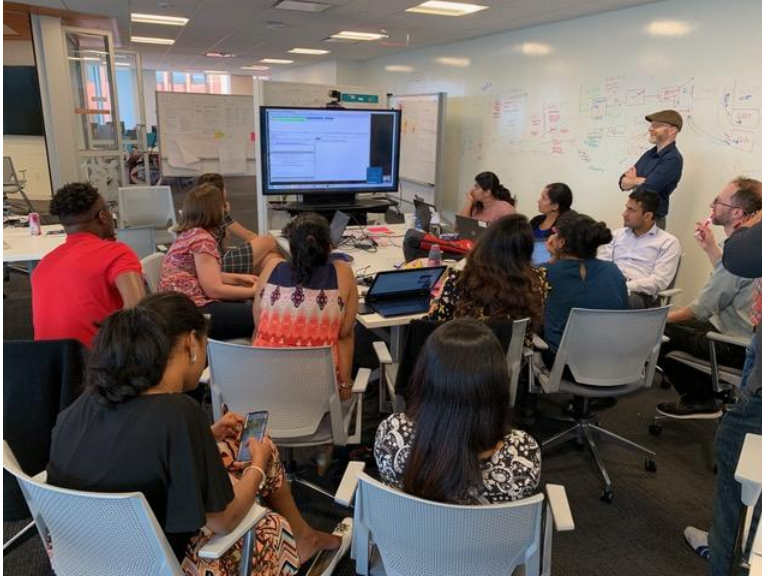
First and foremost Coaches become team members and are embedded with the team for the duration of the Dojo. What does embedded (aka 'sitting with the team') mean for the coaches?

- During daily coach sync (done prior to dojo starting each day) coaches plan out the team's activities/using sprint planning and stand ups to ensure content or activities are getting done to deliver on outcomes.
- Be 'at the table' which means listening to the team, getting them to share what is on the screen, asking questions about what is displayed, why/how/what they coded/tested what they did, observing, learning their strengths and weaknesses from ideation to releases, support, all of the product activities.
- Help find resources if you cannot provide them to the team as the coach.
- Teach the team strategies for solving their problems with skills they can take away from the Dojo.
- Coaches are for all intents and purposes are members of the team. It's their job to teach the team strategies for solving their problems, help them build skills that they can apply outside of the Dojo, and help them recognize their wins, both big and small.
- Coaches join in sprints, retros (where multiple team members come together to share development progress, learning, and improvement opportunities), ceremonies, and demos and offer timely, helpful advice and encouragement to motivate the team to transform their habits, and think in a way that accelerates and improves their development of quality code.

- Optionally, coaches can do one on ones with team members like (SM/POs) to learn more about the team challenges, moral, strengths and weaknesses, goals, and other parts of the team dynamics that will aid in delivering outcomes.
- Keeping a daily log of observations, history, photos of the team and their activities, artifacts of the team (which can be used with other coaches to build on similar outcomes) all of which serve as a history record if the team returns for follow up Dojos.

## Examples of Immersive Experience & Embedded Coaching





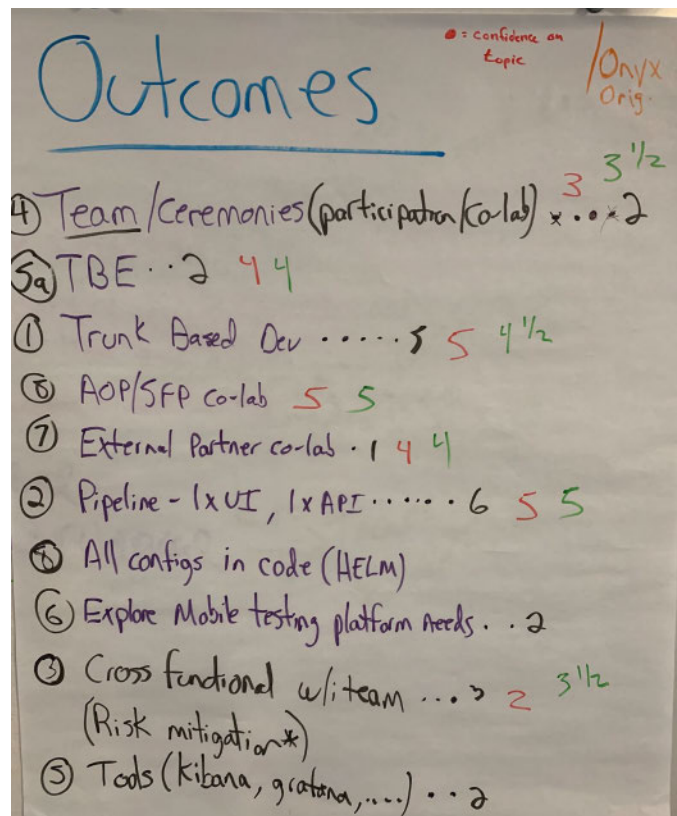
## Outcome Driven

The work accomplished by the team in a Dojo experience should be driven by the outcomes that the team came up with and agreed upon during the charter. With the help of the coaches, the learning goals are created as tickets in the team's backlog and worked alongside their product feature/story development stories/tickets.

