Agile ORLANDO
JULY 24-28 2023

PRESENTED BY Agile Alliance
JOIN US TODAY!

#AGILE2023
A TASTE OF THE DOJO
AN IMMERSIVE LEARNING EXPERIENCE

DOJO

AND CO

EST 2017

Dion Stewart and Joel Tosi

dion.stewart@dojoandco.com  dojoandco.com  joel.tosi@dojoandco.com
We’ve been part of the dojo movement since it started.

OUR STORY

Dion Stewart
- Dojo Coaches for the last eight years
- Cofounders of Dojo and Co
- Both have been coaching teams for 20 years

Joel Tosi

©2019 Dojo and Co
dojoandco.com
@dojoandco
HOW TO CREATE YOUR OWN DOJO

CREATING YOUR DOJO
Upskill Your Organization for Digital Evolution
DION STEWART & JOEL TOSI

Coaching for Learning: The Art and Practice
Joel Tosi and Dion Stewart with Peter Economy
Foreword by Emily Bache
Illustrations by Jeremy Kriegel

©2023 Dojo and Co @dojoandco
ORGANIZATIONAL IMPROVEMENT FROM A SYSTEMS THINKING PERSPECTIVE
TRANSFORMATIONS AND CHANGE INITIATIVES

The Search for Something Better

Underlying all of these is a desire to improve at least one capability.

- Agile transformations
- Move to the cloud
- Project to Product transformations
- DevOps
- DevSecOps
- Innovation
And..
Do all that while cutting costs.
Nobody Ever Gets Credit for Fixing Problems that Never Happened:
Creating and Sustaining Process Improvement

Nelson P. Repenning
John D. Sterman

How much would your organization pay to develop manufacturing capability equal to Toyota’s? How much would a world-class, six-sigma quality program be worth to your company? How about Harley-Davidson’s ability to tap into the hearts and minds of its cus-

http://web.mit.edu/nelsonr/www/Repenning=Sterman_CMR_su01_.pdf

©2023 Dojo and Co @dojoandco hello@dojoandco.com
Causal Loops

"THE PHYSICS OF IMPROVEMENT"

- TIME SPENT ON IMPROVEMENT
- CAPABILITY
  - ACTUAL PERFORMANCE
  - PERFORMANCE GAP
  - DESIRED PERFORMANCE
- PRESSURE TO DO WORK
  - PRESSURE TO IMPROVE CAPABILITY
- TIME SPENT WORKING
  - DELAY

©2023 Dojo and Co  @dojoandco  hello@dojoandco.com
WHAT'S A DOJO?
DOJOS - A DEFINITION

A dojo is a space where teams undergo an immersive, whole-team learning experience. Teams set learning goals for product, technical, and process practices. They learn the practices with the support of skilled coaches while doing real-world work.
The word “dojo” literally means “place of the way”.

The name and metaphor is taken from martial arts studios, in particular Aikido.
THE SEVEN PRINCIPLES
The Seven Principles

SEVEN PRINCIPLES OF EFFECTIVE LEARNING EXPERIENCES

1. Focus on Learning Over Delivery
2. Help Teams Learn Collaboratively
4. Make Learning Holistic and Span Multiple Practices
5. Provide Assessment, Feedback, and Coaching
6. Provide Sufficient Time for Repetitive Practice
7. Ensure Learner Safety

https://www.dojoandco.com/blog/2023/3/7/seven-principles
FOCUS ON LEARNING OVER DELIVERY

Even when an experiment results in no output, it's still a success if it leads to learning.

This is NOT about increasing a team's velocity while in the dojo.
HELP TEAMS LEARN COLLABORATIVELY

Use techniques like pairing and software teaming (ensemble programming) to build shared understanding and foster learning.
The Seven Principles

GROUND LEARNING IN THE CONTEXT OF REAL-WORLD WORK

We need to move beyond intellectual understanding and apply what we're learning.

And we need to apply that learning with all the constraints we face in our daily work.
The Seven Principles

MAKE LEARNING HOLISTIC AND SPAN MULTIPLE PRACTICES
Content is everywhere. We don't suffer from a lack of content.
The Seven Principles

PROVIDE SUFFICIENT TIME FOR REPETITIVE PRACTICE

Hermann Ebbinghaus

THE FORGETTING CURVE

MEMORY

DAYS REMEMBERED

©2023 Dojo and Co  @dojoandco  hello@dojoandco.com
ENSURE LEARNER SAFETY

What happens when someone makes a mistake?

