Choose your WoW!
Disciplined Agile Strategies for Optimizing Your Way of Working (WoW)

• Mark Lines
• Co-creator of the Disciplined Agile (DAD) Toolkit
• Disciplined Agile provides guidance for you on your journey to awesomeness

• mark [at] disciplined-agile.com
• Twitter: @Mark_Lines @Disc_Agile
“I am not here to tell you how to do Agile!”
Our Industry is Fixated on Calling ourselves a Certain kind of “Shop”
It is time to break out of “Methods Prison”
– Ivar Jacobson
Agenda

• Step 1: Understand the Situation
• Step 2: Adopt a Continuous Improvement Strategy
• Step 3: Choose Your WoW!
• Q&A
Step 1:
Understand the Situation
Context Counts

Start by identifying the situation that your team faces so that you can tailor your WoW to address that context.
Unfortunately, there is no "Easy Button"
Your Organization is a Complex Adaptive System

Disciplined Agile Delivery (DAD)

Disciplined Agile for IT (DAIT) and the Enterprise (DAE)
Step 2: Adopt a Continuous Improvement Strategy
Adopting a Prescriptive Method Helps a Bit
You Are Promised

Adopting a Prescriptive Method/Framework Helps a Bit

You Actually Get

Recent study of 3,000+ teams within 155 organizations found:

- Teams adopting agile (mostly Scrum) saw productivity increases of 7 to 12% on average
- Teams adopting prescriptive scaling frameworks, the most popular of which is SAFe, saw average productivity increases of 3 to 5%

InfoQ.com/articles/reifer-agile-study-2017
Guided Continuous Improvement

1. Identify a Potential Improvement
2. Experiment with the new WoW
3. Assess Effectiveness
4. Adopt the new WoW
5. Abandon the new WoW
6. Share Learnings with Others

Apply the Disciplined Agile (DA) toolkit to identify better options.
It Works Even Better With Light-Weight Guidance

Guided Continuous Improvement

Continuous Improvement

Improvement

Time

© Disciplined Agile
You Can Combine Strategies

Guided Continuous Improvement

Continuous Improvement

Adopting a Prescriptive Method/Framework

Time

Improvement
Step 3: Choose your Own Way of Working (WoW)
How to be a better Agile “Chef”
What Recipes does your Organization Use?

- Scrum?
- SAFe?
- Extreme Programming (XP)?
- Lean/Kanban?
- DevOps?
- Traditional/Waterfall?
- Hybrid?
The Ingredients for the Scrum Recipe are Few

• Sprint Planning
• Daily Stand-up (Scrum)
• Sprint Review
• Retrospective
If you are just using Scrum your possibilities are limited
What if your Pantry contained more ingredients?

- Retrospectives
- Behaviour Driven Development (BDD)
- Test-driven Development (TDD)
- Big Room Planning
- Value Stream Mapping
- Mob Programming
- Boundary Spanners
- Feature Toggles
- Etc…
The Seven Principles of Disciplined Agile

- Be Awesome
- Enterprise Awareness
- Pragmatism
- Delight Customers
- Optimize Flow
- Context Counts
- Choice is Good
Disciplined Agile Delivery (DAD)
A Toolkit of Practices
Tailoring Your WoW

• Initially:
  – Understand the situation that you face
  – Choose the most appropriate lifecycle
  – Choose the best strategies/practices that your team can perform

• Over time:
  – Regularly reflect
  – Experiment with new strategies
  – Evolve your WoW
Choosing a Lifecycle

<table>
<thead>
<tr>
<th>Single Team</th>
<th>Team of Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile</td>
<td>Program</td>
</tr>
<tr>
<td>Lean</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project Teams</th>
<th>Long-Standing Teams</th>
<th>Experimental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile</td>
<td>Continuous Delivery: Agile</td>
<td></td>
</tr>
<tr>
<td>Lean</td>
<td>Continuous Delivery: Lean</td>
<td></td>
</tr>
<tr>
<td>Exploratory</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Disciplined Agile
Evolving Your WoW: Lifecycles

Greater discipline
Broader skills

New work arrives often

Drop iterations and increase flexibility of work prioritization

Reasonably stable work (for iteration)

Greater bureaucracy
Specialized skills

Infrequent releases
Project teams
Long feedback cycles
Manual regression testing

Continuous releases
Long-lived stable teams
Short feedback cycles
Automated regression testing

© Disciplined Agile
The Process Goals of Disciplined Agile Delivery

**Inception**
- Form Initial Team
- Align with Enterprise Direction
- Explore Initial Scope
- Identify Initial Architecture Strategy
- Develop Initial Release Plan
- Develop Initial Test Strategy
- Develop Common Vision
- Secure Funding

Get the team going in the right direction.