Here are some guideline to ensure the team is focusing and delivering on their outcomes:

- Coaches, while preparing for the team's Dojo space, bring in the artifacts (especially outcomes, challenges, product pitch) to the Dojo and place it around where the team can see it.
- On the day of the Dojo, revisit the outcomes to identify any changes or adjustments needed.
- At the end of every two and a half day sprint, the entire team should visit the outcomes and do a fist of five vote or confidence vote on it. This will help validate progress and outcomes the team has accomplished.
- Use tags/labels in your work management tool (like JIRA, Version One, etc.) to trace the work back to the outcomes
- Any work that is not directly or indirectly related to the team's outcome should be treated as unplanned work, unless or otherwise, the team as a whole is adjusting or adding new outcomes during the Dojo.
- Based on active work, teams will often discover new findings which may warrant new outcomes and this is absolutely fine as long as the whole team is onboard with the changes/addition. The hyper/mini sprints model will help with any course correction without too much disruption.

Here is an example of a team tracking progress and confidence on their outcome:



## Demo & Celebration

The last and final part of the Dojo experience is the demo and celebratory exit.

### Demo

- Demos are an organic part of the Dojos, as part of the hyper sprints (two and a half day sprints) there is demo and retro every Wednesday and Fridays. This is an opportunity to showcase progress to the team and also throughout the Dojo as needed.
- Towards the end of the Dojo, the team should do a broader demo with the leadership and other stakeholders to showcase the teams work and accomplishments in the Dojo
- This experience is designed to bolster team work, engage the team more fully, and provide insight into more of the app than typically seen by all of the team.
- Leverage the demo lounge and invite other dojo team (current and past) during this demo.
- These demos also help future Dojo teams to see the outcome and experience of the Dojo as delivered by the team directly.

### Celebratory Exit

It is important for the team to celebrate this journey, experience and accomplishments from the Dojo. This helps to solidify and make the experience in the Dojo even more 'sticky' by recognizing the achievements as a group of what was accomplished. We recommend having a celebratory exit for the team when exiting the Dojo, teams typically do a team lunch, potluck, or team event during the final weeks of the Dojo. Coaches, leadership team and other key stakeholders are also invited to participate and celebrate with the team. Leadership participation will reinforce their ongoing support and motivation post Dojo, when they go back to their non Dojo working environment.

## Remote Dojo

---

Co-locating for the entire duration of a Dojo is ideal for collaboration. Co-location offers the most immersive and focused upskill opportunities; makes for better relationships, team interactions, and experiences; and helps team members unlearn what they know and learn new ways to accomplish their desired outcomes together.

On a high level, here is the approach we recommend for remote Dojos:

### *How to Setup Remote Dojos:*

- 1) Co-locate at the start.*
- 2) Meet everyone in person at least once.*
- 3) Choose the right tools and technologies.*
- 4) Organize channels and rooms.*
- 5) Set up a communication infrastructure.*
- 6) Optimize video chat capabilities.*

## Why Remote Dojos?

Distributed teams participating in remote Dojos are inevitable in the current technology age. All types of organizations from startups to larger enterprise businesses are distributed across the globe. Software design, development, and development methodologies and practices have evolved, enabling teams to be distributed across different locations and time zones. Video calls, chat systems, distributed editing, and other technologies have helped bridge the communication gap in these distributed environments.

- Regardless whether an organization has co-located teams or distributed teams, the organization needs a strong culture that promotes continuous learning and improvement.
- A Dojo provides an immersive, distraction-free environment where teams and individuals can experiment, learn and apply towards their real-world product/service. (Such learning and experimentation aren't always encouraged in their current working environment.)
- Remote Dojos are especially helpful for individuals who may not be able to travel or who have special needs work environments.
- An organization's transformation goal should be the driver for the success of a Dojo, whether it be in-person or remote.

## Culture of Remote Working

The success of a remote Dojo depends on the organization's core culture. Before you start designing your Dojo for distributed teams, you need to understand your organization's culture and make sure that culture is ready to nurture remote-style learning. To get the best value from the Dojo, team members need to understand and embrace the tenets of remote working.

- Teams should embrace open communication and collaboration on persistent chat systems (e.g., [Slack](#), [Mattermost](#), or [Microsoft Teams](#)),
- All ceremonies and meetings should be video enabled, and participants should participate actively using video (discussions should be more prevalent than meetings).
- Another critical part of a supportive culture involves leadership support and availability. Leaders should be active in all communication channels and provide necessary support and direction for the team where and when needed.

- Teams should take full advantage of practices like pair programming and mob programming. Ideally, people shouldn't work in a solo, disconnected manner that impacts quality and productivity.

Open communication has its own challenges. The goal is to be inclusive of all roles and all people in a neutral way. Sensitive matters could surface due to constant communication, and team members should have a safe working environment to address any challenges that might affect the team and the organization's culture in a negative way.

If a strong remote work culture is already in place, then the product team's Dojo outcomes and goals can be more advanced and aggressive, making the Dojo a vehicle for real change.