Dr. Timothy Clark
<table>
<thead>
<tr>
<th>Traditional Training</th>
<th>Learning based on the Seven Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises done in &quot;clean&quot; sandboxed environments.</td>
<td>Learning done with real-world work.</td>
</tr>
<tr>
<td>Learning is focused on the individual.</td>
<td>Learning is focused on the team, including how to work together as a team.</td>
</tr>
<tr>
<td>Limited time for assessment, feedback, coaching, and practice.</td>
<td>Schedule allows for assessment, feedback, coaching and repetitive practice.</td>
</tr>
<tr>
<td>Learning is focused on a single set of practices related to one skill (e.g., building deployment pipelines).</td>
<td>Learning spans the entire product development stream. Learners see how the parts relate to the whole.</td>
</tr>
<tr>
<td>Learning is a one-off special event.</td>
<td>Learning and continuous improvement become part of a team's daily work. The organization becomes a learning organization.</td>
</tr>
</tbody>
</table>
EXERCISE - EVALUATE YOUR OWN TRANSFORMATION EFFORTS

The Seven Principles

1. Focus on Learning Over Delivery
2. Help Teams Learn Collaboratively
4. Make Learning Holistic and Span Multiple Practices
5. Provide Assessment, Feedback, and Coaching
6. Provide Sufficient Time for Repetitive Practice
7. Ensure Learner Safety
What Questions do you Have?

REFLECT & SHARE
WHAT'S A DOJO?
(REDUX)
The "Standard" Format

A SAMPLE FLOW

12 x 2½ DAY SPRINTS

- Overview
- Consult
- Chartering
- Retro
- Learn
- Plan Sprint
- Discovery Delivery
- Review
- Exit Interview
- Apply Learnings
# DOJO SCRUM VS CONVENTIONAL SCRUM

## Starting a Team

<table>
<thead>
<tr>
<th></th>
<th>Dojo</th>
<th>Conventional SCRUM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standup</strong></td>
<td>What are our learnings?</td>
<td>3 questions</td>
</tr>
<tr>
<td></td>
<td>How are we doing on learning outcomes?</td>
<td></td>
</tr>
<tr>
<td><strong>Demo</strong></td>
<td>Learning demo (with product)</td>
<td>Product demo</td>
</tr>
<tr>
<td></td>
<td>30 minutes (discourage slides/PPT)</td>
<td></td>
</tr>
<tr>
<td><strong>Retro</strong></td>
<td>Test Driven, Brief - 5 to 15 minutes</td>
<td>3 questions - other exercises</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Up to 3 hours</td>
</tr>
<tr>
<td><strong>Sprint Planning</strong></td>
<td>Now / Next / Later (fits in 2.5 days)</td>
<td>Plan to capacity (using velocity)</td>
</tr>
<tr>
<td></td>
<td>Next Best Investment in Learning</td>
<td>Sprint goal</td>
</tr>
<tr>
<td></td>
<td>Continuously adjusted</td>
<td>Committed and then “locked”</td>
</tr>
<tr>
<td></td>
<td>60 minutes</td>
<td>8 Hours for 1 month sprint</td>
</tr>
<tr>
<td><strong>Sprint Duration</strong></td>
<td>2.5 days</td>
<td>2-4 weeks</td>
</tr>
<tr>
<td><strong>Estimating</strong></td>
<td>Low ceremony - Is it too big?</td>
<td>Planning poker - how big?</td>
</tr>
<tr>
<td></td>
<td>(no sizing)</td>
<td></td>
</tr>
</tbody>
</table>
SIX REASONS WE DO TWO-AND-A-HALF-DAY SPRINTS

1. Repetition and feedback foster learning
2. Multiple opportunities to practice getting to done
3. Teams learn how to break down stories (finally!).
4. Learning is supported by a margin of safety
5. Frequent retrospectives lead to a mindset of continuous improvement
6. More opportunities to ask “What’s the next best investment in learning?”
CHARTERING
Optimizing your Dojo’s Flow

CHARTERING

• Charter at least a few days before a team starts
• Up to 2 weeks before
• Whole team that will be in the experience plus leaders attend
• Outcomes needed (alignment)
• Usually takes about four hours - teams without a clear understanding of their product(s), learning goals, and new “teams” can take longer.
WHY CHARTER

- Create shared understanding of what success looks like
- Frame the dojo experience for the team
- Team members understand expectations
- Get buy in from leaders on learning and delivery goals
Chartering Agenda

- Team Name
- Timeframe
- Elevator Pitch
- Goals and Success Measures
- Logistics/Working Agreements
- Community Map
- Skills Matrix
- (Your own - architecture diagrams? dependencies and constraints?)
Team lists goals around product development and learning.