**Construction**
- Prove Architecture Early
- Address Changing Stakeholder Needs
- Produce a Potentially Consumable Solution
- Improve Quality
- Accelerate Value Delivery

Incrementally build a consumable solution.

**Transition**
- Ensure Production Readiness
- Deploy the Solution

Release the solution into production.

**Ongoing**
- Grow Team Members
- Coordinate Activities
- Address Risk
- Evolve WoW
- Leverage and Enhance Existing Infrastructure
- Govern Delivery Team

Improve and work in an enterprise aware manner.

© Disciplined Agile
Goal: Form Initial Team

- Source
  - Existing product team
  - New
  - Team evolves itself
    - Team lead makes changes
    - Management makes changes
  - Small team (2-15 people)
  - Medium team (10-30 people)
  - Medium team of teams (10 to 50 people)
  - Large team of teams (30+ people)
  - Single team
    - Component teams
    - Feature teams
    - Internal open source
  - Generalizing specialists
    - Specialists
    - Generalists
  - Whole team
    - Specialized team
    - Ad-hoc
  - Stable (product) team
    - Project team
    - Ad-hoc
  - Co-located
    - Partially dispersed members
    - Distributed subteams
    - Fully dispersed members
  - Single-division FTEs
    - Consultants
    - Contractors
    - Outsourcers
  - Multiple-division FTEs
    - Consultants
    - Contractors
    - Outsourcers
    - Same time zone
      - 5 or more hours overlap
      - Less than 5 hours overlap
    - Multiple time zones – No overlap
  - Coaching
  - Mentoring
  - Training
  - Support the team
  - Availability
  - Dedicated
    - Ongoing part-time
    - As needed/available
Goal: Evolve Way of Working (WoW)
Ok, but how do I actually use this toolkit?
There are many ways to explore how people will work with our solution. Although there is a significant focus within the Agile community on user stories and epics, these aren’t our only choices. Disciplined Agilists prefer to use the best technique for the situation they face, and as you can see in the table below there are several options available to us.

<table>
<thead>
<tr>
<th>Options (Not Ordered)</th>
<th>Trade-Offs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Epics</strong> – Large stories that take a lot of effort, often multiple iterations, to complete. Epics are typically organized into a collection of smaller user stories [W]. Sometimes Epics are referred to as Features or User Activities.</td>
<td>• Useful for high-level program planning. • Appropriate level of detail for low priority work since the details are likely not well understood yet and are likely to change anyway.</td>
</tr>
<tr>
<td><strong>Outcomes</strong> – An outcome describes a desired, measurable result that is pertinent to our stakeholders.</td>
<td>• Outcomes describe what stakeholders would like to achieve and why they would like to achieve that, but how to do so. • Provides teams flexibility in how to achieve the desired outcome. • Useful to capture high-level stakeholder needs.</td>
</tr>
<tr>
<td><strong>Persona</strong> – Detailed descriptions of fictional people who fill roles as stakeholders of the solution being developed [W].</td>
<td>• Used as a technique to build empathy for users as real people, and to understand the optimal user experiences for each. • Useful when we don’t have access to actual end users, or potential end users. • Can be used as an excuse not to work with actual users.</td>
</tr>
</tbody>
</table>


- Puts use cases, and potentially usage scenarios and epics if we’re flexible, into context.
- Can promote requirements reuse via <<include>> and <<extend>> relationships.
- Can motivate unnecessary complexity via <<include>> and <<extend>> relationships.

**Usage scenario**. Describes the step-by-step interaction between a user/actor and the solution. Similar to acceptance criteria, although tends to cross the equivalent of several stories. Also known as a use-case.

- Useful to flush out all the different ways that a solution can be used, often putting granular requirements such as stories or features into context.
- Danger of becoming a set of detailed requirements.
- Scenarios are typically less structured than...
• Tailoring sessions
  • Early in your projects
• Retrospectives
  • Throughout your Projects
  • Identify a problem area
  • Look in the DA knowledgebase for suggestions
  • Choose a likely candidate to experiment with
• Experiment
  • Try out a practice/strategy in your environment
  • Measure the results
  • Aim is to determine whether the new strategy works well for you

Evolving Your WoW: Referencing the Practices
Expand your Pantry with new Ideas

• Learn about New Ingredients/Practices
• Disciplined Agile is a rich pantry of ingredients
• Use Different Recipes for your different Contexts
• Putting them together is not hard

© Disciplined Agile
It is not DAD “OR…”, it is DAD “AND…”
Parting Thoughts

• Context counts - Every team is unique, facing a unique situation

• Choice is good - Agile teams own their own process, they get to choose their Way of Working (WoW)

• Pragmatism – There are hundreds of practices/ingredients and strategies to choose from, we need to choose the ones that are right for us so that we can be a better Chef and get better results
QUESTIONS?

Thank you!

@Mark_Lines
@Disc_Agile

info@disciplined-agile.com