## How to Set Up Remote Dojos

1. **Co-locate at the start.** Remote Dojos should start with at least 1 – 2 weeks of in-person, co-located kickoffs and mini-sprints. This temporary co-location enables teams to set a baseline; charter their Dojo expectations, goals, and desired outcomes, and align on their working environment (e.g., communication methods and mini-sprint cadence).
2. **Meet everyone in person at least once.** Everyone participating in the Dojo should meet in order to boost the sense of teamwork and togetherness.
3. **Choose the right tools and technologies.** The technologies you choose should support the needs of your Dojo. First and foremost, choose a persistent chat system like Slack, Mattermost, or Microsoft Teams. Because organizations struggle with a lack of information and transparency across all roles, we recommend encouraging more public channels over private channels and one-on-one messages. The entire organizational culture needs to embrace sharing and open communications.
4. **Organize channels and rooms.** The channels and rooms you create depend on your organization's structure and needs, but at a minimum we recommend having a channel for a team, portfolio, and platforms, along with channels for organization-wide sharing and learning, channels for communities of practices (e.g., developers, designers, scrum-masters, platform, and quality), and

channels for sharing and discussing random technical and non-technical topics (e.g., water cooler talk, personals, and town square).

**5. Set up a communication infrastructure.** Next, set up a solid video chat system to facilitate ad-hoc discussions.

- Choose video chat applications such as [Zoom](#), [Webex](#), and [Adobe Connect](#), all of which support interactive conferencing.
- Create virtual breakout rooms based on their function (e.g., demo lounges, coaches' bullpens, and team syncs), along with temporary rooms for specific topics.
  - Examples: [Sococo](#) & zoom and adobe connect offer break rooms via plugins
- Interactive whiteboards with multiple and simultaneous editing can stimulate discussions and collaboration, as can other support features such as polls, Q&As, and participant pulse checks (hand signals).
  - Examples: [Mural](#), [Miro](#), [Lucidchart](#), [Sketchboard](#)

**6. Optimize video chat capabilities.** Every remote participant should have a solid pair of headphones with a microphone and a good personal webcam.

## Coaching a Successful Remote Dojo

The coaches should make sure the team is working on their real-world product/services in the Dojo. Dojo hyper-sprints or mini-sprints enable fast feedback and rapid flow of work. From a learning perspective, repetition helps ensure fundamental practices stick even after the team leaves the Dojo. Coaches are the key critical role in the team's successful core learning as well.

The required effort from coaches increases exponentially in remote Dojos. Coaches can work remotely or, preferably, in a studio-style setting with multiple monitors/screens so they can observe different breakouts and share learning content simultaneously. In order to maintain the effectiveness of the Dojo, the coaches should:

- Work to keep all participants engaged and help them collaborate, pair, mob, and learn together to accomplish their product and learning goals.
- Be open in their way of teaching in remote Dojos and strive to build a learning environment that dynamically adapts to the team's feedback and behavior.
- All collaborative activities and ice-breakers in the Dojo are thoughtfully designed to make them inclusive and fun for all remote participants including participants with special needs.

- Lead by example, a principle that also applies to the tenets of remote working and availability, including video and audio setup for remote participation and facilitation.

Coaches will sync before each dojo to review the daily agenda, add notes to teams wiki or notes page for the schedule and plan.

Daily sync is helpful to plan out who will be lead coach for each session, who will support with commentary and also handle communication with other channels, observe and assist with obstacles both technical and due to remote communication.

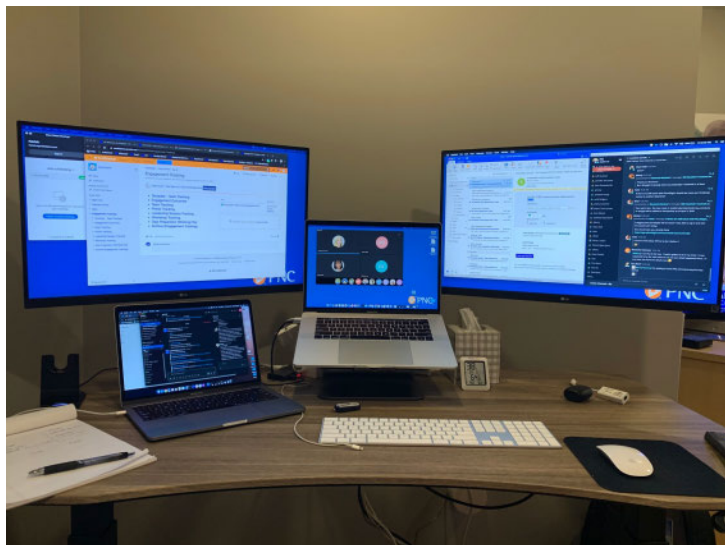
## Coaches Setup For Remote Dojo

Coaches work area is more than a typical office workstation or transient type work space. It is a studio, and as the master of ceremonies/coach/engineer the coach needs visibility across many screens to stay connected to work in progress, multiple communication channels, conference software, collaboration tools, company website, external web sites, code repo's, IDE's, and many other tools that often have to be seen simultaneously as well as the video view of each student, shared screen.

### Workstation

This is an example of one of our coaches' remote Dojo setup using two monitors and laptops to help facilitate and lead a Dojo.