- 2 - 3 success measures for each goal
- Success measures should be binary (we achieved them or we didn’t achieve them)
- No set limit on number of goals. Around 6 is a good place to start.
**SKILLS MATRIX**

<table>
<thead>
<tr>
<th></th>
<th>Test automation</th>
<th>Microservices</th>
<th>Build</th>
<th>Branching</th>
<th>Stories</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaliyah</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>Alexis</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>Ang</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>Darryl</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>David</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>Joy</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>Kiran</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>Lin</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
<tr>
<td>Reggie</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
<td>![Check Mark]</td>
<td>![Circle]</td>
</tr>
</tbody>
</table>

**Key**
- ✔️ Can teach
- 🧐 Wants to learn
- 🙁 Meh

- Team members in rows
- Skills in columns
- Check Mark - know it and can help others learn
- Circle - want to learn
A Tale of Two Teams

Starting a Team

Team A
New Product Development

<table>
<thead>
<tr>
<th>M</th>
<th>T</th>
<th>W</th>
<th>Th</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Discovery</td>
<td>API Design</td>
<td>Develop First Service</td>
<td>Define + Create Automated Integration Tests</td>
<td>CD Pipeline Design</td>
</tr>
</tbody>
</table>

Team B
Cloud Migration

<table>
<thead>
<tr>
<th>M</th>
<th>T</th>
<th>W</th>
<th>Th</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Backlog</td>
<td>Learn Ansible</td>
<td>Config EC2 Instances</td>
<td>Blue/Green Deployment Capabilities</td>
<td>CD Pipeline Design</td>
</tr>
</tbody>
</table>
WHAT IS DOJO COACHING?
What is coaching?

Exercise

How do you define coaching?
How do you define dojo coaching?
Is there a difference?
A dojo coach helps teams improve the way they deliver products.
Some institutions distinguish between coaching for performance and coaching for development. Coaching for performance is about addressing and fixing a specific problem or challenge. It’s putting out the fire or building up the fire or banking the fire. It’s everyday stuff, and it’s important and necessary. Coaching for development is about turning the focus from the issue to the person dealing with the issue, the person who’s managing the fire.

Michael Bungay Stanier - The Coaching Habit: Say Less, Ask More & Change the Way You Lead Forever
FOUR ESSENTIAL SKILLS FOR DOJO COACHES

• Active Listening
• Framing Powerful Questions
• Teaching
• Storytelling
A Model for Coaching

The Seven Roles of a Dojo Coach

- Practitioner
- Guide
- Teacher
- Observer
- Modeler
- Mirror
- Advocate
STARTING A TEAM'S LEARNING EXPERIENCE
THE FIRST WEEK

Starting a Team

The First Week

Revisit Charter
• Refresh everyone’s understanding of what success looks like
• Make sure nothing has changed

Build a Backlog
• Start Product Discovery for new product development
• Create Value Stream Maps for any process the team wants to improve (discovery/delivery process, testing, CI/CD)
• Use Checklists for repeated activities (e.g., migrating a database from one vendor's tech on premises to a different vendor's tech in the cloud)
• Brainstorm with the team to discover any additional backlog items

Run the first Sprint (if product discovery is complete)
Understanding the Product

STORY MAPPING (AND PERSONAS)
Understanding Your Current Processes

VALUE STREAM MAPPING

Submit request

Needs server for app

- Database setup: 5 day wait, 1 hour work
- Middleware setup: 7 day wait, 30 minutes work
- Access, security, network: 10 day wait, 10 minutes work

Wait time = 22 DAYS! + Work Time = 100 minutes \( \rightarrow \approx 1\% \) work/wait ratio
Understanding Your Current Processes

EXERCISE 2 PLACEHOLDER
REFLECT & SHARE

What Questions do you Have?
LEVERAGING SOFTWARE TEAMING
SOFTWARE TEAMING

A Mob Programming, Whole-Team Approach

Second Edition

Woody Zuill and Kevin Meadows
Forward by Kent Beck

Woody Zuill
"For an idea to go from your head into the computer it MUST go through someone else's hands."

-Llewellyn Falco
A 10,000 FOOT VIEW

Driver
Drivers do not decide what code to write, they simply enter the code by following Navigators instructions.

Navigator
The navigator has two responsibilities:
1. Give the next instruction to the driver the instant they are ready to implement it.
2. Talk in the highest level of abstraction the driver can understand.
Understanding Your Current Processes

EXERCISE 3 PLACEHOLDER
What Questions do you Have?

REFLECT & SHARE
MEASURING LEARNING
GOING BEYOND THE KIRKPATRICK MODEL

Level 1: Reaction
Did the learners like the training?

Level 2: Learning
Did the learners learn something?

Level 2: Behavior
Did the learning change behavior?

Level 2: Results
Did the learning have impact?