Using a sit/stand desk is an investment well worth the cost to help coaches physically move around when connected for lengthy periods to a Dojo team.



## Internet Connectivity

Beyond the physical setup, the next area to review is internet bandwidth and backup plans in case of outage. It is recommended that 100 MB up/down is a minimum for handling video/audio and remote work for a Dojo. Higher is better, and stability is key for success. Testing and adjusting these internet speeds prior to starting a Dojo is crucial. If explaining a technical session to a dozen people and the connection is lost or disrupted it disrupts an entire team.

## Partner Coach

Having a backup plan and a partner coach is critical for success in these scenarios. Try to enable hot spotting on your cell phone and install the company apps that your team uses (like slack, mattermost, email, zoom, etc) so if an internet outage scenario occurs you can at least connect enough to redirect them until the outage is corrected. The partner coach role can't be under-estimated in these scenarios as well, if they are engaged they become the driver and take the lead while the other coach recovers from the outage. Similar to co-located Dojos, coaches support each other when one needs to step away from the Dojo for whatever reason.

Since coaches need to sync more purposefully, and cannot do it as randomly as in a co-located setting, the team's breaks often become those touch points to adjust outcomes, change team focus, etc. for the coaches. This often results in lengthy sessions at the coach's chosen work area.

## Working Norm for Remote Dojo

As an expansion of a Dojo's working agreement you need to consider and add some remote working specific norm.

- Learning to speak more metronome based, allowing more pauses between sentences and ideas. Audio communication has a natural lag that has to be accounted for.
- Set timers on the day for breaks and abide by them (use meeting/calendar invites). Recommend 45 min max for sessions and 15 min breaks.
- Expand the usage of persistent chat like slack, mattermost, teams, etc and create more short-lived rooms as needed
- Recommend when going on breaks/lunch/doing independent work/etc each person's video is turned off and when break/lunch/study is over video is turned on signifying that person is ready to go. This speeds up sessions and prevents delays with coaches and teams assuming someone is present and ready when they are not.

Use of video happy hours or team socializing time is a great opportunity to talk about 'not work' and helps bond a remote team more.

## Conclusion

Executing remote Dojos for distributed teams will have its own challenges that will need to be overcome through proper alignment of culture, strong leadership from the coaches and the right tools and technologies in place. However, with advancements in technology and the widely distributed nature of our industry, remote Dojos are the future. Remote team participation in Dojos is a natural progression.

Liatrio has found the Dojo to be the ONLY meaningful vehicle for team modernization where learning and the desire for continuous improvement truly stick. Remote Dojo environments in particular must be strategically designed to be supportive and dynamic, taking into account the needs of distributed teams.

# Metrics, Measure, and Success

---

The primary goal of the Dojo is to foster a culture of continuous learning and improvement that helps in creating more high performing teams and enable them to deliver customer value faster and safer. Based on our experience in creating, executing and scaling Dojos, we believe the primary goal of metrics for the Dojo should be:

- To provide insight on the teams (active and completed their Dojo journey) and how are they improving on delivery
- Showcase the impact of the Dojo as a whole (program/initiative)

In general, we recommend to follow a lean approach for the metrics, use a few key metrics instead of a bunch of complex metrics which are hard to reconcile and make it actionable. We categorize the metrics into 2 broad categories

1. Team Metrics
2. Organizational Reach (Dojo initiative)

## Team Metrics

Team metrics will help provide insight on how the team is improving on their delivery, to assess delivery we follow 4 key metrics in alignment with the industry experts. The key behind measuring these metrics is it should be derived from their delivery pipeline and toolchain automatically to make it truly objective and not subjective.

Team metrics should be easy to produce and readily available to the entire team, leadership and other stakeholders. Organizations can leverage dashboarding tools like Grafana, ELK stack, Hygieia, etc. to create these metrics & dashboards and make them available to the entire organization.

Here are the 4 key metrics we recommend:

### 1. Lead Time

It is the amount of time from a customer request to the request being satisfied. It is a good high level indicator but harder to define with data. As a starting point you can measure this as a mean time from code commit to production (on a master branch)

To calculate the true lead time, you need to trace a flow across multiple events or documents to calculate the start and end times:

- **Ticket:** Lead time starts when a ticket (story/task) is created or when work on the ticket begins (transition of ticket status)
- **Source:** The issue must be linked to the source repository. That can be done via comments in commits/pull requests, branch names or links created between Jira and Bitbucket
- **Build Pipeline:** The build job is linked by the source repository and produces an artifact with a version.
- **Deploy Pipeline:** Is linked by the artifact name and version. When the deploy is successful and target is production that is the end of lead time.

### 2. Deployment frequency

Frequency of deployment to **production** (includes all prod deployments), for additional visibility, measure frequency of deployment to at least 2 lower environments — **Dev & Pre-Prod**

### 3. MTTR Pipeline (Mean Time to Repair)

The mean interval of time between the start of a failed pipeline execution and the start of a successful pipeline execution. This number should be low as it is a measure of a team's ability to "stop the line" when a build fails and swarm on resolving it.

This should be reported for master branch only.

### 4. Feedback Time

The mean amount of time from commit to failure of a pipeline execution.

This provides insight to how quickly a team can identify failures in an automated manner. This should be reported for master branch only.

## Measuring Team Metrics in a Dojo

- Take a baseline of the team metrics before the team enters the Dojo or on day 1 on the Dojo
- While in the Dojo, display the team's dashboard with the key metrics on a screen/monitor
- As the team improves on the metrics while learning and improving in the Dojo, celebrate!
- At the end of the Dojo, take a snapshot of the metric and highlight improvements during demo and leadership readout

## Organizational Reach

The next set of metrics we recommend is around the Dojo initiative/program to get an insight into organizational reach. The purpose of these metrics is to provide insights into the success and impacts of the Dojo as an initiative.

Here are some metrics to consider:

### Net Promoter Score (NPS)

Create a NPS survey and have all dojo participants to participate at the end of their dojo experience. Along with regular feedback and retros in the Dojo NPS data provide insights into the overall impact of the Dojo.

Example of NPS:

1. How likely is it that you would recommend a Dojo experience to a friend or colleague?

Not at all likely

Extremely likely

0	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----

2. The Dojo has enabled our team/group to apply our experience and learnings to improve our ways of working and ability to deliver for our customer.

A great deal

A lot

A moderate amount

A little

None at all

☐

☐

☐

☐

☐

3. Provide any reason and/or examples for your rating

## State of Dojo

At the end of every Dojo experience (mostly Product team Dojos), the Dojo Coaches should provide a write up (1 pager) to summarize the team's journey, accomplishments, lessons learned, and next steps. They can provide the write up by answering the following questions.

- What went well for the team, what were they able to accomplish, specially team's outcomes (OKRs) — Success, Partial, and Needs Improvement
- What does the team need to work on?
- What are the next steps for the team that includes improvement backlog, coaches check-in and future Dojo visits?

## Frequent Team Survey

Frequent surveys at a team level will help provide insights into teams adoption of new ways of working and modern engineering practices. It should also provide symptoms in areas that the team needs some immediate help.

Examples of Team Survey:

Area	Measure	Scale
Business / Product Alignment	Team understand and align to product vision and goals	Sometime
	PO is available for any questions from team	Sometime
	Team asking What, Why to PO during refinement and 3 Amigos	Always
	PO provide immediate feedback as and when stories are completed	Mostly
	Overall	
Working together	Teams swarm when needed (master build fails, etc.)	Rarely
	Pairing regularly on day to day tasks	Never
	Adhering to social & working norms	N/A
	Safely call team out on antipatterns	Mostly
	Avoid single point of failure	Sometime
Managing work smartly	Overall	
	Good meeting hygiene	Always
	Stories are atomic and completed in a day	Sometime
	Able to eliminate unplanned work	Sometime
	Documentation, when necessary (read me, confluence)	Mostly
Modern Engineering practices	Able to reach out to mgmt & leaders for help	Always
	Overall	
	Not afraid to check in code	Mostly
	Deployment to lower environments are stressless	Sometime
	Build and test locally, first	Mostly
Open Communication	Follow norms of ticket based engineering	Mostly
	Initiator manages merge conflicts	Always
	Overall	
	On video chat when remote	Mostly
	Use hipchat for work in progress communication	Mostly
Overall	Share fun messages on team room	Sometime
	PR and reviews are communicated openly	Mostly
	Overall	

## Rolled up Team Metrics

In addition to NPS and surveys you can roll up team metrics and summarize on a value stream, business unit and organizational level. When doing a roll up make sure there is a drill down capabilities to get to a specific team and area that would need immediate attention and most help. Dashboarding tools should by default have drill down capabilities.

## Additional Metrics

Some organizations prefer to collect metrics like:

- # of Dojos executed successfully
- # of participants been through the Dojo

These metrics are easy to collect and can show the volume of the Dojo execution and scale of the Dojo as an initiative/program but not necessarily show the impact of Dojo or provide any actionable data to the leaders and stakeholders.

## Common Mistakes with Metrics

Metrics and measurements are definitely valuable to the team and organizations but only when it is done right. Here are some common mistakes that are made when capturing metrics and driving the organization towards continuous learning and improvement using metrics.

- Driving improvement to the metrics should not be an outcome of the Dojo, rather as teams work and deliver on their outcomes you should see an improvement on their delivery.
- Metrics should be treated as a tool to find areas of improvement and opportunities and metrics by itself does not help with the improvement.
- Different metrics for different people and levels, avoid having separate metrics for executives, program/project, managers and teams.
- Metrics should not be used to measure individual performance or point finger rather it should be helping team to improve on their delivery
- Avoid vanity metrics that do not provide any value or provide actionable insights, they often create noise and confusion.
- Too many metrics and reports are not helpful, with a lot of metrics and reports there will be a lack of clarity and focal point, in addition you would have a lot of overhead in collecting and presenting them.
- Metrics and dashboard is not open for all, avoid a secret team to create and maintain reports. Product teams should have the autonomy to create their own dashboards and report based on a common guideline.