- from Don Kirkpatrick 1959

Training efforts are often evaluated using only a 1 and 2?!?!?!
## SKILLS MATRIX REDUX

### Indexes

<table>
<thead>
<tr>
<th>Index</th>
<th>Baseline</th>
<th>O team retro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross functional index</td>
<td>48.3%</td>
<td>61.7%</td>
</tr>
<tr>
<td>Depth index</td>
<td>17.8%</td>
<td>71.1%</td>
</tr>
<tr>
<td>Diffusion index</td>
<td>40.1%</td>
<td>34.6%</td>
</tr>
</tbody>
</table>

---

### Baseline Team Members

<table>
<thead>
<tr>
<th>Team</th>
<th>CI/CD</th>
<th>Tech writing</th>
<th>Jenkins</th>
<th>Software config mgmt</th>
<th>Coding (C++)</th>
<th>Py (Python)</th>
<th>Agile workflow</th>
<th>Model-based design</th>
<th>Hardware</th>
<th>Test-bench testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team member 1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Team member 2</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Team member 3</td>
<td>9</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Team member 4</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Team member 5</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Team average</td>
<td>5.4</td>
<td>2.8</td>
<td>5.2</td>
<td>6.4</td>
<td>5.6</td>
<td>6.4</td>
<td>5.4</td>
<td>5.4</td>
<td>5.4</td>
<td>5.4</td>
</tr>
</tbody>
</table>

---

### Retro Team Members

<table>
<thead>
<tr>
<th>Team</th>
<th>CI/CD</th>
<th>Tech writing</th>
<th>Jenkins</th>
<th>Software config mgmt</th>
<th>Coding (C++)</th>
<th>Py (Python)</th>
<th>Agile workflow</th>
<th>Model-based design</th>
<th>Hardware</th>
<th>Test-bench testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team member 1</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Team member 2</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Team member 3</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>9</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Team member 4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Team member 5</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Team average</td>
<td>6.6</td>
<td>4.4</td>
<td>6.0</td>
<td>6.4</td>
<td>5.2</td>
<td>7.4</td>
<td>6.8</td>
<td>5.8</td>
<td>4.6</td>
<td>5.4</td>
</tr>
</tbody>
</table>

learningroi.com
# Start Where you Can

## SUCCESS MEASURES

<table>
<thead>
<tr>
<th>Organizational Reach (Simple)</th>
<th>Directional/Team-Based (Strategic)</th>
<th>Impactful/Economic (Measuring Outcomes)</th>
</tr>
</thead>
<tbody>
<tr>
<td># measurements (teams, pipelines create, etc.)</td>
<td>Reduction of Cycle Time</td>
<td>... for a feature <em>that mattered</em></td>
</tr>
<tr>
<td>Increase in Velocity</td>
<td>Increase of flow</td>
<td>Number of bad ideas stopped</td>
</tr>
<tr>
<td>Reduction of Defects</td>
<td>Frequency of Feature Release</td>
<td>Increase in internal sharing</td>
</tr>
<tr>
<td>Increase in Automation</td>
<td>Reduction of dependencies</td>
<td>Increase in customer joy/satisfaction</td>
</tr>
</tbody>
</table>

*Increase in lifecycle profits*


©2023 Dojo and Co  @dojoandco  hello@dojoandco.com
COACHING REDUX
Knowledge is not a thing.

It can’t be “transferred” or “managed” for the most part.

Better to talk about knowledge creation than knowledge transfer.

The best we can do is foster the conditions for knowledge creation to happen.
- skilled coaches
- safe environment
- permission to focus on learning
How Will You Teach?

Be Intentional About Tacit vs Explicit Knowledge

Tacit Knowledge
- in someone’s head
- hard to write down
- may even be hard to verbalize
- harder to share
- requires confirmation of sharing

Explicit Knowledge
- easy to “put into words”
- easy to write down
- easy to share
- can be “externalized”
- easy to share
How Will You Teach?

You Don’t Have To Teach Everything

As a coach you don’t have to figure out everything that people need to learn. You just have to help them identify the work they need to do, give them a safe space to learn in, and the work will teach them what they need to know.
JOIN THE COMMUNITY
Hope for the Future

IMMERSIVE LEARNING

Delighted to see that this style of learning, a place for getting away and learning together by doing, is coming back and that it is getting backing from big companies. I think we’ll be seeing more of it soon. I can’t wait to participate.

Jennifer Guerra @JangryGuerra
Our #DojoConsortium event is over. Shout to to @dojoandco for making it happen. Special thank you to our speakers @MarkGrabin @RossClanton @tmnigard @KentBeck @richburroughs @WoodyZulli @JaneC_H & all the attendees...
Special Thanks

JEREMY KRIEGEL

Thanks for the illustrations!

www.methodsansmadness.com
WHAT QUESTIONS HAVE WE LEFT UNANSWERED?
THANK YOU
Dion Stewart and Joel Tosi

EST 2017

AND CO

dion.stewart@dojoandco.com  dojoandco.com  joel.tosi@dojoandco.com
Join Agile Alliance today!

Become an Agile Alliance member and help support our non-profit mission, while gaining access to valuable benefits like online events, in-person conference discounts, and event session videos.