## Factors for Success

The ultimate success of the Dojo is when enterprises start realizing long term benefits with this culture of continuous learning and improvement as teams become high performing and productive in delivering value to customers. We strongly believe that the following factors will greatly help the Dojo as an initiative to be successful.

### 1. Leadership Support

Leaders are essential in evolving the culture, practices and technology improvements within their organization. Dojo is a vehicle to drive change to culture, ways of working and bring engineering excellence. Leadership support is critical for the success of the Dojo. They need to show their support by empowering and rewarding their teams for working in new and improved ways. Leaders should shift to a servant-leader model and be ready to embrace and support organizational change.

### 2. Platform (Shared Services) Support

Platform team's vision and roadmap should be closely aligned to the Dojo. Coaches and engineers from platform teams should pair before and during the Dojo (specially for new capabilities). Platforms can obtain early and fast feedback from Dojo as they can be early adopters of new features & capabilities. In addition platforms would need to be evolved along with the Dojo and sometimes a step before the Dojo and move towards a more modern and self-service platform in order to support modern engineering practices and rapid delivery.

### 3. Post Dojo Support & Continuous Improvement

Teams after exiting the Dojo would need ongoing support from the platform team on new and changes to existing capabilities and guidelines. For Dojos focuses on changing and evolving practices and technologies we recommend having a squad of coaches and engineers from the platform teams dedicated to provide Dojo support. Coaches should also do frequent check-ins with the team to make sure they are improving and identify any new challenges they may have discovered.

After a team exits the Dojo and they are back into their regular working environment, they should continue:

- To apply newly learned skills & ways of working in their everyday work
- To solve more challenges using dojo model
- To influence and inspire others and bring them along

#### 4. Skilled Coaches

Coaches make or break a dojo experience. Every Dojo demands unique expertise based on the product and technical challenge being solved. Dojo coaching staff and other external experts should be assigned to the Dojo based on the team's needs.

In order to give a successful Dojo experience for a team, we recommend

- Dojo coaching squads to be formed and aligned based on findings from the Dojo Discovery phase that is done for a value stream or business unit level.
- Pairing Dojo coaches for each Dojo experience and being assigned to a team based on the team's outcomes and goals identified during Charter as part of Dojo intake phase.
- Coaches should have opportunities to continuously learn and improve based on their Dojo experiences.
- The biggest learning and uplift for coaches is to ensure they rotate as engineers to actual platform and product teams on a regular cadence

Refer to the [Hiring & Staffing](#) part of the playbook for more information.

#### 5. Dojo as a Product

We believe every team is a product team including those who provide services and support. So the Dojo coaches team should operate like a product team and the Dojo should be treated as a product that will be owned by coaches. Coaches would need to build, learn and add more capabilities to the Dojo product. When coaches are not in Dojo, they should be building/improving their Dojo product.

Another important aspect of treating Dojo as a product is marketing the Dojo and making the entire organization aware of the Dojo offering, benefits and success stories.

What does it mean to treat Dojo as a product?

- Create a brand for your Dojos with logos, stickers and other swag
- Coaches need to own Dojo execution playbook creation and maintenance.
  - You can choose to build your playbooks on git using some documentation framework/libraries
  - Playbooks are helpful to bring common guideline & reference for the Dojo and help the coaches to provide consistent and successful Dojo experiences
- Create a solid pipeline for Dojo intake, communication, this would include automating tool access, platform provisioning, etc.
- A sandbox or demo app would be helpful to showcase features and capabilities of the platform and also act as a reference application

- Identify opportunities to design and build your own custom frameworks to help bootstrap learning for the teams.
- Creating self learning models and scoring systems (git badges, etc) will help in scaling and sustaining Dojos

## 6. InnerSourcing

InnerSourcing is a culture to embrace practices and benefits of open sources tools & frameworks and apply within the organization. Innersourcing will help sustain and improve knowledge sharing and collaboration across teams. Innersourcing improves efficiency across the organization and reduces duplicate development efforts and promotes reusability across the enterprise.

Innersourcing is very broad and has a ton of benefits, we have a detailed **playbook** to implement Inner sourcing practices and embrace a culture of sharing.

# Templates

---

[Dojo Consultation](#)

[Consult Guide](#)

[Welcome to the Dojo Email](#)

[Dojo Outcome Email for Dojo Coaches](#)

[Dojo Outcome Email](#)

[Supporting New Features for Teams](#)

[Leadership/Manager Dojo Template](#)

---

## Dojo Consultation

Hi [[INSERT NAME](#)],

I am reaching out to let you know that I received your Dojo intake request for the [[TEAM NAME](#)] team. The intent of this consultation is for both parties to gain information that enables a decision to be made on appropriate next steps. In this meeting we will discuss the following at a high-level:

- Overview of the Dojo and coaching capabilities
- Objectives the team is looking to accomplish by coming to the Dojo
- The application and/or product the team works on
- The team structure and makeup

Please forward this meeting to other team members as applicable to ensure we are able to effectively cover the above topics.

Best,  
Dojo Team

## Consult Guide

### Purpose

This document describes what will be asked during a consult for Dojo after an intake form has been completed and further information is needed to determine eligibility/success for entering the Dojo. The goal of the consult is to obtain enough information to determine if the team requesting the Dojo should move forward, or if there are blockers that need to be addressed prior to Dojo.

### Date of Consult

[INSERT DATE].

The product owner, scrum master, and technology leads are strongly recommended for each consult.

### Target Outcomes

Question	Response	Dojo Coaches Notes Section
What is motivating your team to come to the Dojo?		
What is stopping your team from doing these things today?		

## Team Information

Question	Response	Dojo Coaches Notes Section
Is your team currently deploying to production?	<ul style="list-style-type: none"> <li>• Yes</li> <li>• No</li> </ul>	
What technology stack is this application leveraging?	<ul style="list-style-type: none"> <li>• [Insert applicable technology Stack]</li> <li>• [Insert applicable technology Stack]</li> <li>• [Insert applicable technology Stack]</li> <li>• [Insert applicable technology Stack]</li> </ul>	
What are your current versions of [insert list of names here] (If applicable)	<ul style="list-style-type: none"> <li>• [Insert applicable versions]</li> <li>• [Insert applicable versions]</li> <li>• [Insert applicable versions]</li> </ul>	
What type of artifact are you generating for your application? (If applicable)	<ul style="list-style-type: none"> <li>• Jar</li> <li>• War</li> <li>• Ear</li> <li>• Other _____</li> </ul>	
Is your team using a pipeline today?	<ul style="list-style-type: none"> <li>• Yes</li> <li>• No</li> </ul>	
Does your team use [Insert SCM platform]?	<ul style="list-style-type: none"> <li>• Yes</li> <li>• No</li> </ul>	
Are there dependencies with any vendor applications?	<ul style="list-style-type: none"> <li>• Yes</li> <li>• No</li> </ul>	
Are multiple teams contributing to your source code?	<ul style="list-style-type: none"> <li>• Yes</li> <li>• No</li> </ul>	

## Tooling Access

Question	Response	Dojo Coaches Notes Section
Does your team have access to [collaboration tools]?	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>	
Does your team have access to [Insert wiki platform]?	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>	
Does your team have access to [Insert SCM platform]?	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>	
Does your team have access to Jira/Boards?	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>	

## Anything else we should know?

---

---

---

---

---

---

---

---

## Welcome to the Dojo Email

[CIO NAME] and I are very excited to send this message to you. Over the past few weeks, you have met with experts from the Dojo team to discuss your product vision, technology, and team dynamics as part of the selection process. We are happy to announce [TEAM NAME] have been selected to participate in the Dojo.

**Your Dojo will begin on [DOJO START DATE] and conclude on [DOJO END DATE]**

### What should you expect?

By entering the Dojo, you will have the opportunity to immerse yourself in the DevOps culture, partner with experts and focus on collaboration, learning and strengthening your expertise in product planning, development, automation, and CI/CD **while working on your product and services.**

### As a team, you are being asked to:

- Co-locate for the duration of the Dojo as coaches walk you through core curriculum
- Attend every day and remember that your team counts on your full participation

### We are preparing the Dojo for your arrival and in the next few days you:

- Will attend an orientation session to review logistics and discuss expectations
- Will receive an invitation for the Dojo
- Should work with your manager to ensure access requests have been approved for new tools

This is an exceptional opportunity to enable your team to become cross-functional and deliver better, faster and safer products to our customers. [CIO NAME] and I are excited for your participation in the Dojo and look forward to partnering with you along this journey.

Sincerely,

[Sponsor] and [CIO NAME]

## Dojo Outcome Email for Dojo Coaches

Hi Coaches,

As part of our efforts to create transparency for leadership stakeholders into Dojo team outcomes, we will be sending out emails summarizing outcomes the team achieved during their stay. This email will come from the Dojo mailbox, and the coaching team will be CC'd.

### Ask of Coaches:

- Please insert any relevant information or content into the green highlighted fields
- If you feel anything else in the message below should be added/removed/revised please feel free to do so

Your prompt review is appreciated.

Let me know if there are questions,

Dojo OpEx Team

## Dojo Outcome Email

[REVISE SUBJECT LINE]

Hi [LEADERSHIP NAMES],

At The Dojo we believe in celebrating the improvements and success of teams across the enterprise who have acquired new skills and ways of working. That said, we wanted to provide insight into the great things that the [LOB] [TEAM NAME] team accomplished during their time working in the Dojo.

[TEAM NAME] began their Dojo experience on [DOJO START DATE] and worked with us through [DOJO END DATE]. The [# COACHES] Dojo coaches that worked side-by-side with the team were [COACH NAMES]. The team also benefited from the expertise of [EXPERT NAMES] who helped with [PRACTICES].

During their chartering session, [TEAM NAME] defined specific objectives to focus on:

- [OBJECTIVES]

By the end of their stay in the Dojo, [TEAM NAME] achieved the following outcomes:

- [OUTCOMES]

[PICTURES, BLOG, ETC]

Continued opportunities for the [TEAM NAME] Team:

- [OPPORTUNITIES]
- [OPPORTUNITIES]
- [OPPORTUNITIES]
- [OPPORTUNITIES]

Things [TEAM NAME] Leadership can help with:

- [OPPORTUNITIES FOR HELP]
- [OPPORTUNITIES FOR HELP]

We enjoyed partnering with [TEAM NAME] and wanted to express our appreciation for your continued support in enabling the team to travel and dedicate time for continuous learning. Please let us know if there is additional opportunity to partner with [TEAM NAME] in the future!

Best,

The Dojo Team

## Supporting New Features for Teams

Hello,

We are touching base with you because it has been a while since we last discussed your team's interest in coming to The Dojo! Since we last spoke, there have been multiple improvements to the DevOps Pipeline and The Dojo that have enabled us to help even more teams across the company. Some of the things we have improved include:

- [\[INSERT NEW FEATURES\]](#)
- [\[INSERT NEW FEATURES\]](#)

If you would like to discuss whether The Dojo is right for your team you can submit a quick intake form [\[INSERT INTAKE LINK\]](#). By filling out the intake form, we will be able to understand your team's needs and schedule a conversation to discuss them further.

The Dojo also offers a number of additional services including, workshops, Dojo Primers, tours, and more! You can view all of our offerings [\[LINK TO DOJO OFFERING\]](#).

Hope to hear from you soon!

Dojo Team

# Leadership Dojo

## Templates for Agenda

### Leadership Example Agenda

(customized based on current organizational priorities)

Time	Topic	Notes
9 – 9:45am	Business Strategy/Alignment	Business to share company goals for quarter and for year
10 – 10:45am	Technology Roadmap to meet business goals	Outline areas that technology will enable, build, support to deliver on business outcomes
11 – 11:45am	Culture  Metrics	What do we want our team members to understand, how to deliver on expectations  What do we want to measure, how will we use it, how will we gather it?
12 – 12:45pm (working lunch)	Friction/Opportunities	Break out, small groups
...		

**Manager/Engineering Manager Example Agenda**  
(customized based on current organizational priorities)

Time	Topic	Notes
9 – 9:45am	Technology Strategy	Outline areas that technology will enable, build, support to deliver on business outcomes
10 – 10:45am	Hands on Lab	Work with demo app, learn to build, test, deploy for example
11 – 11:45am	Metrics/Measures	What are they, how to use them, etc.
12 – 12:45pm (working lunch)	Feedback on Issues	Breakout sessions to work on solutions to common issues (prioritized)
...		

# Appendix

---

Additional Section include

[Quotes from Dojo Participants](#)

[References](#)

## Quotes from Dojo Participants

---



*This excites and motivates me to come to work (in person) everyday, I haven't felt this way for a long time.*

— **Developer**

*Liatrio engineers and coaches inspire me to come to work everyday in spite of longer commutes, you all have so much energy even at 6PM. And also, encouraging me to turn on videos while on meetings.*

— **DevOps Program Manager**

*All of this is new and really fun and exciting. We are working together to write new code, troubleshoot, build & deploy. I'm glad to be part of this whole DOJO experience.*

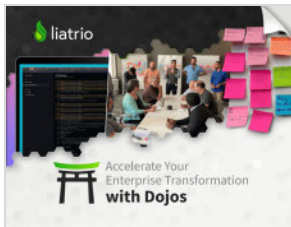
— **Lead Developer**

*This is very productive. I can see things moving from idea to complete more clearly and faster. The team is working more together than before.*

— **Product Owner**

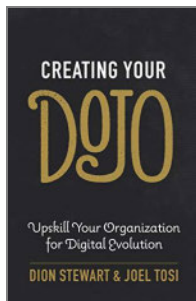
## References

---



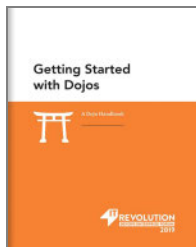
LIATRIO

### [Our Dojo Practice Brochure](#)



DION STEWART & JOEL TOSI

### [Creating Your Dojo](#) [Upskill Your Organization for Digital Evolution](#)



IT REVOLUTION

### [Getting Started with Dojos](#)



LIATRIO

### [View The Latest Insights From Our Blog](#)

# About Liatrio

---



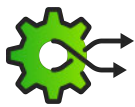
**Liatrio is a catalyst for change. We enable our customers to deliver value faster and safer by empowering people, teams, and organizations through modern DevOps practices and Cloud Technologies.**

We do this by bringing deep industry experience, engineering excitement, and a vested interest in our customers' success — which becomes a foundation for continuous improvement.

We embody and foster a culture of empathy, authenticity, and transparency. We focus on delivery and always ensure that our talent brings passion and excitement to our customers.

---

## Our Core Capabilities



### Enterprise DevOps Transformation

Accelerate business results and scale your organization with a lean, value-driven approach to software delivery and IT operations.



### Cloud Native Delivery

Empower your teams to build scalable apps in dynamic environments and make high-impact changes frequently and predictably with little toil.



### Modern Platform Engineering

Reliable applications are built on modern self-service platforms that reduce engineering friction.



### DevSecOps

Speed of delivery while always staying safe and secure in an automated way. Remove untimely, manual, last gate siloed approvals and validations.



# liatrio

DEVOPS AND CLOUD TRANSFORMATIONS

[LIATRIO.COM](https://liatrio.com)